The Journal of Computing Sciences in Colleges

Papers of the 35th Annual CCSC Eastern Conference

October 25th-26th, 2019
Robert Morris University
Moon Township, PA

Baochuan Lu, Editor
Southwest Baptist University

John Wright, Regional Editor
Juniata College

Volume 35, Number 3 **October 2019**

The Journal of Computing Sciences in Colleges (ISSN 1937-4771 print, 1937-4763 digital) is published at least six times per year and constitutes the refereed papers of regional conferences sponsored by the Consortium for Computing Sciences in Colleges. Printed in the USA. POSTMASTER: Send address changes to Susan Dean, CCSC Membership Secretary, 89 Stockton Ave, Walton, NY 13856.

Table of Contents

The Consortium for Computing Sciences in Colleges Board of Directors

Following is a listing of the contact information for the members of the Board of Directors and the Officers of the Consortium for Computing Sciences in Colleges (along with the years of expiration of their terms), as well as members serving CCSC:

Kevin Treu, Southeastern Representative (2021), (864)294-3220, kevin.treu@furman.edu, Furman University, Dept of Computer Science, Greenville, SC 29613.
Bryan Dixon, Southwestern Representative (2020), (530)898-4864, bcdixon@csuchico.edu, Computer Science Department, California State University, Chico, Chico, CA 95929-0410.

Serving the CCSC: These members are serving in positions as indicated:
Brian Snider, Membership Secretary, (503)554-2778, bsnider@georgefox.edu, George Fox University, 414 N. Meridian St, Newberg, OR 97132.
Will Mitchell, Associate Treasurer, (317)392-3038, willmitchell@acm.org, 1455 S. Greenview Ct, Shelbyville, IN 46176-9248.
John Meinke, Associate Editor, meinkej@acm.org, UMUC Europe Ret, German Post: Werderstr 8, D-68723 Oftersheim, Germany, ph 011-49-6202-5777916.
Shereen Khoja, Comptroller, (503)352-2008, shereen@pacificu.edu, MSC 2615, Pacific University, Forest Grove, OR 97116.
Elizabeth Adams, National Partners Chair, adamses@jmu.edu, James Madison University, 11520 Lockhart Place, Silver Spring, MD 20902.
Megan Thomas, Membership System Administrator, (209)667-3584, mthomas@cs.csustan.edu, Dept. of Computer Science, CSU Stanislaus, One University Circle, Turlock, CA 95382.
Deborah Hwang, Webmaster, (812)488-2193, hwang@evansville.edu, Electrical Engr. & Computer Science, University of Evansville, 1800 Lincoln Ave., Evansville, IN 47722.

CCSC National Partners

The Consortium is very happy to have the following as National Partners. If you have the opportunity please thank them for their support of computing in teaching institutions. As National Partners they are invited to participate in our regional conferences. Visit with their representatives there.

Platinum Partner

Turingscraft
Google for Education
GitHub
NSF – National Science Foundation

Silver Partners

zyBooks

Bronze Partners

National Center for Women and Information Technology
Teradata
Mercury Learning and Information
Mercy College

Foreword

The following five CCSC conferences will take place this fall.

Midwestern Conference	October 4-5, 2019 Benedictine University in Lisle, IL
Northwestern Conference	October 4–5, 2019 Pacific University, Forest Grove, OR
Rocky Mountain Conference	October 11-12, 2019 University of Sioux Falls in Sioux Falls, SD
Eastern Conference	October 25-26, 2019 Robert Morris University in Moon Township, PA
Southeastern Conference	October 25-26, 2019 Auburn University in Auburn, AL

The papers and talks cover a wide variety of topics that are current, exciting, and relevant to us as computer science educators. We publish papers and abstracts from the conferences in our JCSC journal. You will get the links to the digital journal issues in your CCSC membership email. You can also find them in the ACM digital library and in print on Amazon.

Since this spring we have switched to Latex for final manuscript submission. The transition has been smooth. Authors and regional editors have worked hard to adapt to the change, which made my life a lot easier.

The CCSC board of directors have decided to deposit DOIs for all peer-reviewed papers we publish. With the DOIs others will be able to cite your work in the most accurate and reliable way.

Baochuan Lu
Southwest Baptist University
CCSC Publications Chair

Welcome to the 2019 CCSC Eastern Conference

On behalf of the 2019 CCSC Eastern Conference Committee, welcome to the 2019 CCSC Eastern Conference being hosted at Robert Morris University located in Pittsburgh, Pennsylvania. While you get an opportunity to participate in this year's conference, we hope you'll also take time to travel our wonderful city and explore the 261 years of history behind it. We look forward to seeing each of you at the conference and excited about the sessions available to faculty and students. Whether this is your first year attending or you are returning from a prior year, we think you will find the conference will provide content that is new and exciting to your field of interest.

This year, we welcome 4 different tracks presenting Faculty / Student Papers, Tutorials, Workshops, Panels, and Poster Presentations plus much more. This is our second year incorporating student presentations into our conference. We hope students will find this opportunity rewarding to present their work and get appropriate feedback. We look forward to future years of the conference where our student track continues to grow. Alongside the student presentations, we welcome the continuation of the programming competition this year which will feature college and high school student teams from around the eastern region. We will also have an enlightening workshop presented by NSF.

This year we received numerous submissions for both faculty and student papers. While it was highly competitive, we were able to accept 15 Faculty papers and 6 student papers. Topics this year covered computer science, education, and cybersecurity. The selected papers will be included in the Journal of Computing Sciences in Colleges. Additionally, we will have panel and tutorial sessions covering Mentoring, Cybersecurity, Databases for Market Basket Analysis, and Hand-Coding for Basic Graphical Transformations.

This conference could not have been done without the help of our volunteers and the committee. We are grateful for their help and experience with the conference along with reviewing the faculty and student submissions. A special thank you to John Wright and Mike Flynn who helped keep us on track for delivering and executing a wonderful conference experience this year. Of course, a final thanks to Robert Morris University for hosting our conference this year along with our wonderful sponsors.

We hope you will enjoy your time in Pittsburgh. Please let us know about your conference experience so we can continue improving it each year. If you are interested in helping for future conferences by hosting, participating, volunteering on the committee, or serving as a paper reviewer, please contact any committee member. We hope to see each of you at the conference. Enjoy!!

Karen Paullet
Robert Morris University
Adnan Chawdhry
California University of Pennsylvania
Conference Chairs

2019 CCSC Eastern Conference Committee

Karen Paullet, Co-Chairs .... Robert Morris University
Adnan Chawdhry, Co-Chairs .... California University of Pennsylvania
Karen Paullet, Papers .... Robert Morris University
Donna Schaeffer, Papers .... Marymount University
John Wright, Papers .... Juniata College
John Wright, Panels, Workshops, and Tutorials .... Juniata College
Adnan Chawdhry, Student Faculty Posters .... California University of Pennsylvania
Steven Kennedy, Programming Contest .... Frostburg State University
Karen Paullet, Registration, Local Arrangements .. Robert Morris University
Nathan Green, Treasurer .... Marymount University
John Wright, Regional Board Representative, Web Site ...... Juniata College

2019 CCSC Eastern Steering Committee

Elizabeth Adams .... James Madison University
Steven Andrianoff .... St. Bonaventure University
Karen Anewalt .... University of Mary Washington
George Benjamin .... Muhlenberg College
Elizabeth Chang .... Hood College
Vincent Cicirello .... Stockton University
Michael Flinn .... Frostburg State University
Sister Jane Fritz .... St. Joseph's College
Nathan Green .... Marymount University
David Hovemeyer .... Johns Hopkins University
John Meinke .... University of Maryland University College
Donna Schaeffer .... Marymount University
Jennifer Polack-Wahl .... University of Mary Washington
John Wright .... Juniata College

Reviewers — 2019 CCSC Eastern Conference

Rana Alabdan	Robert Morris University, Moon, PA
Faleh Alshameri	Marymount University, Arlington, VA
Ian Barland	Radford University, Radford, VA
Mónica Costa	Polytechnic Institute of Castelo Branco, Portugal
Ian Finlayson	The University of Mary Washington, Fredericksburg, VA
Reva Freedman	Northern Illinois University, DeKalb, IL
Alessio Gaspar	University of South Florida, Tampa, FL
Nathan Green	Marymount University, Arlington, VA
Ivaylo Ilinkin	Gettysburg College, Gettysburg, PA
Alex Mbaziira	Marymount University, Arlington, VA
Robert McCloskey	University of Scranton, Scranton, PA
John McManus	Randolph-Macon College, Ashland, VA
James Moscola	York College of Pennsylvania, York, PA
Muhammed Nadeem	Mississippi State University, Starkville, MS
Patrick Olson	National University, Campbell, CO
Karen Paullet	Robert Morris University, Moon, PA
Donna Schaeffer	Marymount University, Arlington, VA

Database for Conducting Market Basket Analysis*

Conference Workshop

Stephen Larson
Healthcare Administration & Information Systems
Slippery Rock University
Slippery Rock, PA 16057
stephen.larson@sru.edu

1 Abstract

This assignment can be presented two different ways:

1. A point-and-click process to teach market basket analysis. This is a lab which includes an Access database and ready-made queries which are used with a lab worksheet to learn how market basket analyses work.

2. A data database development tool to teach how to build queries and learn how market basket analyses work. This second lab follows a lab worksheet in which the students start with the raw data in a spreadsheet, perform ETL, import the data into a blank Access database and build queries to perform a market basket analysis.

Attendees will receive a USB drive containing the lab worksheets, the Access database, and the raw data in an Excel spreadsheet.

This is a hands-on workshop. Attendees should bring their own laptop with Windows OS, MS Access and MS Excel installed.

2 Intended Audience

Any teachers of data analysis or database development.

3 Presenter's Background

Stephen Larson worked in industry as an IT/IS consultant for over 15 years, earning a number of industry certifications before getting a PhD in Business Information Systems. He has worked with spreadsheets and databases to perform data analysis for several years. He has developed and taught data analysis courses at the undergraduate and graduate levels, and enjoys creating new ways to teach data analysis and make it interesting for students.

Cloud computing and Running code on Google Cloud*

Conference Workshop

Wesley Chun
Developer Advocate, Google
Mountain View, CA 94043
wesc+api@google.com

1 Abstract

Cloud computing skills are critical for student career readiness, and more cloud should be integrated into the curriculum. This lecture+hands-on workshop shows you how! The lecture begins with a vendor-agnostic tour of cloud computing to ensure common vocabulary and pedagogy focus. An overview of Google Cloud Platform (GCP) and G Suite developer tools follows. Products covered can be applied to many courses. Lecture also covers Google education grants (in qualifying countries) which include both teaching and research grants. In the hands-on component, attendees learn how to use our REST APIs and run code on our serverless platforms via hands-on tutorials

2 Description of Workshop

This 3-hour workshop is comprised of two halves, 90 minutes of comprehensive lecture as well as 90 minutes of hands-on coding. The lecture covers cloud computing, Google Cloud products, Google API usage, serverless computing on Google Cloud, and sample use-case applications as inspiration. Technical overviews aside, lecture also covers the education grants as described in the abstract as well as a career-readiness program for students & faculty.

The hands-on portion is made up of individual hands-on tutorials demonstrating how to use various technologies in Google Cloud. These "codelabs," as they're referred to, are generally self-paced, but will be led in real-time by the speaker in this workshop. Each codelab runs about a half-hour each. The goal is to provide a broad overview of both the technologies available as well as in-practice use with the hands-on portion.

3 Intended Audience

This workshop is open to anyone, whether you're a professor, researcher, university IT staff, grad student, or lecturer. You can be in either an engineering or non-engineering discipline. As long as students/staff are expected to code as part of your coursework or graduate research, this workshop is meant for you. Undergraduates are also welcome to attend and add cloud skills to their repetoire immediately.

4 Speaker's Biography

WESLEY CHUN (@wescpy) is the author of Prentice Hall's bestselling "Core Python" (corepython.com) series, co-author of "Python Web Development with Django" (withdjango.com), and has written for Linux Journal, CNET, and InformIT. In addition to being a Developer Advocate at Google focused on Google Cloud (for higher education) and a host of the G Suite Dev Show (goo.gl/JpBQ40), he runs CyberWeb (cyberwebconsulting.com), a consultancy specializing in Python training. Wesley has decades of programming, teaching, and writing experience, and was one of the original Yahoo!Mail engineers. He holds degrees in CS, Math, and Music from the University of California, is a Fellow of the Python Software Foundation, and Adjunct Computer Science faculty at Foothill College in Silicon Valley.

5 Speaker's Background

The speaker has over 35 years of teaching experience at all levels from elementary school to retired technical professionals and all levels in-between. He has nearly three decades of experience in industry as a software engineer. He is also a published author, including several conference and journal papers as well as several technical books. He has been an engineer building tools and advocating publicly for Google Cloud for nearly a decade and has subject matter expertise in its primary product groups.

Introduction to Jetstream - A Research and Education Cloud*

Conference Tutorial

Sanjana Sudarshan and Jeremy Fischer
Research Technologies
Indiana University
Bloomington, IN 47401
{ssudarsh, jeremy}@iu.edu

1 Abstract

Jetstream is the first production cloud funded by the National Science Foundation (NSF) for conducting general-purpose science and engineering research as well as an easy-to-use platform for education activities. Unlike many high-performance computing systems, Jetstream uses the interactive Atmosphere graphical user interface developed as part of the iPlant (now CyVerse) project and focuses on interactive use on uniprocessor or multiprocessor. This interface provides for a lower barrier of entry for use by educators, students, practicing scientists, and engineers. A key part of Jetstream's mission is to extend the reach of the NSF's eXtreme Digital (XD) program to a community of users who have not previously utilized NSF XD program resources, including those communities and institutions that traditionally lack significant cyberinfrastructure resources. One manner in which Jetstream eases this access is via virtual desktops facilitating use in education and research at small colleges and universities, including Historically Black Colleges and Universities (HBCUs), Minority Serving Institutions (MSIs), Tribal colleges, and higher education institutions in states designated by the NSF as eligible for funding via the Established Program to Stimulate Competitive Research (EPSCoR).

While cloud resources won't replace traditional HPC environments for large research projects, there are many smaller research and education projects that would benefit from the highly customizable, highly configurable, programmable

cyberinfrastructure afforded by cloud computing environments such as Jetstream. Jetstream is a Infrastructure-as-a-Service platform comprised of two geographically isolated clusters, each supporting hundreds of virtual machines and data volumes. The two cloud systems are integrated via a user-friendly web application that provides a user interface for common cloud computing operations, authentication to XSEDE via Globus, and an expressive set of web service APIs.

Jetstream enables on-demand access to interactive, user-configurable computing and analysis capability. It also seeks to democratize access to cloud capabilities and promote sharable, reproducible research. This event will describe Jetstream in greater detail, as well as how its unique combination of hardware, software, and user engagement support the "long tail of science." This tutorial will describe Jetstream in greater detail, as well as how its unique combination of hardware, software, and user engagement support the "long tail of science." Attendees will get a greater understanding of how Jetstream may enhance their education or research efforts via a hands-on approach to using Jetstream via the Atmosphere interface.

2 Tutorial Description

This tutorial requires two to three hours.

- Prerequisites: Basic Linux command line knowledge a plus (but not required)
- Required: Laptop, modern web browser (Chrome, Firefox, Safari)
- Targeting: Educators, Researchers, Campus Champions/ACI-Ref Facilitators, Campus research computing support staff

This tutorial will first give an overview of Jetstream and various aspects of the system. Then we will take attendees through the basics of using Jetstream via the Atmosphere web interface. This will include a guided walk-through of the interface itself, the features provided, the image catalog, launching and using virtual machines on Jetstream, using volume-based storage, and best practices.

We are targeting users of every experience level. Atmosphere is well-suited to both HPC novices and advanced users. This tutorial is generally aimed at those unfamiliar with cloud computing and generally doing computation on laptops or departmental server resources. While we will not cover advanced topics in this particular tutorial, we will touch on the available advanced capabilities during the initial overview.

3 Tutorial Program

This is a sample tutorial program. Time required for this tutorial is approximately 3 hours.

- What is Jetstream?
- Q & A and what brief hands-on overview
- Getting started with Jetstream, including VM launching
- Break
- Accessing your VM, creating and using volumes
- Customizing and saving images, DOIs
- Cleaning up
- Final Q & A

A Hand-Coding Lecture for Basic Graphical Transformations*

Conference Tutorial

Penn Wu
Computer Information Systems
DeVry University
Sherman Oaks, CA 91403
pwu@devry.edu

1 Abstract

Being able to write codes to produce visual effects of graphical transformations is a highly desired skill in the industry, although many programming languages, application programming interface (APIs), or libraries provide professional and read-made toolkits to transform graphics.

The presenter strongly believes that students majoring in computer graphics, gaming, and, simulation must acquire this programming skill. However, students often focus on learning how to render transformed graphics using ready-made tools and miss the opportunity to learn to hand-code programs to perform the demanded transformations.

This presentation will describe how to prepare a lecture, at a level suitable for lower-division of undergraduate students, with a series of hand-coding projects designed for students acquire programming skills. The instructional contents include the review of matrix-based math concepts for translating, scaling, rotating, and shearing graphics in a 2D plane and 3D world. The presentation includes live demonstration of Visual C# codes that demonstrate how to apply mathematical concepts to produce visual effects of graphic transformations without using ready-made transformation tools. If time allowed, the presenter will also discuss pedagogical issues and barriers.

Audiences will take home with handouts to help them prepare for the lectures as well as step-by-step instructions to compile samples into executable Windows forms applications (WFPs).

---

2 Intended Audience

College educators, instructional content developers, and administrators of Computer Sciences, Computer Engineering, Gamin, and Simulation

3 Presenter's Comments

There are many programming languages, application programming interfaces, and libraries available in the market to facilitate the coding of graphical transformation. They provide abundant tools to learners and educators of computer graphics, gaming, and simulation. However, if the learning is based on using ready-made tools, students will not have a foundation to resolve issues when they encounter a graphics problem that must be solved with programmability. Interestingly, this is a topic overlooked by many educators.

The presenter intends to reiterate the awareness of this pedagogical issue by sharing how he led students in a career-oriented college with hands-on, learn-by-doing coding projects to explore and understand how mathematical concepts are used in graphical transformations. The presenter is eager to advocate the need to guide students to hand-code features that produce visual effects of translation, scaling, shearing, and rotating without the use of professional toolkits.

Introduction to Cyber Security*

Conference Tutorial

Veena Ravishankar
Computer Science
University of Mary Washington
Fredericksburg, VA 22401
vravisha@umw.edu

1 Description

What is cyber security? Why is there a need to study cyber security? What does it mean to study and invest in cyber security these days? What are the tools and technologies available to teach various components of cyber security? We will be addressing all these questions in this tutorial.

The participants will be introduced to different components and topics of cyber security, with details on assignments and in-class activities, in each category. Topics include the policy making, securing systems, and web-based threats. We will specifically focus on two of the assignments in different categories. We would encourage the participants to share their assignments and tools they use in their classrooms.

Intended audience includes faculty who are interested in teaching security related courses within the technology field and promoting discussion on related topics.

2 Presenter's Background

Veena Ravishankar is an Assistant Professor of Computer Science at the University of Mary Washington. She teaches courses in discrete mathematics, object-oriented design, programming, and cyber security. Her research focuses on formal analysis of cryptographic protocols and tools, and also has broad interests in Natural Language Processing. Her work has been published in The International Workshop on Unification, Description Logic, Theory Combination, and All That and International Symposium on Frontiers of Combining Systems and The International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction.

Evaluating Potential Solutions for Facility Location Problems in Parallel on GPUs*

Erik Wynters
Department of Mathematical and Digital Sciences
Bloomsburg University of Pennsylvania
Bloomsburg, PA 17815
ewynters@bloomu.edu

Abstract

Two different facility location problems with many potential solutions are executed serially and in parallel. One problem is where to place a new facility to get the most customers if they choose the facility closest to them. The other problem is where to place a communications device to minimize signal loss to existing sites it needs to communicate with. The parallel implementation executed on a powerful graphics card's processor (GPU) runs hundreds or thousands of times faster than a serial implementation executed on a central processing unit (CPU). The speedup factor for the GPU versus the CPU varies depending on the CPU and GPU and some characteristics of the problem. The approach used in this paper (C++ AMP), is an easy way to start using massive parallel processing in a research project or college course.

Introduction

Some expensive CPUs for personal computers might be two or three times faster than a cheaper low-end one. And CPUs typically have 2-8 cores that can run in parallel, which means their speedup factor for parallel processing versus serial processing would be in the 2-8 range. In contrast, powerful GPUs have hundreds or thousands of cores that can run in parallel. They are not equivalent to CPU cores or we wouldn't need both. They have different strengths and weaknesses. The GPU cores are not independent. They need to execute the same instructions in parallel or they block other threads and execute sequentially. And they perform best on tasks that emphasize single-precision floating point calculations with lots of parallelism, since that's what graphics processing is mostly about.

An early approach to general parallel programming on GPUs was CUDA ([7, 4]). A newer approach that was more high-level and handled more details for the user was Thrust ([2]). But both of those approaches only work with the Nvidia-brand graphics cards. Another high-level approach called C++ AMP (Accelerated Massive Parallelism) ([5]) can be used with different GPU brands and architectures (e.g., NVidia GTX cards and AMD Radeon cards). It was originally implemented as part of Microsoft's Visual C++ programming language for use on PC's using the Microsoft Windows operating system. It's an open standard and other implementations exist for other platforms now.

C++ AMP lets a programmer easily parallelize some serial C++ programs that do something for each element of a vector, a template array that's part of the C++ standard template library, a.k.a. the STL ([6]). C++ ([3]) also lets you do a task for each element of the vector using the for_each function that lets you specify what to do for each element inline without creating a named function. C++ AMP has a similar function called parallel_for_each, that distributes what to do for each element over the GPU cores that execute in parallel. Before executing that function, a vector to be processed has to be copied to an array object on the graphics card's memory. After doing that and executing the parallel task, a single assignment statement can be used to wait until the task completes on the GPU and copy the results back to a standard vector container in main memory. Another Journal of Computing Sciences in Colleges article ([8]) demonstrates parallel speedups on a GPU using C++ AMP.

A Facility Location Problem

This problem evaluates potential locations for a new facility. As shown in Figure 1, if customers are evenly distributed and prefer to use the facility that's closest to where they live, the size of the gray-shaded region that's closest to each site (a white dot) is a measure of how many customers each site gets. This example uses a pixel-based representation of the area that's being divided into regions closest to each site with a certain resolution (rows and columns). It calculates which site each pixel is closest to and colors each pixel based on which site it's closest to.

Given the information already calculated and displayed in Figure 1, this problem chooses where to place a new facility to get the most evenly distributed

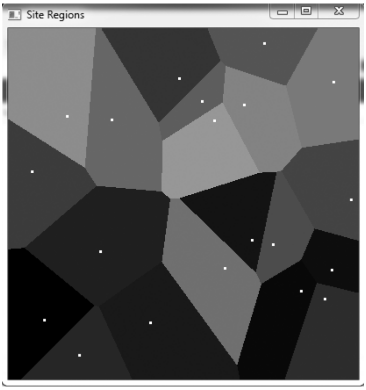


Figure 1: Original Sites.

customers that will be closer to the new facility than an existing one. To do that it calculates for each row and column in the pixelated region, how many pixels would be closer to the new potential site than the one they're currently closest to. After calculating the number of "steals" from current sites for each potential new site, the one with the most "steals" is where the new facility should be placed. That's shown below in Figure 2 for the example shown above in Figure 1. The black dot is the new location and the white region is all the points closer to the new facility than a previous one.

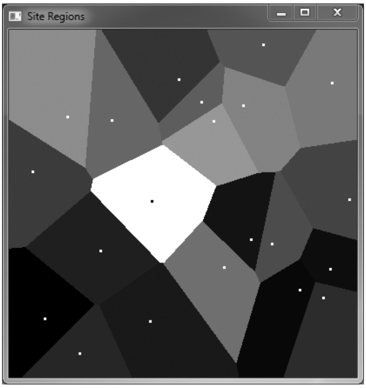


Figure 2: New Chosen Site.

As shown in Table 1 below, the parallel version executed on the GPU was hundreds of times faster than the serial version executed on the CPU on the platform used for testing (run times are in seconds).

Table 1: Run Times for First Problem.

Sites	CPU	GPU	Speedup
750 X 750	751.263	2.902	274.5
1000 X 1000	2372.550	7.949	298.5
1250 X 1250	5798.338	17.691	327.8
1500 X 1500	12025.986	34.611	347.5

Testing Methodology

The platform used for testing was a powerful gaming PC with a 7$^{\text{th}}$-generation Intel Core i7-7700K CPU running at 4.20 GHz and an NVidia GTX 1080Ti GPU with 3584 cores that can run in parallel. A single-threaded serial version was executed on the CPU and a parallel version with a thread for each pixel was executed and distributed over the GPU cores. Microsoft's Visual Studio 2017 IDE was used to produce release-mode executables with the O2 flag set to optimize code for maximum speed and with fast single-precision floating point calculations on both platforms.

C++ AMP was used to parallelize the serial version for execution on the GPU. It handles the details of creating a thread for each element of an array. After initially calculating serially which existing site each pixel is closest to, it evaluates each pixel as a potential new site by calculating how many pixels would be closer to the potential new site than the current one it's closest to. The "steals" for each pixel were calculated serially on the CPU and in parallel on the GPU for the run times shown in Table 1.

Another Facility Location Problem That Has Better Speedup

When choosing a facility location for where to place a communications tower, e.g., a radio station transmitter or a cell phone tower, the loss in signal strength through free space is a major factor to be considered ([1]). Since signal loss is more important for customers further away from the tower than ones that are closer, the bigger distances should be emphasized when choosing where to place the tower. Given locations of all the customers, the evaluation of each potential tower location uses a sum of numbers that are distances to each customer scaled to a low number and then used as an exponent (with base 2) to emphasize the bigger distances.

Similar to the first example, this example compares a finite set of options for the facility divided into rows and columns. It calculates the total signal loss for each potential tower location using the sum of the exponential scaled distances mentioned above. This example evaluates a million potential locations divided into a thousand rows and a thousand columns. Given different numbers of randomly-generated customers located in those million locations, it calculates the total signal loss for each potential tower location serially on the CPU mentioned above and in parallel on the GPU mentioned above.

As shown in Table 2, the parallel version of this problem has much better speedups than the first facility location problem. Instead of being around 300 times faster in parallel (much less than the cores on the GPU), this problem is about 4000 times faster in parallel (more than the cores on the GPU). CPUs and GPUs have different strengths and weaknesses, which is why different problems have different speedups when parallelized on a powerful GPU.

Table 2: Run Times for Second Problem.

Sites	CPU	GPU	Speedup
16,000	239.561	0.065	3685.6
32,000	479.037	0.109	4394.8
64,000	958.293	0.204	4697.5
128,000	1918.197	0.373	5142.6

Conclusions

This paper shows that a powerful GPU executing a program in parallel using C++ AMP can be hundreds or thousands of times faster than a serial program run on a CPU. This is an easy way to start using massive parallel processing in a research project or college course.

It might seem unrealistic that the speedups in Table 2 exceed the number of cores in the GPU, but GPU cores are not equivalent to CPU cores. They have different strengths and weaknesses. Many factors affect speedups that can be achieved with parallel processing on GPUs. They include characteristics like high parallelism (GPU threads are not independent), computational intensity, data transfer needs, and memory access patterns. The second example had more parallelism since there wasn't a conditional statement and it had more single-precision floating point calculations by adding scaling and the powf function to each distance calculation. The greater parallelism and those additional single-precision floating point calculations are what GPUs are best at and why the speedups were about 13 times bigger than what they were in the first example.

Both examples had better speedups at the bottom of the run time tables, where they did more calculations. The results stored in the array on the graphics card have to be copied back to main memory as part of the timing. When more calculations are done on the GPU, the time needed to transfer the results back to main memory becomes a smaller portion of the total run time on the GPU, which explains why the instances with more calculations have better speedups.

The second example in this paper took less than half a second for the largest instance on the GPU, but took about 32 minutes on the CPU. Bigger instances or other problems that take more time could be much slower. If a larger instance or a harder problem took a whole day (24 hours) to solve it in parallel with the biggest speedup achieved in Table 2, it would take over 14 years to solve it serially. That shows how parallel speedups can make it practical to solve some problem instances that wouldn't be practical to solve serially.

References

[1] Constantine A Balanis. *Antenna theory: analysis and design*. John wiley & sons, 2016.

[2] Nathan Bell and Jared Hoberock. *Thrust: A productivity-oriented library for CUDA*. Morgan Kaufmann Publishers Inc., 2011.

[3] Paul Deitel and Harvey Deitel. *C++ how to Program*. Pearson, 2017.

[4] Rob Farber. *CUDA Application Design and Development*. Morgan Kaufmann Publishers Inc., 2011.

[5] Kate Gregory and Ade Miller. *C++ AMP: accelerated massive parallelism with Microsoft Visual C++*. Microsoft Press, 2012.

[6] Nicolai M Josuttis. *The C++ standard library: a tutorial and reference*. Addison-Wesley, 2012.

[7] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda. *Queue*, 6(2):40–53, March 2008.

[8] Erik Wynters. Parallel particle swarm optimization can solve many optimization problems quickly on GPUS. *Journal of Computing Sciences in Colleges*, 33(6):114–123, 2018.

Using Three-Dimensional Terrain Models to Measure Terrain Change*

John McManus
Computer Science Department
Randolph-Macon College, Ashland, VA 23005
`johnmcmanus@rmc.edu`

Abstract

This paper presents the successful completion of the first phase of a multi-disciplinary research program. The research program utilizes low-cost unmanned aerial vehicles to collect high-resolution terrain photographs. Computational methodologies developed by the Principal Investigator and students process the photographs into three-dimensional digital elevation (aka terrain) models. Changes between digital elevation models are used to measure changes in the terrain over time. The algorithms provide capabilities not available in existing commercial tools. The computation methodologies allow for the correct alignment of the high-resolution three-dimensional digital elevation models, analysis of changes between models, and visualization of changes in the models. This paper covers the development, verification, and validation of the data collection and analysis tools to measure terrain change at $\pm$ 1 cm. Two types of digital elevation model data sets were collected at the same location over a two-year period. Twelve "Accuracy" data sets were collected in a fixed area at the same altitude. The accuracy data sets were used to measure the accuracy (in cm) of the terrain data and the repeatability of the data analysis process. Fifteen "Sensitivity" data sets were collected in the same area; three data sets at each of five different altitudes. The sensitivity data sets were used to measure the sensitivity of the data collection and analysis process to changes in the altitude used to collect the data. Surface difference matrices that measure the difference between each point in two digital elevation models were calculated. The area of each point in the three-dimensional digital elevation

elevation model, measured in cm^2, is determined by the altitude of the unmanned aerial vehicle when the data is collected, and the resolution of the camera used to take the photographs.

The next phase of this research program is to apply the three-dimensional terrain modeling tools to measure the impact of successive storms on the Virginia Barrier Islands. The digital elevation data and surface difference matrices will be used to validate a recently developed mathematical model that quantifies the impact of successive storms on sandy beaches.

Keywords: Terrain Modeling, Computational Methods, Image Processing, Unmanned Air Vehicles

1 Introduction

This research program has developed and validated quantitative computational methods and software for measuring changes between three-dimensional Digital Elevation Models (DEM's). DEM's are created by processing high resolution aerial images. Sets of overlapping high-resolution digital images of a data collection area are processed to develop DEM's with the temporal and spatial resolution required to analyze changes in the terrain being studied. The high-resolution photographic images are collected, processed, and three-dimensional point cloud data is extracted. The point cloud data contains the location (latitude, longitude, and altitude) of each data point in the data collection area. Each data point represents approximately one cm^2 of terrain. The point cloud data is processed and used to generate high resolution three-dimensional DEM's. Surface difference models are matrices created by comparing two DEM's on a point by point basis, calculating the altitude differences between each data point in the two DEM's.

The tools and techniques will be used by a multidisciplinary research team (Environmental Studies, Physics, and Computer Science) to measure the impact of successive storms on the Virginia Barrier Islands (VBI's). The VBI's were selected for this study because of their pristine (undeveloped) condition and the increased amount of scientific attention given to this region during the past two decades [6] [17] [7] [8]. The DEM's and surface difference models will be used to measure the impact of multiple storms and the attendant changes that occur to the physical and ecological systems of beaches and barrier islands. The VBI data will be used to validate a mathematical model used to quantify the spatial-temporal impact of successive storms on sandy beaches [6].

The high-resolution aerial images used to validate the models were collected by the Principal Investigator (PI) and undergraduate research students using Unmanned Aerial Vehicles (UAV's). The use of commercial-off-the shelf UAV's offers a significant cost savings over conventional fixed-wing aircraft, helicopters, and proprietary UAV's currently used to collect shoreline imagery and three-dimensional maps [15] [25]. The UAV's require smaller take-off and landing areas and can be deployed swiftly after storms or other major events [16]. The lower cost of deployment and operations allow researchers to collect data more often, improving the temporal resolution (the amount of time between collections) of the models [25]. When coupled with the appropriate software, UAV's can create DEM's and take volumetric terrain change measurements 85 percent faster and at one-tenth of the per-survey cost of traditional survey approaches [23].

1.1 Background

The research program utilizes a low-cost UAV to capture high resolution images. The data, in the form of three-dimensional point clouds, is extracted from the high-resolution photographs. Computational methodologies developed and validated by the PI and students process the point cloud data into DEM's. Successive DEM's are used to calculate surface difference matrices that capture the altitude difference in the terrain at each point in the model. Preliminary research [5] [24] has demonstrated that the DEM's provide sufficient temporal and spatial resolution (data collected approximately every cm^2) to accurately measure terrain changes. The first phase of the research validated the accuracy of this research approach. Analysis [5] of the data processing and DEM alignment methods at a single location with fixed features with known dimensions determined the error between the measured terrain features and each DEM. The computation methodologies and data visualization techniques allow for the correct alignment, analysis, and visualization of changes with high resolution (1 cm^2 spatial resolution) DEM's. The data visualization techniques enable comprehension of shoreline / terrain changes across a large population of both technical and non-technical stakeholders.

While UAV based research has been used to analyze the impact of individual storms [26] [10], no studies have analyzed the impact of successive storms on beaches or barrier islands. The data collected will validate a new numerical model developed to quantify the spatial-temporal impact of successive storms on sandy beaches [6]. The model, termed the Cumulative Storm Impact Index (CSII) uses a weighting scheme that incorporates the timing between successive storms and a storm erosion potential index (SEPI) to quantify the cumulative impact of successive storms on beaches. The model requires volumetric data for validation. The volumetric data are time consuming and costly to acquire – especially in difficult to access areas. Surface difference maps from pre- and post-storm surveys will provide the volumetric loss or gain to a beach as a function of successive storms striking the same region. These data will be compared to erosion potential output from the CSII, enabling re-parameterization of the theoretical model to fit reality. This process will be repeated for multiple, successive storms. Once validated, the theoretical model will be applicable to sandy beaches around the world and provide a more accurate representation of historical storm impacts and predictions of cumulative storm impacts for any sandy beach. This information is especially important given studies which have conjectured that recent increases in the magnitude of large coastal storms (tropical and extratropical) have occurred in response to climate change [9] [1].

The stereographic images and DEM's capture the impact of multiple storms and the attendant physical changes that occur to the physical and ecological systems of beaches and barrier islands. This allows costal systems researchers to relate antecedent conditions to post-storm impacts (e.g., beach or dune erosion, deposition / overwash, shoreline migration, nearshore erosion / deposition and vegetative abundance / absence). This will answer questions relating natural physical and ecological coastal system responses (changes) and how potential acceleration to these processes may impact these fragile and critical systems. Achieving these research objectives will allow determination of threshold events which engender system (barrier island) state changes.

The spatial and temporal resolution of the data sets is critical to the success of the analysis of shoreline change. The spatial resolution of a data set specifies the area in square centimeters or meters of the sensor data. High spatial resolution is defined as 0.41 m^2 to 4 m^2, and low spatial resolution is defined as 30 m^2 to 1000 m^2 [2]. When applied to remote sensing, temporal resolution depends on the characteristics of the sensor platform and the characteristics of the sensor systems. Temporal resolution is high when the revisiting delay (the time between each data collection effort) is low. Temporal resolution for modern remote sensing systems is usually expressed in days. The temporal resolution of a data set specifies the revisiting frequency of the data collection sensor (e.g., UAV mounted camera) for a specific location. High temporal resolution is defined as a revisiting frequency between 24 hours and 3 days. Medium temporal resolution is defined as a revisiting frequency between 4 and 16 days. Low temporal resolution is defined as a revisiting frequency that is greater than 16 days [2].

Existing shoreline change research uses data with low temporal resolution, low spatial resolution, and low spatial accuracy. The spatial accuracy varies from $\pm$ 20 cm for GPS survey data to $\pm$ 10 m for satellite and National Oceanographic Service topographic sheets [17] [4]. The temporal resolution for the historical data is very low, usually measured in years or decades [11]. Satellite imaging data is becoming available at higher temporal resolution and higher spatial resolution, but satellites are difficult to redirect in response to storms or natural disasters [16].

Research in the field of computer vision has developed highly efficient image matching methods and commercial products to generate point clouds [25]. Point clouds can be generated at unprecedented spatial resolution due to the low altitude of data collection flights (20m and 40m), precision flight path planning, and high-resolution (currently 4K video, 12-megapixel photo) cameras available. Multi-view stereo photographic data is collected and used to generate the point clouds. This data collection approach is easily supported by UAV platforms; the flying speed is low, and the image acquisition rate is high. The high image overlap required to produce accurate point cloud data can be achieved. More information on point cloud accuracy assessment and image matching algorithms, their classification and comparison can be found in [25], [19], [21], [22].

This research program developed a comprehensive set of data capture, processing, and analysis tools that address the cost, spatial accuracy, and temporal and spatial resolution of existing shoreline change analysis tools. The use of UAV's allows high-resolution temporal and spatial data capture. The high-resolution photographs are processed into orthophotos, photos that are geometrically corrected so the scale is uniform. Point clouds and three-dimensional terrain models are extracted from the orthophotos. The high temporal resolution of the UAV based data collection system allows researchers to monitor a site in near real time [16]. Traditional data collection and analysis platforms (satellites, fixed wing, and rotary wing aircraft) present problems related to temporal and spatial resolution [20], [15]. UAV's have emerged as the preferred alternative to acquire data at the optimal time and with minimal time gaps between data collections. UAV's allow the collection of high spatial and temporal resolution data sets [15]. The high spatial resolution terrain models facilitate a measurement of shoreline changes at a resolution not currently possible. The surface difference matrices facilitate a point by point comparison of changes at a spatial accuracy of $\pm$ 1.0 cm. The data visualization techniques enable comprehension of shoreline changes across a large population of stakeholders.

2 Methodology

Prior research with multi-rotor UAV's [20] tested two data collection approaches based on the UAV following a preprogrammed flight path. The authors tested a preprogrammed flight path with preprogrammed image capturing spots. The UAV stopped to hover in a fixed position to capture an image. This approach was impractical for mapping purposes. Stopping to hover in a fixed position requires decelerating before each stop and accelerating after the stops. The approach consumed excessive time and battery power.

The preliminary research conducted [3] [5] [24] during 2017 and 2018 confirmed this result. The research flights presented in this paper employ a revised data collection approach. The revised approach is more time and energy efficient. The UAV flies the preprogrammed flight route at a constant speed without stopping for image capture. Images are captured at fixed intervals with high (75 percent) image overlap. This approach conserves power, allowing the UAV to travel a greater distance during a single flight.

The Pix4D 3D data package [18] used to extract the raw point cloud data provides positional accuracy of ± 1cm in the x and y axes and ± 1.5cm in the z axis. Spatial resolution is high at 0.99 cm / pixel at 75 ft altitude. Spatial resolution can be adjusted by collecting data at different altitudes. Temporal resolution of a UAV-based data collection system is high. Multiple data collection flights can be accomplished within a 24-hour period. The spatial accuracy, spatial resolution, and temporal resolution of this system exceed the capabilities used by current shoreline change research programs [17].

The point cloud data collected has an irregular shape (Figure 1.) and requires significant processing to produce the surface difference matrices used for terrain analysis. The software developed to process the point cloud data into DEM's has efficient methods for storing, aligning, resizing, and processing the increased volume of higher spatial resolution data. The surface difference matrices and analysis tools allow for point by point comparisons at 1 cm^2 resolution, a resolution not supported by current solutions. All software developed under this project will be released under a Creative Commons CC BY-NC-SA license.

Figure 1: Data Collection Site

The point cloud alignment and analysis tools are currently implemented in the Python programming language. The first phase of processing reads in the point cloud data points produced by Pix4D and determines the range of latitude and longitude data contained in the point cloud. Two vectors, one containing the latitude data, and one containing longitude data are created. A matrix containing the altitude at each data point is also created. As shown in Figure 1, the terrain areas analyzed are irregular. The DEM matrix must be correctly oriented (north is up, and east is to the right) to allow efficient point by point comparisons of their data elements. The minimum and maximum latitude and longitude values contained in a point cloud are determined. A "bounding box" matrix with the full range of the latitudes as the x-axis values and full range of longitudes as the y-axis values is formed. The matrix surrounds the point cloud data, padding any empty elements with `None` values. The point cloud altitude data is then loaded into the matrix at the correct latitude and longitude.

The second phase of processing reads in two DEM data sets and resizes them so both data sets include the same area of terrain (the same set of latitude and longitudes. The resized matrices are the union of the latitude and longitude ranges. The minimum and maximum latitude and longitude values from the two data sets are used to create revised matrices that include the ranges of data from both DEM's. Data points are added as required to complete the matrices. The additional data points in the revised DEM's are filled with `None` values to distinguish them from the point cloud data.

The last phase of processing calculates the surface difference matrices which contain the point by point altitude differences between two data sets. The surface difference matrices are used to validate the results of the first two phases. The algorithm processes two DEM's that have the same latitude and longitude ranges and creates a surface difference matrix that contains the difference of the two DEM's. If one of the DEM data points in a comparison is a `None` value (no DEM data existed in the original model) a `None` value is stored in the surface difference matrix. This prevents the comparison of points in the DEM models that do not exist in the raw point cloud data. The comparison of a DEM with itself should produce a surface difference matrix of all zeros and `None` values.

MATLAB is used to visualize the three-dimensional terrain data. The latitudes, longitudes, and altitudes within the matrix are used to create a 3D grid that contains the mesh of altitudes. The `surf(x, y, z)` function uses the latitudes, longitudes, and matrix to produce a 3D model.

3 Results

The first-generation point cloud processing and analysis algorithms were developed during the fall of 2017 and revised in the spring of 2018. Additional data collection occurred in the fall of 2018 and spring of 2019. The first generation point cloud processing and analysis algorithms [5] [24] were refined to address the issues identified during the first round of data collection and analysis. The point cloud processing algorithms were refined to reduce errors in properly aligning points into the matrices. The algorithms and data structures were optimized for efficient storage and analysis of high volumes of data.

A total of twenty-seven data sets were collected and analyzed. The Senior Capstone students collected nine data sets in the fall of 2017. The PI collected eighteen data sets in the fall of 2018 and the spring of 2019. All student collected data sets were reprocessed in the spring of 2019 after the algorithms were revised. There are two types of data sets: Accuracy data sets and Sensitivity data sets.

The twelve accuracy data sets were flown at an altitude of 60 ft and have a spatial resolution of 0.81 cm^2. They were used to determine the spatial accuracy and repeatability of the data collection process and algorithms.

Table 1: Sensitivity Flights

Number of Flights	Altitude ft	Area cm^2
3	45	0.60
3	60	0.81
3	75	0.99
3	90	1.19
3	105	1.40

The fifteen sensitivity data sets were flown at five different altitudes (Table 1). Each altitude tested produced a point cloud where the data elements represented a different sized area (ranging from 0.60 cm^2 to 1.40 cm^2) of terrain. The sensitivity data sets were used to measure the sensitivity of the algorithms to changes in altitude and spatial resolution.

3.1 Preliminary Terrain Modeling Research, Fall 2017

Preliminary terrain modeling research was conducted in the fall of 2017 by two Senior Computer Science majors as their Senior Capstone projects. One student performed research on the accuracy of the point cloud data used to develop the three-dimensional terrain models. The second student performed sensitivity analysis on the impact of changing the spatial resolution of the data collected on the spatial accuracy of the models, the storage required for the terrain models, and the processing time required to analyze the terrain models.

The students collected a set of baseline data on the campus of Randolph-Macon College using a test site with a set of fixed, premeasured features (Figure 1.). The test site has an area of 6,142.08 m^2. The images for each data set were processed using the Pix4D point cloud generation tool. The point cloud and terrain maps generated were compared to the known terrain data. The baseline accuracy data was collected at approximately one data point per square centimeter, or 10,000 data points per square meter. An average of seventy high-resolution images was used to create each DEM. The DEM's contained an average of 61.4 million data elements. The size of the DEM's varied based on the spatial resolution of the DEM. Each student performed a set of system validation flights prior to collecting research data. The flights allowed the students to gain operational flight experience with the UAV and experience with the point cloud extraction tools.

The students collected nine data sets (four accuracy data sets and five sensitivity data sets) at the test site. The UAV flew a grid pattern (Figure 2.) to ensure full coverage of the terrain being analyzed. The students implemented algorithms [5] [24] to: correctly translate the raw data into matrices of correctly aligned (in both the x and y axis) altitudes; calculate the error between each point in the data sets; and determine the location and size of the premeasured features.

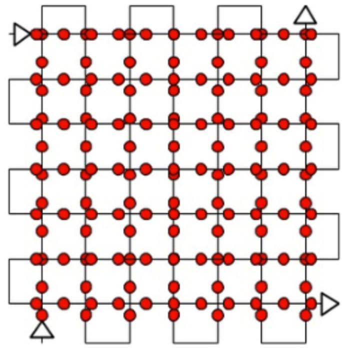


Figure 2: Data Collection Pattern

The accuracy analysis [5] measured the differences between four accuracy data sets collected at the test area. The data was collected on four different dates. The statistical analysis of the surface difference matrix between each DEM and the other accuracy DEM's showed a median spatial accuracy error of 0.0 cm and a standard deviation of 6.0 cm in the best case and 12.9 cm in the worst case. The maximum error occurred at the points along the edge of the roof of the building (Figure 1). Some points along the very edge of the roof were captured as a part of the roof (approximate altitude of 3.0 m) in one data set and at ground level (0.0 m) in the comparison data set. The DEM construction algorithms have been revised to correct the source of the alignment anomalies.

The sensitivity analysis of the impact of changes in spatial resolution on the spatial accuracy of three-dimensional models was performed [24]. The sensitivity analysis focused on how changes in the altitude used to collect the point cloud data (Table 2) effect the DEM's and the surface difference maps produced from the stereographic images. Collecting data at higher altitudes results in reduced spatial resolution (each point represents a larger area) in the point cloud data. Data was collected at several altitude / spatial resolution pairs (Table 1) to evaluate the impact of spatial resolution on the accuracy of the resulting DEM's and surface difference models. Preliminary sensitivity analy-

Table 2: Altitude Sensitivity Analysis

Altitude (in ft)	Median (in cm)	Processing (in second)	Percentage Decrease
45	0.50	416.657	0.00
60	0.50	183.115	56.05
75	0.00	118.249	71.62
90	0.50	118.903	71.46
105	1.00	87.072	79.10

sis was performed to determine the spatial resolution that produces the most accurate DEM's while reducing processing time and storage requirements. The sensitivity analysis showed that flights flown at 75 ft had the smallest overall percentage of error between the data points collected and the actual terrain and had an acceptable processing time.

3.2 Terrain Modeling Algorithm Updates, Fall 2018, and Spring 2019

The first generation point cloud processing and analysis algorithms [5] [24] were refined during the fall of 2018 and spring of 2019. Revised analysis algorithms were incorporated into the point cloud processing software. The surface difference matrix generation software and data visualization software [5] [24] was revised. The algorithms and data structures were optimized for efficient storage and analysis of high volumes of data. The second set of point cloud data (eighteen addition data sets) was collected at the same location. The images were processed with the Pix4D point cloud extraction tools. The point cloud data was compared to the known terrain data and to the point cloud data from the first set of validation flights.

The accuracy analysis was rerun after each set of software revisions. Eight additional accuracy data sets were collected on four different days, each with different weather conditions. The results of the accuracy analysis, measuring the error at the corners of the reference building, are shown in (Table 3). The maximum error was 1 cm, the median difference was 0.00 cm, with a standard deviation of 0.48 cm.

Each of the twelve accuracy data sets (the four student data sets and the eight additional data sets) was compared to itself and to each of the other DEM's. This produced seventy-eight unique surface difference matrices. The statistical analysis of the points in each DEM compared with their respective point in the other DEM's show a median spatial accuracy error of 0.00 cm with a standard deviation of 0.10 cm across all seventy-eight DEM difference matrices analyzed.

The sensitivity analysis of the accuracy of the model at differing altitudes and spatial resolutions was repeated to determine a set of efficient altitude / spatial resolution pairs to use for collecting data. A total of fifteen DEM's were analyzed. A student collected one set of DEM's (five flights, one at each altitude) in the fall of 2017. Two additional sensitivity data sets (ten flights, two at each altitude) were collected in the fall of 2018 and spring of 2019. The results of the altitude sensitivity analysis are shown in Table 2. An altitude of 75 feet produced the lowest median error. Above 75 feet the pixel size in the image is over 1 cm ^{2} and there is a slight increase in the average error.

Each of the five DEM's (one for each altitude) in the three sensitivity data

Table 3: Revised Model Accuracy Data

Data Set	Corner 1	Corner 2	Corner 3	Corner 4
1	0.00	-1.00	-1.00	0.00
2	0.00	0.00	-1.00	0.00
3	0.00	0.00	0.00	1.00
4	0.00	0.00	-1.00	0.00
5	-1.00	-1.00	0.00	0.00
6	0.00	0.00	0.00	1.00
7	0.00	0.00	0.00	0.00
8	-1.00	0.00	0.00	0.00
9	0.00	0.00	0.00	1.00
10	0.00	0.00	0.00	0.00
11	0.00	-1.00	0.00	0.00
12	-1.00	0.00	0.00	0.00

sets was compared to the corresponding DEM (same altitude) in the other two data sets and to itself. This resulted in a total of six comparisons at each altitude, and a total of thirty comparisons. The statistical analysis of the surface difference models produced by comparing each model with the other data sets at the same altitude showed a median error of 0.00 cm and a standard deviation of 0.12 cm across all data sets collected. The comparison of a DEM with itself should produce a difference matrix containing only zeros and None values. The processing time required to generate and analyze the DEM's showed a leveling off at altitudes above 75 ft.

4 Conclusions

The primary goal of the first phase of the research was to develop and validate the research approach, computation methods, and our experimental procedures. The results demonstrate that low cost commercial UAV's can be used to collect high spatial resolution data with high temporal resolution and high spatial accuracy. The validation data shows that the data can be collected with 1 cm accuracy, and that data from multiple flights can be properly aligned to compute surface difference matrices. The surface difference matrices allow a point by point comparison of each data element in two matrices to determine changes in the terrain.

The research and educational goals of this research program are tightly integrated and provide students with a cohesive experience that stresses the core theoretical concepts of computer science in the classroom and provides extensive research opportunities as a part of the Senior Capstone Experience and the Schapiro Undergraduate Research Fellowship summer research program. The joint Computer Science / Environmental Studies research program demonstrates computer science's role as a "hub" or foundational discipline. The joint research project is focused on expanding the number of students from other disciplines that gain a basic understanding of computational methods and computer science to support research in their chosen discipline.

The low cost of the research system and the ability to deploy multiple data collection platforms in parallel demonstrates the challenges of coordinating the collection and processing of the research data. The multiphase research program allows students to apply lessons learned in earlier phases to revise the point cloud extraction, processing, and visualization techniques.

4.1 Future Directions

Following the successful completion of the data collection and model validation, the PI will continue research in the use of UAV's to gather high temporal and spatial resolution data. Paired with the algorithms and visualization tools developed in the research program the UAV's provide a lower cost, lower risk method for collecting terrain change data. The remote sensing platforms can be rapidly deployed to collect data after large storms or natural disasters. The next phase in the development will apply cooperative problem-solving techniques [13] [14] to develop an intelligent flight management system [12] to manage flight planning for the data collection flights. The application of cooperative problem-solving techniques will allow a set of UAV's to autonomously map a predefined field of interest.

The computational and quantitative methods will be used to validate a recently developed mathematical model used to quantify the impact of successive storms on sandy beaches [6]. The data collected will create a visual representation of the impact of multiple storms and the attendant physical changes that occur to the physical and ecological systems of beaches and barrier islands. The high-resolution models facilitate a comprehension of shoreline changes across a large population of stakeholders that is accessible from anywhere in the world.

Surface difference maps from pre- and post-storm UAV surveys will allow a comparison of modeled erosion potential of successive storms to actual erosion, enabling a re-parameterization of the theoretical model to fit the measured impact. Once validated, the theoretical model will be applicable to sandy beaches around the world and provide predictions of cumulative storm impacts to any beach. This information is especially important given recent research studies which have shown increases in the magnitude of large coastal storms.

5 Acknowledgments

The author thanks Randolph-Macon College for supporting this research.

References

[1] J. C. Allan and P. D. Komar. Increasing hurricane-generated wave heights along the u.s. east coast and their climate controls. *Journal of Coastal Research*, 24:4–86 – 488, 2008.

[2] Satellite Imaging Corporation. Characterization of satellite remote sensing systems. `http://www.satimagingcorp.com/services/resources/characterization-of-satellite-remote-sensing-systems/`.

[3] Philip J. Costello. Project orion: Investigating and developing a cost-effective modular drone-based sensor package with a raspberry pi. Schapiro undergraduate research fellowship research report, Randolph-Macon College, 2016.

[4] Mark Crowell, Stephen P. Leatherman, and Michael K. Buckley. Historical shoreline change: Error analysis and mapping accuracy. *Journal of Coastal Research*, 7:839 – 852, 1991.

[5] Morgan Davis. Measuring accuracy of 3d models. Senior capstone project, Randolph-Macon College, Ashland, VA, USA, 2017.

[6] Rachel Dominguez and Michael S. Fenster. A new index for quantifying storm impacts on sandy beaches. In *American Geophyscial Union Fall 2017 Meeting*, 2017.

[7] Michael S. Fenster and Robert Dolan. Assessing the impact of tidal inlets on adjacent barrier island shorelines. *Journal of Coastal Research*, 1:294 – 213, 1996.

[8] Benjamin T. Gutierrez, S. Jeffress Williams, and E. Robert Thieler. Potential for shoreline changes due to sea-level rise along the u.s. mid-atlantic region. Technical report, U.S. Department of the Interior, U.S. Geological Survey, Reston, Virginia, 2007.

[9] G. J. Holland and P. J. Webster. Heightened tropical cyclone activity in the north atlantic: natural variability climate trend? *The Royal Society*, pages 2695 – 2716, 2007.

[10] Daniel Ierodiaconou, Alexandre C.G. Schimel, and David M. Kennedy. A new perspective of storm bite on sandy beaches using unmanned aerial vehicles. *Zeitschrift für Geomorphologie*, 60:123 – 137, 2016.

[11] Eric Lambin and Jules Wilmet. Land-cover changes in west africa: Multitemporal change-vector analysis at coarse scale and change processes categorisation with high spatial resolution data. Technical report, Laboratoire de Télédétection et d'Analyse Régionale, Louvain-la-Neuve, 1996.

[12] John W. McManus and Kenneth H. Goodrich. Application of artificial intelligence (ai) programming techniques to tactical guidance for fighter aircraft. In *Proceedings AIAA Guidance, Navigation, and Control Conference*. AIAA, 1989.

[13] John William McManus. *Design and Analysis Techniques for Concurrent Blackboard Systems*. PhD thesis, College of William and Mary, Williamsburg, VA, USA, 5 1992.

[14] John William McManus and William Lee Bynum. Design and analysis techniques for concurrent blackboard systems. *IEEE Transactions on Man, Machine, and Cybernetics*, pages 669 – 6807, 1996.

[15] Francisco-Javier Mesas-Carrascosa, Jorge Torres-Sánchez, Inmaculada Clavero-Rumbao, Alfonso García-Ferrer, Jose-Manuel Peña, Irene Borra-Serrano, , and Francisca López-Granados. Assessing optimal flight parameters for generating acurate multispectral orthomasicks by uav to support site specific crop management. *Remote Sensing*, pages 12793 – 12418, 2015.

[16] Robin Murphy, Brittany Duncan, Tyler Collins, Justin Kendrick, Patrick Lohman, Tamara Palmer, and Frank Sanborn. Use of a small unmanned aerial system for the sr-530 mudslide incident near oso, washington. *Journal of Field Robotics*, 2016.

[17] Stephanie H. Nebel, Arthur C. Trembanis, and Donald C. Barber. Shoreline analysis and barrier island dynamics: Decadal scale patterns from cedar island, virginia. *Journal of Coastal Research*, 28:332 – 341, 2010.

[18] Pix4D. Pix4d mapping solutions. `https://pix4d.com/industry/mapping/`.

[19] Fabio Remondino, Maria Grazia Spera, Erica Nocerino, Fabio Menna, Francesco Nex, , and Sara Gonizzi-Barsanti. Dense image matching: comparisons and analyses. In *Proceedings Digital Heritage International Congress*, 2013.

[20] Tomi Rosnell and Eija Honkavaara. Point cloud generation from aerial image data acquired by a quadrocopter type micro unmanned aerial vehicle and a digital still camera. *Sensors*, pages 453 – 480, 2012.

[21] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, pages 7 – 42, 2002.

[22] Steven Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. Comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, EICS '09, 2006.

[23] Skycatch. How drones are building a better construction industry. *Construction Global*, pages 14 – 21, November 2016.

[24] Dave Thomas. Sensitivity analysis of the impact of altitude changes on the percentage error. Senior capstone project, Randolph-Macon College, Ashland, VA, USA, 2017.

[25] Charles Toth, Grzegorz Jozkow, , and Dorota Grejner-Brzezinska. Mapping with small uas: A point cloud accuracy assessment. *Journal of Applied Geodesy*, pages 213 – 226, 2015.

[26] Arthur C. Trembanis. Personal communication. unpublished Correspondence, 2016.

Increasing Minority Youths' Participation in Computing through Near-Peer Mentorship*

Michael J. Lee
Department of Informatics
New Jersey Institute of Technology
Newark, NJ 07102
mjlee@njit.edu

Abstract

It is critical to focus on diversity and increasing participation of underrepresented groups in computing. To address this need, we must better understand minorities' access to role models and mentors, especially at a young age, as research and practice shows that these relationships can affect students' self-efficacy and motivation in the educational fields and careers they choose to pursue. We provided a 9-Saturday programming camp to middle school students in Newark, New Jersey with near-peer mentors (first year, college student instructors) to learn more about the younger students' initial access to role-models and mentors, and how an intervention might change this. Our camp served a total of 28 minority students (17 males and 11 females; grades 5-7) from a low-income, urban area. We found that when asked at the beginning of the camp, our middle students largely reported that they did not have any role-models or mentors in computing. However, at the conclusion of the camp, these same students indicated that they developed strong connections with their near-peer mentors and even saw them as role-models. These findings highlight the need for more mentorship opportunities for students of all ages, and the importance of providing resources and support to help develop and nurture these connections.

1 Introduction

Computing jobs continue to be in significant demand across the world [23], with programming jobs being among the fastest growing career areas [4]. Unfortunately however, many youth—especially those from underrepresented minority groups—are not pursuing related educational opportunities or careers [23]. Without adequate representation from all groups, we run the risk of severe inequities and bias in software design and use [24]. Therefore, we must investigate new learning opportunities that both engage a wider range of the population and sustain their learning and engagement over time so that we can better address the lack of diversity and low supply of technology workers.

This project attempts to address inequities in computing by learning more about low-income, minority, middle school students' access to mentors and role-models in computing, and how a 9-Saturday programming camp with university student instructors might affect this. We surveyed middle school students, before and after they were exposed to programming with a debugging game, website design curriculum, and block programming environment. We specifically chose to focus on middle school students, as research has shown that this is the age when youth begin to form strong, lasting ideas about their future careers [6, 8, 10]. Our goal was to provide underrepresented minority students with a positive experience with programming along with similarly-aged instructors, to develop productive mentee-mentor relationships, and produce evidence that these types of activities yield benefits beyond learning how to code.

2 Related Work

2.1 Middle Schools

Although many computing related enrichment activities target high school students, research has shown that youth begin to form ideas about future careers earlier, during their formative years in middle school [6, 8, 10]. Many educational research efforts targeting younger children have focused on how to engage them in Computer Science (CS) by using programming environments that are user-friendly and have the potential to lower the cognitive threshold for novice programmers, such as Scratch (e.g., [21]), Alice (e.g., [13, 14]), Gidget [17, 18, 20], and others [9]. Overall, studies repeatedly found that these types of environments are effective in the acquisition of basic programming skills and concepts (e.g., [16, 20, 21]) and support computational thinking. For example, Meerbaum-Salant et al. [22] found that Scratch helped middle schoolers learn most of the targeted CS concepts. Lee & Ko [20] found similar results with Gidget, where users showed significant, measurable learning gains for targeted CS concepts. In our study, we aim to learn more specifically about underrepresented minority middle school students and how a coding camp using many of these tools with similarly-aged instructors might affect their views about access to computing mentors and role-models.

2.2 Near-Peer Mentorship

Bandura [2] wrote that "seeing or visualizing people similar to oneself perform successfully, typically raises self-efficacy beliefs in observers that they themselves possess the capabilities to master comparable activities." However, if the youth perceives the model to be dissimilar or have disparities in experience, they "are likely to view skills exemplified by an experienced model as beyond their reach and are therefore disinclined to invest the effort needed to master them fully" [1]. Youth are therefore more likely to imitate those who they perceive as similar to themselves [2]. Denner [6] and D'Souza et al. [7] found that role-models are an important motivator for middle school girls' and high school students' interest in computing, respectively. Clarke-Midura et al. [5] found that middle school students relate particularly well to near-peer mentors—mentors that were only a few years older than the students and not necessarily the most skilled with CS—increasing the younger students' self-efficacy and interest in computer science. Unfortunately, many female and minority students may not have role-models or mentors in computing that they can relate to directly. Based on these observations, we aim to learn more about minority youths' access to role-models and/or mentors, and how this might affect their pursuit of computing-related education and careers.

3 Method

3.1 Participants and Instructors

We ran a 9-Saturday camp at a local K-8 public school in Newark, New Jersey. The school principal recruited students by recommendations from the 6th and 7th grade teachers at her school, who made announcements about the camps in their classes. We did not specifically recruit for underrepresented minorities, as the school's demographic makeup reflects the community it serves, composed of 52% Hispanics, 47% African Americans, and 1% others; 82% of these students are low-income—eligible for free or reduced-price lunch.

For the camp, we had access to one classroom, the cafeteria, equipment (e.g., laptops with WiFi access), middle school teachers (one male and two female technology and mathematics teachers), and first-year undergraduate student instructors. The instructors ran all the events and taught the middle school students, while the teachers were available primarily for classroom support (e.g., to help keep the children on task) and did not provide any computing-related instruction. The day was split into two parts, a morning session and an afternoon session. The camp included 11 instructors (4 females and 1 male in the morning session; 1 female and 5 males in the afternoon session). The instructors all took a university shuttle to and from the middle school, and all of the middle school students were dropped off and picked up by their respective parents/guardians.

3.2 Procedure

Each of our nine Saturdays consisted of a seven-hour day including breakfast, a morning programming activity, lunch, and an afternoon (block) programming activity. Breakfast was 15 minutes and lunch was 45 minutes. On the first day before starting any programming activities, students filled out a pre-test questionnaire. On the (ninth) final day, students filled out a post-test questionnaire before concluding the camp. Each of the questionnaires took 20 minutes to complete. The game/HTML (morning) and block programming (afternoon) activities lasted approximately three hours each, every Saturday.

3.2.1 Morning Session

Since one's first experience with code is important [15], we chose two different activities for the morning sessions. On the first day, students played *Gidget* [20] (www.helpgidget.org)—a free, online, educational programming game—which has been shown to be engaging for a wide range of programming novices and a good introduction to (text-based) programming for after-school programs [11, 19]. We encouraged our middle school students to complete as many levels as they could during the first day, and allowed them to continue playing the game from home and during any free time on subsequent camp days. For all other days, we had students complete code.org's *CS Discoveries*' curriculum, which was designed for and used by thousands of middle school students. This curriculum focuses on teaching students about HTML, CSS, best practices for posting on the internet, and ultimately for them to create a personal website.

3.2.2 Afternoon Session

We used Harvard University's *Creative Computing for Scratch* curriculum [3] in our afternoon sessions to give the middle school students experience with a block programming language. Scratch has been used extensively to engage youth with block programming through animation authoring and storytelling [21, 22]. This curriculum focused on teaching students introductory programming concepts and to ultimately create a personal music video.

3.2.3 Guest Speakers

Every even week (i.e., Weeks 2, 4, 6, and 8), a guest speaker came to talk about their computing-related job for 30 minutes after lunch. All speakers were recruited to have grown up, live(d), and work in the local metropolitan area. We had a total of four speakers, all African American/Black professionals (three males and one female) who worked in game/app development, computer security, disaster response, or research, respectively. The guests spoke about their background, education, and career, giving examples of their responsibilities (and sometimes demoing some of their tools or products). After the guest finished speaking, students asked them questions for 10-15 minutes.

3.2.4 Pre- and Post- Test Questionnaires

The middle school students completed a pre-test questionnaire before the start of the camp, and a post-test questionnaire as the final activity on the last day of the camp. The questionnaires consisted of three questions asking students if they wanted to go to college, and whether they would attend more classes *at* school or *outside* of school to learn more about programming. Next, we asked them the following questions about mentorship and role-models:

1. *Do you know of anyone famous related to computing? If so, who?*
2. *Do you have a role-model or role-models related to computing? If so, who?*
3. *Do you know anyone personally who has a computing related job? If so, who?*
4. *Have you ever had a mentor for anything related to computing? If so, who?*

Then, only for the pre-test (since we did not expect responses to change for the duration of the camp), we collected demographic information (i.e., grade level, ethnicity, gender, and eligibility for free/reduced lunch), and asked four yes/no questions examining if they: owned a smartphone, had prior programming experience, and had access to a computer and/or internet at home (since the availability of computing resources has been reported to be significantly related to household income and inversely affected by minority status [12]).

4 Results

It took the middle school students approximately 20 minutes to complete each of the pre- and post- test questionnaires. In total, we had 28 students ($n = 28$) participate in our event (10-13 years old). The camp included 17 boys and 11 girls (one 5th grader, twelve 6th graders and fifteen 7th graders; see Figure 1). All of the students were underrepresented minorities in STEM, identifying as either African American/Black (13 students) or Hispanic/Latino (15 students), and all were eligible for free/reduced cost lunch.

Figure 1: Participants and near-peer mentor instructors interacting in our camp.

All of our 11 instructors were first year college students (5 females and 6 males; aged 18-19 years old) from a local university, majoring in various fields including computer science, biology, informatics, and history. They also identified with a wide range of ethnicities, including South Asian (3 students), White/Caucasian (2 students), Pacific-Islander (2 students), Hispanic/Latino (3 students), and Middle-Eastern (1 student). Without any prompting, each group of instructors selected a lead instructor for the morning session and afternoon session (both were female students), respectively.

When applicable, we use nonparametric Chi-Squared likelihood ratio tests with $\alpha = 0.01$ confidence throughout our analyses—as our data was nominal and not normally distributed—to compare participants' responses. We report our statistically significant results with the understanding that our sample size is small and that the resulting findings may not be widely generalizable. For all questions, we ran comparisons between demographics (grade level, ethnicity, or gender) and questionnaire (pre-test vs. post-test). Unless indicated otherwise, there were no significant differences between responses by demographics within or between the pre-test or post-test for the results described below.

4.1 Questionnaire Results

All students indicated that they wanted to attend college in both the pre- and post- test questionnaires. Similarly, all students indicated that they would be willing to take more computing-related courses *at* school if available. As there was no difference or change in responses for either of these questions, there were no significant differences between responses by demographics (grade level, ethnicity, or gender) within or between the pre-test or post-test questions.

Two students (both African American/Black, female students) indicated in their pre-test that they would *not* want to take a computing-related courses *outside* of school. However, their responses changed for the post-test, indicating they *would* take a computing-related courses *outside* of school.

4.1.1 Mentorshop & Role-Model Results

For our question, *Do you know of anyone famous related to computing? If so, who?,* in the pre-test, 21 students (75%) said 'no' or left it blank. The remaining 7 students (25%) put either *Steve Jobs* (4 students), *Bill Gates* (2 students), or both (1 student)—both White/Caucasian males. This did not change significantly for the post-test, with 11 students (39.3%) writing *Elon Musk* (2 students), *Steve Jobs* (3 students), *Bill Gates* (2 students), or any combination of these names (4 students)—all famous White/Caucasian males.

Next, for our question, *Do you have a role-model or role-models related to computing? If so, who?,* in the pre-test, 26 of 28 students (92.9%) said 'no' or left it blank. For the two students (one Hispanic/Latino male and one African American/Black male) who indicated that they had a role-model, they mentioned a family member (a cousin or an older brother who was a game developer). There was a statistically significant change in response between the pre- and post- test, with 23 students (82.14%) indicating in the latter test that they had a role-model ($\chi^2(1, N = 56) = 44.083, p < .01$). Students indicated a guest speaker (11 students), a near-peer instructor (9 students), or any combination of these categories (3 students; e.g., one guest speaker and one near-peer mentor, or two near-peer mentors), as their role-model(s).

The results for our next question was similar, *Do you know anyone personally who has a computing related job? If so, who?,* in the pre-test, 26 of 28 students (92.9%) said 'no' or left it blank. The same two students from the previous question mentioned a family member with a computing related job. There was a statistically significant change in response between the pre- and post- test, with all 28 students in the latter test indicating that they knew someone with a computing related job ($\chi^2(1, N = 56) = 62.937, p < .01$). Here, students indicated a family member (2 students), a guest speaker (25 students), or a near-peer instructor (1 student), as someone they knew personally with a computing related job.

For the final question, *Have you ever had a mentor for anything related to computing? If so, who?*, in the pre-test, 24 of 28 students (85.7%) said 'no' or left it blank. The same two students from the previous question mentioned a family member with a computing related job as someone who mentored them, and two additional students (two Hispanic/Latina females) mentioned one of their middle school teachers as a mentor. There was a statistically significant change in response between the pre- and post- test, with *all* students in the latter test indicating that they had a computing mentor ($\chi^2(1, N = 56) = 53.519, p < .01$). Here, students indicated a teacher (1 student), a guest speaker

(1 student), a near-peer instructor (23 students), or any combination of these categories (3 students), as someone that mentored them in computing.

4.1.2 Pre-Test Only Results

For our pre-test only questions, 21 of 28 (75%) students reported that they owned a smartphone. Additionally, 22 of 28 (78.6%) students indicated that they had a computer at home. These same students also had access to the Internet at home. Those 6 of 28 (21.4%) who stated not having a computer at home also did not have Internet access at home. 7 of the 28 (25%) students reported they had some previous programming experience (all using Scratch).

5 Discussion & Conclusion

In this study, we found that nearly all of our minority middle school students initially reported that they did not have any role-models or mentors in computing prior to participating in our coding camp. Moreover, prior to participating in the camp, the majority of these students (26 of 28) did not know anyone who had a computing related job, and most (21 of 28 students) could not readily identify any famous people related to computing (and for those that did, these celebrities were all White/Caucasian males, which may not be helpful for minority students seeing themselves in these leadership positions [2]). These are serious issues, as youth are at a major disadvantage in pursuing education/careers in computing not knowing anyone with computing related jobs, and not having role-models or mentors that they can look up to, relate to, or talk to.

However, we also found that a 9-Saturday coding camp—emphasizing instruction by near-peer mentors and including guest speakers from the local metropolitan area who the students could better relate and identify with [2]—significantly increased *all students* reporting of having role-models and mentors by the conclusion of the camp. Helping youth, especially minority youth, identify these types of individuals early in their academic experience is essential in keeping them engaged with topics such as computer science [6, 8, 10] and may ultimately contribute to increasing diversity and representation in computing careers [24]. These findings have important implications, demonstrating that relatively short interventions such as after-school programs can provide lasting positive outcomes on underrepresented minority youth.

Some limitations of this work include the representativenses and generalizability of our sample, our instructor demographics, and the quantitative nature of our data. We believe that our sample is representative of the community we aim to serve, as our students were all underrepresented minorities in computing. Moreover, we are confident that our results can generalize to other similar contexts focusing on increasing engagement and providing mentorship for minority youth since we had such strong effects. Next, our instructors represented a wide range of ethnicities and academic interests, and were close in age with the middle school students. We found that our younger students overwhelmingly reported that they viewed these college students as computing mentors, suggesting a good fit between mentors and mentees. Finally, we only collected quantitative survey data from the participants, and only from the perspective of the younger students. Collecting qualitative data, through interviews or focus groups, could give us a richer picture of both the middle school and college students' views and feelings about the nature of their mentor-mentee relationships, and what it means to them. We plan to collect this type of data, from both middle school and college students' perspectives, in our future camps.

6 Acknowledgements

We thank the Newark Public Schools and the Urban League of Essex County for their contributions and participation. This work was supported in part by the National Science Foundation (NSF) under grants DRL-1837489 and IIS-1657160. Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the views of the NSF or other parties.

References

[1] Albert Bandura. Self-efficacy: toward a unifying theory of behavioral change. *Psychological review*, 84(2):191, 1977.

[2] Albert Bandura and Richard H Walters. *Social learning theory*, volume 1. Prentice-hall Englewood Cliffs, NJ, 1977.

[3] Karen Brennan, Michelle Chung, and Jeff Hawson. Scratch curriculum guide: A design-based introduction to computational thinking with scratch, 2011.

[4] Anthony P Carnevale, Nicole Smith, and Michelle Melton. Stem: Science technology engineering mathematics. *Georgetown University Center on Education and the Workforce*, 2011.

[5] Jody Clarke-Midura, Frederick Poole, Katarina Pantic, Megan Hamilton, Chongning Sun, and Vicki Allan. How near peer mentoring affects middle school mentees. In *ACM SIGCSE*, pages 664–669, 2018.

[6] Jill Denner. What predicts middle school girls' interest in computing? *International Journal of Gender, Science and Technology*, 3(1), 2011.

[7] Daryl D'Souza, Margaret Hamilton, James Harland, Peter Muir, Charles Thevathayan, and Cecily Walker. Transforming learning of programming: a mentoring project. In *Australasian Computing Education-Vol.78*, pages 75–84, 2008.

[8] Jacquelynne S Eccles and Rena D Harold. Parent-school involvement during the early adolescent years. *Teachers College Record*, 1993.

[9] Shuchi Grover, Roy Pea, and Stephen Cooper. Factors influencing computer science learning in middle school. In *ACM SIGCSE*, pages 552–557, 2016.

[10] Nancy E Hill and Ming-Te Wang. From middle school to college: Developing aspirations, promoting engagement, and indirect pathways from parenting to post high school enrollment. *Developmental Psychology*, 51(2):224, 2015.

[11] William Jernigan, Amber Horvath, Michael Lee, Margaret Burnett, et al. General principles for a generalized idea garden. *Journal of Visual Languages & Computing*, 39:51–65, 2017.

[12] Yasmin Kafai, Kylie Peppler, and Grace Chiu. High tech programmers in low-income communities: Creating a computer culture in a community technology center. In *Communities and Technologies*, pages 545–563. Springer, 2007.

[13] Caitlin Kelleher and Randy Pausch. Using storytelling to motivate programming. *Communications of the ACM*, 50(7):58–64, 2007.

[14] Jordana Kerr, Mary Chou, Reilly Ellis, and Caitlin Kelleher. Setting the scene: Scaffolding stories to benefit middle school students learning to program. In *IEEE VL/HCC*, pages 95–98, 2013.

[15] Andrew J Ko. Attitudes and self-efficacy in young adults' computing autobiographies. In *IEEE VL/HCC*, pages 67–74, 2009.

[16] D Midian Kurland and Roy D Pea. Children's mental models of recursive logo programs. *Journal of Educational Computing Research*, 1(2):235–243, 1985.

[17] Michael J Lee. How can a social debugging game effectively teach computer programming concepts? In *ACM ICER*, pages 181–182, 2013.

[18] Michael J Lee. *Teaching and engaging with debugging puzzles*. PhD thesis, 2015.

[19] Michael J Lee. Exploring differences in minority students' attitudes towards computing after a one-day coding workshop. In *ACM ITiCSE*, pages 409–415, 2019.

[20] Michael J Lee and Andrew J Ko. Comparing the effectiveness of online learning approaches on cs1 learning outcomes. In *ACM ICER*, pages 237–246, 2015.

[21] John H Maloney, Kylie Peppler, Yasmin Kafai, Mitchel Resnick, and Natalie Rusk. *Programming by Choice: Urban Youth Learning Programming with Scratch*, volume 40. ACM, 2008.

[22] Orni Meerbaum-Salant, Michal Armoni, and Mordechai Ben-Ari. Learning computer science concepts with scratch. *Computer Science Education*, 23(3):239–264, 2013.

[23] U.S. Bureau of Labor Statistics. Occupational outlook handbook, 2018.

[24] David A Patterson. Restoring the popularity of computer science. *Communications of the ACM*, 48(9):25–28, 2005.

Using the VAST Challenge in Undergraduate CS Research*

Christopher P. Andrews[1], R. Jordan Crouser[2]
[1]Computer Science Department
Middlebury College Middlebury, VT, 05753
candrews@middlebury.edu
[2]Computer Science Department
Smith College, Northampton, MA, 01063
jcrouser@smith.edu

Abstract

The Visual Analytics Science and Technology (VAST) Challenge is a yearly competition designed to push forward visual analytics research through synthetic, yet realistic analytic tasks. In this paper, we discuss the challenges and the successes we have experienced incorporating the VAST Challenge and associated datasets into undergraduate research programs at two liberal arts colleges. We advocate for increased undergraduate participation in this and similar competitions, arguing they afford unique opportunities for positive development in early researchers.

1 Introduction

There have been many benefits ascribed to undergraduate research (UR) programs, including enhanced domain knowledge and skills, honing of analytic skills, increased interest and retention, greater self-confidence, gains in critical thinking, and validation of post-graduation choices [14, 7, 9]. While Seymour et al. found that some of these may be more anecdotal than rigorously demonstrated [14], there is a general agreement that students benefit particularly from the apprentice model of research that is carried out in many small undergraduate institutions. In addition, for these gains to be realized, students must have **legitimate participation** in the research – they must be engaged, making decisions, and have personal stake in the research program rather than simply following directives [7].

Few undergraduate computer science (CS) programs offer (and even fewer require) courses in research methods, and many students begin their study of CS at the undergraduate level. As a result, students just starting out on a research project frequently lack foundational research skills, the remediation of which can consume valuable time [6, 17]. There is a need to balance getting students' technical abilities to a point where they can pursue a meaningful research question and ensuring that sure they have a project in which they can take some ownership, while simultaneously continuing to advance faculty research agendas at a reasonable pace.

While multi-year projects that establish continuity and pull in new students who can grow to be contributors have been proposed as the ideal [17], the lure of many compelling internship opportunities has made recruiting and retaining CS students for multiple years increasingly difficult. Hadfield and Schweitzer suggest addressing these issues by distilling the research into small (but non-trivial) problems to promote a sense of ownership within each student, with the faculty member providing mentorship over the larger process [6]. Koeller suggests that, "the most efficient use of undergraduate resources is in implementation," which should be accessible to all students who have progressed beyond data structures [8]. While these suggestions converge to a consistent message, practical and reproducible models which can be readily adapted by new faculty at primarily-undergraduate institutions (PUIs) are sorely lacking.

In this paper, we discuss our experiences using the annual IEEE Visual Analytics Science and Technology (VAST) Challenge [3] as a component of undergraduate research at two PUIs, and how it has helped us to address some of these challenges. We hope that this paper will serve as a resource for others in the VIS community wishing to increase participation of undergraduate students in their research programs, and that participation of undergraduate students in the VAST Challenge will continue to grow.

2 The VAST Challenge

The VAST Challenge is an annual competition in which visual analytics researchers, software developers, and designers pit their best tools and techniques against realistic analysis tasks. It has been held annually since 2006, providing synthetic tasks and datasets that mirror the real world in both scale and complexity. The data, tasks, and selected solutions are made publicly available through the Visual Analytics Benchmark Repository[1].

[1] `https://www.cs.umd.edu/hcil/varepository`

2.1 Origin of the Challenge

The field of visual analytics has evolved into a complex hybrid of information visualization, interaction design, machine learning, and human psychology. Due to the dynamic interplay between human and machine analysis, standard benchmarking tasks and traditional measures of success have limited utility. The need for new evaluation mechanisms and realistic benchmark tasks provided the motivation for the challenge [3]. The challenge also provides researchers who would otherwise have no access to practicing analysts or data that is frequently sensitive or confidential with realistic tasks and datasets.

2.2 Data and Mechanics

The challenge typically consists of a collection of "mini-challenges" and an overarching "grand challenge" that ties the challenges together. Submissions are reviewed by the committee, professional analysts, and peer researchers. Each competing team has the opportunity to present a poster, participate in the challenge workshop held at the VAST conference, and submit a short two-page paper for inclusion in the workshop proceedings.

Tasks have included piecing together stories from text fragments (e.g., news articles, intelligence reports, and social media), finding patterns in GPS traces and sensor logs, and analyzing satellite images, video and sound recordings. Frequently, the challenges make use of heterogeneous data sets, requiring analysis that combines and relates different data sources and types.

Each challenge also comes wrapped in an accompanying story. Over the years, participants have had to contend with bio-terrorists, exotic animal smugglers, arms dealers, corporate hijinks, and polluters (to name just a few). The background stories serve as motivation to find the solution, as well as making the analysis fun and engaging [10, 13]. Finally, a key feature of each of the challenge data sets is that each one has an **embedded ground truth**, which allows for comparison between solutions, and motivates participants by indicating that a solution exists and that it is achievable by researchers and students [5].

2.3 Other Uses of the VAST Challenge Data

Due to the availability of the datasets through the benchmark repository, participation in the challenge is not required to make use of the data, and it is used fairly extensively [11]. Companies report using the data to test their analytic tools [13, 10], and researchers use the data sets to explore new analytic techniques and attempt to better understand the analytic process [5].

The challenge has also been used in a number of university courses as a source of projects [19]. Rohdantz et al. report that the novelty of the scenarios “requires students to synthesize their learned knowledge to create novel approaches”. The students are reported to overwhelmingly find the scenarios to be engaging and useful in identifying requirements and evaluating the analytic tools they developed in the classes [12].

3 Participating in the Challenge

While the VAST Challenge data sets can be used in a variety of ways to support UR, it can be very rewarding to participate in the actual challenge. The reviewing process provides valuable feedback, and the opportunity to present a poster and write a short paper is a nice entry into conference participation. The students also get the experience of working on the same problems as graduate and industry researchers. However, assembling a submission is not without its own challenges. To date, Middlebury College has sent two official submissions and Smith College has sent one.

3.1 Challenges to Participation

A key challenge is timing. The challenges are typically released late in late April, early May, with submission due early in July, which does not line up well with the academic calendar. The reality for many research programs conducted at PUIs is that we often do not have multi-year student continuity in our research labs, and summer is the main time to engage in research. Thus, the start of summer typically needs to be devoted to onboarding new students who, in addition to being new to research, may also lack a background in many of the skills required to develop a submission. Producing a competitive submission requires a great deal of effort, and it is quite challenging to work toward an early July deadline under these conditions. The deadline roughly bisects the summer, which leaves time for a second project, but one that must be limited in scope based on the schedule.

3.2 Some Alleviating Techniques

Our hope is that our experiences show that these challenges are not insurmountable. One strategy is to try to place useful electives (e.g., information visualization, data science, software development, etc...) in the fall when planning course offerings, to create a pool of students with known skills.

Another technique is to be strategic about which portions of the challenge to undertake. The Smith entry was done by a pair of rising sophomores after only a single computer science course. They focused on the 2016 Mini-Challenge 1, a design challenge that required participants to design an innovative interactive visual interface which could enable security investigators to conduct real-time analysis of multimodal streaming data about a resort property [4]. The completed design, rendered with PowerPoint was not particularly novel in the context of more advanced submissions to the challenge, but the development of a multiple-display, coordinated system was novel *to the students themselves*, and gave them the opportunity to learn valuable skills such as rapid prototyping in the context of interface design.

The two Middlebury entries were for the 2015 Mini-Challenge 1 (MC1), which dealt with patterns of behavior at an amusement park [18], and 2016 Mini-Challenge 2 (MC2), which required participants to analyze environmental sensors and "prox-card" records for the main building of the fictitious GASTech company [4]. In both cases, the solutions were built up as a series of connected visualizations using Middguard, our in-house framework. By pursuing designs consisting of simple linked visualizations and without extensive "computational analytics" required, students could be given ownership over small, self-contained pieces that become useful tools when connected together.

The Middlebury entries could be a little more ambitious both because the students were more advanced (a rising senior on MC1 and a rising senior and a rising junior on MC2), but also because we made use of an apprentice model with a faculty member working alongside the students. Hunter et al. note that "apprentice-style UR fits a theoretical model of learning advanced by constructivism" [7]. This style of research allows the student to make their own meaning through interaction with and support from their faculty mentor. This leads to a number of direct benefits to the students: the students gain new technical knowledge and skills, learn how research is conducted and disseminated in their field, and develop their ability to approach complex problems. In all cases, by the end of the challenge, the students were equal contributors in terms of both implementation and analysis. This model also makes it easier to keep them from losing a lot of time following unpromising paths by steering their efforts. In addition, working alongside a faculty member on the same problem (and one for which the faculty member doesn't already have an answer) is a novel experience for most students — one that seems to give them more of a stake in the process and the outcome.

In the same vein, we have found it useful for the faculty member to start work on the challenge well in advance of the students. This allows time to strategize and pick the most relevant challenge, and to clean up the realistically "messy" data. The heterogeneous nature of the data sets also means that some early work can focus on finding ways to correlate the data. While these are very realistic data problems, solving them ahead of time allows the students to be able to focus on the actual analytic problem.

4 Balancing Education and Research Agendas

A major challenge in supporting UR is in balancing the sometimes orthogonal goals of student education and furthering faculty research agendas. While participating the challenge makes a good, self-contained project that teaches students about one aspect of research, it is more difficult to shape them into a more sustained, productive research agenda, especially in light of the way the challenge bisects the summer research time. In this section we will discuss some of the other ways that we have engaged with the VAST Challenge.

4.1 Onboarding New Undergraduate Researchers

Between 6 and 8 students conduct research in the Human Computation and Visualization Lab at Smith College each year, with a subset of these positions being reserved specifically for undeclared students in their first two years. During the first week of the program, we use the VAST Challenge as an opportunity for these mixed groups of students to work together toward a common goal, flex their programming muscles, and run into roadblocks that require them to investigate unfamiliar techniques. We have found that opportunities to interact with peers within the low stakes / high energy context of the VAST Challenge are useful in helping the adviser to identify strengths and growth areas within the cohort, and working collaboratively can help new students integrate into the lab culture.

4.2 Engaging Students in Open Source Projects

At Middlebury, our first attempt at the challenge was essentially an onboarding exercise – an attempt to both get a new lab up and running and a new undergraduate researcher thinking analytically. While we did not have time to put together an official submission, the difficulties we faced building the system that we envisioned inspired us to build Middguard[2].

Middguard is a web-based framework for building bespoke visual analytics tools requiring a collection of heterogeneous, but connected visualizations – the basic target being the development of tools to solve new, heterogenous data challenges, such as those poised by the VAST Challenges. The framework allows developers to create visualization modules that can be bundled together and deployed to create more complex tools without the need for the individual tools to have knowledge of one another. The framework is responsible for connecting data models to the visualizations, fetching data from the server, and broadcasting selection events to all visible modules so that the views can work together.

[2]https://github.com/AnalyticArtsLab/middguard

In addition to making a contribution to the lab's research agenda, the project became an important part of the education of our lead developer, as he built upon it for both a senior project and senior thesis.

4.2.1 Data Generation

In 2019, Smith partnered with Pacific Northwest National Laboratory (PNNL) to generate a challenge dataset. The 2019 VAST Challenge centers on a natural disaster. A group of 28 volunteer Smith undergraduate students joined 1 Smith professor and 8 PNNL researchers in a role-playing exercise, interacting on a social media feed to generate over 2,100 realistic posts covering the time before, during, and after an earthquake. These exercises are particularly interesting because they flip the **consumption model** frequently adopted in undergraduate data science education, engaging undergraduate students as legitimate co-creators not only of data that might benefit students like themselves, but which would be used to advance the state of the art in visual analytics research. This is a powerful shift as it allowed the students to feel they were making direct contributions to real research.

5 Reflections on student engagement and outcomes

One tangible measure of success for interventions such as these is the generation of research products during the undergraduate research experience. By that metric, participating in the challenge was a success. The Middlebury students have made contributions to multiple short papers [1, 2], and the work on Middguard led to a poster [15], and a thesis [16]. Further, Middlebury won an award for the 2015 submission, indicating that our model could produce a competitive team and that working at the undergraduate level is not in and of itself a barrier to success (two further undergraduate teams have since won awards as well).

Of course, it is also important to consider other metrics in our assessment of the utility of the VAST Challenge in promoting undergraduate scholarship. A commonly stated motivation for supporting undergraduate research is to increase the pipeline to graduate school [7, 14]. However, while enrollment in computer science courses is healthy and the number of majors continues to rise, the current economic climate means that only a small fraction of computer science majors go on to attend graduate school immediately after completing their undergraduate degree. Indeed, none of the students described above were on a path toward doctoral programs in CS, and their experience working on the challenges did not change that. Instead, the student who have graduated have gone on to successful positions in industry (at Google, Morgan Stanley, and Epic, to name a few).

Of course, we mentioned a number of less tangible effects attributed to UR. Our sample size is too small to replicate any of the earlier results, however, our informal conversations with the students afterward revealed signs of many of these. One student reported, "I think I gained a weird sense of confidence through this challenge. The comments are very informative and sincere so they oddly encouraged me to try new problems and take on challenges that were beyond my scope."

As for the challenge itself, a common sentiment was that it "is very fun because the challenge simulates real world problems" and the realism inspired "a sense of interest and excitement". They talked about all of the important skills they gained from visualization and design, to working with databases and servers, and working on existing codebases. The students also appreciated the different perspective on visualization they gained, with one describing how it encouraged her to be more mindful of users: "visualization requires you to be an empathetic problem solver with an eye for design".

6 Conclusion

Our goal in writing this paper is to encourage the use of the VAST Challenge and its datasets in UR. While principally designed to engage researchers and developers at research institutions and businesses, the problems are within reach for undergraduates. The scenarios have many compelling features: they model real analytic problems, they are sufficiently complex so that they can't be solved by in a couple of hours, they have a ground truth so results can be evaluated, and the scenarios themselves are engaging and fun.

The diversity of the data also provides a number of different challenges and could provide interesting material for research outside of the visual analytics domain. While the challenges are obviously designed to push forward research in visual analytics, we feel that engagement with the challenge datasets need not be limited to visual analytics or even data science researchers. The data could also be fodder for undergraduate projects in a number of different areas within computer science such as natural language processing, machine learning, and image processing.

We have laid out some of the potential benefits and challenges of participating the the challenge itself, as well as discussing some of the other ways we have incorporated the datasets into our research. We would also like to encourage more undergraduate teams, especially from PUIs, to participate in the challenge itself[3]. An ideal outcome would be the development of a community of faculty members working to bring visual analytics research into the undergraduate sphere.

[3] We encourage teams from PUIs to mark their submissions as such

7 Acknowledgments

This work was made possible by undergraduate researchers at Smith College and Middlebury College, in particular the contributions of Dana Silver '16, Julian Billings '16, Lily Taub '17, Shannon Ovitt '18, Ji Won Chung '18, Zheng ("Alice") Mu '19J, Tiffany Xiao '20J, and Ananda Montoly '22. The authors also wish to thank the Smith Undergraduate Research Fellowship Program, the Middlebury Undergraduate Research Office and the Bicentennial Fund for Research Partnerships.

References

[1] C. Andrews and J. Billings. Middguard at DinoFun World. In *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*, pages 129–130, October 2015.

[2] C. Andrews, L. Taub, and S. Ovitt. Middguard at gastech. In *Visual Analytics Science and Technology (VAST), 2016 IEEE Conference on*, Oct 2016.

[3] Kristin Cook, Georges Grinstein, and Mark Whiting. The vast challenge: History, scope, and outcomes, 2014.

[4] R Jordan Crouser, Kristin Cook, John Fallon, Georges Grinstein, Kristin Liggett, Danko Nebesh, Diane Staheli, Mark A Whiting, and Kirsten Whitley. VAST Challenge 2016: Streaming Visual Analytics. Technical report, MIT Lincoln Laboratory, October 2016.

[5] Carsten Görg, Zhicheng Liu, and John Stasko. Reflections on the evolution of the Jigsaw visual analytics system. *Information Visualization*, 13(4):336–345, October 2014.

[6] Steven Hadfield and Dino Schweitzer. Building an undergraduate computer science research experience. pages 1–6. IEEE, October 2009.

[7] Anne-Barrie Hunter, Sandra L. Laursen, and Elaine Seymour. Becoming a scientist: the role of undergraduate research in students' cognitive, personal, and professional development. *Science Education*, 91(1):36–74, January 2007.

[8] Andreas Koeller. Experiences with Student Research at a Primarily Uundergraduate Institution. page 7, 2005.

[9] D. Lopatto. Survey of Undergraduate Research Experiences (SURE): First Findings. *Cell Biology Education*, 3(4):270–277, December 2004.

[10] Pascale Proulx and Casey Canfield. The beneficial role of the VAST Challenges in the evolution of GeoTime and nSpace2. *Information Visualization*, 14(1):3–9, January 2015.

[11] Swetha Reddy, Manas Desai, Catherine Plaisant, and Jean Scholtz. A usage summary of the VAST Challenge Datasets. Technical Report, HCIL, University of Maryland and PNNL, October 2010.

[12] Christian Rohrdantz, Florian Mansmann, Chris North, and Daniel A Keim. Augmenting the educational curriculum with the Visual Analytics Science and Technology Challenge: Opportunities and pitfalls. *Information Visualization*, 13(4):313–325, October 2014.

[13] Jean Scholtz, Catherine Plaisant, Mark Whiting, and Georges Grinstein. Evaluation of visual analytics environments: The road to the Visual Analytics Science and Technology challenge evaluation methodology. *Information Visualization*, 13(4):326–335, October 2014.

[14] Elaine Seymour, Anne-Barrie Hunter, Sandra L Laursen, and Tracee DeAntoni. Establishing the benefits of research experiences for undergraduates in the sciences: First findings from a three-year study. *Science education*, 88(4):493–534, 2004.

[15] Dana Silver. MiddGuard: A Framework for Collaborative and Extensible Visual Analytics. In *Proceedings of CCSCNE-2016*, page Poster #5049, April 2016.

[16] Dana Silver. *Middguard: Collaborative and Extensible Visual Analytics.* PhD thesis, Middlebury College, May 2016.

[17] Elaine Wenderholm. Challenges and the Elements of Success in Undergraduate Research. 36(4):3, 2004.

[18] Mark Whiting, Kristin Cook, Georges Grinstein, John Fallon, Kristen Liggett, Diane Staheli, and Jordan Crouser. VAST Challenge 2015: Mayhem at dinofun world. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 113–118, Chicago, IL, USA, October 2015. IEEE.

[19] Mark A. Whiting, Chris North, Alex Endert, Jean Scholtz, Jereme Haack, Carrie Varley, and Jim Thomas. VAST contest dataset use in education. pages 115–122. IEEE, 2009.

A Basic RISC-V Verilog Datapath Project Experience for Software Engineers*

Dee A. B. Weikle
Computer Science Department
James Madison University
Harrisonburg, Virginia 22807
weikleda@jmu.edu

Abstract

Designing processors for special purpose applications such as accelerators is an important skill in the computing community. Broadening the number of students who are exposed to the basic concepts and skills of high-level hardware design, especially developing an understanding of trade-offs between hardware and software feature implementation, is critical for future success in the field. Most software engineers do not have the opportunity, however, to take a course in modern hardware design. Available courses often focus on simulation using a hardware design language or, if they build a system on a chip using field programmable gate arrays (fpgas), are so extensive that they encompass a whole course. This paper describes an advanced computer architecture course in a computer science department at a medium-size liberal arts university where hardware design is only a portion of the content covered. This course and set of projects use the open source RISC-V architecture enabling students to learn a modern processor architecture forming a basis for potential future research projects without acquiring an expensive license. Even with this basic set of projects, students learned the process of hardware design including an introduction to Verilog, the use of testbench code, and how to synthesize a design to configure a fpga using an industry standard simulation environment, Vivado. The author found three key factors in teaching hardware design to software engineers: 1) that it is important to emphasize the differences between writing in HDL (a high level design language) to generate a circuit and coding in programming languages, 2) to scaffold carefully the introduction to Verilog and simulation, and 3) that configuring the fpga and running code provided additional insight and motivation beyond simple simulation. This paper provides an outline of the course, it's projects and references to additional resources.

1 Introduction

Teaching computer architecture by having students program field programmable gate arrays (fpgas) has become more prevalent as boards and systems improve. Implementing these courses can be expensive in time, equipment cost and space. In particular, if a commercial instruction set architecture (ISA) and software development tools are used, it can be cost prohibitive for many institutions due to licensing fees. Also, these tools and environments are relatively complicated to navigate. Traditionally these courses are in electrical or computer engineering departments and target hardware designers. Examples include EECS 151/251A at Berkeley [23] [19], Universite de Lille [13], the University of Cadiz [12], Korea University, Seoul, Korea [18] and in another detailed reference using MIPS members from the University Complutense of Madrid, Spain, Harvey Mudd College, and the University of Nevada. [11].

At James Madison University (JMU), we have redesigned our curriculum using backward design to include a strong systems core. This curriculum [16] includes options for several upper level systems electives that include large software design projects to apply previously learned concepts in addition to new material. One of these, Advanced Computer Architecture, includes hardware design language and fpga programming. This changes the context from one of introducing computer architecture to one where students will have already been exposed to basic architecture concepts, C programming, circuits, and the X86 instruction set in a previous course. They will also have already learned concurrent programming with both data and threads, including exposure to network communication models. As a result, it is unnecessary to cover everything from a traditional computer architecture course and it is possible to dive deeper into the organization and design using a realistic processor example.

References [13] and [12] use processors developed specifically for education including one that is intended for first-year students. Reference [11] includes excellent materials using the MIPSfpga from Imagination University. The materials and equipment are well designed, but we chose to use the RISC-V architecture and modify the materials in [23] because the processor is the most modern and the materials are open-source making it more cost-effective. Training students to use the RISC-V can enable undergraduate research in computer architecture. Because advanced architecture concepts such as reordering, advanced cache configurations, detailed pipelining with hazard avoidance, and data-level parallel architectures are also being covered, to design a fully pipelined processor proved to be too time-intensive. We found even a simplified subset of the labs in the Berkeley design course [23] and a simplified datapath project of the single-cycle RISC-V gave students significant hands-on experience improving observed engagement and understanding in the initial implementation of the course in Spring 2019. This paper provides a starting point for others who might like to begin using fpgas including references to resources, a description of the current course, and a discussion of the project and lab environment.

2 Related Work

2.1 Courses or Projects in Computer Architecture

As already mentioned, [23] [13] [18] and [11] all use fpga or reconfigurable hardware to teach computer architecture. Reference [23] provides students with a lecture and lab components. The lab provides an application specific integrated circuit (ASIC) option and an fpga option using the PYNQ-Z1 development board. Reference [13] describes building the S3 microprocessor in 12 lab sessions and provides a description of the labs as problem based learning mapping outcomes in Bloom's taxonomy. The MIPS processor is used in both [18] and [11]. [11] provides an excellent set of activities for a whole course from which modules could be chosen.

2.2 Courses or Projects in FPGA design methodology

In [14] the authors use an FPGA configured with a soft-core ARM Cortex-M1 32-bit processor and a custom hybrid RISC/CISC 12-bit processor called VIP. Students explore and compare multiple soft-core implementation issues within an embedded context including software development tools. In [24] authors use the Altium Designer design flow for labs and group projects to teach embedded design with applications in image, digital audio, and video processing. In [21] fpga projects are used to teach system-on-chip design with a common hardware platform across several courses to improve motivation. Emphasis is on making sure students schedule appropriately by breaking projects into smaller tasks and allowing partners. They use automated testbenches and start-up examples.

2.3 Other Resources

References [19] and [15] both teach hardware design languages (HDL) as part of textbooks on computer architecture with some hardware design included. Computer Architecture and Design (COD) [19] includes this material as part of an appendix and has versions for the ARM, MIPs, and RISC-V processors, though the HDL is for MIPs even in the RISC-V version of the textbook. Digital Design and Computer Architecture [15] has a good introductory chapter outlining background for HDLs and several basic examples for combinational and sequential circuits as well as finite state machines using the two most common HDLs: SystemVerilog and VHDL. These textbooks have interactive versions through Zybooks [10] which provides useful activities to reinforce the concepts. Chapter 1 of FPGAs for Software Programmers [17] provides an excellent general overview for understanding the importance of introducing fpgas to software engineers as well as a rationale for when and why they should be used instead of a general purpose or graphics processor. Finally, the 2019 Workshop on Computer Architecture Education (WCAE19) [9] first paper session includes new RISC-V teaching infrastructure.

3 Course Description

As described in [16], this course in Advanced Computer Architecture is taught after students have taken two basic systems courses where they have been introduced to concepts in computer architecture, operating systems, and networking including C programming, basic circuits, the X86 instruction set, caching, concurrent programming and network communication models. This course dives deeper into the content and focuses on applying concepts in large projects where students must decompose, test, schedule and present their work in design reviews and write-ups instead of submitting to an auto-grader. The course learning objectives and content outline are given below.

The course objectives are:

1. Summarize the construction of a pipelined processor
2. Describe and categorize hardware techniques for parallel implementation at the instruction, data, and thread levels.
3. Summarize storage and I/O interfacing techniques.
4. Apply address decoding and memory hierarchy strategies.
5. Evaluate the performance impact of hardware designs, including caches.
6. Describe how hardware implementations can improve system performance.
7. Justify the use of hardware-based optimization where the optimization fails occasionally.

8. Compare and contrast the actual execution of code with software designs.
9. Analyze how a person's logical flow of thinking (sequential) differs from the processor implementation.
10. Demonstrate the ability to communicate design trade-offs.

Table 1: Advanced Computer Architecture Content

Module	Hours	Description
Assembly Language	6	RISC assembly language and decoding
Building a Datapath	6	Logic gates, control unit ALU Construction, register banks, von Neumann implementation
Hardware Descriptor Languages	3	Verilog, VHDL, RTL
Pipelined datapath and hazards	6	Pipelined datapath and control, data hazards (forwarding) vs. stalling, control hazards, exceptions
Memory hierarchy and cache design	6	Quantitative performance measures, cache mapping techniques, cache coherence protocols
Storage and I/O interfacing	4.5	Storage devices, bus protocols I/O performance
Instruction-level parallelism	4.5	Branch prediction, dynamic scheduling
Data-level parallel architectures	3	Vector, SIMD, GPU architectures
Thread-level parallel techniques	3	Hyperthreading, shared-memory multiprocessors

4 Project Descriptions

The three projects for Spring2019 could be done in pairs, although some elected to do the first project alone. The first project was on RISC assembly language and decoding, the second was a small arithmetic logic unit (ALU) and the final project was a single-cycle RISC-V datapath.

4.1 Project 1 - RISC-V Programming

This first project introduced students to RISC-V assembly language programming requiring them to write three different programs and include an analysis of the execution time as well as an explanation of the hardware resources used.

This project had three deliverables: 1) code on GitHub, 2) a write-up explaining the code functionality, and 3) a 10 minute design review in class. The explanations and in-person design review discouraged copying from the internet without learning. The analysis connects the project to other course content and moves the project up in Bloom's Taxonomy for increased student learning. In the design review, it is also possible to provide feedback in a context where multiple students will benefit. Rubrics etc. for these deliverables and those in Projects 2 and 3 are available by emailing the author.

Project 1 encouraged students to use either the RISC-V Venus simulator [22] or the RIPES simulator [20]. In addition to assembling and running RISC-V, RIPES provides a simulation of the execution of instructions in a 5-stage RISC-V pictorial pipeline helping students visualize instruction execution. Project 1 familiarizes students with the RISC-V instructions and capabilities in preparation for the fpga projects.

4.2 FPGA Projects 2 and 3

Projects 2 and 3 were supported by stripped down versions of several labs from the Berkeley course [23] complemented by selections from Xilinx University [1]. Each was reworked to include our system instructions and to remove math concepts not relevant to the learning outcomes of this course while still scaffolding students into the content needed for these simplified projects.

Lab1 was created from Berkeley lab0. Lab2 is based on Xilinx University's first lab in the 2018x workshop materials. Lab3 is based on Berkeley's lab2. Lab4 is a rework of Berkeley Lab3, adding a simplified ALU as specified in COD Appendix A. Lab5 was a rework of the memory parts of Berkeley's lab2. Lab6 is a rework of the UART portions of Berkeley's lab5.

Project 2 had students implement a complete ALU circuit and test it using the buttons and leds on the PYNQ-Z1 board. It had deliverables of 1) Verilog code on GitHub, 2) a lab write-up including timing diagrams and analysis of fpga resources used, and 3) an interactive design review with the instructor.

Project 3 (the final project) had students create the single cycle RISC-V datapath described in Figure 4-17 of COD [19]. Students were allowed to use ROM hard-coded with two different sample programs for the instruction memory and a register file for the data memory rather than implement DRAM. This project also required each team to learn and demonstrate something new that they had not been explicitly taught. Deliverables were the same as in Project 2 with the addition of creating and presenting a poster for a school-wide poster session. At the poster session students were asked to explain trade-offs in their design to other department faculty and students.

Table 2: FPGA Labs Advanced Computer Architecture Spring 2019

Lab	Basic Description	Learning Outcomes
L1	Initial set up, basic logic gates with button interactions	Exposure to stages of fpga design flow, Predicting simple circuit output, Downloading source from GitHub, Establishing personal GitHub account.
L2	Detailed design flow including timing diagrams & schematics, different basic logic gates	Create and save timing diagrams and schematics at synthesis & implementation stages, Mark timing diagrams to demonstrate functional understanding
L3	Building and testing a 14-bit adder	Demo knowledge of behavioral Verilog including basic combinational and sequential circuits using multiplexors. Demonstrate the use of a testbench.
L4	Building and testing a 14-bit simplified ALU	Apply learning outcomes of Lab3 to slightly more difficult concept and without explicit directions, analyze fpga resources used.
L5	Building inferred ROM	Demonstrate another testbench, Decode small RISC-V program to load, Load and verify program in memory.
L6	Serial I/O - UART	Exposure to I/O interfacing

4.3 Project 2 and 3 System Configuration

We chose to use the PYNQ-Z1 from Xilinx [6] primarily because this is the system used by the Berkeley fpga design course with detailed labs and assignments [23] that uses RISC-V and additional resources are available from the XILINX University program. The board is reasonable cost ($230) and comes with an ARM processor, native support for multi-media applications with on-board audio and video interfaces, and extensible with Pmod, Arduino, and Grove peripherals. The PYNQ-Z1 can also be programmed with Python via Jupyter notebooks or used with the XILINX Vivado Design Suite [8]. This enables high-level open-source designs using packaged parts much like software libraries. Details on using the PYNQ-Z1 boards with Python and Jupyter, are in the PYNQ documentation. [5] We had students use HDL to make sure they learned hardware design and the impact of poorly written implementations.

Our set up was for a class of 25-30 students in a basic classroom environment without a dedicated lab space. Setting up the PYNQ-Z1 boards so to use the USB port of the laptop for power allows labs to be run in the classroom interspersed with other class activities. We used Dell laptops with a Linux Mint 19 operating system. We found it necessary to re-image the SD cards that came with the board. We used the Linux version of the Vivado HLx 2018.2 WebPACK, which no longer requires a license for use.[3] We downloaded the board files from [2] 2018x Workshop Materials - Board Files. These files must be placed in /opt/Xilinx/Vivado/2018.2/data/boards/board_files/. We also reinstalled the cable drivers by using the readme.txt in [7].

5 Observations

Several observations from this experience are: 1) It is critical to emphasize the differences between writing in HDL to generate a circuit and coding in programming languages. Specifically, HDLs describe a circuit. All of the commands to generate circuits are executed, laying the circuit out all at once, rather than having code "sequentially" execute. In contrast, testbenches have an element of timing, driving inputs to the circuit being tested in a predefined sequence. This difference must be made explicit and is brought home when students see the actual implementation on the fpga and in schematics. 2) Writing HDL can be seen as a form of parallel programming, providing another opportunity for students to recognize parallel processing in circuit design. 3) Scaffolding the introduction to Verilog and simulation is necessary. Verilog and other HDLs can take years to master. Paring down the experience to a small subset and mapping projects to specific learning outcomes for the course are critical to success in a single semester. In particular it is important to selectively choose the Verilog assignments from resources such as the Berkeley course, the MIPS practical experiences in [11], or in the Verilog tutorial [4]. 4) Configuring the fpga provided additional insight and motivation beyond simulation only. Students had to learn what circuits their Verilog described. Implementation mistakes often meant they could not create a bitstream for the fpga even when it could be simulated. Students were also motivated by the hands-on aspect of making something work, especially when they knew they would have to present to the instructor. 25 students (12 teams) made this reasonable. A larger class would require teaching assistants to complete design reviews.

6 Conclusions and Future Work

In summary, the use of the PYNQ-Z1 was worth the resources used to develop an advanced course in computer architecture. The hardware gave a single hands-on method for students to design several different circuits from simple combinational circuits to a more complicated datapath and serial interfacing.

In future work, the labs will be streamlined and labs that teach how to create finite state machines and a project using caching will be added. Python programming of the PYNQ-Z1 to create a lab experimenting with parallel processing using provided processor IP will also be investigated.

Acknowledgements: Thank you to system administrators Brandon McKean and Pete Morris who provided valuable support in setting up the PYNQ-Z1 systems.

References

[1] Advanced embedded system design using vivado. `http://www.xilinx.com/support/university/vivado/vivado-workshops/Vivado-adv-embedded-design-zynq.htm`.

[2] Board files. `https://www.xilinx.com/support/university/vivado/vivado-workshops/Vivado-adv-embedded-design-zynq.html`.

[3] License. `https://www.xilinx.com/support/answers/42066.html`.

[4] Online verilog tutorial. `http://vol.verilog.com/`.

[5] Pynq: Python productivity for zynq. `http://www.pynq.io/`.

[6] Pynq-z1 board from diligent. `https://store.digilentinc.com/pynq-z1-python-productivity-for-zynq-7000-arm-fpga-soc/`.

[7] Pynq-z1 cable drivers. `/opt/Xilinx/Vivado/2018.2/data/xicom/cable_drivers/lin64/install_script/install_drivers/`.

[8] Vivado webpack. `https://www.xilinx.com/support/download.html`.

[9] Wcae19. `https://people.engr.ncsu.edu/efg/wcae2019.html`.

[10] Zybooks. `https://www.zybooks.com/`.

[11] Daniel Chaver, Yuri Panchul, Enrique Sedano, David M. Harris, Robert Owen, Zubair L. Kakakhel, Bruce Ableidinger, and Sarah L. Harris. Practical experiences based on mipsfpga. In *Proceedings of the 19th Workshop on Computer Architecture Education*, WCAE'17, pages 1–9, New York, NY, USA, 2017. ACM.

[12] Ma De Los Ángeles Cifredo-Chacón, Ángel Quirós-Olozábal, and José María Guerrero-Rodríguez. Computer architecture and fpgas: A learning-by-doing methodology for digital-native students. *Comput. Appl. Eng. Educ.*, 23(3):464–470, May 2015.

[13] Jean-Luc Dekeyser and A. Shadi Aljendi. Adopting new learning strategies for computer architecture in higher education: Case study: Building the s3 microprocessor in 24 hours. In *Proceedings of the Workshop on Computer Architecture Education*, WCAE '15, pages 6:1–6:8, New York, NY, USA, 2015. ACM.

[14] David C. Dyer and Yan Lin Aung. A multi-paradigm approach to teaching students embedded systems design using fpgas and cplds. In *Proceedings of the FPGA World Conference 2014*, FPGAWorld '14, pages 4:1–4:8, New York, NY, USA, 2014. ACM.

[15] David Money Harris and Sarah L. Harris. *Digital Design and Computer Architecture*. Morgan Kaufmann, Cambridge, Massachusetts, 2013.

[16] Michael S. Kirkpatrick, Mohamed Aboutabl, David Bernstein, and Sharon Simmons. Backward design: An integrated approach to a systems curriculum. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, SIGCSE '15, pages 30–35, 2015.

[17] Dirk Koch, Daniel Ziener, and Frank Hannig. *FPGAs for Software Programmers*. Springer, Switzerland, 2016.

[18] Jong Hyuk Lee, Seung Eun Lee, Heon Chang Yu, and Taeweon Suh. Pipelined cpu design with fpga in teaching computer architecture. *IEEE Trans. on Educ.*, 55(3):341–348, August 2012.

[19] David Patterson and John Hennessy. *Computer Organization and Design RISC-V Edition*. Morgan Kaufmann, Cambridge, Massachusetts, 2017.

[20] Morten Petersen. Ripes. `https://github.com/mortbopet/Ripes/`.

[21] Erno Salminen and Timo D. Hämäläinen. Teaching system-on-chip design with fpgas. In *Proceedings of the 10th FPGAworld Conference*, FPGAworld '13, pages 1:1–1:7, New York, NY, USA, 2013. ACM.

[22] Keyhan Vakil. Venus risc-v simulator. `http://www.kvakil.me/venus/`.

[23] John Wawrzynek. Introduction to digital design and integrated circuits. `http://inst.eecs.berkeley.edu/~eecs151/sp19/`.

[24] Markus Weinhardt. Teaching hardware/software codesign on a reconfigurable computing platform. In *Proceedings of the 8th International Conference on Reconfigurable Computing: Architectures, Tools and Applications*, ARC'12, pages 264–275, Berlin, Heidelberg, 2012. Springer-Verlag.

Protecting Personal Information and Data Privacy - What Students Need to Know*

Susan S. Conrad
Information and Cybersecurity Department
Marymount University
Arlington, VA 22207
sconrad@marymount.edu

Abstract

Since the European Union's adoption of the General Data Protection Regulation (GDPR) on June 28th, 2018, organizations around the globe have been required to develop policies and procedures to protect personally identifiable information (PII) or face tough financial sanctions. The GDPR has four overarching components as they relate to the safeguarding of personal information.

The first clarifies that the definition of PII extends far beyond a person's name, social security number, driver's license or other discrete identifier. Instead it can be a combination of multiple personal information vectors that when examined in entirety can be used to identify an individual. The second element of GDPR defines eight rights that an individual can expect regarding the collection, storage and access of one's private information. Three of the most discussed rights include the right to be informed about the collection and use of PII; the right to access what data about an individual has been collected; and the right to be forgotten. The third component identifies a data breaches as a serious offense and has clear guidelines for reporting the breach within 72 hours to country authorities as well those whose data has been breached. Lastly, security can no longer be an afterthought when designing products. GDPR specifies that security must be included in the design phase

of product development. Under GDPR, failure to follow these regulations may result in stiff financial penalties for the perpetrators.

Given the ever-growing abundance of digital personal data, it is imperative to educate students on the sensitivity of PII and the proper methods for storage and safeguarding this information. This paper provides an overview of GDPR and discusses what concepts students need to know in order to comply with these regulations.

1 Introduction

Keeping an individual's data private and safe has taken center stage in the minds of consumers, companies and governments, with statistics as high as 10 Billion records being breached since 2013 [10]. In 2018, The European Union (EU) created the General Data Protection Regulations (GDPR) to strengthen previous laws and give individuals control of their personal data. Use of personal data without the consent of individuals has become headline news with the scandals such as Cambridge Analytica using PII for political purposes; Polar's fitness app revealing the location of military and security personnel; Exactis, a Florida marketing firm, leaving 230 Million survey respondent records on an unsecured server; and the list goes on [8]. Lack of accountability and the never-ending reports of privacy scandals are costing businesses big money and have raised public ire, forcing governments to enact legislation to protect citizens.

The European Commission began crafting data privacy laws as far back as 1995 to encourage and recognize organizations for secure handling of data. In 2003, Japan enacted laws to safeguard personal information being used for business purposes and in 2017 updated the law to include protections for biometric data such as finger prints, DNA and facial recognition [2]. In 2012 the Philippines passed the Data Privacy Act 2012 to protect the use of personal information without owner's consent [11]. These laws were the forerunners to the EU adoption of GDPR, which takes a more comprehensive approach to data security, data collection and personal rights by defining personally identifiable information (PII) and setting standards of care for this data. One month after the passage of GDPR on June 28th, 2018, California enacted the California Consumer Privacy Act (CCPA) which is aimed at protecting consumer rights by strengthening data privacy laws and providing transparency mandating that consumers must consent to the use of their personal data [7].

The concern about data privacy and security is also under discussion in the United States House and Senate where several key initiatives have been introduced to protect consumer data. One comprehensive piece of legislation was introduced November 2018 by Senator Ron Wyden (D-OR) which would significantly expand the role of the Federal Trade Commission to create a Technology Commission charged with the enforcement of laws related to privacy, cybersecurity and Internet of Things (IOT) devices. Implementation of stiff penalties including jail time for executives from companies violating privacy laws, has also been included in other legislation [9].

In order to comply with new GDPR regulations, companies have been forced to invest in data security by updating infrastructure and policies. In a survey of more than 300 companies from the US, UK and Japan, 88 percent reported to have spent more than 1 Million on GDPR preparation and 40 percent reported to have spent more than 10 Million [3]. In addition to data protection, the use of an individual's data has been significantly limited. No longer can organizations collect and sell data without providing easy-to-read user consent forms with information about opting-in or opting-out. Users have the right to know who and where the data is being used under GDPR. This concept termed "data portability" allows users to request a report identifying data stored about themselves in a readable format. Organizations that typically would collect and sell personal data, such as Google, Yahoo, Microsoft, Amazon and Facebook, must under GDPR identify all entities it shares information [5].

Given the importance of protecting personal data and knowing the stiff penalties for organizations which fail to comply with international laws, this paper will provide a high-level overview of key GDPR components and discuss how educators can incorporate GDPR requirements into their teaching.

2 Background

The GDPR legislation takes a broad reach in defining not only who is protected under the law but also what that protection entails. Since the law is uniformly enforced across the 28 EU countries, the impact is global, affecting any organization that does business within any of the countries of the EU. To best understand the impact, students must know who is protected and what data falls under the regulation.

2.1 Who is Protected

The intent of GDPR IS to protect individuals living in any of the 28 countries within the European Union. Three other European countries have also adopted the regulations: Norway, Iceland and Liechtenstein [4]. The law does not differ depending whether you are an EU citizen or non-resident. It protects data of those individuals regardless of where the information is stored or processed . While offering broad protection to the majority of EU citizens, the GDPR does not protect certain individuals under the following circumstances: (1) It does not protect EU citizens who reside outside the EU. (2) It does not protect EU citizens who "avail themselves" to goods or services aimed at people not located in the EU. (3) It does not protect non-human legal entities. (4) It does not protect anonymous individuals. (5) It does not protect deceased individuals. There are some organizations exempt from complying with GDPR and they broadly fall into the category of the courts or organizations concerned with national security [6].

2.2 What Data is Protected

At the very core of GDPR is the intent to protect "personally identifiable information" (PII). The key question here is what is PII? According to the EU Regulation Article 4.1 personal data is: "any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person." [5]

This definition includes non-sensitive information such as an email address or phone number as well as more sensitive information such as genetic, biometric or other information about an individual. PII can be one single data point such as a name or a collection of discrete data elements that create a profile of an individual. This very broad definition of "identifiable natural person" creates a complex web of data which can be considered PII.

One way to look at this concept is by creating two categories of data. The first being single data points which identify an individual and the second being data vectors, connectable data points which when linked with other data can identify a person. Below is a list of data items some of which will directly identify an individual but others that when linked together with multiple data elements will identify, trace or locate a person. GDPR considers these elements to be PII:

(1) Biographical information or current living situation, including names, maiden names, mother's maiden names, aliases, dates of birth, Social Security numbers, passport number, tax identification number, driver's license number, phone numbers, addresses and email addresses.

(2) Looks, appearance and behavior, including photos, fingerprints, handwriting and biometrics such as retina scans, voice signatures, facial geometry, eye color, weight and character traits.

(3) Workplace data and information about education, including salary, tax information and student numbers.

(4) Private and subjective data, including religion, political opinions, blog posts, group membership, travel destinations and geo-tracking data.

(5) Health, sickness and genetics, including medical history, genetic data and information about sick leave.

(6) Information identifying owned property: Assessment id, VIN number, Title number, Log-in details.

(7) Financial information: Bank Account Numbers, Credit scores, Account activity.

GDPR also includes certain device identification as non-PII data protected under GDPR. Recital 30 identifies the following device identifiers for inclusion in this regulation: "Natural persons may be associated with online identifiers provided by their devices, applications, tools and protocols, such as internet protocol addresses, cookie identifiers or other identifiers such as radio frequency identification tags. This may leave traces which, in particular when combined with unique identifiers and other information received by the servers, may be used to create profiles of the natural persons and identify them."[5]

Although the EU has chosen to designate these device identifiers as non-PII, California has designated them in the PII category. This further illustrates the necessity to safeguard digital data and device identifiers in a secure manner. For all PII information the onus is on the collector and repository owner to ensure that the data is encrypted and/or pseudonymized. Pseudonymization masks data replacing identifying information with fake information, while encryption encodes messages so that only those with proper authorization (keys) can access the data. Together both methods provide for the highest level of data protection.

3 GDPR Provisions

The primary intent of GDPR is to safeguard PII and give users control over their personal data. Under the language of GDPR there are two essential entities: the data controller and the data processor. Article 4 defines data controllers and data processors as: (a) "controller" means the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data; (b) "processor" means a natural or legal person, public authority, agency or other body which processes personal data on behalf of the controller For example, if Company XYZ sells pencils to consumers and uses Gmail to email consumers on their behalf and track their engagement activity, then with regard to such email activity data, Company XYZ is the data controller, and Gmail is the data processor. Understanding this distinction is critical for GDPR compliance. Per the GDPR, the data controller is determined to be the party responsible for collecting user consent, managing consent-revocations, enabling users right to access and all of the identified eight rights defined under the GDPR. From the example above if Mary Jones contacts Company XYZ (the controller) and wishes to revoke her consent for collecting personal data, this requires Company XYZ to contact Gmail (the processor) to then remove the revoked data from their servers. Failure to comply with Mary's request may result in stiff penalties to Company XYZ.

3.1 Individual Rights

GDPR has identified eight rights afforded the individual. These rights are the backbone of the legislation and clearly focus on protecting an individual's right to privacy. The following text outlines each of these rights. Since the right to restrict processing and the right to object to processing are closely aligned this paper combines the discussion of both rights into a single topic.

3.1.1 Right to be informed

Individuals have a right to be informed about the collection and use of their personal data. There are four components with this right: (1) The content of this notification must be immediate based upon when the data is obtained and it must specify the source of the data. (2) The name of the data collector and the name of the data protection officer must be identified. (3) The individual must be notified when personal data is transferred to a third country (not the collecting or processing country). This notification must include: processing purpose, intent, and legal basis for transmitting personal data, to third countries. (4) Specifics regarding the duration for data storage, the rights of the data subject, the ability to withdraw consent, the right to lodge a complaint with the authorities and whether the provision to hold this personal data is a statutory or contractual requirement (Article 13 of GDPR).

3.1.2 Right of access

An individual can request a report of all the data that a data controller has collected and has stored. This personal data must be provided within one month of the data subject's requests, in most cases, there should be no charge for the provision of this information (Article 15 of GDPR).

3.1.3 Right to rectification

An individual can request to have inaccurate personal data rectified, or completed if it is incomplete. An individual can make a request for rectification verbally or in writing. The controller has one calendar month to respond to a request (Article 16 of GDPR).

3.1.4 Right to be forgotten

An Individual can withdraw consent and request that data be permanently erased from all computer servers. There are several conditions in which an individual can request that data be deleted: (1) The individual withdraws consent for the collection of the data. (2) There is no reason for the collector of the data to keep the data. (3) The data was obtained without consent. If the controller of the data has made the data public, it is the controller's responsibility to inform all entities processing the data to take action to remove all links to the data and destroy data copies. (Article 17 of GDPR).

3.1.5 Right to restrict processing and right to object

An individual can put limitations on the way that an organization uses their data. This is an alternative to requesting the erasure of their data. A controller is permitted to store the personal data, but not use it. It is possible for a controller to reject this request of an individual. (Article 18 of GDPR). If a controller rejects an individual's request to restrict processing (Article 18 of GDPR), the individual has the right to object to being denied (Article 21 of GDPR).

3.1.6 Right to data portability

An individual has the right to have data transferred to oneself or to a third party in a machine-readable format (meaning something a user can understand). This data could be ascertained by observing an individual's activities when using a device or service such as: (a) History of website usage or search activities; (b) Traffic and location data; or (c)'Raw' data processed by connected objects such as smart meters, health equipment and wearable devices (Article 20 of GDPR).

3.1.7 Right with automated decision making and profiling

An individual is not to be subject to solely automated decisions or profiling, which may have a legal or similarly significant effect on them. A solely automated decision means that the decision-making process is totally automated and excludes any human influence on the outcome. A process might still be considered solely automated if a human enters the data to be processed, but the decision-making component is carried out by an automated system with no human interpretation. A decision that has a "similarly significant effect" is something that has an equivalent impact on an individual's circumstances, behavior or choices. Some examples of non-legal significant effects would be (a) automatic refusal of an online credit application and (b) e-recruiting practices without human intervention (Article 22 of GDPR).

4 Data Breach Requirements

In the event of a personal data breach, GDPR Article 33 requires that the data controller notify the appropriate supervisory authority within 72 hours after becoming aware of the event. A supervisory authority is an independent organization established by each country in the EU to monitor data privacy activities and protect the rights and freedoms defined by GDPR. In the case of a breach the GDPR specifies several steps to be taken by the organization that has been compromised: (1) Carry out an investigation. (2) Inform regulators and individuals impacted by the breach. (3) Provide specific details with respect to what data was compromised. (4) Describe how the issue will be addressed moving forward. (5) Inform the supervisory authority of the specific country where the breach occurred within 72 hours.

A breached organization must inform the supervisory authority with important details about the breach such as: who accessed what data and when; what categories of people were affected; how their data was being used; and name the impacted users. The regulators will want a record of all patches applied to the system; any forensic details about the breach; estimated impact of the breach and the remediation plan. Failure to follow these steps can result in stringent penalties.

The GDPR has two categories of fines, one for lesser breaches and the other for severe breaches. For lesser breaches the maximum fine cannot be more than 10 million euros or 2 percent of company's annual revenue. For severe breaches the maximum can be the greater of 20 million euros or 4 percent of a company's annual revenue. The regulators may use discretion when determining fines and are evaluate a range of factors such as the nature, gravity, and duration of the infringement; the number of people affected and the extent of the damage to them; whether the breach was intentional or negligent; any previous history of noncompliance; any action taken to mitigate the damage; and whether the controller notified and cooperated with the supervisory authority regarding the breach.

4.1 Privacy by Design

Although the concept of incorporating security in the design of a product is not new, the GDPR has strengthened this concept under Article 25 specifying that the controller shall consider the security and the exposure of personal data when creating the design of the product. Recital 76 of the GDPR further states that the principles of data minimization should be considered in the design in order to limit exposure of PII. Technical and organizational measures such as encryption and pseudonymization are encouraged to be incorporated in the design of collecting, processing and storing of sensitive data. This is especially relevant with the proliferation of Internet-of-things (IOT) devices being quickly developed and brought to market often without any consideration as to security of PII [1].

5 Conclusion

Clearly the intent to protect an individual's privacy is at the core of these regulations. No longer will it be tolerated for organizations to haphazardly handle customer data they have collected and stored. Strong penalties serve as threats to encourage organizations who collect PII to handle it with care. In our data driven society, students from all majors, must have a comprehensive understanding of the value and sensitivity of the data collected, analyzed, processed and stored by the products they develop and use. There are four major take-aways from this paper that teachers must incorporate into their curriculum and students must learn based upon the requirements listed in the GDPR.

The definition of PII extends far beyond a person's name, social security number, driver's license or other discrete identifier. Instead, it can be a combination of multiple personal data vectors that when examined in entirety can be used to identify an individual. Some of these information sources can include information about the computer device such as IP address, or physical location. Using extreme care when collecting and storing this information is required.

Second, the GDPR defines eight rights that an individual can expect regarding the collecting, storing and accessing of one's PII. Three of the discussed rights include the right to be informed about the collection and use of PII; the right to access what data about an individual has been collected; and the right to be forgotten and have data erased from online platforms. Failure to adhere to the regulations for each of these rights will result in significant penalties to the perpetrator.

Third, GDPR treats data breaches as a serious offense and has clear guidelines for reporting the breach within 72 hours to country authorities as well those whose data has been breached. Any organization collecting data must adhere to the regulations and be able to provide a full report within the specified 72-hour timeframe.

Lastly, security can no longer be an afterthought when designing products. GDPR specifies that security must be included in the design phase of product development or face stiff penalties. Teaching engineering and programming students the need to think security first when creating products will move the design of PII security into the front-end of the development cycle.

The EU has taken the lead to protect an individual's privacy with the data handling requirements specified GDPR. The rights of the individual trump the rights of organizations who wish to sell an individual's data without obtaining the consent of the user. GDPR and other privacy laws put the ownership of personal data back with the individual. Violating the rights of the individual or a breach of collective data will pack a powerful punch to the bottom-line of the violating organization. Anyone working with PII must understand the implications for failing to take necessary precautions to secure this sensitive data which may not only cost their employer big fines but their own jobs. Therefore, understanding what constitutes PII and technically knowing how to protect an individual's privacy must be incorporated into the curriculum.

References

[1] Internet of things status and implications of an increasingly connected world. 2017.

[2] Beyond the GDPR: What You Should Know about Japan's Act on the Protection of Personal Information. 2018.

[3] Pulse Survey: GDPR Budgets Top 10 Million for 40 percent of Surveyed Companies. 2018.

[4] The GDPR Applies for Liechtenstein, Norway and Iceland. 2018.

[5] European Commission. Data Protection Rules for the Protection of Personal Data inside and Outside the EU. 2018.

[6] Joe Dehner. Data protection rules for the protection of personal data inside and outside the eu. 2018.

[7] Cyrus Farivar and David Ingram. California's new data privacy law could change the internet in the us. 2019.

[8] Michael Grothaus. How our data got hacked, scandalized, and abused in 2018.

[9] Jennifer Huddleston. *What GDPR's First Year Says about Data Privacy Regulation.* The Hill, 2019.

[10] The Ponemon Institute. 2018 Cost of a Data Breach Study. 2018.

[11] Alex Wall. Summary: Philippines data privacy act and implementing regulations. 2017.

GenCyberCoin: An Engaging, Customizable, and Gamified Web Platform for Cybersecurity Summer Camps and Classrooms*

Vitaly Ford[1], Ambareen Siraj[2]
[1]*Computer Science and Math Department*
Arcadia University
Glenside, PA 19095
fordv@arcadia.edu
[2]*Department of Computer Science*
Tennessee Tech University
Cookeville, TN 38505
asiraj@tntech.edu

Abstract

Teaching cybersecurity requires dedicating a substantial amount of time and effort to combine both practical and theoretical notions into a coherent and clear chain of thoughts. As a result, educators have been exploring various gamification techniques to spark interest among students and engage them with interactive activities leading to a cybersecurity career. In this paper, we present a GenCyberCoin open-source web platform that can be used as a complementary module to the existing teaching material in cybersecurity summer camps and classrooms. GenCyberCoin aims to facilitate the development of students' interest in cybersecurity by providing students with opportunities to earn and spend digital currency, practice bug hunting, and get rewarded for helping peers and completing tasks. This platform introduces students to real-world concepts such as the blockchain, digital currency markets, banks, cybersecurity principles, open source intelligence gathering, passwords, bug bounty, and social norms and values.

1 Introduction

The cybersecurity field has seen a significant growth of interest from academia and government in recent years. Numerous cybersecurity education programs and calls for grant proposals have been developed, for example, GenCyber[12], CLARK[1], C5[2], NSF SaTC EDU[4], and CyberCorps SFS[5].

Teaching cybersecurity requires dedicating a substantial amount of time and effort to combine both practical and theoretical notions into a coherent and clear chain of thoughts. Consequently, to address the need for developing a cybersecurity workforce pipeline, educators have been exploring various creative gamification techniques to spark interest among students and engage them with interactive activities leading to a cybersecurity career – a career that demands much focus, patience, grit, critical thinking, and aspiration for continuous learning[7, 8, 9, 10, 11, 13, 14, 15].

In this paper, we present a publicly-available GenCyberCoin web platform[6] that can be used as a complementary module to the existing cybersecurity teaching material in cybersecurity summer camps and classrooms. This platform facilitates increasing students' interest in cybersecurity, incentivizes students to complete assignments, promotes exploring cybersecurity on their own, and provides a practical approach to teaching cybersecurity principles with interactive elements. GenCyberCoin is an open-source platform and it assumes no prior experience in cybersecurity from students and teachers.

1.1 Related Work

Weanquoi et al.[15] developed an educational 2D game called "Bird's Life". This game focuses on teaching phishing attacks and defense techniques. The game works on different platforms (PC, web, and mobile). They deployed the game in several courses related to Internet Systems and Computer Usage. The feedback they received demonstrated that students enjoyed the game, learned anti-phishing tips, and increased their interest in learning more.

Jin et al.[10] incorporated game elements into their GenCyber[12] summer camp. They developed four games: a 3D social engineering game, 3D VR secure online behavior game, cyber defense tower game, and 2D GenCyber card game. Their post-camp survey demonstrated promising results in terms of utilizing games as camp activities.

Švábenský et al.[14] designed a cyber range where undergraduate students learned about network attacks and defenses by creating educational games that teach about certain vulnerabilities. Students learned cybersecurity attack and defense methodologies and worked on their own game projects that they later presented in front of students and professors.

Gonzalez et al.[9] developed a classification taxonomy for cybersecurity training resources that implemented gamified elements. Based on that taxonomy, they classified a number of gamification-related cybersecurity education projects and aligned them with their curriculum in several courses.

Giannakas et al.[8] developed a mobile app called CyberAware for K-6 children for cybersecurity awareness. The topics of the mini-games in this app include firewalls, antivirus, security patches, updates, and email spam filters.

Fouché et al.[7] proposed utilizing the existing Code Hunt framework[13] to teach application security and IT auditing by adding gamification elements. They also discussed how others have been incorporating secure coding into the Code Hunt framework.

Olano et al.[11] developed a multiplayer cybersecurity game called SecurityEmpire for high school students. It teaches about information assurance practices.

2 Motivation and Contribution

In 2017, our initial goal was to develop a web platform that would engage and incentivize students, providing a continuous cybersecurity learning experience throughout a one-week long GenCyber summer camp. GenCyberCoin has since evolved to address the following challenges in a practical way:

- Encourage students to pursue a cybersecurity career.
- Introduce students to the blockchain and digital currency markets.
- Introduce students to the major cybersecurity concepts, including "confidentiality, integrity, availability, defense in depth, keep it simple, and think like an adversary".
- Introduce students to social engineering and reconnaissance.
- Introduce students to bug bounty hunting and software bugs.
- Introduce students to social and ethical norms and values.
- Introduce students to password attacks and management practices.
- Provide an incentive for students to go an extra mile in participating in the activities and assignments.

2.1 Features

Additional features of GenCyberCoin include:

- Rewarding participation by allowing students to earn digital coins.
- Allowing students to show appreciation to peers.
- Allowing GenCyberCoin administrators to customize:
  - the marketplace with real products of their choice that students can purchase with their earned GenCyberCoins;

  - the reconnaissance and social engineering questions that students can answer, earning GenCyberCoins;
  - the activities that can be assigned to students who receive a custom number of GenCyberCoins when they successfully complete an activity.

- Being open-source and freely available on GitHub[6] with detailed instructions on how to set it up locally, through docker[3], and in the cloud.
- Having an intuitive, responsive, and easy to use web/mobile interface.

3 GenCyberCoin Web Platform

Some of the major goals of GenCyberCoin are to raise interest in cybersecurity, capture students' attention, and make the cybersecurity assignments engaging. To achieve these goals, the platform provides students with opportunities to explore and learn the cybersecurity principles in a hands-on way via the integrated account, wallet, blockchain, bug bounty, hall of fame, reconnaissance, feedback, and market modules. Additionally, it provides teachers (administrators) with modules to generate registration and reward codes, add and edit market/activities/reconnaissance, nominate activities, view student orders, manage students, change settings, and view feedback. The student view of the home page is shown in Figure 1.

3.1 Initial Setup

When teachers log into their accounts, they can perform multiple actions, such as preparing Activities, Market Items, and Reconnaissance/social engineering questions. It is not necessary to set up any of these for students to start learning about cybersecurity, however, we recommend to at least add a few new Activities and Reconnaissance questions to fully engage students.

Activities can represent badges, achievements, or completed assignments (gamification features) that can be later nominated to students upon teachers' discretion. Teachers have full control of what information they put on each Activity (picture, name, and description). Every Activity can also have a different GenCyberCoin value assigned to it.

Market Items are real goods that teachers decide to give away. Those items can be things that teachers receive from conferences, school swag, etc. Students can purchase those items on the Market, using their GenCyberCoins.

Reconnaissance and social engineering questions can be used to increase curiosity, break the ice, and teach about open source intelligence gathering. These questions can be anything that the teachers would like to ask; they can be about the teachers themselves, some random facts related to someone

Figure 1: Home page of GenCyberCoin in “student” view.

famous, cybersecurity puzzles, etc. For example, “where did Dr. Leo spend her vacation this year”, “how old is our school”, “decode: scaegneig oilniern”, and “what is the company name of the watch that Dr. Craig is wearing”. When a student correctly answers a reconnaissance question, the student automatically receives a GenCyberCoin reward, the value of which is set by the teacher in Settings.

3.2 Registration and Reward Codes

Teachers can create registration and reward codes that can be generated automatically or manually. The registration codes would allow students to register on the website whereas the reward codes (custom-named by teachers or auto-generated by the platform) would allow teachers to give those codes to students for various reasons, such as successfully completing assignments, finishing within a certain time period, helping peers, being a good team player, being a good listener – in other words, teachers have the absolute freedom to choose for what reasons they give the reward codes.

3.3 Bug Bounty and Hall of Fame

Bug Bounty is an autonomous module that introduces bugs on the website in various places on the platform. Students are provided with hints on where and what types of bugs to search for, whereas teachers do not need to know how to find those bugs. This module does not break anything on the website but when a student finds a bug, the platform automatically lets the student know what this bug allows to do in the real world and then rewards the student with GenCyberCoins for discovering the bug. Additionally, the student's bug hunting record is timestamped and captured on the Hall of Fame page that is visible to all students and teachers. At the time of writing, the platform contains 18 different bugs and we are in the process of adding more.

Additionally, the Hall of Fame links every found bug with the external resources so that students could further investigate the secure coding topics.

3.4 Blockchain

Every transaction happening on the platform is captured in a Blockchain. Students and teachers can transfer coins to anyone in their Wallets. Also, when students complete Reconnaissance questions or find a bug, they automatically receive GenCyberCoins. All of the aforementioned transactions are recorded in the Blockchain, representing a public digital ledger that is available for everyone to view. The transactions in the Blockchain are not reversible.

3.5 Settings

Teachers have full control over all settings including Market On/Off, Bug Bounty On/Off, Reconnaissance On/Off, custom reward value for bug bounty, custom reward value for correctly answering reconnaissance questions, and others. They can also customize the title and description of the platform's header text, located at the top of every page.

3.6 Feedback

Students can leave anonymous comment/feedback about anything they have in mind and teachers can view that feedback on a separate page.

3.7 Account, Password, and Wallet

Students can view their Activities and personal information on the Account page. The password manager is also available on that page, leading students to learn about the importance of password complexity and where to securely check if a real personal password has been compromised as a result of a data breach. The Wallet page represents a digital bank where students can transfer money and redeem reward codes as GenCyberCoins that they can later use to purchase real goods on the Market.

GenCyberCoin has a password recovery option and if a student answers two out three security questions correctly, the student will be automatically logged into the account without a password prompt. This could be an effective learning example of how hackers can log in to a bank account by obtaining personal information of a victim. Consequently, teachers can talk about the ethical implications of possibly knowing the answers to security questions of peers, friends, and relatives.

3.8 Social Ethics and Norms

GenCyberCoin lays the foundation for a social ethics and norms discussion. For example, Bug Bounty teaches students about ethical hacking versus unethical hacking, the difference between which can be described with one word: permission (did the person receive permission before performing the attack?) Also, Bug Bounty demonstrates the concept of responsible disclosure. Reconnaissance and Password Recovery teaches about privacy and its potential violation as a consequence of targeted social engineering and open source intelligence gathering attacks.

4 Learning Outcomes

At the end of all activities in this platform, the students will learn about:

- Confidentiality, integrity, availability, "defense in depth", "keep it simple", and "think like an adversary" concepts.
- Fundamentals of the blockchain technology and digital currency.
- Open source intelligence gathering and social engineering tactics.
- Password complexity and data breaches that leak passwords.
- Social ethics and norms.

5 Case Studies

We have deployed GenCyberCoin in multiple venues and its first use case was at the NSA/NSF GenCyber summer camp in 2017, Tennessee Tech University. It has also been deployed at the same camp in 2018. Both camps were one week long and GenCyberCoin was introduced on the first day, engaging students from the very beginning. Separately from GenCyberCoin, Tennessee Tech University GenCyber team (faculty, staff, students) has prepared many different activities with Raspberry Pis, virtual machines, console-based chat systems, CryptoBoxes, and cyber defense games. We integrated some of those activities (especially cryptography-related) into GenCyberCoin in such a way that upon completing them, the students automatically obtained reward codes that they were able to redeem in the Wallet on the platform.

In spring 2019 semester, we integrated GenCyberCoin in the Introduction to Practical Security Assessment course at Arcadia University. We used the platform for making extra credit assignments, teaching about web attacks, and rewarding GenCyberCoins based on how active students have been in the class.

Additionally, we ran a three-hour long outreach event at a Lower Merion High School in Philadelphia, PA in January 2019. A part of the event was dedicated to a mini-competition on GenCyberCoin, in which students had to find answers to reconnaissance questions and search for as many bugs as they could find, engaging them to interact with each other and ourselves.

The overall feedback we received from students across all above-mentioned events was that they felt much more interested in studying cybersecurity. The GenCyberCoin motivated them to continuously learn more about security topics, search for bugs on the platform, complete the activities, and be good team players. The following is the anonymous anecdotal feedback from high school students and teachers who attended our camps and outreach activities:

> I think GenCyberCoin is a great way to learn about bugs and collect points.
>
> Wow!! Hope you get to make the coin for all camps.
>
> The website was a creative way to get us to learn about digital currency and other stuff. Love it. This was pretty cool.
>
> The coins both make the camp competitive and fun. It takes a lot to get teens focused, and even more to keep focus. You've done both.
>
> I'd love to figure out how to modify it for my classes.

Some other feedback we got was more peculiar:

> I will give you all of my coins for information of a bug.
>
> If I pay you 15$, can I have 15 coins.

The gameplay metrics gathered automatically in GenCyberCoin demonstrated that more than 90% of all students (across all venues where GenCyberCoin was presented) answered all Reconnaissance questions regardless of their difficulty and found more than half of the Bounty Bugs.

6 Conclusion and Future Work

We have developed a GenCyberCoin web platform that engages students in practical learning of the major cybersecurity principles, blockchain, digital currency, reconnaissance, ethics, and bug bounty. The engaging elements of the platform make it a promising tool for cybersecurity education at summer camps (for instance, GenCyber[12]) and classrooms. This platform is freely available on GitHub[6] and is constantly maintained. New features are being added and code-base is being improved. Also, GenCyberCoin has been used in multiple GenCyber camps in 2019 across the nation. Our next steps are to disseminate this work, receive and address the feedback, and continue making it a better web platform for raising students' interest in cybersecurity.

7 Acknowledgments

We thank the National Security Agency and National Science Foundation for supporting this work as part of the NSA Grant#H98230-19-1-0147.

References

[1] CLARK. Effective cybersecurity curriculum at your fingerprints. `https://clark.center`.

[2] C5 Cyber Curriculum Development. The c5 project has brought together computer science faculty and cybersecurity educators to develop new course content that integrates the two disciplines. `https://www.c5colleges.org/index.php/cs-course`.

[3] Docker. Enterprise container platform for high-velocity innovation. `https://www.docker.com/`.

[4] NSF SaTC EDU. Proposals that focus entirely on cybersecurity education. `https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=504709`.

[5] CyberCorps Scholarship for Service. Scholarship for service is designed to recruit and train the next generation of it and security professionals to meet the needs of the cybersecurity mission for federal, state, local, and tribal governments. `https://www.sfs.opm.gov`.

[6] Vitaly Ford. GenCyberCoin GitHub Repository. `https://github.com/vitalyford/gencybercoin`.

[7] Sandro Fouché and Andrew H Mangle. Code hunt as platform for gamification of cybersecurity training. In *Proceedings of the 1st International Workshop on Code Hunt Workshop on Educational Software Engineering*, pages 9–11. ACM, 2015.

[8] Filippos Giannakas, Georgios Kambourakis, and Stefanos Gritzalis. Cyberaware: A mobile game-based app for cybersecurity education and awareness. In *2015 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL)*, pages 54–58. IEEE, 2015.

[9] Hugo Gonzalez, Rafael Llamas, and Francisco Ordaz. Cybersecurity teaching through gamification: Aligning training resources to our syllabus. *Research in Computing Science*, 146:35–43, 2017.

[10] Ge Jin, Manghui Tu, Tae-Hoon Kim, Justin Heffron, and Jonathan White. Evaluation of game-based learning in cybersecurity education for high school students. *Journal of Education and Learning (EduLearn)*, 12(1):150–158, 2018.

[11] Marc Olano, Alan Sherman, Linda Oliva, Ryan Cox, Deborah Firestone, Oliver Kubik, Milind Patil, John Seymour, Isaac Sohn, and Donna Thomas. Securityempire: Development and evaluation of a digital game to promote cybersecurity education. In *2014 {USENIX} Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*, 2014.

[12] NSA/NSF GenCyber Program. Inspiring the next generation of cyber stars. `https://www.gen-cyber.com`.

[13] Microsoft Research. Code hunt is a serious gaming platform for coding contests and practicing programming skills. `https://www.microsoft.com/en-us/research/project/code-hunt/`.

[14] Valdemar Švábenský, Jan Vykopal, Milan Cermak, and Martin Laštovička. Enhancing cybersecurity skills by creating serious games. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, pages 194–199. ACM, 2018.

[15] Patrickson Weanquoi, Jaris Johnson, and Jinghua Zhang. Using a game to improve phishing awareness. *Journal of Cybersecurity Education, Research and Practice*, 2018(2):2, 2019.

Telepresence: Evaluation of Robot Stand-Ins for Remote Student Learning*

George Dimitoglou
Department of Computer Science & Information Technology
Hood College
Frederick, MD 21701
dimitoglou@hood.edu

Abstract

Distance learning comes with many benefits and challenges. The lack of face-to-face interaction and difficulty to instill a sense of belonging to a learning community can lead to student disengagement and attrition. This pilot study examines the use of telepresence to bridge the gap of distance learning and class interaction. With telepresence, or robot-mediated learning, a remote student attends an in-person class using an in-classroom robot. The study evaluated the use of the technology and the the remote and in-class learners' experience. The results are very encouraging both from the the pedagogical and technology viewpoints, making the prospect of deploying robots in the classroom a viable and attractive alternative to distance learning.

1 Introduction

Distance learning has gone through several phases of development and continues to evolve. The first, *physical*, phase was asynchronous and based on postal mail exchanges of instructional material. The second, *analog*, phase first used radio and later television to broadcast educational programs that allowed both synchronous and asynchronous instructional delivery (e.g. live and taped broadcasts). The third, *digital*, current phase is Internet-based synchronous and asynchronous delivery of educational content using a mix of multimedia (e.g. audio, video) formats.

The need for distance education at the college level has been increasing, as more students are holding jobs or they are working adults often with family and professional commitments. The time and burden on these students to meet their academic responsibilities by having to travel and be physically present on campus has prompted institutions to offer alternative instructional delivery methods: online, via video conferencing and hybrid (blended) to name a few.

These alternative methods are convenient and flexible even though in- person classes still offer a richer learning experience. Physical classroom presence provides structure, social presence and opportunity to learn from verbal and non-verbal interactions with instructors and peers. The social and tacit aspect of learning from the interactions can be easily lost with online, video conference or other distance education modalities despite efforts to engage students with forum-based discussions, group activities and synchronous sessions. The hybrid delivery method where students mix in-person with online attendance addresses, to an extent, some of these concerns.

A solution that could fully address these issues is telepresence or robot-mediated learning. With telepresence, a remote student attends class via an in-classroom robot. Telepresence is a variation of online learning, best described as *synchronous face-to-face distance learning*. Using robots has become feasible due to the rapid evolution of robotic technologies in terms of capability, affordability and public acceptance. Robotic platforms in the consumer market come in a variety of form factors and offer sophisticated navigation, communication and sensing capabilities.

At our institution we offer a limited mumbler of on-line graduate courses despite the increasing demand for them by current and prospective students. In the meantime, the online graduate education landscape in our geographical region is very competitive with many online offerings that vary in modality (online, hybrid), cost and quality. The technology advances and the growing body of education research on the efficacy of telepresence have been a motivating factor to explore using robots as an additional, flexible distance learning modality.

The objective of this limited-scale pilot study is to explore the practical challenges and evaluate the prospect of delivering distance, robot-mediated learning at a bigger scale.

2 Background

Telepresence technology has been studied extensively and deployed in various fields, including healthcare ([6], [39], [41], [28], [30], [42]), the workplace ([19],

[36], [15], [20], [35]), education ([8], [28], [24], [1], [7], [11]), helping those with special needs [46], helping the elderly [9] and even in library reference desks [23]. The affordability and ease of use of modern robots has also led to their deployment for even more lighthearted uses such as sending a robot to social work-related events [14].

In the United States, K-12 education has been a telepresence pioneer. The field of K-12 education has a long history of providing distance learning services with the first documented case of such service was offered in the 1930s for pregnant students in Connecticut [21]. Later, similar services were offered using different distance education ways to help seriously ill home-bound students. In the context of telepresence, robots have been deployed to help chronically ill children ([38], [45]) or to support learning foreign languages [29]. The motivation for using telepresence has been to reduce the educational impact of long physical absences, ensure learning continuity and reduce social isolation. In most K-12 telepresence deployments the home or hospital-bound student joined a physical classroom via a robot although there has also been experimentation with having only the instructor in the classroom and all the students participating remotely using robots [31].

In higher education, the rapid growth of online programs has prompted institutions and researchers to examine pedagogical strategies and technologies to deliver comparable learning experiences in all classroom modalities: in physical space, virtually or in hybrid format ([2], [4], [18], [22], [40]). Despite the inherent convenience of online attendance, it has been reported that students may feel disconnected due to their limited social presence ([10], [44], [12], [1]). Similar sentiments of isolation also appear in other distance learning modalities such as video-based and tele-conferencing.

One way to address the isolation, is to leverage robot-mediated communication to introduce *embodiment* in the classroom. Several studies have underscored the value of embodiment to reduce isolation of remote learners ([26], [13], [16], [17]) by enabling remote students to obtain social presence [3]. Embodiment, interaction and social presence have all been found to enhance cognitive engagement and performance in shared activities [32]. Preliminary results from telepresence studies in higher education ([1], [25]) are finding encouraging results with remote students gaining direct classroom access with stimulating peer and instructor-mediated educational experiences [5]. With telepresence, students get a sense of *virtual inclusion*, helping them feel physically present ([27], [33]) a feeling further amplified using robots with mobility capabilities [37].

3 Methodology

To examine the viability of robot-mediated learning at our institution, we conducted a small pilot study in an introductory graduate course with one remote student. The intention of the study was to examine the use of a robot without altering the course learning strategy or pedagogy but examine how telepresence is viewed from: (a) an educational perspective by evaluating the student perceptions, and, (b) a technology perspective, evaluating the robotic platform. The following sections report on the tools, methods, participants, results and lessons learned.

3.1 The Robot

The hardware platform used in our study is a two-wheeled robotic platform (Fig. 1) with a 10-inch screen that includes a camera, a microphone and a speakerphone. The platform's mobility and small footprint allows for easy movement between desks and chairs. The remote learner uses a tablet or mobile phone application to control the robot's motion and the screen's tilt. Through the robot's camera, microphone and speakerphone, the learner can participate in classroom activities as though they were present.

Height: 1.1m
Dimensions: 270 x 372 x 1100mm
Visual range: 30 degrees tilt (up or down).
Top Speed: 1.64 mph (2.64 k/h)
Bandwidth: WiFi, 4 Mbps (min)
Video Resolution: 720p HD
Run Time: 10 hours continual (60 hrs Standby)
Charge Time: 6 hours
Misc.: Foldable
Software Features: Facial Recognition. Face Detection. Anti-Falling System. Obstacle Detection. Crash Avoidance.

Figure 1: Left: the robot with the mast fully extended. Right: the hardware and software features and specifications.

3.2 Course Participants

The selected course for the pilot study is an introductory graduate-level course attracting students in information technology and management information systems. It is a lecture-based course with 185-minute weekly class sessions.

During the study, thirteen students were enrolled including the one remote student.

The course and the expected study participants were deliberately selected because of their expected higher level of comfort with technology. The purpose was to exclude -as practically possible- student populations that may have any techno-phobias and focus on the technology evaluation. It also meant that most likely the remote student would require minimal training to connect, operate and control the robot. An additional student played the role of the Setup Assistant with the responsibility to ensure that the robot is charged, arrived in class on time, is turned on and the audio and video feeds work. The Assistant would also handle any technology challenges during lectures as long as the handling was not distracting for anyone in the class. A backup communication plan was devised to enable the remote student to continue participating in class in case of a robot failure. Selecting a student rather than the instructor for this role was also by design to avoid inadvertently influencing students due to authority bias.

The instructor was expected to conduct the class as if the remote student using the robot was physically present: no special attention or accommodations were to be made beyond what would be warranted for a physically present student.

The physical layout of the classroom is a rectangular room (30' x 25.5') with four rows of floor-bolted desks, able to seat 24 students, an instructor's podium, a digital projector and a whiteboard.

The robot during lectures was placed in the front of the class at an angle so the remote student could rotate and be able to see both the instructor, the projection screen or the whiteboard and the rest of the class.

3.3 Data Collection and Analysis

Two surveys were prepared to collect data from the students. One version of the survey was designed for the in-person students while an adjusted, second survey version was designed for the remote student. Both were administered at the end of the course and were designed based on similar surveys from the literature review. Besides basic demographic questions, participants were asked to respond to Likert-style statements. One question related to familiarity with technology had a custom scale (*None*, *Somewhat*, *Familiar*, *Mastery*) while all others use a six-level scale (*Strongly Agree*, *Agree*, *Neutral* [Undecided], *Disagree*, *Strongly Disagree* and *Non-applicable*).

4 Results

The remote student was a male in the 40-49 age group and his survey included additional questions about embodiment, social presence, psychological involvement along with questions about behavioral, affective and cognitive engagement. The student's responses were positive, with strong agreement on the benefits of telepresence across all survey questions. However, while valuable, the feedback of a single student (n=1) may give the wrong impressions and lead to wrong conclusions so his responses are not presented here.

From the in-person students (n=12), nine (75%) responded to the survey. Demographically, the gender distribution was fairly balanced, with 44.44% females and 55.56% males. The age distribution was as expected for a graduate course with 22% in the 18-24 age group and 77.78% in the 25-34 age group.

The first set of questions aimed to determine the breadth and depth of student exposure to robotics and artificial intelligence (AI). The responses about their exposure (Fig. 2) clustered between two categories, "*Somewhat*" and "*Familiar*" across the three popular technologies: a household robot, intelligent personal assistants and self-driving cars. The area with the least familiarity was the advanced AI algorithms with responses split equally between students who have only heard of the them and those that have not. This was encouraging as it indicated a lower likelihood of confirmation bias from in-depth understanding of robotics and AI.

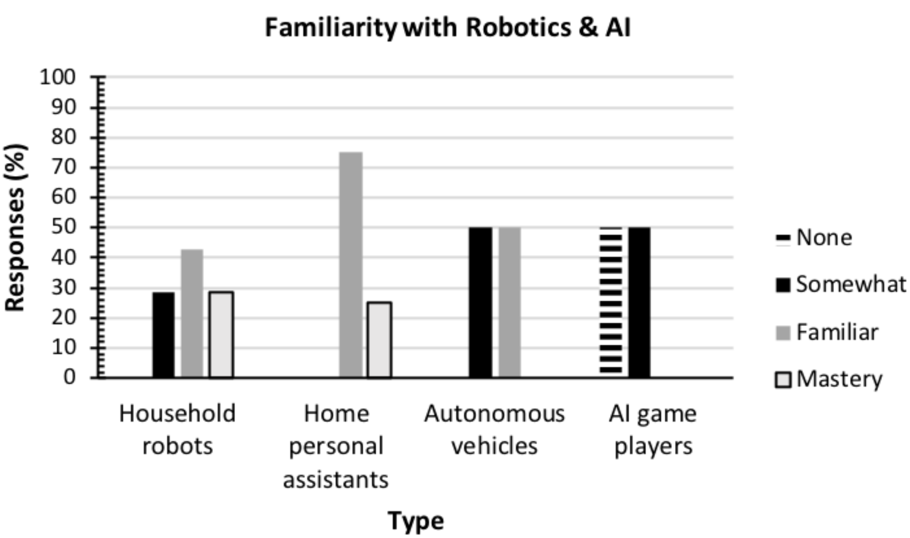


Figure 2: Range of student familiarity with robotics and AI.

The next two questions focused on communication. First, asking if in-class students were able to communicate verbally and non-verbally (e.g. gestures, facial expressions, movement) with the remote student. The responses clustered in the Strongly Agree (22.22%) and Agree (44.44%) categories while 33.33% responded as Neutral. Second, asking if in-class students felt any difference communicating with students physically and not physically present. Two-thirds (62.50%) strongly agreed (25%) and agrees (37.50%) that they experienced no difference in communication and the rest (37.50%) disagreed (Fig. 3).

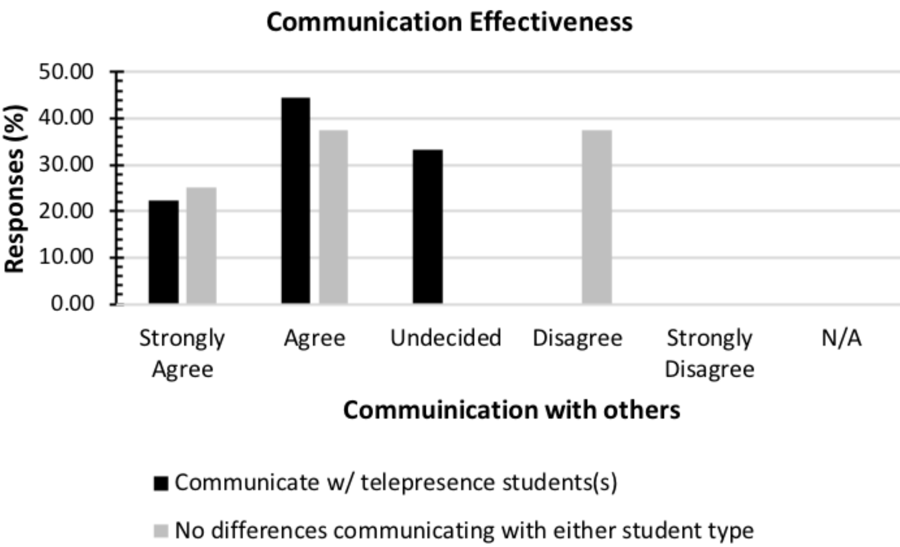


Figure 3: Most students responding that their ability to communicate both verbally and non-verbally was not hindered by telepresence.

The next group of four questions was designed to evaluate how telepresence affected working on class assignments: in-class participation, in-class assignments, homework and group projects (Fig. 4).

For in-classroom participation, the majority of responses (71.43%) strongly agreed (14.29%) and agreed (57.14%) that telepresence did not affect participation while 28.57% disagreed. For in-classroom assignments, the responses were distributed throughout the scale with less than half (44.44%) evenly split in the Strongly Agree (22.22%) and Agree (22.22%) categories and the rest were Neutral (11.11%) or deemed the question as non-applicable (11.11%). Similar responses were received about homework assignments: half (50.00%) of the responses were in the strongly agree (37.50%) and agree (12.50%) categories and the rest were Neutral (25.00%), disagreed (12.50%) or deemed the question as non-applicable (12.50%). The widest distribution of responses was observed with the responses about working on group projects. More than half (57.15%) of the responses strongly agreed (14.29%) and agreed (42.86%) and the rest disagreed (14.29%) or deemed the question as non-applicable (28.57%).

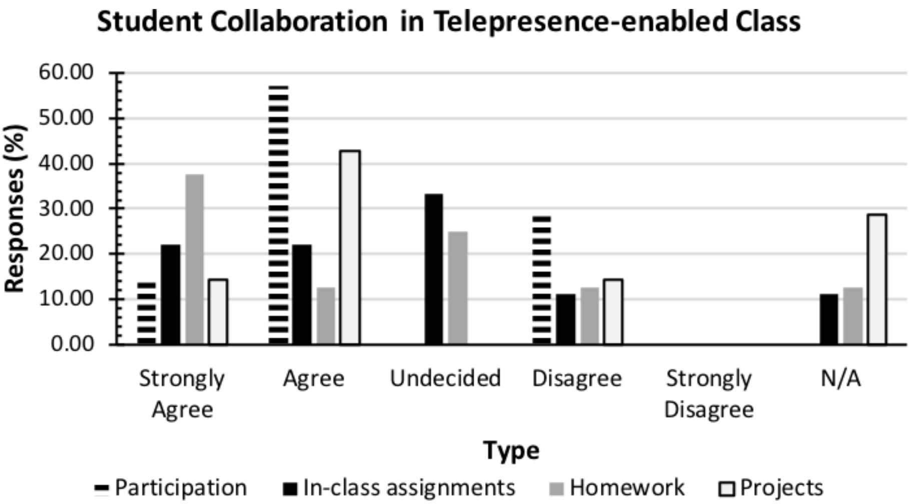


Figure 4: Student responses were across categories with respect to participation in different in-class assignment types.

The final set of questions was designed to evaluate the overall student perception and comfort after being in a classroom with a telepresence robot.

On the question about the *level of discomfort with telepresence before the study*, students disagreed with the premise of having any discomfort (71.43%), few (14.29%) had some level of discomfort and the rest (14.29%) deemed the question as not applicable. For the question of *feeling comfortable after participating in the study*, the majority of responses (87.50%) agreed (25%) and strongly agreed (52.50%) with feeling comfortable and the rest (12.50%) deemed the question as not applicable (Fig. 5).

On the question to determine whether *there was any difference between telepresence students and those physically present*, two thirds of the responses (66.66%) were evenly split between the Agree and Strongly Agree categories. The rest were also evenly split (each at 16.67%) being neutral and deeming the question as non-applicable.

The next question asked students to generalize if they thought *using a telepresence robot is equally effective as being physically present.* Students strongly agreed and agreed (42.86%) that it is equally effective while some of them

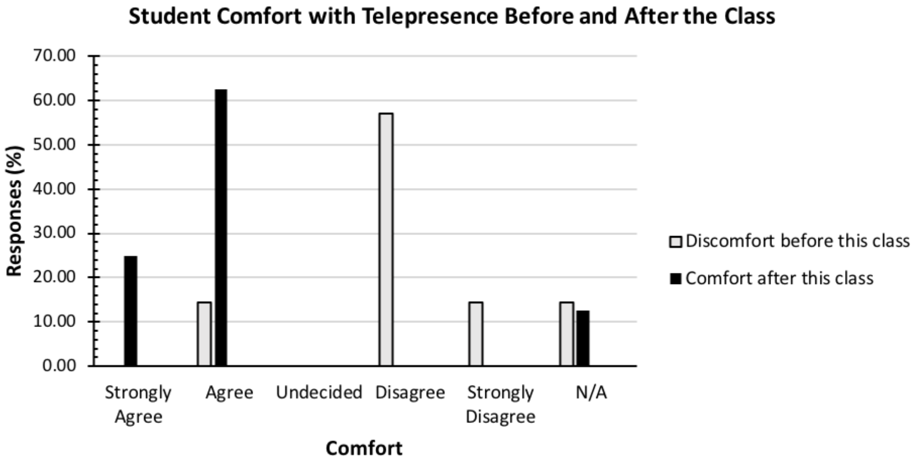


Figure 5: Student responses regarding their level of comfort before and after class. Students who felt uncomfortable about telepresence before class seem to have changed their mind after the class was completed.

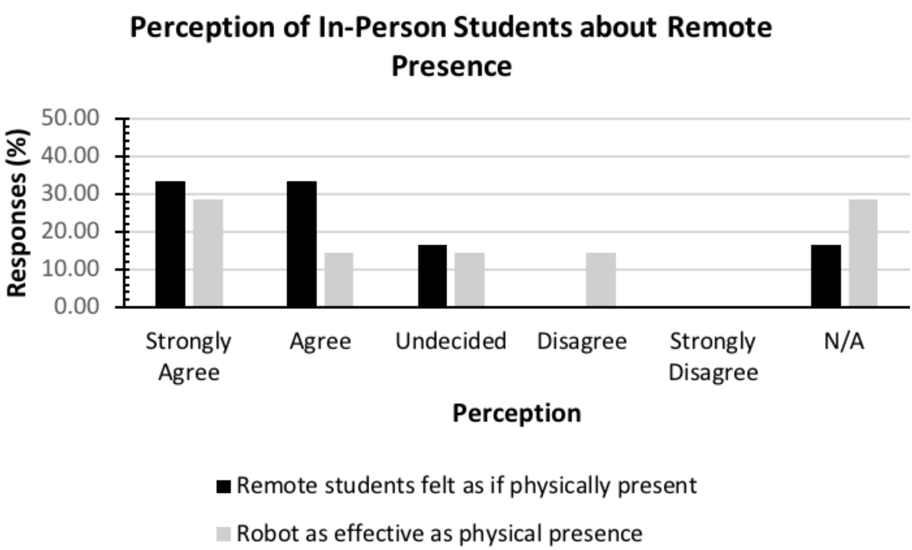


Figure 6: Student responses regarding their perception of differences between those physically present and the remote student.

(14.29%) were neutral, 14.29% disagreed and about one third (28.57%) deemed the question as not applicable. The final question asked if *students would like to be the remote students in a future telepresence class* and they overwhelmingly indicated (75.86%) they would like to, while only 12.50% disagreed, and 12.50% deemed the question as not applicable.

5 Discussion

Introducing a robot in a traditional, face-to-face classroom was a risky endeavor with many potential challenges. There were many unknown parameters for almost every facet of the study. The technology was tested in the lab, providing promising results about network connectivity, multimedia fidelity and easy remote navigation of the robot. However, the tests were executed in empty classrooms, during off hours and not during a real lecture.

The wireless network "stress test" showed no delays or communication failures. Yet there was no indication if the network and the robot's communication capabilities would respond equally well when multiple classes are in session and tens of students connect their devices to the wireless access points. There were concerns about the robot's video capability to let the remote student read the whiteboard or view the digital projector screen. Similarly, there were audio concerns about the ability to hear the instructor, be heard, ask a question, participate in a discussion or do something as simple as "raising their hand".

There were also concerns about the psychological aspect of the physically present students with having a robot embodying a classmate. There was no indication if the sterile, industrial design of the robot would alienate classmates or intrigue them. Human-computer interaction studies have a rich history about the range of human response to robots. On some occasions, humans have positive attachment to robots when the robots were personified (e.g. given names or had human or animal-like physical properties). On other occasions human reactions have varied from indifference to eerie sensations for robots with "too human" properties: a well-understood reaction in the robotics literature since the 1970s, referred to as the *uncanny valley* [34] effect.

Overall, judging from the survey responses, the experience during the class was overall positive for both the remote student and the students physically present in the class. The course was delivered as planned, with no interruption or distraction by having the robot in the class.

An underlying goal of this study was to examine if the learning experience could seamlessly include telepresence, without having to change the course pedagogy strategies. The feedback received by the instructor and the students indicated that everyone seemed unfazed by the existence of the robot and treated its existence in the classroom as nothing extraordinary. The instructor followed the course syllabus and planned methods of instruction without deviation or adjustment. From the remote user's viewpoint the technology met the embodiment expectations of feeling as "being there" without loss in verbal and non-verbal communication.

The performance of the video feed worked well in both directions, although a higher than the 720p resolution would have been welcome. From the remote student's perspective there was one minor technical issue with audio that requires further experimentation to resolve. The built-in microphone on the robot is mono-directional which means that until the robot (and the microphone) are facing the direction of the speaker, the sound resembles background noise. We are exploring options with multi-directional microphones although this solution requires careful balance to avoid introducing too much audible input and sound sensitivity that could overwhelm the remote user.

From a deployment logistics perspective, it became clear that a physically present handler is required for something as simple as making sure that the robot is charged, turned on and out of its storage closet. It became also apparent that while the mobility aspect of the robot gave flexibility to the remote student in adjusting their direction and distance to the whiteboard, projector and desks, it would be more challenging in large amphitheater-style classrooms with steps and tight corridors.

6 Conclusion and Future Work

Reflecting on the overall experience from this small scale telepresence study, the results are very encouraging. The relatively low cost barrier to entry and the ability to use existing infrastructure (e.g. campus wireless network) gave us the opportunity to provide a remote student synchronous classroom access with negligible loss of the classroom experience.

From an educational perspective, the study was a success, albeit conducted using a very small sample (n=12) of student participants. The remote student interactions with the instructor and other classmates were smooth and effective. Our students adapted quickly as peers to a robot-mediated student. From a technology perspective, the technology provided sufficient capabilities to carry out the study and for the remote student to complete the course without any problems.

Looking forward, more experimentation will be necessary using larger student groups and across different course structures. One possible direction would be to examine the efficacy of telepresence in courses that have in-class and out-of class collaborative assignments. Another option would be to increase the number of remote students using telepresence to explore the dynamics between both physically present and remote students but also between the fully remote students. From the technology perspective, dealing with the minor audio issue should further improve the overall experience. Perhaps a possible formal study could evaluate the audio capabilities against well established International Telecommunication Union (ITU) subjective quality evaluation standards [43].

The completion of this pilot study confirmed that the use of robots in the classroom provide a sense of physical presence for a remote student without making any course strategy or pedagogical changes or impacting course delivery for students.

7 Acknowledgements

This work was supported by an Academic Innovation Grant from the Hood College Center for Teaching and Learning.

8 Appendix

The following are student comments from the survey.

- *I believe that use the technology is a great idea specially for education.*
- *Telepresence robots are effective for students who have to travel long distance. But it will be little difficult in case of group class assignment or project works as that robotic presence cannot give effective focus.*
- *I think it was the same as he was in The class. He was able to participate with questions, and actually move the robot to his convenience which was interesting to watch. I think is great to know that we that at Hood. Even more for people who wants to get educate, but live miles away as our classmate is. It is interesting that made that possible, and also to check on the experience! I appreciate you asked us as well! Thank you so much!*

References

[1] John Bell, William Cain, Amy Peterson, and Cui Cheng. From 2d to kubi to doubles: Designs for student telepresence in synchronous hybrid classrooms. *International Journal of Designs for Learning*, 7(3), 2016.

[2] John Bell, Sandra Sawaya, and William Cain. Synchromodal classes: De- signing for shared learning experiences between face-to-face and online students. *International Journal of Designs for Learning*, 5(1), 2014.

[3] Frank Biocca, Chad Harms, and Judee K Burgoon. Toward a more robust theory and measure of social presence: Review and suggested criteria. *Presence: Teleoperators & virtual environments*, 12(5):456–480, 2003.

[4] Matt Bower, Barney Dalgarno, Gregor E Kennedy, Mark JW Lee, and Jacqueline Kenney. Design and implementation factors in blended synchronous learning environments: Outcomes from a cross-case analysis. *Computers & Education*, 86:1–17, 2015.

[5] John Bransford, Nancy Vye, Reed Stevens, Pat Kuhl, Daniel Schwartz, Philip Bell, Andrew Meltzoff, Brigid Barron, Roy Pea, Byron Reeves, Jeremy Roschelle, and Nora Sabelli. Learning theories and education: Toward a decade of synergy. 01, 2005.

[6] Elizabeth Broadbent, Rebecca Stafford, and Bruce Macdonald. Acceptance of healthcare robots for the older population: Review and future directions. *I. J. Social Robotics*, 1:319–330, 11 2009.

[7] W. Cain, J. Bell, and C. Cheng. Implementing robotic telepresence in a synchronous hybrid course. In *2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT)*, pages 171–175, July 2009.

[8] William Cain and Danah Henriksen. Pedagogy and situational creativity in synchronous hybrid learning: Descriptions of three models. 01, 2013.

[9] Amedeo Cesta, Gabriella Cortellessa, Andrea Orlandini, and Lorenza Tiberio. Long-term evaluation of a telepresence robot for the elderly: methodology and ecological case study. *International Journal of Social Robotics*, 8(3):421–441, 2016.

[10] AW Chickering and ZF Gamson. Seven principles for good practice in undergraduate education. *american association for higher education bulletin, 1987*, 1987.

[11] Chris Fowler and John Mayes. Applying telepresence to education. *BT Technology Journal*, 15:188–195, 10 1997.

[12] D Garrison, Terry Anderson, and Walter Archer. Critical inquiry in a text-based environment: Computer conferencing in higher education. *The Internet and Higher Education*, 2:87–105, 03 1999.

[13] Benjamin Gleason and Christine Greenhow. Hybrid education: The poten- tial of teaching and learning with robot-mediated communication. *Online Learning Journal*, 21(4), December 2017.

[14] E. Guizzo. How i became a texai robot and went partying. *IEEE Spectrum*, May 2010.

[15] E. Guizzo. When my avatar went to work. *IEEE Spectrum*, 47(9):26–50, Sep. 2010.

[16] Antal Haans and Wijnand Ijsselsteijn. Embodiment and telepresence: To- ward a comprehensive theoretical framework. *Interacting with Computers*, 24:211–218, 07 2012.

[17] Tilo Hartmann, Werner Wirth, Peter Vorderer, Christoph Klimmt, Holger Schramm, and Saskia B¨ocking. Spatial presence theory: State of the art and challenges ahead. 01 2013.

[18] Danah Henriksen, Punya Mishra, Christine Greenhow, William Cain, and Cary Roseth. A tale of two courses: Innovation in the hybrid/online doctoral program at michigan state university. *Techtrends*, 58:45–53, 07 2014.

[19] Pamela J. Hinds, Teresa L. Roberts, and Hank Jones. Whose job is it anyway? a study of human-robot interaction in a collaborative task. *Human-Computer Interaction*, 19:151–181, 2004.

[20] Pamela J. Hinds, Teresa L. Roberts, and Hank Jones. Whose job is it anyway? a study of human-robot interaction in a collaborative task. *Human-Computer Interaction*, 19(1-2):151–181, 2004.

[21] Mary E. Holmes, Lorraine V. Klerman, and Ira W. Gabrielson. A new approach to educational services for the pregnant student. *The Journal of school health*, 40:168–172, 05 1970.

[22] Valerie Irvine, Jillianne Code, and Luke Richards. Realigning higher education for the 21st century learner through multi-access learning. *MERLOT Journal of Online Learning and Teaching*, 9:1–15, 06/2013 2013.

[23] Eric Johnson and Johannes Kromer. Testing telepresence: Remote reference service via robotics. 2015.

[24] Jason Kaufman and G Mcnay. Community in the telepresence classroom. *Educause*, 51, 04 2016.

[25] Hideyuki Nakanishi Kazuaki Tanaka and Hiroshi Ishiguro. Comparing video, avatar, and robot mediated communication: Pros and cons of embodiment. 60, 05 2014.

[26] Konstantina Kilteni, Raphaela Groten, and Mel Slater. The sense of embodiment in virtual reality. *Presence: Teleoperators and Virtual Environments*, 21(4):373–387, 2012.

[27] Taeyong Kim and Frank A. Biocca. Telepresence via television: Two dimensions of telepresence may have different connections to memory and persuasion. *J. Computer-Mediated Communication*, 3(2):0, 1997.

[28] Annica Kristoffersson, Silvia Coradeschi, and Amy Loutfi. A review of mobile robotic telepresence. *Advances in Human-Computer Interaction*, 2013:1–17, 02 2013.

[29] Jian Liao and Xiaofei Lu. Exploring the affordances of telepresence robots in foreign language learning. *Language, Learning and Technology*, 22:20–32, 10 2018.

[30] Nonofo Losike-Sedimo. Telepresence teaching: Implementation issues in universities and other institutes of higher learning. 03 2010.

[31] Eduardo Luévano, Eduardo López de Lara, and Juan Edward Castro. Use of telepresence and holographic projection mobile device for college degree level. *Procedia Computer Science*, 75:339–347, 2015.

[32] Brian E. Mennecke, Janea L. Triplett, Lesya M. Hassall, Zayira Jordán Conde, and Rex Heer. An examination of a theory of embodied social presence in virtual worlds. *Decision Sciences*, 42(2):413–450, 2011.

[33] Marvin Minsky. Telepresence. *OMNI Magazine*, pages 45–54, June 1980.

[34] Masahiro Mori, Karl MacDorman, and Norri Kageki. The uncanny valley [from the field]. *IEEE Robotics Automation Magazine*, 19:98–100, 06 2012.

[35] B. Mutlu and J. Forlizzi. Robots in organizations: The role of workflow, social, and environmental factors in human-robot interaction. In *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 287–294, March 2008.

[36] Bilge Mutlu and Jody Forlizzi. Robots in organizations: The role of workflow, social, and environmental factors in human-robot interaction. In *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 287–294. ACM, March 2008.

[37] Hideyuki Nakanishi, Yuki Murakami, and Kei Kato. Movable cameras enhance social telepresence in media spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09*, pages 433–442. ACM, 2009.

[38] Veronica Newhart, Mark Warschauer, and Leonard Sender. Virtual inclusion via telepresence robots in the classroom: An exploratory case study. *International Journal of Technologies in Learning*, 23:9–25, 01 2016.

[39] Hayley Robinson, Bruce MacDonald, and Elizabeth Broadbent. The role of healthcare robots for older people at home: A review. *International Journal of Social Robotics*, 6(4):575–591, Nov 2014.

[40] Cary Roseth, Mete Akcaoglu, and Andrea Zellner. Blending synchronous face-to-face and computer-supported cooperative learning in a hybrid doctoral seminar. *TechTrends*, 57, 05 2013.

[41] Martin Salzmann-Erikson and Henrik Eriksson. Tech-resistance: the com- plexity of implementing nursing robots in healthcare workplaces. *Contemporary Nurse*, 52(5):567–568, 2016. PMID: 27295546.

[42] Martin Salzmann-Erikson and Henrik Eriksson. Tech-resistance: the com- plexity of implementing nursing robots in healthcare workplaces. *Contemporary Nurse*, 52(5):567–568, 2016. PMID: 27295546.

[43] Muhammad Shahid, Mekides Assefa Abebe, and Jon Yngve Hardeberg. Assessing the quality of videoconferencing: from quality of service to quality of communication. *Electronic Imaging*, 2018(12):235–1, 2018.

[44] Glenn Smith and Marypat Taveras. The missing instructor: does e-learning promote absenteeism? *Elearn*, 2005, 01 2005.

[45] Neelkamal Soares, Jeffrey C Kay, and Geoff Craven. Mobile robotic telepresence solutions for the education of hospitalized children. *Perspectives in health information management*, 14:1e, 10 2017.

[46] Kate Tsui, Adam Norton, Daniel Brooks, Holly A Yanco, and David Kontak. Designing telepresence robot systems for use by people with special needs. In *International Symposium on Quality of Life Technologies 2011: In- telligent Systems for Better Living, held in conjunction with RESNA 2011 as part of FICCDAT*, 06 2011.

Integrating Git into CS1/2*

David C. Conner[1], Matthew McCarthy[2], and Lynn Lambert[1]

[1]*Department of Physics, Computer Science and Engineering*
Christopher Newport University
Newport News, VA 23606
{david.conner,llambert}@cnu.edu

[2]*Department of Mathematics*
University of Virginia
Charlottesville, VA 22904
mm8fb@virginia.edu

Abstract

We recently introduced a Git-based workflow into our early CS curriculum, which allows its use throughout our curriculum and provides all students with tools they need as they pursue internships and undergraduate research. A side benefit of our approach is that we provide an early introduction to command line and Unix-based commands by using Git-Bash. In this paper, we discuss our motivation, and our current course setup based on lessons learned over the past two years. We describe our approach to manage the student's cognitive burden as they adapt to our workflow, which makes use of continuous-integration unit testing for local testing, and automatic upload to the Web-CAT centralized grading system. We present our open source tools developed to streamline our course management workload by managing creation of user accounts and groups on our local GitLab server.

1 Introduction

Since its beginnings in 2005, Git has become one of the most widely used source code management (SCM) systems in the world[5]. By 2014, almost 43% of professional developers used Git[16]. In addition, Git can be used collaboratively, and can run unit tests using continuous integration tools. Still, Git can be overwhelming when students are expected to learn it on their own, and only highly motivated students tend to be successful. By managing this cognitive burden, we feel that we can better prepare all of our students.

For these reasons, starting in spring 2017, we have integrated the basics of Git into our early CS curriculum at our predominately undergraduate liberal arts university. Our intent in teaching these skills early is to prepare our students for internships and research opportunities, while also improving their work in future classes.

This paper describes our lessons learned over the past several years, and presents open source tools that enable us to effectively manage our courses. In Section 2, we describe related work on using Git in a classroom environment. Section 3 describes the current structure of our introductory course, lessons learned from earlier efforts, and student responses. Section 4 presents a collection of Python scripts that we use to manage our GitLab groups and student accounts; we have made these available open source. Finally, Section 5 summarizes our results and provides recommendations for future work.

2 Background

Others have used Git in teaching Computer Science courses [2, 3, 4, 7, 8, 9, 10, 11, 12, 14, 15]. Most of these courses, however, were upper level, not CS1 and CS2 courses. They used Git for many of the same reasons that we did: it is an industry standard that allows students to improve their resume and better prepare for future internships and jobs; it allows students to use class machines that are not their own and then transfer to their personal laptops without losing work; and, it allows the instructor to track a student's progress (by checking for regular pushes). Because Git is a distributed system, a student can work without Internet access, and then push changes once the student regains network access [2]. Most, like us, found that the command line worked well with students; [10] found that students were more receptive to the command line Git interface than the graphical user interface that was provided. This allows us to ease students into using the command line early in our curriculum.

Most using git in class used GitHub rather than GitLab (though both [9] and [14] used GitLab). There are several advantages to using GitLab over GitHub to manage the remote server, while still using Git. First, GitLab is open source. GitLab has private repositories by default, making it better for student use as instructors can control the amount of sharing. By hosting it locally, we can create scripts[1] that both set up the course and student repositories, and enable continuous integration with each pushed commit, which allows a student to receive immediate feedback about how many unit tests were passed. In addition, we have used a pair programming model in our introductory courses [13], and GitLab easily allowed partners to share a repository while keeping it private from other students.

The downsides of locally hosting a GitLab site are some system administration overhead, and the lack of student exposure to GitHub, which is the more commonly used platform for open source code. We do not see this latter issue as significant as we use GitHub in upper level courses, and students transition to it seamlessly. As to the former, we are fortunate to have dedicated resources to manage this.

Other papers focused on aspects of Git that we do not emphasize, such as branching and merging [2, 10, 11], collaborative development [9, 14], and pull requests and issue tracking [11]. Although these are valuable skills and necessary to fully exploit Git's power, to reduce the cognitive burden, we chose to not use these tools in our introductory courses. We defer these advanced Git skills to our later software engineering course.

While these papers confirm our experience, in all honesty, we adopted Git for our introductory classes because we believed it would benefit our students, not because of an exhaustive search of the literature. In retrospect, it is gratifying to see others confirm our intuition and experience.

3 Course Structure

Christopher Newport University is a public 4-year predominately undergraduate liberal arts institution that graduates approximately 500 STEM majors each year; of these, approximately 120 are in Computer Science (CS) or related fields. All CS, engineering, physics, and information science majors are required to take our beginning programming sequence of two courses, generally in their first year. Only Computer Engineering and CS majors are required to take the subsequent Data Structures and Algorithms courses.

In spring 2017, we began to distribute assignments and labs using Git and to integrate automated submission to a Web-CAT[2] server for automatic grading [6]. In Fall 2018, we revamped our CS curriculum by switching the introductory programming sequence to Python from Java. Thus, we have two

[1] See Section 4 for more info.

[2] https://web-cat.cs.vt.edu

semesters of Python experience and three semesters of Java experience using git.

Hosting our own internal server running GitLab[3] gave us complete control over project visibility and user accounts. The system runs on an departmental Linux cluster in a virtual machine with 4 assigned cores; a separate "Runner" instance where student code is run in separate shells is assigned eight cores to handle the continuous integration (CI) load.

In Section 4 we present our open source scripts[4] for course management. For each course, our script creates a course "Group" on our GitLab server with all students and instructors as "Reporters" (Reporter is read only access for GitLab). All common course materials are distributed here as individual project repositories (repo). A separate group is created for each section of the course for distributing time sensitive materials such as exams. Each student has an individual GitLab account, and we create an individual student group for the particular course; instructors are added as "Reporters" to the student's individual group. Students "Fork" each distributed project repo from the course group into their individual group; this is required once per project. Figure 1 shows our workflow.

Each project repo contains a `src/` folder that students modify as needed, a `given/` folder for instructor distributed code that should not be modified, and a `test/` folder for distributed unit tests used in grading the assignment. Additionally, the repo contains a `README.md` file containing the project description and related information, a `.gitignore` file, and a `.gitlab-ci.yml` file used by the GitLab Continuous Integration (CI) system. Our open source release provides a sample project.

Students "Clone" the project repo onto their personal machines, and open the repo as an existing project in their IDE; for our class, we use PyCharm[5] Community Edition (we used Eclipse for Java classes with similar results). Students write and debug their code in the IDE, and follow a standard `git add`, `git commit`, and `git push` work flow. We teach students to use the command line Git interface as this is the most general. On Windows systems, we make use of the GitBash[6] command line interface. This allows us to teach consistent commands across platforms and provides an early introduction to command line interfaces and Linux commands to students who are primarily familiar with Windows GUI interfaces.

With each push to the GitLab remote server, the GitLab CI runs the unit tests as configured in the `.gitlab-ci.yml` file. As part of the CI, we automatically upload the student code in the `src/` folder to Web-CAT server.

[3]`https://www.gitlab.com` Community Edition version 11.8.1

[4]`https://github.com/pcse/gitlab_tools`

[5]`https://www.jetbrains.com/pycharm/`

[6]`https://git-scm.com/downloads`

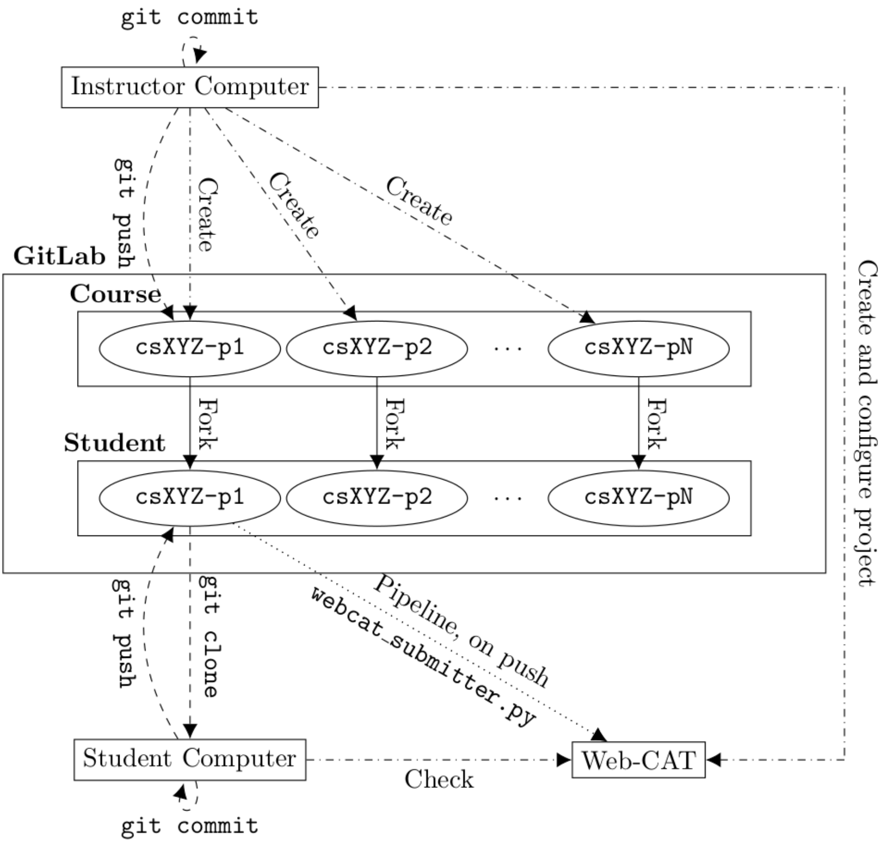


Figure 1: Assignment workflow.

The student's Web-CAT passwords are stored in GitLab as local environment variables assigned to each student's personal group.

We use Web-CAT to grade the code based on style[7] and correctness using the provided unit tests [6]. For some projects, we release all unit tests to students, while others have hidden tests only used in Web-CAT. Most of the time there are variations in the Web-CAT tests to adjust grade weighting. The Web-CAT Dashboard provides an overview of the course at a glance, and allows for simplified download of grades for importing into our course management software.

[7]Currently the Web-CAT system only checks style for Java; we expect future support for Python style checking.

Implementing this workflow necessitated training students on the rudiments of Git to follow a Fork-Clone-Add-Commit-Push cycle. This obviously increases the cognitive load on students. Thus in Spring 2019, we discontinued the use of GitLab in our first introductory programming class, so the rest of the section will focus on our second semester course. To overcome the cognitive load in CS2, we emphasize the tool chain in the first few lectures, as we are reviewing prior material that students are familiar with, and reinforce the workflow through guided practice in early labs. We emphasize to the students that this is part of professional practice, is good for their resumes, and is a fundamental expectation of the course. To further reduce cognitive load at this early stage, we simplify Git to focus only on Add-Commit-Push cycle. We avoid merging and branching in CS2, although we do suggest that interested students learn this on their own. We follow this consistent workflow for all lectures, labs, and exams in this course.

Our CS2 course uses a mix of lecture and active coding in class. One issue we had with earlier courses was that some students would choose to not follow along in class, nor complete the practice exercises as they were ungraded. gitlab and our autograding scripts have allowed us to follow a "many small programs" model that allows us to collect and grade the weekly in-class demonstrations as a participation grade, as well as more frequent smaller projects [1]. For the weekly in-class demonstration code, we allow open collaboration; The projects are independent with discussion allowed under an "empty hands" rule, where no code is shared and no notes are taken away from the discussion. Students have been receptive to Git using this approach.

From an instructor's point of view, having the full Git commit history is useful for a number of reasons. First, students can email questions about specific issues; it is easy for the instructor to track down coding issues on the GitLab web interface. An instructor can leave comments directly attached to a specific commit using GitLab's built-in code review features. Second, Git gives a history of when students were working on the code. We emphasize that our students should commit each significant change, and push frequently, at a minimum of once per coding session. Students are repeatedly reminded that "If you don't push, you were not working on code!" In cases of suspected plagiarism, the Git commit history can be the best witness.

4 Open Source Tools

To help manage our courses, we developed several Python scripts to assist in setting up student accounts, course groups, manage member permissions, and submit code. These convenience scripts are a Python 3 wrapper to an existing Python interface[8] to the GitLab Community Edition version 12.0.3 API,[9] which works by sending and receiving HTTP/S messages. Our Python scripts are open sourced on GitHub;[10] detailed instructions are provided online in the associated `README.md` file.

For each course, we must create a GitLab student group used to distribute all assignments including in-class work, projects, exams. We provide a basic `gitlab_account_setup.py` Python script to handle creation of the required group, student accounts, and student group for the course. All currently enrolled students in the course roster file are added as 'Reporters' in this main student group. The assignments are distributed as individual Git repositories with private visibility on the main student group.

The students work on forks of the assignment repositories and commit to Git using the standard workflow. Our setup uses a `webcat-submitter.py` script to automate the process of uploading code to Web-CAT for automatic grading.

Included in our open source repo is a sample project distributed to students during the first lecture. The simple "Hello World" example is used to demonstrate the tool chain to our students during the first two lectures in our CS2 course. We use this trivial example as a refresher so that students can focus on the workflow, which includes unit testing, and not the code. By the end of this exercise, the students have installed and demonstrated use of the complete toolchain on their personal machines. While personally checking the workflow and installed toolchain on all student machines is time consuming, it helps to establish an early face-to-face connection with students during the first week of classes, identify any students struggling with the workflow, and ensure a more level playing field students in later assignments.

In addition to account creation, we provide scripts for managing removal of instructors from student groups at the end of the term, and to monitor group and project visibility.

5 Results and Conclusion

Overall, our students have accepted the additional cognitive load of learning Git. In Spring 2019, we decided to move away from Git in our introductory course, and use a commercial online system for managing submissions and grading. This was largely done to reduce the initial cognitive load for students without prior exposure to coding and computer concepts. We will continue to assess the impact of this change as we move forward.

---

[8] `https://python-gitlab.readthedocs.io/en/stable/#`

[9] `https://docs.gitlab.com/12.0/ce/api/`

[10] `https://github.com/pcse/gitlab_tools`

In our second course, we ease the cognitive burden by introducing Git at the beginning of CS2, while we are reviewing prior concepts from our first introductory course. By following a consistent workflow across all course sections, and integrating the same workflow into our lecture practice, lab-based project work, and in-class exams, students receive regular and consistent reinforcement of the skills required to succeed. While the first week can be challenging, students quickly adapt.

We did not do a controlled study with students using git and not using git, but we did measure using git as a way to implement the many small programs approach by grading weekly in-class practice assignments. There, a strong majority of students agree or strongly agree that this has helped their understanding. Figure 2 shows the results from this spring in five different sections of the course with four different instructors where 57 percent of students agreed; this percentage is lower than prior semester surveys, likely due to growing pains in the new Python course design, but still indicative of the benefit.

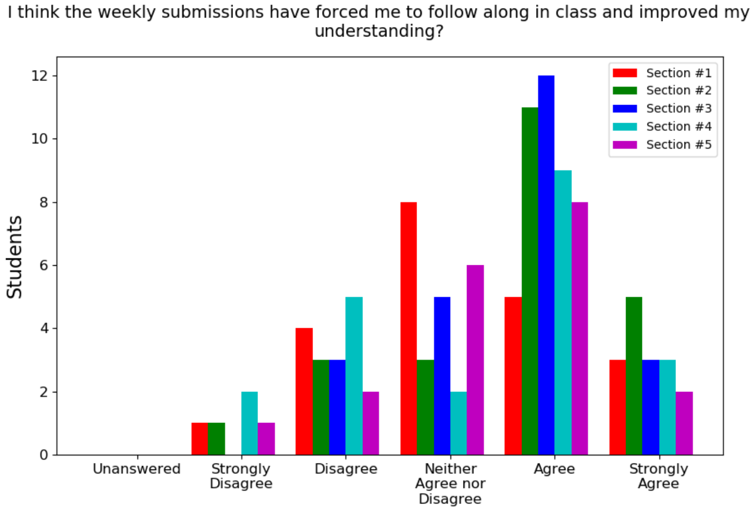


Figure 2: Survey of 107 students across five sections of CS2 in Spring 2019.

The most common complaint is not about Git, but the grading within Web-CAT. This mainly stems from early confusion about the automatic submission process and password management, or about the grading scheme based on different unit tests that may not be fully released with our distributed projects (e.g. “How come I only missed one question in PyCharm, but ten on Web-CAT?”). Students recognize the importance of SCM, and the prevalence of Git “in-the-real-world,” and are thus motivated to learn.

Since we first introduced Git in Spring 2017, our upper level students are now familiar with Git. This has allowed us to expand its usage throughout our curriculum. Overall, our efforts have been well received by both faculty and students.

As part of our work presented in this paper, we released a collection of course management scripts open source. These initial scripts described in this paper focus on account/group creation and user permissions. Additionally, we released scripts that work with GitLab's CI system to provide automatic upload to the Web-CAT grading system. In future work, we plan to present efforts toward project development and automated deployment.

References

[1] Joe Michael Allen, Frank Vahid, Alex Edgcomb, Kelly Downey, and Kris Miller. An analysis of using many small programs in CS1. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, pages 585–591, New York, NY, USA, 2019. ACM.

[2] F. F. Blauw. The use of Git as version control in the South African software engineering classroom. In *2018 IST-Africa Week Conference (IST-Africa)*, May 2018.

[3] Esmail Bonakdarian. A cost-effective multi-track introduction to Linux, Git and GitHub. *J. Comput. Sci. Coll.*, 32(2):34–41, December 2016.

[4] Esmail Bonakdarian. Pushing Git & GitHub in undergraduate computer science classes. *J. Comput. Sci. Coll.*, 32(3):119–125, January 2017.

[5] Scott Chacon and Ben Straub. *Pro Git*. Apress, Berkely, CA, USA, 2nd edition, 2014.

[6] Stephen H. Edwards and Manuel A. Perez-Quinones. Web-CAT: Automatically grading programming assignments. *SIGCSE Bull.*, 40(3):328–328, June 2008.

[7] J. Feliciano, M. Storey, and A. Zagalsky. Student experiences using GitHub in software engineering courses: A case study. In *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*, pages 422–431, May 2016.

[8] Terry Griffin and Shawn Seals. Github in the classroom: Not just for group projects. *J. Comput. Sci. Coll.*, 28(4):74–74, April 2013.

[9] Lassi Haaranen and Teemu Lehtinen. Teaching Git on the side: Version control system as a course platform. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '15, pages 87–92, New York, NY, USA, 2015. ACM.

[10] J. Kelleher. Employing Git in the classroom. In *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*, pages 1–4, Jan 2014.

[11] C. Kertész. Using GitHub in the classroom - a collaborative learning experience. In *2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pages 381–386, Oct 2015.

[12] Joseph Lawrance, Seikyung Jung, and Charles Wiseman. Git on the cloud in the classroom. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, pages 639–644, New York, NY, USA, 2013. ACM.

[13] Nachiappan Nagappan, Laurie Williams, Miriam Ferzli, Eric Wiebe, Kai Yang, Carol Miller, and Suzanne Balik. Improving the CS1 experience with pair programming. In *Proceeding of the 34th ACM Technical Symposium on Computer Science Education*, SIGCSE '03, pages 359–362, New York, NY, USA, 2003. ACM.

[14] J. Rodriguez, C. Apsey, S. Rees, T. Boudreau, and G. Raileanu. Courseware as code setting a new bar for transparency and collaboration. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–4, Oct 2018.

[15] Arnold Rosenbloom, Sadia Sharmin, and Andrew Wang. Git: Pedagogy, use and administration in undergraduate CS. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '17, pages 82–83, New York, NY, USA, 2017. ACM.

[16] Wikipedia contributors. Git — Wikipedia, the free encyclopedia, 2004. [Online; accessed 8-May-2019].

Comparison of Classical and CNN-based Detection Techniques for State Estimation in 2D*

Christopher Adleson and David C. Conner
Department of Physics, Computer Science and Engineering
Christopher Newport University
Newport News, VA 23606
{christopher.adleson.12,david.conner}@cnu.edu

Abstract

Tracking the state of a moving object is a challenging but useful application of computer vision. The problem is two-fold: efficiently and effectively locating the object within an image and deriving accurate state values despite measurement error or external interference. This paper investigates two state estimation methods for a radially symmetric object moving on a two-dimensional plane within fixed (x,y) boundaries; i.e., an air hockey puck and table. The first method uses classical object detection algorithms, while the second uses a specific deep learning-based Convolutional Neural Network (CNN) called YOLOv2. A state estimation algorithm was designed that, given consecutive images of the moving object, estimates the object's state, where state is the 2D position and velocity. The algorithm uses a Kalman filter to mitigate uncertainty. The aim of this comparison, is to examine the performance differences between classical and deep learning-based object detection in regards to this state estimation algorithm in order to determine a satisficing solution for object tracking.

1 Introduction

This project compares two state estimation methods that estimate the 2D state of a moving object, where state is defined as the object's x and y components of position and velocity at a given time: $[x, y, \dot{x}, \dot{y}]$. The object's movement is constrained to a fixed area, which in our case is an air-hockey table as shown in Figure 1. The table surface defines the 2D motion plane and an air-hockey puck is the object to be tracked. To capture the object's motion, a camera is centered above the field. Given each frame of camera data, each state estimation method must first detect the object to be tracked from the rest of the image before using that data to estimate the current state.

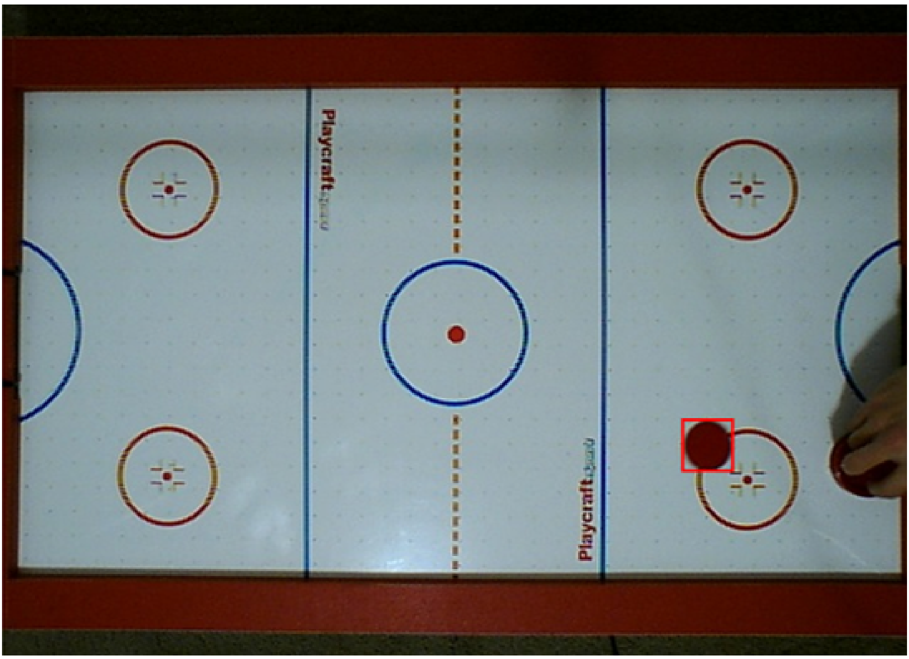


Figure 1: Image of air-hockey table with detected puck shown with outlined bounding box.

The first state estimation method uses a classical computer vision approach to object detection and the other uses a specific convolutional neural network (CNN) trained with deep learning. Though each method differs on their approach to object detection, they share the same state estimation algorithm. The algorithm is designed to handle cases of detection failure and collision; it uses a Kalman filter to refine the estimated state values.

The aim of this paper is to evaluate the differences between the two object detection approaches and how these differences affect the state estimation algorithm. In Section 2, we briefly describe the key topics and related works for this project. Section 3 highlights the key parts of the state estimation algorithm; Section 4 describes the results relative to defined metrics. Finally, Section 5 presents conclusions and recommendations for future work.

2 Background

This section is meant to offer a high-level overview of the algorithms and methods used. Details can be found in the cited resources. The topics covered are classical computer vision algorithms, Convolutional Neural Networks (CNN), and Kalman filtering.

2.1 Classical Computer Vision

These algorithms comprise the classical object detector and perform background subtraction, to remove extraneous detail from the image and blob detection to identify the moving foreground object. The classical algorithms are provided by OpenCV (Open Source Computer Vision Library), an extensive library, containing optimized algorithms for both classic and state-of-the-art computer vision [5]. The object detector initializes OpenCV's SimpleBlobDetector() which has been tuned to filter blobs by area or how many pixels comprise the blob, circularity or how close the blob's shape is to a circle, and a grayscale threshold [5]. The classical detector also initializes a background subtractor class, using OpenCV's background segmentation module (bgsegm), which will remove extraneous detail from the image before the SimpleBlobDetector() operates on it [5].

2.2 Convolutional Neural Networks

An (Artificial) Neural Network (ANN) is the type of model used in "deep learning" [2]; an ANN can be thought of as a hyper-parametric function that learns the mapping between a set of inputs and outputs. For this project, the inputs are pixel values and the output is the position of the puck. A Convolutional Neural Network (CNN) is a specific type of ANN, that specializes in handling image data as input. In a CNN, the network includes multiple sets of learned "kernel" weights that are used in convolution filters. YOLO is a state-of-the-art CNN that offers real-time object detection [8]. Using the deep learning framework, Darknet, provided by the author's of YOLO, the second iteration of YOLO (i.e. YOLOv2) was fine-tuned with custom training data for the purposes of this project [7, 8, 10]. The network receives a camera frame as input and outputs the object's coordinates.

2.3 Kalman filtering

The Kalman filter is an algorithm designed to improve estimates of state given uncertain or noisy measurement data[3, 9]. This project uses a linear Kalman filter, which follows the standard predict-correct cycle. The Kalman filter tracks the state estimate and an estimate of state uncertainty (covariance) given a linear state transition model, model of state transition uncertainty, and measurement uncertainty. For reference the Kalman filter equations and nomeclature are displayed in Table 1 and Table 2 respectively. Specific matrix values are discussed in Section 3.

Table 1: Linear Kalman Filter Equations

Prediction	Equation
State Vector	$\bar{\boldsymbol{x}}_{\boldsymbol{t}} = \boldsymbol{A}\boldsymbol{x}_{\boldsymbol{t-1}} + \boldsymbol{B}\boldsymbol{u}$
State Covariance	$\bar{\boldsymbol{P}}_{\boldsymbol{t}} = \boldsymbol{A}\boldsymbol{P}_{\boldsymbol{t-1}}\boldsymbol{A}^{\boldsymbol{T}} + \boldsymbol{Q}$
Correction	
Innovation	$\boldsymbol{y}_{\boldsymbol{t}} = \boldsymbol{z}_{\boldsymbol{t}} - \boldsymbol{H}\bar{\boldsymbol{x}}_{\boldsymbol{t}}$
Kalman Gain	$\boldsymbol{K}_{\boldsymbol{t}} = \bar{\boldsymbol{P}}_{\boldsymbol{t}}\boldsymbol{H}^{\boldsymbol{T}}(\boldsymbol{H}\bar{\boldsymbol{P}}_{\boldsymbol{t}}\boldsymbol{H}^{\boldsymbol{T}} + \boldsymbol{R})^{-1}$
State Vector (Updated)	$\boldsymbol{x}_{\boldsymbol{t}} = \bar{\boldsymbol{x}}_{\boldsymbol{t}} + \boldsymbol{K}_{\boldsymbol{t}}\boldsymbol{y}_{\boldsymbol{t}}$
State Covariance (Updated)	$\boldsymbol{P}_{\boldsymbol{t}} = (\boldsymbol{I} - \boldsymbol{K}_{\boldsymbol{t}}\boldsymbol{H})\bar{\boldsymbol{P}}_{\boldsymbol{t}}$

Table 2: Kalman Filter Nomenclature

Symbol	Definition
$\boldsymbol{x}_{\boldsymbol{t}}$	State Vector
$\boldsymbol{A}$	State Transition Matrix
$\boldsymbol{B}$	Control Matrix
$\boldsymbol{u}$	Control Vector
$\boldsymbol{Q}$	Process Covariance Matrix
$\boldsymbol{z}_{\boldsymbol{t}}$	Measurement Vector
$\boldsymbol{R}$	Measurement Covariance Matrix
$\boldsymbol{H}$	Observation Matrix
$\boldsymbol{I}$	Identity Matrix

2.4 Additional Approaches

The computer vision algorithms used are just some of many possible solutions to the tracking problem. Another classical technique considered was using dense optical flow to measure the distance traveled between frames. Details for optical flow as well as the other classical algorithms can be found in the

OpenCV documentation [5]. Early experiments with optical flow did not improve results, so this approach was not included in our final state estimation approach.

There are multiple versions of CNN, including multiple versions of YOLO, that have been created and have potential application [7]. Another option to consider is the deep learning model called a Recurrent Neural Network (RNN) which is designed to process sequential data such as image frames, in which time data or memory is incorporated as part of the model [4]. This RNN could use sequential images and time data to estimate velocity as well as object coordinates. This approach was not explored here.

3 Methods

We developed a state estimation algorithm that calculates the state of the puck for each frame and filters the raw state data through a Kalman filter, which follows a standard Predict-Measure-Correct cycle. This section highlights the key points of the state estimation algorithm. Before describing the state estimator algorithm, it is important to clarify terms (Table 3).

Table 3: Method Nomenclature

Symbol	Definition	Example
$\boldsymbol{x}$	State Vector	$\hat{\boldsymbol{x}}_k$, $\bar{\boldsymbol{x}}_k$, $\check{\boldsymbol{x}}_k$, $\tilde{\boldsymbol{x}}_{\boldsymbol{k}}$
$\hat{}$	State Measurement	$\hat{\boldsymbol{x}}_k = [\hat{x}, \hat{y}, \hat{\dot{x}}, \hat{\dot{y}}]_k$
$\bar{}$	State Prediction	$\bar{\boldsymbol{x}}_k = [\bar{x}, \bar{y}, \bar{\dot{x}}, \bar{\dot{y}}]_k$
$\tilde{}$	State Estimate (Kalman)	$\tilde{\boldsymbol{x}}_k = [\tilde{x}, \tilde{y}, \tilde{\dot{x}}, \tilde{\dot{y}}]_k$
$\check{}$	Ground Truth	$\check{\boldsymbol{x}}_k = [\check{x}, \check{y}, \check{\dot{x}}, \check{\dot{y}}]_k$
k	Current frame	$\hat{x}_{k-1}, \hat{x}_k, \hat{x}_{k+1}$ or $(\hat{x}, \hat{y})_k$
Δt	Time difference	$t_k - t_{k-1}$
$\Delta\hat{x}, \Delta\hat{y}$	Positional Difference	$\hat{x}_k - \hat{x}_{k-1}$

The state measurement is raw or unfiltered initial calculation of state based on the puck detection and the discrete-time approximation of velocity. State prediction refers to the expected state at k as it is predicted to be after the conclusion of the previous iteration $k-1$. State estimate is the filtered and corrected best estimate of state for a given frame. Both the state measurement and Kalman state estimate are tracked throughout the estimation process. The term, Δt, refers to the difference in timestamps between frames and terms $\Delta\hat{x}$ and $\Delta\hat{y}$ will always refer to the difference in measured position between frames $k-1$ and k.

The components of the Kalman filter that are essential to the state estimator are as follows:

The Control matrix $\boldsymbol{B} = \mathbf{0}$ and the Observation matrix $\boldsymbol{H} = \boldsymbol{I}$. The **A**, **Q**, and **R** parameters for the Kalman filter were chosen experimentally to work well for this project. More formal techniques can be used to improve the Kalman model [3]; however, this is left for future work.

State Transition matrix ($\boldsymbol{A}$): Since negative acceleration due to friction tends to cause a steady decline in velocity over time, it's considered a part of state transition rather than a control input. As friction is minimal, the velocity components of the proceeding time step were chosen to be 99% of their previous value, which was shown to give good results. Additionally, rather than use a constant time value, the State Transition matrix uses the time difference, Δt, between the current and previous frame. For state prediction, $\bar{\boldsymbol{x}}_{\boldsymbol{t}} = \boldsymbol{A}\boldsymbol{x}_{\boldsymbol{t-1}} + \boldsymbol{B}\boldsymbol{u}$, the control matrix $\boldsymbol{B}$ and control vector $\boldsymbol{u}$ are eliminated entirely because the only remaining control or user input to the system is player contact, in which case, the filtering process restarts.

Process Covariance matrix ($\boldsymbol{Q}$): The $\boldsymbol{Q}$ matrix represents the uncertainty increase due to a state transition step; the uncertainty always increases with the prediction step. We model the $\boldsymbol{Q}$ matrix with a single variable q along the main diagonal. This q value is modeled as a time varying quantity. Initially $q = 0.2$ after any collision, and it decreases with each Kalman estimation step or iteration to 90% its previous value and will continue to decrease until a minimum threshold of $q = 0.01$.[1] This can be interpreted as instilling more confidence in the filter's ability to estimate state while the puck is in free travel.

Measurement Covariance matrix ($\boldsymbol{R}$): The values of $\boldsymbol{R}$ should reflect the confidence of the value being measured. The current uncertainty for position and velocity are 0.01 and 0.3 respectively as position is rather certain relative to velocity. In this study, the Kalman Filter matrices are:

$$\underset{\mathbf{A}}{\begin{bmatrix} 1.0 & \Delta t & 0.0 & 0.0 \\ 0.0 & 1.0 & \Delta t & 0.0 \\ 0.0 & 0.0 & 0.99 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.99 \end{bmatrix}} \quad \underset{\mathbf{Q}}{\begin{bmatrix} 0.2 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.2 \end{bmatrix}} \quad \underset{\mathbf{R}}{\begin{bmatrix} 0.01 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.01 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.3 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.3 \end{bmatrix}}$$

The state estimation algorithm is comprised of four components: measurement, comparison, correction, and prediction.

[1] This minimum threshold is an adjustable parameter that functions well for the purposes of this project.

3.1 Measurement (Detection)

The Measurement step uses computer vision-based object detection to locate the puck in the image to determine the state positional components: $(\hat{x}, \hat{y})_k$. We compare two computer-vision-based object detectors. The classical detector uses background subtraction and blob detection; the deep learning detector uses YOLOv2.

The specified detector receives the current image and if a puck is detected it returns the position of the puck for that frame, $(\hat{x}, \hat{y})_k$. Assuming the previous frame, $k - 1$, also has a detection, and the elapsed time between frames is known, the algorithm can approximate velocity. As the proceeding steps rely on consecutive frames having a measure of position, for sake of explanation, presume this initialization has already occurred. If no detection occurs, then the Comparison and Correction steps are skipped.

3.2 Comparison

If the puck was detected, the Comparison step validates that the puck is valid for estimation; that is, consecutive detections are found and the puck is not in collision. The predicted position, $(\bar{x}, \bar{y})$, is compared with the current state measurement, $(\hat{x}, \hat{y})$, to determine whether player or boundary collisions have occurred in the interim. A large enough deviation between the current measured position $(\hat{x}, \hat{y})_k$ and predicted position $(\bar{x}, \bar{y})_k$ would suggest a collision [1]. A player collision is considered to be any external interference that intercepts and modifies the puck's current trajectory. A boundary collision is puck deflection off a playing field wall. If the measured position is within an acceptable threshold, no collision is suspected and the current velocity measurement is estimated as the average velocity between frames, $(\hat{\dot{x}}, \hat{\dot{y}})_k = (\frac{\Delta \hat{x}}{\Delta t}, \frac{\Delta \hat{y}}{\Delta t})$. Collisions require special handling; see [1] for more information.

3.3 Correction (Refinement)

Assuming consecutive detections, the Correction, or Refinement, step reduces the uncertainty within the raw state values, $[\hat{x}, \hat{y}, \hat{\dot{x}}, \hat{\dot{y}}]_k$, using a Kalman filter.

3.4 Prediction

The Prediction step estimates the puck's state for the next frame, $[\bar{x}, \bar{y}, \bar{\dot{x}}, \bar{\dot{y}}]_{k+1}$ using the current state estimate to predict the next state for use by the following comparison step. Predicting the location of the puck in the next frame, $(\hat{x}, \hat{y})_{k+1}$, enables the subsequent comparison step to verify player and boundary collisions.

In the early stages of this work, we separated the prediction step for comparison from the state transition-based prediction used by the Kalman filter; future work will unify these predictions. For the Kalman filter, $\bar{\boldsymbol{x}}_{\boldsymbol{t}} = \boldsymbol{A}\boldsymbol{x}_{\boldsymbol{t-1}}$ as there is no input after the initial impulse due to player collision. Given a player collision, the position is set to the position of first subsequent detection with significant uncertainty.

For our early Comparison step, we assumed that friction is minimal, which makes the velocity across the previous, current, and next frames relatively constant; therefore, the distance the puck will travel between frame k and $k+1$ will be similar, if not identical to, the distance traveled between the frame k and $k-1$. As such, these approximated travel distances where, $\Delta\hat{x} = \hat{x}_k - \hat{x}_{k-1}$ and $\Delta\hat{y} = \hat{y}_k - \hat{y}_{k-1}$, are used to predict the next position: $(\bar{x}, \bar{y})_{k+1} = (\hat{x}_k + \Delta\hat{x}, \hat{y}_k + \Delta\hat{y})$. Having established this preliminary prediction it is necessary to then check if a boundary collision will occur as this will alter the prediction. The state estimator expects the boundary of the playing area to be defined in which dimensions x and y have an upper and lower bound. Should $\bar{x}_{k+1}$ or $\bar{y}_{k+1}$ remain inside these boundaries the original prediction will remain unaltered. Otherwise, a reflection is calculated [1].

4 Validation and Results

After developing a system to capture images and provide them in sequence to our chosen methods, we evaluated the detection methods and the impact on overall estimate quality. We manually determined the puck positions for each image to establish ground truth position, and used the ground truth positions to estimate ground truth velocity. Ground truth velocity is considered to be the Kalman filtered velocity values derived from ground truth positions. We did not have an independent measurement of velocity.

4.1 Validation Metrics

Mean average precision (mAP) is the common metric used to measure the effectiveness of object detectors [6]. For this project it is re-purposed to also evaluate the quality of the velocity measurements [1]. The velocity measurements, $(\hat{x}, \hat{y})$, will be ranked from least to greatest error, where error is defined as the sum of squared errors between the method's raw unfiltered velocity measurements and ground truth velocity measurements $(\hat{\dot{x}} - \check{\dot{x}})^2 + (\hat{\dot{y}} - \check{\dot{y}})^2$.

The derived velocity values alone are not considered ground truth because they still maintain a degree of noise due to timestamp error and so the filtering is an effort to bring them even closer to their true values. Though bounding boxes in this case don't represent points in image space they still can be used as a measure of distance between measured and ground truth velocity values.

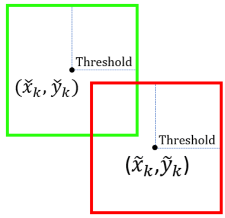


Figure 2: The green box or leftmost box represents ground truth. The width of each box is the error threshold.

The ground truth bounding boxes are determined by the ground truth velocity measurements $(\check{\dot{x}}, \check{\dot{y}})$ and an error threshold value. The measured bounding boxes are calculated in the same manner using the method's Kalman state velocity measurements $(\tilde{\dot{x}}, \tilde{\dot{y}})$. A set of velocity measurements will only be considered a true positive if the Intersection over Union (IoU) [6] is greater than 0.25, otherwise the velocities are considered too widely spread and are a false positive.

Since the error is between the raw value and ground truth and true positives are determined between the filtered value and ground truth, if the filter does a good job of mitigating noise, even as error increases precision will still remain high. In essence, the mAP for velocity measures how well the Kalman Filter performs as error between the raw measured velocities and ground truth velocities increases.

4.2 Training

The YOLO network requires application specific training. When training YOLO, the primary goal was to establish that the network could be trained to detect the puck's appearance. A large dataset was collected by taking a series of recordings in which the environmental conditions for each recording were varied slightly such as changes in lighting or adding shadows. Puck speed was varied for both straight trajectories and collisions ensuring that the network had both clear and blurry images of the puck to train from. Some of the images included obstructions that might be expected during gameplay such as a hand or the paddle. After data collection, which all together totaled around 21,000 frames, a subset of 2000 images were randomly selected to be used for training. Each of these 2000 samples were passed through our classical detector, and then were manually verified and corrected as needed to define the ground truth bounding box for each image. After splitting the 2000 labeled images into a train and test set, YOLO was fine tuned using Darknet.

4.3 Results

The evaluation results for position and velocity are summarized in Tables 4 and 5 respectively. The Ideal scenario refers to relatively slow and consistent puck movement with little to no external interference; The Realistic scenario refers to what could be considered to be average gameplay. The puck movement and speed varies greatly and player interference is expected in the form of puck-paddle collision.

Table 4: Detection Evaluation: mAP @ IoU

	Method AP			
	Ideal		Realistic	
IoU	CLASS	YOLO	CLASS	YOLO
0.2	88.59	95.61	86.77	13.68
0.3	88.59	94.97	86.77	13.13
0.4	88.55	92.07	86.71	12.64
0.5	88.55	85.91	86.59	11.51
0.6	87.39	67.25	86.40	8.01
0.7	86.99	27.21	86.21	2.32
0.8	80.98	3.40	83.72	0.11

Table 5: Velocity Evaluation: AP @ Velocity Error Threshold (pixels/sec)

	Method AP			
	Ideal		Realistic	
Error Thresh	CLASS	YOLO	CLASS	YOLO
100.0	90.81	72.29	93.11	8.27
90.0	90.78	69.86	93.08	7.19
80.0	90.78	67.61	93.08	6.28
70.0	90.78	63.82	93.01	5.47
60.0	90.74	58.35	92.97	4.84
50.0	90.66	52.08	92.83	3.82
40.0	90.58	42.63	92.67	2,78
30.0	90.54	30.37	92.37	1.71
20.0	90.21	14.55	92.17	0.48
10.0	89.50	1.99	91.50	0.04

In regards to the Ideal capture, YOLO has an advantage over Classical across lower IoU values, $IoU \leq 0.4$, since it is less sensitive to interference (Table 4). At lower IoUs, YOLO's detections do not have to be as accurate to be considered a true positive whereas for $IoU \geq 0.5$ its performance diminishes rapidly. Classical remains relatively constant across IoUs, only showing diminishing returns at an IoU of 0.8. In a realistic scenario, YOLO's high rate of false positives make it unable to compete with the mAP of Classical.

Similar trends are found in relation to the performance evaluation of velocity. This is most likely related to the accuracy of each method's object detector as velocity is derived from these detections. Classical mAP remains consistent across varying velocity error threshold values for the IoU values greater than 0.25. YOLO mAP varies widely, as seen by an Ideal best performance of 72.29 and a worst of 1.99 (Table 5). The error between the Classical method's raw measurements and ground truth are significantly smaller than that of YOLO. This difference in the magnitude of error means the Kalman filter will be able to better estimate velocity given raw Classical measurements rather than raw YOLO measurements. This means Classical will be able to maintain a higher mAP across varying thresholds because its velocity estimates differ from ground truth by only a small amount.

5 Conclusion

In summary, for both the Ideal and Realistic captures, Classical outperforms YOLO, in estimating both position and velocity. The high mAP values for Classical across IoU values is indicative of its ability to consistently and accurately locate the puck. Though YOLO is comparable in performance for the Ideal capture, its performance is limited by its more variable detections. For the Realistic capture YOLO's performance suffers from an overwhelming number of false positive detections. The Classical method is robust enough to detect the puck for a majority of frames despite occasional failures due to interference and it is un-matched when it comes to precision. Specific performance specifications and requirements will have to be developed in future work.

This preliminary study is not sufficient to prove that YOLO or other CNN-based techniques are always worse, as more data and better training may lead to better performance by those methods; it does show that we should not forget the classical methods.

References

[1] Christopher Adleson. A comparison of computer-vision-based state estimation methods for a 2-dimensional dynamical system. Master's thesis, Christopher Newport University, Newport News, VA, April 2019.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[3] Rodger Labbe. Kalman and Bayesian filters in Python. `https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python`, Jul 2017.

[4] Zachary Chase Lipton. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019, 2015.

[5] Open Source Computer Vision Library. Opencv. `https://opencv.org/`.

[6] Rafael Padilla. Metrics for object detection. `https://github.com/rafaelpadilla/Object-Detection-Metrics`, March 2019.

[7] Joseph Redmon. Darknet: Open source neural networks in c. `http://pjreddie.com/darknet/`, 2013–2016.

[8] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You Only Look Once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788, 2016.

[9] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[10] Nils Tijtgat. How to train YOLOv2 to detect custom objects. `https://timebutt.github.io/static/how-to-train-yolov2-to-detect-custom-objects/`, 2017.

Recursion Refined: Results From an ABET Continuous Improvement Cycle*

Jonathan M. Cazalas
Department of Computer Science
Florida Southern College
Lakeland, FL 33801
jcazalas@flsouthern.edu

Abstract

Traditional pedagogical approaches towards recursion leave much to be desired. This paper presents a retrospective analysis of a three-year continuous improvement cycle conducted in a Data Structures course, where recursion was one of the major course learning outcomes (CLOs) needing dramatic improvement. While less than one-quarter of the students were achieving the recursion CLO at the outset of the cycle, the proposed pedagogical changes resulted in over eighty percent achieving the CLO during the final two semesters. An empirical analysis of over one thousand undergraduates during the course of the study demonstrates the efficacy of these very easy and practical changes.

1 Introduction

Recursion is one of the most important, fundamental Computer Science concepts, while arguably also being the most notoriously difficult for beginning programmers.[3, 7, 9] Traditional pedagogical approaches towards recursion are too often teacher-centric, providing little engagement, and leaving students ill-equipped, forced to muddle through seemingly impossible programming assignments.[11, 12] Several works have examined the mental models of recursion, the classical approaches in teaching recursion, and even how those approaches have evolved using engagement and learner-centric strategies.[4, 11, 12] While these studies were valuable and enhanced our pedagogical approach to recursion, we arrive at a stark reality: recursion, like any other programming paradigm, requires practice. Said practice effectively translates to programming assignments and an often prohibitively large grading workload for instructors or teaching assistants. In practice, this is unsustainable, with many instructors relegating programming assignments to a small set of inferior exercises, leaving students with very little experience in solving meaningful problems requiring recursion.

After receiving ABET accreditation in 2013, all courses within the department were immediately enrolled in a continuous improvement cycle. Recursion was one of five highly-valued course learning outcomes (CLOs) for the CS3 course, Data Structures I. Direct assessment results after the first semester were bleak; the overwhelming majority of students were unable to achieve the recursion CLO, illustrating the need for dramatic and immediate change. Proposed changes were disseminated by all faculty members teaching the course, via end-of-semester reports, were discussed as a group prior to the start of the following semester, and were again assessed via direct assessment during the following semester's final examination. While this continuous cycle of improvement never stopped, this retrospective study examines a three year period from Fall 2014 to Spring 2017. The proposed pedagogical changes evolved over the course of the study, with the second and third year only having small refinements or enhancements to the new program. During the final two semesters, over eight percent of the students achieved the recursion CLO, demonstrating the efficacy of these very easy and practical changes.

Our contributions can be summarized as follows:

- We evolved our pedagogical approach to teaching recursion over a period of three years, maintaining full records of student examinations, instructor pedagogical recommendations for improvement, and how these recommendations were implemented in subsequent semesters.
- We provide an evaluation of the changes, how they were executed, and the impact on student success in the course.
- We provide a statistical analysis of the improvement, demonstrating the efficacy of the proposed changes.

The remainder of the paper is organized as follows. Section 2 reviews related work on recursion pedagogy. Section 3 details our proposed pedagogical changes. Results are discussed in Section 4, and we give concluding remarks in Section 5.

2 Related Work

The pedagogical changes implemented during the three-year cycle were informed by previous studies detailing best practices when teaching recursion.[2, 4, 8, 11, 12] Students often understand recursion as a looping model, similar to iteration, instead of the correct copies model.[7, 13] Several studies also investigated in-class experiments with varying levels of success.[1, 4, 6] That said, we found two studies whose proposed approaches resulted in a statistically significant and measurable improvement in both student engagement and success in the course, although both studies were published after the onset of our pedagogical changes, with the latter, RecurTutor[5], published in 2018. For completeness, we detail each below.

Cargo-Bot was introduced by Tessler et al.[6] to provide a structured, learner-centric approach that engaged students both outside the classroom and before they were formally presented with the material! Backwards course design was first used to identify course learning goals with respect to recursion. These learning goals were then mapped to a set of Cargo-Bot puzzles, which were assigned to students prior to direct classroom instruction. An empirical analysis from over two hundred students showed statistically significant improvement from those students who solved eight or more of the assigned Cargo-Bot puzzles.

RecurTutor is a web-based interactive tutorial for learning recursion.[5] Hamouda et al. present a simple, yet highly effective argument: recursion is best learned by practice. Practice equates to coding, which, from an instructor or teaching assistant perspective, translates to grading. In short, quality practice problems require significant time to manually grade. The result: recursion assignments are often limited in both size and scope in an effort to facilitate efficient assessment, resulting in students having limited experience in writing and tracing recursive code. RecurTutor was proposed to explicitly address these pitfalls. Through a web-based tutorial, students engage with the material, viewing a variety of examples and visualizations, and then solving several small-scale recursion exercises, which are automatically assessed by the system. Students who used the software performed demonstrably better on the recursion exam questions and exited the course with a high confidence level as evidenced by the indirect assessment questionnaires.

3 Details of Study

This retrospective study is based on a three-year continuous-improvement cycle at King Abdulaziz University. After attaining ABET accreditation in 2013, all courses were closely monitored based on both the direct assessment and indirect assessment of the articulated course learning outcomes (CLOs). Recursion was one of five highly-valued CLOs for our CS3 course, Data Structures I. Throughout the study, this CLO was assessed, via direct assessment, on the final exam of the semester. Questions included, but were not limited to, multiple-choice questions on their understanding of recursion, recursive tracing, and the coding of small recursive methods. Student confidence levels were also measured via indirect assessment on exit surveys at the end of the semester, as research has demonstrated valuable links between student confidence and both engagement and participation.[10] End-of-semester reporting was conducted by all faculty teaching the course as well as the course coordinator. Reporting included an objective analysis of the assessment data, a subjective evaluation of the results, and suggestions for improvement for the upcoming semester. At the outset of the study, the recursion CLO was achieved by only twenty-three percent of students, illustrating the need for dramatic change and improvement.

A formal meeting with course instructors and coordinators resulted in the following points of concern:

- The topic of recursion, along with the majority of other course topics, was overwhelming teacher-centric, lecture heavy, and with little student engagement.
- While students were encouraged to practice outside of class, little direction was given other than to review the course notes.
- The only meaningful hands-on experience students had was with a large, seemingly daunting programming assignment.
- Perhaps uniquely challenging was an enrollment of over 200 students per semester, partitioned into eight to ten individual sections, each taught by an independent instructor. Course content and pedagogical approaches were not unified among the sections, resulting in an often disparate student learning experience.

Suggestions for improvement were gathered via the end-of-semester reports and are shown in Table 1.

3.1 Evaluation

This recursion CLO was evaluated via direct assessment on the final exam. Questions were usually focused on recursive tracing and the coding of small recursive methods. For clarity, one such exam question would be as follows:

```
Write a method, sumEven(), which takes one positive int
number as a parameter, and then returns the sum of all
even digits in the number.
```

Table 1: Summary of Proposed Suggestions

Year Implemented	Suggestion
Year 1	Redesign the course notes to make them learner-centric, providing several opportunities of knowledge discovery.
Year 1	Instead of merely detailing the classical examples, engage the students by having them group-solve similar examples.
Year 1	Provide a two-hour tutor session (Lab), where students can practice writing smaller, recursive methods to solve a variety of problems.
Year 2	Spend a minimum of one week, both in class and in the lab, writing upwards of two dozen, small-scale recursive methods to solve simple exercises.
All years	The overall suggestion was also to fully unify the learning experience for students. All decisions regarding course content and pedagogy should be implemented by all sections and at all campuses.

A sample solution is shown below in Listing 1.

Listing 1: Example solution

```
public int sumEven(int number) {
    if (number == 0) {
        return 0;
    }
    else {
        if ((number % 10) % 2 == 0) {
            return number % 10 + sumEven(number/10);
        }
        else {
            return sumEven(number/10);
        }
    }
}
```

An analysis of the Fall 2014 and Spring 2015 direct assessment data did show incremental improvement based on the proposed Year 1 changes. However, as shown in Section 4, the vast majority of students were objectively unable to attain the recursion CLO. Anecdotally, the struggle was profound, as most were unable to write meaningful recursive code. Such was the motivation for the final suggestion itemized above, namely the need to spend at least one week, both in class and in the lab, engaging students while writing small-scale recursive methods to solve short exercises.

3.2 CodingBat to the Rescue!

One proposal was to create, from the ground up, a novel, interactive tutor, similar to the work published by Tessler et al.[6]. While the end result of their work is something to be proud of, our goal was to have a solution that could be easily duplicated and implemented by all sections of the course, and, by extension, other campuses and institutions. Revisiting the major crux of the problem, students need more practice, and they need timely feedback on their performance. With so many students and so many exercises to be completed, manual grading was not an option. After perusing online resources, CodingBat.com came out as the clear winner. For clarity, the use of CodingBat is not novel; in fact, many instructors have now been using it for years, some quite successfully. To the best of our knowledge, no research, or retrospective study, demonstrates the efficacy of this simple hands-on approach, one that can easily be leveraged by faculty at other institutions.

CodingBat has over thirty-five recursive problems to solve. The problems are short, clear, and provide several example inputs and outputs to facilitate understanding of the problem. Students can then code the solutions (usually less than fifteen lines) right in their browser and then have it tested against a set of judgment data. In the event that a solution fails, CodingBat provides feedback, alerting the student as to the error, the location, and even the reason! For the Fall 2015 and Spring 2016 semesters, students spent one week, in class and in the lab, solving these CodingBat recursion exercises. Individual problems at CodingBat were introduced by the instructor, discussed as a class, and then solved over a period of five to ten minutes each, depending on the difficulty. Improvements were dramatic as shown in the following section, but a passing level of achievement of the recursion CLO was still not found across all sections. Seeing the improvement and the vehicle for change, instructors immediately and unanimously agreed on the continuous problem: students still needed more practice! While they were solving problems in class and benefiting from CodingBat, there was little to no effort exerted outside of class. Accordingly, for the Fall 2016 and Spring 2017 semesters, completing any unfinished CodingBat exercises outside of class and the lab was no longer simply optional; students were required to solve all exercises, with a percentage of the lab grade allocated to CodingBat.

4 Results

The results are shown in Figure 1 below. Student performance was measured via direct assessment on the final exam by way of several recursion questions, usually focused on tracing and the coding of small methods. For consistency, during the course of the study, each final exam had one tracing question and one coding question. Results from the Fall 2014 and Spring 2015 semesters were bleak and dictated dramatic change. The percentage of students achieving the recursion CLO almost doubled during the second year with the use of CodingBat, although little structure was given to course instructors. When additional structure and guidance were provided to both instructors and students alike during the third year of the study, coupled with making completion of CodingBat a requirement and part of their final grade, students achieved the recursion CLO as measured with direct assessment.

The indirect assessment results and student confidence echoed that of the direct assessment results, with students having a perceived increase in confidence each year. Students were asked the following question on the exit survey, "I am confident in my ability to write recursive methods to solve short problems." The possible answers were Strongly Agree (5), Agree (4), Neither Agree nor Disagree (3), Disagree (2), and Strongly Disagree (1). Results over the course of the study are given in Figure 2. Students who took the course during the Spring 2017 semester had an average confidence of 4.3 out of 5, a dramatic increase from just two years earlier.

5 Conclusions

Recursion, like any other programming paradigm, requires practice. Said practice effectively translates to programming assignments and an often prohibitively large grading workload for instructors or teaching assistants, resulting in instructors relegating programming assignments to a small set of inferior exercises, leaving students with very little experience in solving meaningful problems requiring recursion. Our goal was clear: find or a develop a solution that encourages high levels of student engagement and practice, while leveraging auto-grade functionality for timely feedback. CodingBat, with its carefully crafted problems, responsive feedback, and auto-grade functionality, was perfectly suited for our needs. Further value was that it requires no third-party software, has minimal setup, and easily be applied at other institutions due to its free and readily available nature. An analysis of the direct assessment results

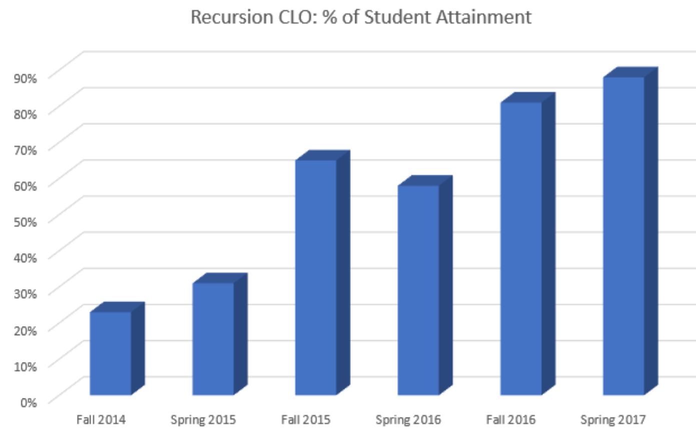


Figure 1: Results of Three-year Changes.

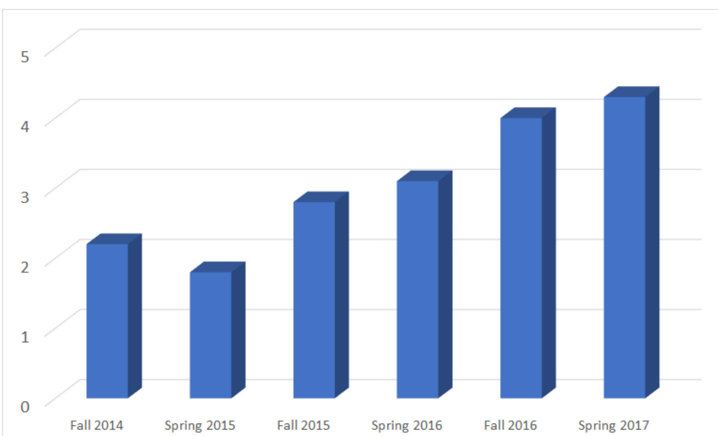


Figure 2: Indirect Assessment Results of Student Confidence Level

of the study indicates that the proposed pedagogical changes and structured use of CodingBat were paramount in both student success and engagement, and this was further echoed by indirect assessment feedback.

References

[1] Drew Hicks Amanda Chaffin, Katelyn Doran and Tiffany Barnes. Experimental evaluation of teaching recursion in a video game. In *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games*, Sandbox '09, pages 79–86, New York, NY, USA, 2009. ACM.

[2] Alan C. Benander and Barbara A. Benander. Student monks—teaching recursion in an IS or CS programming course using the Towers of Hanoi. *Journal of Information Systems Education*, 19(4):455–467, 2008.

[3] Nell B. Dale. Most difficult topics in CS1: Results of an online survey of educators. *SIGCSE Bulletin*, 38(2):49–53, 2006.

[4] David Ginat and Eyal Shifroni. Teaching recursion in a procedural environment—how much should we emphasize the computing model? *SIGCSE Bulletin*, 31(1):127–131, 1999.

[5] Sally Hamouda, Stephen H. Edwards, Hicham G. Elmongui, Jeremy V. Ernst, and Clifford A. Shaffer. RecurTutor: An interactive tutorial for learning recursion. *ACM Trans. Comput. Educ.*, 19(1):1:1–1:25, November 2018.

[6] Bradley Beth Joe Tessler and Calvin Lin. Using Cargo-Bot to provide contextualized learning of recursion. In *Proceedings of the 9th Annual International ACM Conference on International Computing Education Research*, ICER'13, pages 161–168, New York, NY, USA, 2013. ACM.

[7] Hank Kahney. What do novice programmers know about recursion? In E. Soloway and James C. Spohrer, editors, *Studying the Novice Programmer*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1989.

[8] Todd Milford Katherine Gunion and Ulrike Stege. Curing recursion aversion. *SIGCSE Bulletin*, 41(3):124–128, 2009.

[9] Claudius M. Kessler and John R. Anderson. Learning control flow: Recursive and iterative procedures. In E. Soloway and James C. Spohrer, editors, *Studying the Novice Programmer*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1989.

[10] M. Norman and T. Hyland. The role of confidence in lifelong learning. *Educational Studies*, 29(2-3):261–272, 2003.

[11] Sue Fitzgerald Renée McCauley, Scott Grissom and Laurie Murphy. Teaching and learning recursive programming: A review of the research literature. *Computer Science Education*, 25(1):37–66, 2015.

[12] Christian Rinderknecht. A survey on teaching and learning recursive programming. *Informatics in Education*, 13(1):87–119, 2014.

[13] Tamarisk Lurlyn Scholtz and Ian Sanders. Mental models of recursion: Investigating students' understanding of recursion. In *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE '10, pages 103–107, New York, NY, USA, 2010. ACM.

SQL: An Introduction to SQL Lesson and Hands-On Lab*

Gayle Jesse
West Liberty University
West Liberty, WV, 26074
`gayle.jesse@gmail.com`

Abstract

We live in a world run by databases. Thus, knowing Structure Query Language (SQL) is vital to extract data from a database. The following teaching case is a hands-on introduction to SQL lab activity based on a scenario of working at a help desk for Foods, Inc using W3 Schools. W3 schools web-based environment is stable, consistent, and allows numerous users at one time without ever altering the database. Students begin the case by learning the background information, database structure, and working through a teaching lesson. The lesson teaches students the basics of writing queries from basic select statements through joins. After students complete the lesson on querying basics they are ready to begin the assignment. The assignment requires students to write 25 queries which, are to be recorded in an answer template designed for easier grading and reduction of cheating.

1 Background

A database is an organized collection of data that consists of tables, queries, views, reports, and other objects. Structure Query Language (SQL) is a standard language for storing, manipulating and retrieving data in databases (W3 Schools, 2018). As stated in the abstract, W3 schools web-based environment is stable, consistent, and allows numerous users at one time without ever altering the database. The database can easily be restored by clicking the "Restore

Database" button. Meaning each user gets a separate instance they can reset at any time.

In this lab, you are a recently hired database professional working at a helpdesk for Foods, Inc., a grocery store distributor located in Manhattan. Foods, Inc, is a specialty store selling a variety of unique upscale imported food. From coffee to tofu as well as deserts to dairy products. Currently, there are twenty-nine (29) suppliers for Food, Inc. located worldwide with several shippers. Foods, Inc. employs 10 people with an average of 100 customers. As a help desk employee, it is your responsibility to answer questions from customers, suppliers, and internal managers of your company by querying your Foods, Inc.'s database using SQL.

2 Database Structure

Foods, Inc.'s database consists of eight (8) tables that are already created and populated with data.

1. Customers
2. Categories
3. Employees
4. OrderDetails
5. Orders
6. Products
7. Shippers
8. Suppliers

3 Lesson: Querying Basics

To access your database, click on the following link: `http://www.w3schools.com/sql/`

W3schools.com is a site for web developers that provides tutorials on everything from creating web pages to database programming in SQL.

To query (i.e. ask a question) from a database, you must type in a command in a SQL editor. W3schools.com provides its own built in SQL editor that you can access by clicking on the "Try it yourself" button as shown in the Appendix Figure 1. The basic format of a query is as follows:

```
SELECT [fieldname]
FROM [tablename]
[WHERE condition]
```

Note: Where clause is optional.

For example, to view every customer's name in the Customers table, you would enter the command as shown below:

```
SELECT CustomerName
FROM Customers;
```

Clicking the **Run SQL»** button causes the SQL editor to execute your query and display the results in the lower half of the screen.

Before beginning this exercise, I encourage you to click on each of the table names in the upper right-hand corner of your screen to familiarize yourself with the structure of each table. See Appendix Figure 2.

SELECT clause

To view more than one column (field) of data in a table, you can specify the fieldnames in the Select clause, separated by commas:

```
SELECT CustomerName, City, Country
FROM Customers;
```

To view all of the columns in a table, you can use the asterisk (*) as a wildcard character that represents all columns:

```
SELECT *
FROM Customers;
```

Aliases in the SELECT clause

Did you notice that CustomerName looks really "techie"? If you were to create a report that displayed CustomerName as the title of a column, the report would not look very professional. Instead, you can make your query display this field as a different name (i.e. an "alias") using the keyword "AS":

```
SELECT CustomerName AS [Customer Name]
FROM Customers;
```

Note: If you have a space in your alias, you must enclose the alias in square brackets.

DISTINCT keyword in the Select Clause

Sometimes a table includes multiple occurrences of the same value in a field (not a primary key field, of course), and you only want to see each occurrence listed one time in your output. Assume we want to see a unique listing of all of the countries in the Customers table. We can use the DISTINCT keyword to ensure that every country is only listed once:

```
SELECT DISTINCT Country
FROM Customers;
```

Aggregate operators in the Select Clause

In many queries, you are not interested in the actual values of each individual row in a table, but rather a summary of them. For example, if someone asked you how many customers had ever ordered from you, you could use the COUNT operator in your select clause. For example:

```
SELECT COUNT(CustomerID)
FROM Customers;
```

This query would return the count (or the total) number of rows in the Customers table.

Note that whenever you specify any other field in the Select clause other than the item that you are aggregating, you must also include a GROUP BY clause at the bottom of your query. For example, assume you want to know how many customers are in each country in your table, you can count the number of customers and group them by Country:

```
SELECT Country, Count(CustomerID) as [Customers by Country]
FROM Customers
GROUP BY Country;
```

Count is not the only aggregate operator available to you. SUM, AVG, MIN, and MAX are a few examples of others that you can use. Note, however, that statistical functions (e.g. SUM, AVG, etc.) can only be used on fields that contain numbers and are defined with a numeric data type. For instance, it would not make sense to SUM the customer name field. However, it would make sense to SUM the Quantity of a particular product in the OrderDetails table:

```
SELECT Sum(Quantity)
FROM OrderDetails
WHERE Productid = 11;
```

This query produces the sum of the quantity field for the product whose ID is 11. Note that this query includes the use of a WHERE clause (discussed next).

WHERE Clause

What if you want to view only those customers that meet a certain criterion? This is where the “Where” clause comes in. Let’s assume you want to view customers that live in Berlin:

```
SELECT *
FROM Customers
WHERE Country = "France";
```

Multiple conditions in a WHERE clause

What if you want to narrow down your results even more? Let's assume you want to view customers that live in the country of France AND the city of Paris?

```
SELECT *
FROM Customers
WHERE Country = "France" AND City = "Paris";
```

What if you want to expand your results to include everyone in France OR the USA?

```
SELECT *
FROM Customers
WHERE Country = "France" OR Country = "USA";
```

Note that you can combine AND's and OR's as much as needed to satisfy your question.

IN keyword

Assume you want to specify an elaborate OR condition like this:

```
SELECT *
FROM Customers
WHERE City = "Portland" OR City = "Paris"
  OR City = "San␣Francisco" OR City = "Boise"
  OR City = "London" OR City = "Madrid" OR city = "Walla"
  OR City = "Cork" OR City = "Berlin";
```

SQL gives you a shortcut that eliminates much of the typing. This shortcut uses the keyword "IN". The list of items following the IN keyword is the list of items for which you are filtering:

```
SELECT *
FROM Customers
WHERE City IN("Portland", "Paris", "San␣Francisco", "Boise",
  "London", "Madrid", "Walla", "Cork", "Berlin");
```

Pattern matches

In our previous queries, we have been searching for records that meet a specific criterion, such as City = “Portland” or Country = “France”, etc. What if, however, we do not know exactly how a particular item is spelled? Instead of searching for an “exact” match, we can look for a “pattern”. For example, I want to find a customer whose name is “Ricardo” but I don’t his last name. I can use the following query to extract all of the records whose customer name starts with “Ricardo”:

```
SELECT *
FROM Customers
WHERE CustomerName LIKE "Ricardo%";
```

Note: The use of the percent (%) sign. This symbol stands for any character or sequence of characters.

ORDER BY clause

Assume you want to sort your results in a particular order. You can use the ORDER BY clause to do this. You can specify whether you want the results to be in ascending order (A-Z) with the “ASC” argument or in descending order (Z-A) with the “DESC” argument. If you leave out the argument, SQL assumes you mean ascending order.

```
SELECT *
FROM Customers
WHERE Country = "USA"
ORDER BY CustomerName ASC;
```

You can even sort by more than one field at a time. Because there are multiple cities and countries listed in the Customers table, you could sort first by Country in descending order and then by City in ascending order with the following query:

```
SELECT *
FROM Customers
ORDER BY Country DESC, City ASC;
```

Note: ASC does not have to be listed here as an argument because SQL assumes that if there is no argument specified, it will sort that field in ascending order.

JOINING tables

In all of our previous examples, we have been querying a single table, Customers. It is obvious, however, that our one Customers table does not give us the whole picture. In order to get the "bigger picture", we need to be able to combine data from multiple tables.

The way this is accomplished is by "joining" tables based on fields that they have in common. For example, click on the Products table. Notice that it has the following fields in it:

```
ProductID, ProductName, SupplierID, CategoryID, Unit, Price.
```

Now, click on the Categories table. Notice that it has the following fields in it:

```
CategoryID, CategoryName, Description
```

What field exists in both tables? The CategoryID field, right? Therefore, we can write a query that will allow us to retrieve data out of both tables by joining both tables on that common field:

```
SELECT *
FROM Products
INNER JOIN Categories ON
  Categories.CategoryID = Products.CategoryID;
```

Note the keywords "INNER JOIN" above. These keywords are necessary to tell SQL that the two tables are to be linked based this common field.

There are other types of joins in database terminology (e.g. Left Outer Joins, Right Outer Joins, and Full Joins), but those are beyond the scope of this assignment.

The preferred way of joining tables in today's database industry is by using the keywords "inner join", "outer join", etc. In this assignment, I want you to learn an older way of performing inner joins in case you ever run into them in industry.

An inner join can be accomplished in one of two ways: in the FROM clause (as we illustrated above) or in the WHERE clause.

The WHERE clause method allows you to specify the all of the tables to be included in your FROM clause, but the joins are specified in the WHERE clause. For example, assume I want to join the products, orders, and orderdetails tables. I specify all three of them in the FROM clause, but I join them on their common fields in the WHERE clause as follows:

```
SELECT *
FROM Products, Orders, OrderDetails
WHERE Products.ProductID = OrderDetails.ProductID
  AND OrderDetails.OrderID = Orders.OrderID;
```

If you are having difficulty understanding the concept of joins, (or any of the commands presented above), w3schools.com does an excellent job of explaining them. To view the SQL command tutorials, click on any of the SQL Tutorial topics on the left hand side of this screen: `http://www.w3schools.com/sql/`

4 Assignment

Scenario

You are working at the helpdesk of Foods, Inc. and are given the task of answering questions using the database we have been querying during the previous lesson on querying basics. The 25 questions you have received and need to answer are as follows.

This is your first job working with databases, so your manager wants you to familiarize yourself with all of the tables in the database. Therefore, he asks you to produce a query for each of the eight (8) tables displaying all of the columns (fields) and all of the rows (records) using a separate SELECT query for each table. The eight (8) tables are listed below.

25 SQL Questions

1. Customers To get you started, here is the first query.

```
SELECT *
FROM Customers
```

Instructions:

- Download the Answer Template.
- Save as: YourFirstName_YourLastName_IntroSQLLab.
- List all 25 queries and their results in the Answer Template (Appendix Figure 3) that is provided for you.
- Paste a Screen Capture of output; Screen Capture must include three items: SQL Statement, Results, and computer date and time (Appendix Figure 4). Note: You only need to display the first screen of output for each query.

2. Categories
3. Employees
4. OrderDetails
5. Orders
6. Products
7. Shippers
8. Suppliers

After familiarizing yourself with the eight (8) tables, your manager knows that you can handle the remaining requests that come in to the helpdesk.

For each of the remaining items, use the same format for your queries and results as you did in the previous 8 questions and continue putting your answers in the Answer Template provided.

9. The manager of Human Resources wants a list of all of the employees who work at the company. He only wants to see the employees' last name and first name. [You may assume that everyone who works at the company is included in the Employees table].

10. Your manager asks you for a list of the product names that the company sells. He does not want everything related to each product – only each product's name. He wants them sorted alphabetically (in descending order from Z-A). [The Products table contains all of the products that are sold by the company.]

11. The manager of the Purchasing department needs the phone number of the United Package shipper.

12. The Accounting department manager requests a list of all of the suppliers, contacts, and phone numbers of the suppliers in Japan.

13. A customer calls in and wants a list of all of your products and their prices. Your results should be sorted in alphabetical order from A-Z.

14. Another customer calls in requesting all of your seafood products. Note that he only wants the name of the product in his report – not all of the fields. [Hint: You will need to join the Products table to the Categories table to get this information. Your WHERE clause should include the criterion WHERE CategoryName = "Seafood".]

15. The manager of the accounting department needs a list of all of the suppliers (Supplier name only) in either the USA or the UK. [Hint: You must use the OR operator in your WHERE clause to receive credit for this question.]

16. The Accounts Receivable clerk needs a list of all the customers (including customer names, addresses, city, country, and postal codes) who are located in Germany, France, or Spain. [Hint: You must use the IN operator in your WHERE clause to receive credit for this question.]

17. Just out of curiosity, you want to produce a listing of the products (product name only) that are supplied by SupplierID 1 and are in CategoryID 1.

18. The Human Resources department requests a list that contains the number (i.e. count) of orders taken on each date, grouped by the order date. [Be sure to use an aggregate operator in your query to receive credit for this question.]

19. One of the employees in the IT department asks for the highest order ID that is in the Orders table. [Hint: To receive credit for this question, you must use the MAX function to perform this task.]

20. To double check your results from the previous question, you decide to run a query that returns all of the rows and columns from the Orders table sorted in descending order. If you see that the order ID of the first record in your

output equals the answer you obtained in your previous question, you know you were correct.

21. Your boss wants to know the total quantity of Boston Crab Meat that has ever been ordered. [Hint: You will have to join the OrderDetails and Products tables. Your WHERE clause should contain the criterion ProductName = "Boston Crab Meat", and you will have to use the SUM function in your SELECT clause.]

22. The manager of your Customer Relationship department wants a list of all of the customers whose name begins with the letter “D”.

23. Speedy Express, one of your shippers, wants to know how many orders it has ever shipped for you. [Hint: You need to join the Shippers and Orders tables and include ShipperName = "Speedy Express" in the WHERE clause. Remember to use an aggregate operator to receive full credit for this question.]

24. Someone from the tax department wants a unique list of all of the customers who have ever placed an order with the company. He only needs the customer name, sorted from A-Z. [Hint: You will need to join the Customers table with the Orders table because it is possible that a customer may exist in the customers table who has not yet placed an order with your company. Joining the two tables together with an inner join ensures that your list of customers only includes those customerid’s that exist in both tables (i.e. customers who have actually placed an order).]

25. The president of the company wants a list of all orders ever taken. He wants to see the customer name, the last name of the employee who took the order, the shipper who shipped the order, the product that was ordered, the quantity that was ordered, and the date on which the order was placed. [Hint: You will need to join the following tables: Customers, Employees, Shippers, Orders, OrderDetails, Products, and get all of the necessary information.]

Congratulations!!! You have just helped numerous people inside and outside of your company to solve all kinds of problems! You deserve a raise!

Deliverables

Submit your completed Answer Template to your professor by the due date.

References

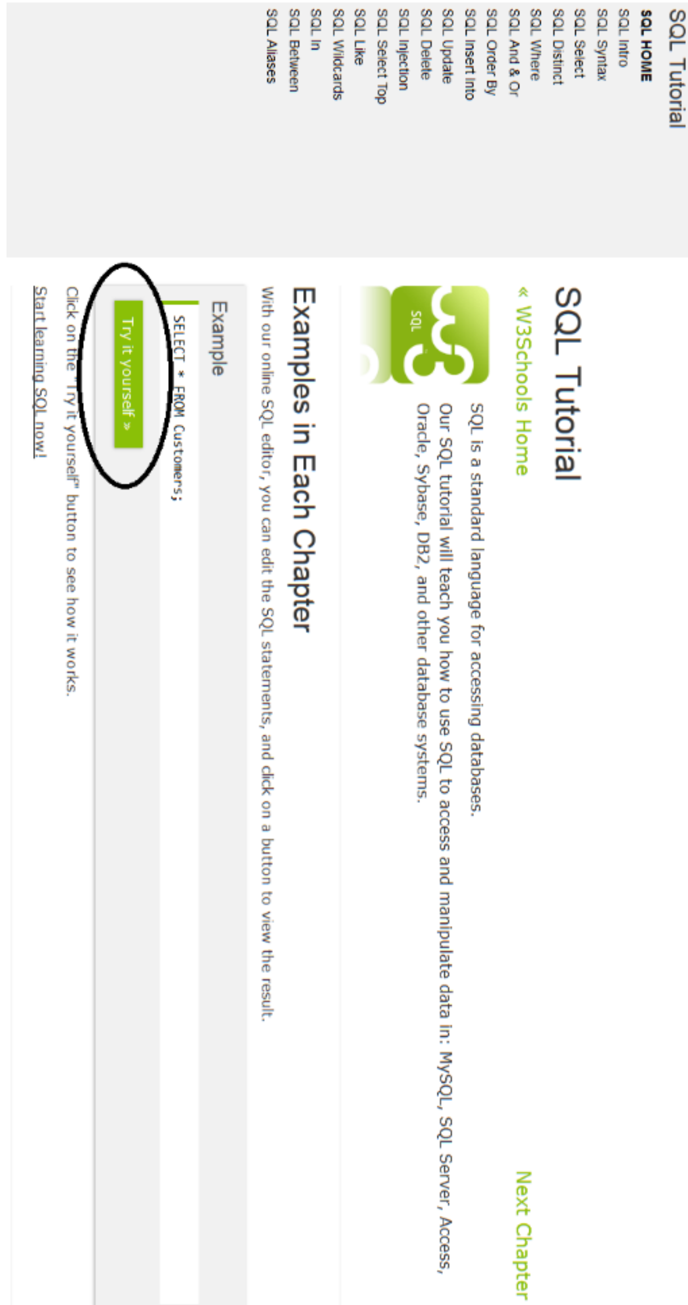

SQL Tutorial
SQL HOME
SQL Intro
SQL Syntax
SQL Select
SQL Distinct
SQL Where
SQL And & Or
SQL Order By
SQL Insert Into
SQL Update
SQL Delete
SQL Injection
SQL Select Top
SQL Like
SQL Wildcards
SQL In
SQL Between
SQL Aliases
SQL Tutorial
« W3Schools Home
Next Chapter »
SQL
SQL is a standard language for accessing databases.
Our SQL tutorial will teach you how to use SQL to access and manipulate data in: MySQL, SQL Server, Access, Oracle, Sybase, DB2, and other database systems.
Examples in Each Chapter
With our online SQL editor, you can edit the SQL statements, and click on a button to view the result.
Example
SELECT * FROM Customers;
Try it yourself »
Click on the "Try it yourself" button to see how it works.
Start learning SQL now!

Figure 1

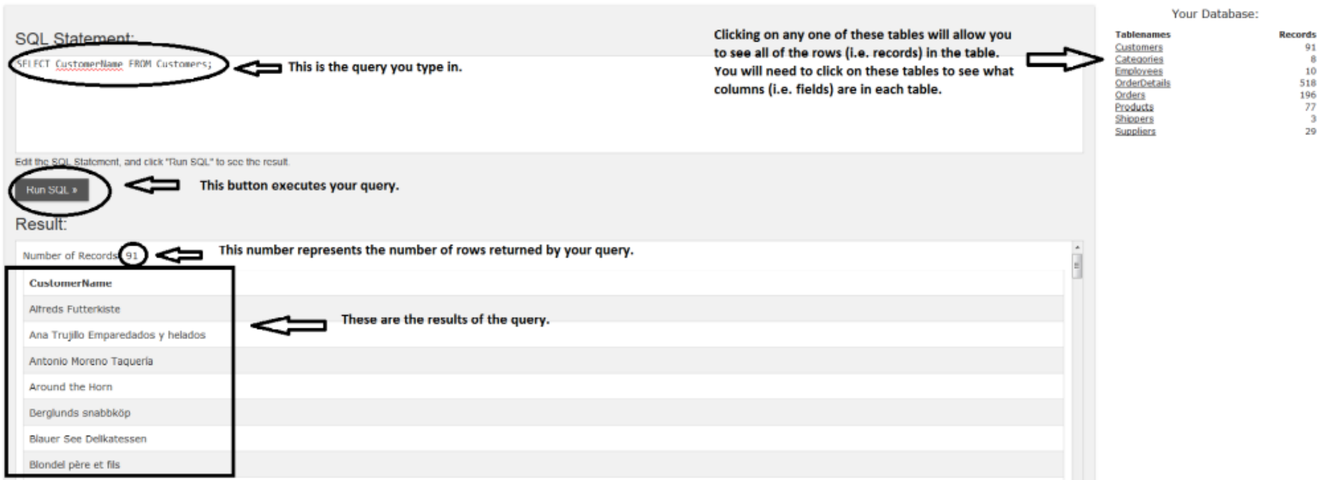
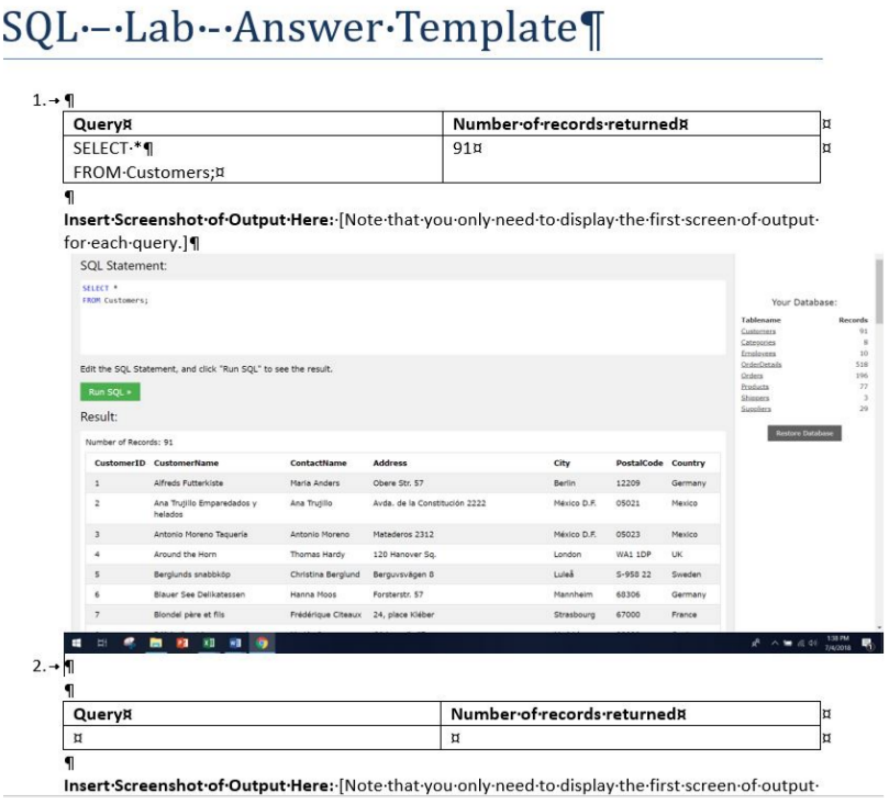

SQL Statement:
SELECT CustomerName FROM Customers;
This is the query you type in.
Clicking on any one of these tables will allow you to see all of the rows (i.e. records) in the table. You will need to click on these tables to see what columns (i.e. fields) are in each table.
Your Database:
Tablenames
Records
Customers 91
Categories 8
Employees 10
OrderDetails 518
Orders 196
Products 77
Shippers 3
Suppliers 29
Edit the SQL Statement, and click "Run SQL" to see the result.
Run SQL »
This button executes your query.
Result:
Number of Records: 91
This number represents the number of rows returned by your query.
CustomerName
Alfreds Futterkiste
Ana Trujillo Emparedados y helados
Antonio Moreno Taquería
Around the Horn
Berglunds snabbköp
Blauer See Delikatessen
Blondel père et fils
These are the results of the query.

Figure 2



SQL – Lab – Answer Template

1.

Query	Number of records returned
SELECT * FROM Customers;	91

Insert Screenshot of Output Here: [Note that you only need to display the first screen of output for each query.]

SQL Statement:

```
SELECT *
FROM Customers;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 91

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
6	Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306	Germany
7	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

Your Database:

Tablename	Records
Customers	91
Categories	8
Employees	10
OrderDetails	518
Orders	196
Products	77
Shippers	3
Suppliers	29

Restore Database

2.

Query	Number of records returned

Insert Screenshot of Output Here: [Note that you only need to display the first screen of output

Figure 3

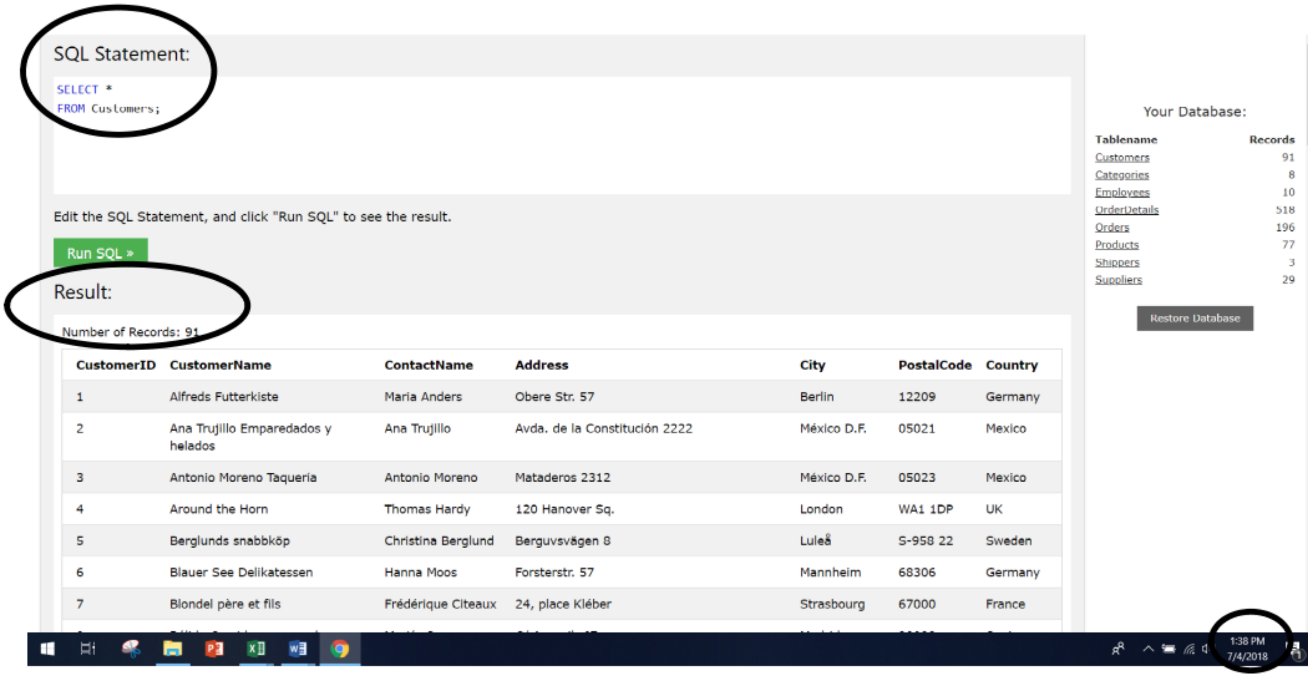

SQL Statement:
SELECT *
FROM Customers;
Edit the SQL Statement, and click "Run SQL" to see the result.
Run SQL »
Result:
Number of Records: 91
CustomerID CustomerName ContactName Address City PostalCode Country
1 Alfreds Futterkiste Maria Anders Obere Str. 57 Berlin 12209 Germany
2 Ana Trujillo Emparedados y helados Ana Trujillo Avda. de la Constitución 2222 México D.F. 05021 Mexico
3 Antonio Moreno Taquería Antonio Moreno Mataderos 2312 México D.F. 05023 Mexico
4 Around the Horn Thomas Hardy 120 Hanover Sq. London WA1 1DP UK
5 Berglunds snabbköp Christina Berglund Berguvsvägen 8 Luleå S-958 22 Sweden
6 Blauer See Delikatessen Hanna Moos Forsterstr. 57 Mannheim 68306 Germany
7 Blondel père et fils Frédérique Citeaux 24, place Kléber Strasbourg 67000 France
Your Database:
Tablename Records
Customers 91
Categories 8
Employees 10
OrderDetails 518
Orders 196
Products 77
Shippers 3
Suppliers 29
Restore Database
1:38 PM
7/4/2018

Figure 4

A Flipped Classroom Approach for Engaged Learning using Arduino and LabVIEW*

Benjamin R. Campbell
School of Engineering, Mathematics and Science
Robert Morris University
Moon Township, PA 15108
campbellb@rmu.edu

Abstract

A course called "Signals and Sensors" has been designed and taught to upper level engineering students using the flipped classroom approach to teach software and hardware integration concepts for mechatronics (the intersection of electrical, software and mechanical engineering). The course does not use a textbook, rather the students invest in a lab kit that includes an Arduino Uno with a set of 37 sensors and other I/O devices. LabVIEW is introduced as a graphical programming language in contrast to the Arduino's text based code. Once both technologies are explored, the Arduino is used as a low cost data acquisition device for LabVIEW projects which require hardware to interface with software. Students complete pre-labs outside of class to learn the basics of using each hardware component or software concept. In class there is a short lecture on a topic relevant to that week's lab and a recap of what they should have accomplished in pre-lab. Using the basic skills learned in the pre-lab, the students are asked to complete a more difficult in-class project, often combining multiple pre-lab concepts. The in-class projects can be individual or group, and are due prior to the next class if not finished in the allotted class time. More complex projects may span several weeks and have weekly design reviews and demonstrations, presenting to the entire class. Final projects are designed by the students with guidance from the instructor to allow students the freedom to

creatively explore applications of the software and hardware they have learned using the devices of their choice. Based on the student feedback, this teaching approach has been highly successful, with evidence showing it is both educational and engaging.

1 Background

Students studying engineering and computer information systems need to have a grasp of technology at a systems level with software and hardware seamlessly working together. Beyond designing and building these systems, students need to develop the skill set to troubleshoot and find solutions to unique problems. Teaching these skills requires instructor supervision and availability for questions as (seemingly) insurmountable problems arise. The flipped classroom model provides a framework to move the basic instruction into the homework before class and allow the advanced building and testing of systems to occur in the classroom under supervision, with readily available assistance from the instructor and from peers. A flipped classroom is defined as one which has "interactive group learning activities inside the classroom and direct computer-based individual instruction outside the classroom [1]." This often results in the traditional lecture being delivered outside of class and filling class time with interactive learning activities [2]. The concept of a flipped classroom in engineering education has only been popular for the past decade, with many studies conducted to document course design and share student feedback [5]. The process of flipping the classroom has challenges; the students who are not used to the format often resist it and the instructor can have difficulty in finding or developing appropriate content to be learned individually outside of class [3]. Another challenge students face is self-regulating their schedules and study times to be adequately prepared for class, since more burden is placed upon them to learn the basic concepts on their own [12]. In general, studies of flipped classrooms have found they "brought positive impacts toward students' learning activities such as achievement, motivation, engagement, and interaction [11]." In upper level engineering classes it has been found that flipped classroom approaches have allowed instructors to cover more material while student performance stayed the same or improved over a traditional classroom format [7]. A review of 24 studies pertaining to flipped classrooms in engineering showed high student satisfaction and increased performance [6]. At Purdue University, two undergraduate courses in systems measurements and controls were successfully flipped making use of the NI (National Instruments) myRIO device and LabVIEW [10] but the cost of the myRIO can make this approach prohibitive for the students or the school. The Universidade Federal do Maranhão in Brazil developed a flipped approach to teach Instrumentation using LabVIEW and the ELVIS II (Engineering Laboratory Virtual Instrumentation Suite) platform following closely with Bloom's taxonomy using webinars in their "before-class stage", guided activities for their "in-class stage", and then enhanced challenges in an "after-class stage" for undergraduates in engineering [8]. Mechatronics courses that have aspects of electrical, software, and mechanical engineering require mastery of a diverse set of topics to understand each portion of a system. At Deakin University a mechatronics class for seniors which uses project oriented, design based learning was flipped, but learning was differentiated requiring one student on each team to become an expert on a randomly chosen aspect of the project and that student is responsible for educating others on the team [4]. A graduate course at Chalmers University in Sweden called "Signals and Systems" flipped their instruction using the 5E model, assigning videos outside class for engagement, exploration and explanation while the class time was reserved for elaboration and evaluation; the student feedback was overwhelmingly positive [9]. Although flipped classrooms are a relatively new initiative in engineering education, the results to date have shown that the change from traditional lecturing has many positive benefits to enhance learning and results in more active education and engagement of the students.

2 Methods

This course was originally designed for a mechatronics minor at Robert Morris University and also serves as an elective for engineering majors within a general engineering program. It was named "Signals and Sensors" to convey that the course is built around detecting information and encoding it as electrical signals, but the broader underlying concepts are system design and troubleshooting. Students learn about different types of transducers (devices that convert one type of energy into another type of energy) and their applications. They are taught about analog and digital signals, sampling rates, signal processing, and decision making based on input signals (Boolean and threshold functions). They apply these lessons though text-based and graphically-based programming. The only prerequisites are a general circuits course and an introductory programming course (Java or C++). Students purchase an Arduino Uno and Sunfounder Sensor Kit [13] as seen in Figure 1 for about $80. The course descriptions is:

> For this course, knowledge of basic circuits and computer programming will be needed as a foundation for exploring additional topics relating to sensors generating signals. The course will be lab based, with the majority of the time spent constructing, testing, and debugging systems that use sensors to collect information and

> writing programs to perform signal processing, analysis and decision making. Students will learn to use LabVIEW to create virtual instruments and work with microcontrollers and sensors for data acquisition and device control. Topics to be covered include but are not limited to:
>
> - LabVIEW graphical programming environment
> - Arduino hardware and software
> - Signals types and signal processing/analysis
> - Various types of transducers/sensors
> - Joystick control, keypads, remotes
> - LCD displays
> - Motor control
>
> In addition, various laboratory techniques will be used to aid in the learning process using Arduino microcontrollers and the National Instruments LabVIEW systems and circuit simulations using the National Instruments Circuit Design Suite: Multisim. Grades will be based on attendance, homework, and labs/projects with an emphasis on the lab activities.

The course was designed to meet weekly in a 4 hour block during a 14 week semester followed by a two hour meeting in finals week. There is a weekly pre-lab assignment introducing new concepts based on the Arduino platform and LabVIEW software (done separately). The student have to complete the task and submit documentation a day before the class to give the instructor time to review the submissions. The first hour of the class is dedicated to review of previous material and pre-labs and the introduction of new lecture topics. For the first nine weeks of the semester the remaining 3 hours is usually divided between 90 minutes for an Arduino Lab activity and 90 minutes for a LabVIEW lab activity (or a single three hour activity). If not finished in class, these in-class activities become homework which are due before the next class. For the final six weeks of the semester, lecture topics are usually replaced with individual or group student presentations on topics they have researched (Applications of LabVIEW, types of Add-ins for LabVIEW, Types of Shields for Arduino, project proposals, and project design reviews. The remaining three hours is then for project work, usually unifying Arduino and LabVIEW through the LINX add-in for LabVIEW. For the last class meeting during the final exam week, and in lieu of an exam, the students present and demonstrate their final projects. The entire semester of Arduino projects from the original course proposal can be seen in Table 1 with the desired education outcomes from each activity.

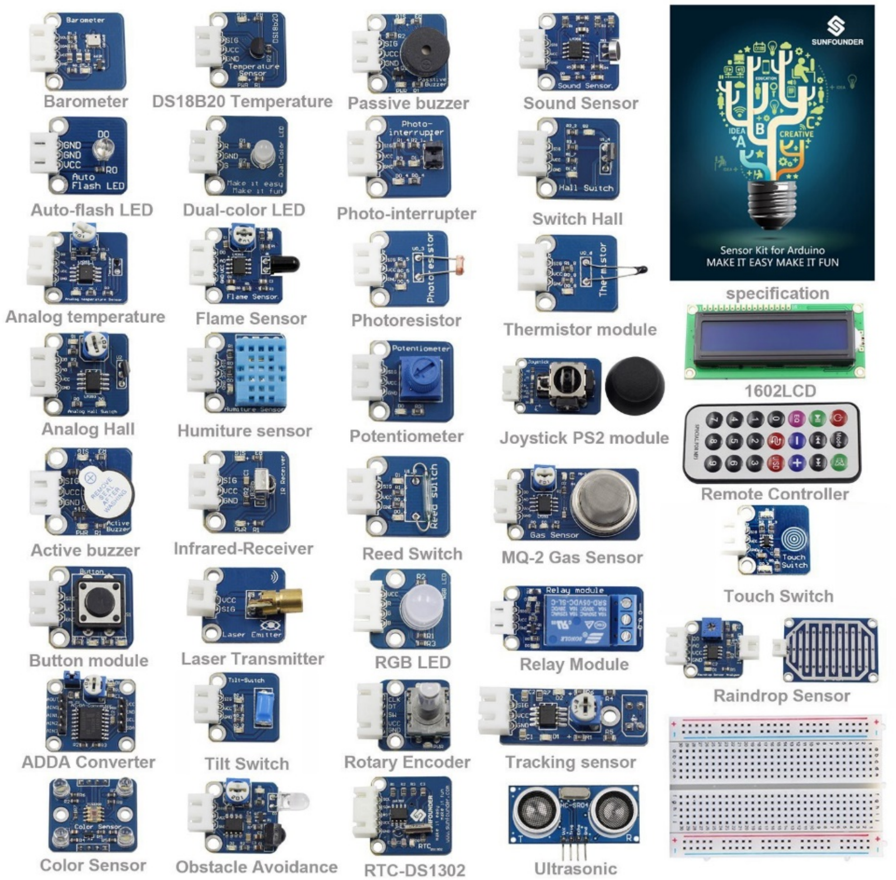


Figure 1: Arduino version of the 37 sensor kit V 2.0 from Sunfounder (Arduino Uno Included)

2.1 Arduino Lessons

The first few Arduino assignments are selected from the openly available standard Arduino examples that are pre-loaded in the IDE software (a free programming interface from Arduino). For most of the assignments the sensor kit is used to supply the hardware for the task; the instructions and code associated with the task is provided online [14]. In addition to the standard Arduino examples the students are asked to modify the code and hardware in some way to show understanding of the technology. The first few weeks of standard

Week	Lab activity	Educational Outcomes
2	Controlling Lights: programming a sequence of LEDs (differentiate with encouraging various levels of difficulty depending on prior experience)	Demonstrate the ability to initialize and declare I/O status, and use it to send signals out to device.
3	Producing Sound: Attach a speaker, program the Arduino to generate a series of pitches. Use smartphone FFT apps to analyze the sound and confirm the frequencies.	Understand the relationship between frequency and audible pitch. Be able to program a modulated signal to produce sounds. Use analysis tools to observe the frequency domain and identify signals.
4	Controlling Motors: Students will learn how to program an Arduino to control various types of motors. They will demonstrate control of multiple types of motors and move the motors through a prescribed pattern of motion.	Understand the properties of DC signals and pulse width modulation for motor control. Understand power requirements and limitations of motor drivers. Understand encoders and incremental vs absolute motion commands. Learn about brushed, brushless and stepper motors. Have the ability to program motorized position control.
5	Optical transmission and Receiving: Coupled with the binary/logic signal topics, students will design and build and optical digital signal transmission and receiving system. Students work in pairs and attempt to encode secret messages and decipher them.	Demonstrate understanding of binary systems by encoding information in a binary string and decoding an unknown message. Show the ability to transfer information wirelessly or through fiber optic cables.
6	LCD Display, ultrasonic distance: Students will build a system to measure distance with an ultrasonic distance sensor and display that distance on an LCD screen.	Have the ability to program an LCD display for numeric information. Have the ability to collect and interpret data from a distance sensor. Have the ability to translate signals from a distance sensor to display data in the correct units.
7	Sensors (temp, humidity, pressure). Student groups will pick a weather related sensors and build a device to collect and display information from the sensor. They will present to the class and explain their method for the sensor they chose.	Students will get exposed directly to one of a set of sensors and then learn about the other sensors from presentations made by their classmates. Through their presentation they will demonstrate that they understand the signals generated by the sensor, how to interpret the signal, how to convert it to the correct units and how to display it in a virtual or physical device.
8	Joystick, Tilt switch, buttons: groups will chose one interface device to work with and demonstrate the ability to collect information generated from manual input and use that information to control a motor, light, optical data transmitter or other device that they have previously learned about.	Students will learn about human-computer interactions with their device and then learn about the other devices from presentations made by their classmates. Through their presentation they will demonstrate that they understand the signals generated by the input device, how to interpret the signal, and how to apply it to control another device.
9	Thermostat: Students will use a temperature sensor fed through an Arduino into LabVIEW and create a method for converting the signal data to temperature in Arduino language and then sending the output to LabVIEW for display, and also sending the raw sensor data into LabVIEW and having LabVIEW perform the calculations before displaying the values.	Demonstrate the ability to code the conversion of a sensor signal to a temperature (in °F, °C, and K) in both in the LabVIEW Language and in the Arduino Language. Use a LabVIEW graphical user interface to display the temperature in multiple units. Use the Arduino as a DAQ device feeding LabVIEW information.
10-12	Rewrite previous Arduino program(s) to run from LabVIEW. Using devices the students already successful demonstrated on the Arduino platform they will now have data sent into LabVIEW and create programs to handle that data (logging, analyzing, and responding) and displaying results.	Demonstrate the ability to use Arduino as a DAQ device for LabVIEW. Demonstrate the ability to construct a graphical user interface in LabVIEW to display results and allow a level of control.
13-15	Design & Build LabVIEW/Arduino system with sensors, signal processing, control, and data log/display	Students show their creativity and design ability by conceiving their own system from the concepts they have learned in the class. They will propose, design, build, test, document, and present their project.

Table 1: Arduino lab activities and educational outcomes for 15 weeks of the Signals and Sensors course

examples and modifications are shown in Table 2 with a connection to the in-class lab activity for that week. For example, the first assignment is to run the Arduino "Blink" program and use the built-in Arduino LED, then students are instructed to connect an actual LED using a breadboard. The final part of the prelab that week is to modify the standard lesson code and hardware to add additional LEDs and have them light up in a particular sequence. The students must then upload a word document in blackboard as a pre-lab report (Table 3) the day before the class meeting. The pre-lab report is not done for every Arduino pre-lab task assigned that week, just for the final modified task that incorporates the pre-lab lessons. An example of a completed pre-lab assignment is shown in Table 4. When students struggled with pre-lab they can get help from other students, email the instructor, visit office hours, or post questions on the course discussion board (which would also help other students who had encountered the same problem). The pre-lab report is graded by the instructor in the hours between the due date and when the class meets, mainly for attempted completion. The pre-lab is meant to be easy points for the students and give them the foundational material they need to accomplish their in-class group assignment.

Week	Arduino Lessons	Pre-lab Task(s) with Modifications	Lab Assignment in Class
2	Arduino Examples Built In: Blink, Button	Use 7 LEDs to make the face of a die and use the "randNumber" command to choose a random number 1-6 and illuminate the correct lights when a button is pressed.	Build LED circuits controlled by potentiometers, photo-resistors, and pulse width modulation
3	Serial Communication [15]	Build a program to perform a calculation of your choice using inputs and the serial monitor display	"Tone Melody" and "Pitch Tone Follower" Built-in Examples, compose melody of your choice, add volume knob.
4	Sensor Kit Lesson 24: Rainbow LED	Pulse width modulation (PWM) with RGB LED, make at least 5 different colors using AnalogWrite. Have a sequential transition through those colors. Program a random color generator.	Use PWM to program and operate servo and DC motors, add a potentiometer to the servo for the "Sweep" example.
5	Sensor Kit Lesson 25: IR Receiver	Assemble the IR Receiver. Read out the hexadecimal codes sent by your IR remote. Make a table mapping each button to its hex code. (can also ask to decode any other IR device students have)	Build a system with the IR remote, IR receiver, servo motor, DC motor, RGB LED and buzzer/speaker. Assign buttons on the remote to trigger different actions with each device.

Table 2: A sample of weekly Arduino lessons and the additional items assigned. Week 1 has no pre-lab.

2.2 LabVIEW Lessons

The LabVIEW portion of the class runs in parallel with the Arduino portion. In designing the course there was much consideration given to using a LabVIEW textbook vs online video based tutorials available through National Instruments or an instructor designed tutorial. Many of the texts were highly priced, and those that were lower cost had older copyrights and hence older content which might not align with the current version of LabVIEW. For no cost, the students could watch instructional videos on the NI website presented by experts in

Pre-Lab Report Item	Purpose for Item
Name	Identify Student
Assignment Name	Identify Assignment
Description of project (in your own words):	Check that the student understands the concept of the assignment, what they are trying to accomplish
Description of code (in your own words or pseudocode)	Check that the student understands the basic segments and code needed to solve the problem
Insert code here (picture or text)	Be able to see the modifications student made to the base code from the assignment, also it serves as a record for students to refer back to when needing to reuse code, also a plagiarism deterrent
Insert image of Arduino and breadboard here	Proof of physical competition of the circuit, also a plagiarism deterrent since identical pictures can stand out when reviewing submissions.
Description of problems encountered and solutions you found to them (must have at least 1, if no errors were made describe a potential source of error)	This is valuable feedback to the instructor about what was most difficult about the project and how the student overcame it. When discussing the assignment in class, this a good item to review to see where most students struggled.
Approximate time to complete this assignment (minus breaks and distractions)	This is also valuable feedback to the instructor to gauge how quickly a student is grasping the concepts and accomplishing tasks. High performers and the students in need of assistance can easily be identified.

Table 3: Arduino Pre-Lab Report

LabVIEW [16]. Due to time constraints the instructor did not create their own content and decided to leverage the NI tutorials which had both video and illustrated written guides to learn various functions and techniques in LabVIEW to create Virtual Instruments (VIs). A preselected set of videos that were easy to follow, most relevant to the week's projects, and appropriate for novices were chosen to be watched outside of class and replicated by the student before coming to class. Due to the challenges of learning LabVIEW, no additional modification of the examples were required, it was sufficient for the students to be able to replicate the example from the video. The sequence of pre-labs and class projects relating to LabVIEW for the first few weeks of the semester can be seen in Table 5 and an example of a pre-lab assignment is shown in Table 6.

2.3 Combining Arduino and LabView

After students have a firm grasp of using LabVIEW and working with the Arduino Hardware system, they can be taught about the LINX add-in for LabVIEW that allows Arduino to be a data acquisition (DAQ) device feeding information to LabVIEW and communicating back to external components. The LINX example VIs require that a firmware be flashed onto the Arduino in order to properly communicate. The process of flashing the software has an easy-to-follow tutorial to set up the Arduino for LINX or return it to the original IDE format. LINX has a built-in set of examples which mimic the examples that the Arduno IDE provides but more limited in scope. Basic programs like "Blink" can be set up with the LabVIEW Front Panel interface for control with a real LED connected to the Arduino as the indicator of the

Pre-Lab Report Item	Submitted Item
Name	Robert Morris
Assignment Name	Light Sequencing
Description of project (in your own words):	A random number generator is used to "roll a dice" and get a number from 1-6.
Description of code (in your own words or pseudocode)	The number rolled by the random number generator is displayed by LED lights (set up in a way that mimics the number you would see on the face of the six sided dice). By pressing the button switch, you start the random number generator again. The lights will blink out for half of a second before coming back on in the configuration that matches the newly generated number.

Insert code here	int roll;
	int buttonState = 0;
	const int buttonPin = 7;
	const int ledCenter = 0;
	const int ledBR = 1;
	const int ledBL = 2;
	const int ledEnds = 3;
	
	void setup()
	{
	// to run once:
	pinMode(0, OUTPUT);
	pinMode(1, OUTPUT);
	pinMode(2, OUTPUT);
	pinMode(3, OUTPUT);
	
	pinMode(7, INPUT);
	}
	
	void loop()
	{
	// to run repeatedly:
	// checks for button press
	buttonState = digitalRead(7);
	
	if (buttonState == LOW)
	{
	//clearing display
	digitalWrite(0,LOW);
	digitalWrite(1,LOW);
	digitalWrite(2,LOW);
	digitalWrite(3,LOW);
	//number generation
	roll = random(1, 7);
	delay(400);
	if (roll == 1)
	{
	// turn LED on:
	digitalWrite(0, HIGH);
	}
	else if (roll == 2)
	{
	digitalWrite(2, HIGH);
	}
	else if (roll == 3)
	{
	digitalWrite(2, HIGH);
	digitalWrite(0, HIGH);
	}
	else if (roll == 4)
	{
	digitalWrite(2, HIGH);
	digitalWrite(1, HIGH);
	}
	else if (roll == 5)
	{
	digitalWrite(2, HIGH);
	digitalWrite(1, HIGH);
	digitalWrite(0, HIGH);
	}
	else if (roll == 6)
	{
	digitalWrite(2, HIGH);
	digitalWrite(1, HIGH);
	digitalWrite(3, HIGH);
	}

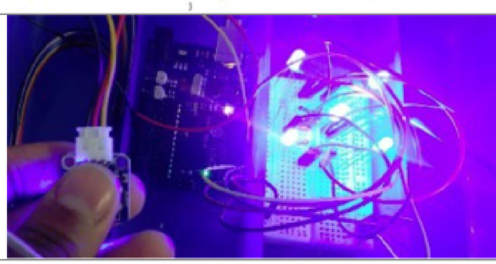
Insider image of Arduino and breadboard here	
Description of problems encountered and solutions you found to them (must have at least 1, if no errors were made describe a potential source of error)	I needed to reacquaint myself with the commands and programming format, so that took a little time to get back into the swing of it. I also encountered problems because I couldn't remember how to set the port in order to upload my codes to the Arduino.
Approximate time to complete this assignment (minus breaks and distractions)	1 hour

Table 4: Arduino Pre-Lab Report Example

state of the LabVIEW virtual button or switch. Other example programs allow inputs from photoresistors/ potentiometers, control of RGB LEDs, drive motors, and other basic device functions. As with all LabVIEW code, these VIs can be adapted, edited and altered to provide other functionality for customized projects and applications.

The final project requires a combination of LabVIEW and hardware to produce a system of their own design. They decide in class what hardware their group will use with LabVIEW, and create a proposal on the system which is presented to the class. Their peers ask questions and make suggestions. The instructor will increase or decrease the complexity of the project to ensure it

Week	LabVIEW Pre-Lab Lesson	Lab Assignment in Class
2	Loops and Execution Structures, videos 1-4	Execution Structure video 5 will be done is class due to difficulty
3	Data Types and Structures, videos 1-4	Create a program to replicate your Arduino Pre-lab in LabVIEW. Have a button that triggers the random number generator to select a number from 1-6. Use Case structures to have a display of your output as a set of boolean variables on the front panel in the shape of a die.
4	Signal Processing,	Manipulate VI generated sin and cos signals and review the differences between charts and graphs in LabVIEW, display using both types.
5	MathScript and Text-based programming	Create a GPA calculator program for a single semester of grades with an attractive and functional front panel. Inputs: Course ID, # credits, grade.

Table 5: LabVIEW Flipped Classroom Lessons Sequence for the first few weeks of class

LabVIEW Activity 2.1	
Watch the videos for Lesson 2, Loops and Execution Structures. (http://www.ni.com/academic/students/learn-labview/execution-structures/) Requirements: Create a random number generator with a FOR loop. Set a 1 second delay and create a constant to set the number of loops. The indicator should change every second. Capture the data in an auto index array. Make a second FOR loop that shows each value in the array after the array is populated. Submit the following:	
Pre-Lab Report Item	**Purpose for Item**
Description of VI (in your own words):	Can the student verbalize what they are trying to accomplish with this assignment?
Image of Front Panel:	Do they have the correct controls and indicators with a logical and aesthetic layout?
Image of Block Diagram:	Is their graphical programming logical, well organized and correct?
Errors encountered and solutions found (at least 1):	With what aspects of the pre-lab did students struggle? Does the assignment need changed or would better instructions prevent other students from making similar mistakes?
Approximate time to complete:	To assess which students may need addition assistance if struggling to complete the pre-lab

Table 6: Example of Pre-Lab assignment using LabVIEW

is achievable within the scope of the class and of sufficient difficulty for the students. The project has three main requirements:

- The project must involve using LabVIEW to process, store, and display data.
- It must take in data from a sensor and produce some type of response external to LabVIEW based on a set of programmed conditions (two or more of: lights, sound, motor movement)
- The project needs to integrate with external hardware (Arduino, Video game controller, Xbox Kinect, ELVIS board, Mindflex headset, etc.)

The project proposal presentation is vital to control the scope of the projects and set up timelines for the projects. Weekly updates are required along with demonstrations of intermediate results which serve as proof-of-concept milestones. In final exam week, each group gives a presentation discussing the design and construction of their system along with a demonstration of functionality. A list of past final projects can be seen in Table 7.

Name	Hardware Used	Inputs	Outputs	Function
Alarm Clock	Arduino	Brightness	Light, sound motor control	Alarm clock that senses sunrise and creates audio/visual alarm with moving parts
Muscle controlled car	Arduino, RC car.	Sensing Muscle flexing	Steering, throttle	Use muscle sensors to drive a remote controlled car by flexing arms
Xbox controlled car	Arduino, Xbox Controller	Controller buttons	Steering, throttle, sound and lights	Map Xbox joystick and buttons to drive a remote controlled vehicle and activate lights and sound on it.
Automatic pet feeder	Arduino	Keypad, pushbutton	LCD screen, motor	Pet can trigger feeder with the button, but not to exceed user chosen amount per time period.
Connect Four	Arduino	Joysticks	Lights and sounds	Construct a Connect 4 game in LabVIEW that is joystick controlled with external signals for player notifications
Escape Room Trivia Game	Arduino	Keypad	Lights, sounds, LCD display	LCD displays trivia questions, keypad used to enter answers. Need several correct answers in 30 seconds to pass.
Escape Room Musical Challenge	Arduino	Keypad	Lights, sounds, LCD display	Generates random 6 tone pattern, user must press numbers on keypad that produces same tone pattern. New pattern every user.

Table 7: Examples of student designed final LabVIEW based final projects

3 Results

The course has now been taught three times. There have been some changes related to keeping the technology current. The course was designed using the LIFA add-in for LabVIEW to communicate with Arduino. Between the design of the course and the launch in SP17 (Spring 2017), LIFA was retired and replaced with LINX which caused a reworking of several assignments and verification of functionality with the new software. Also, after the first offering of the course the Sunfounder Sensor Kit was retired and replaced with Sensor Kit V2.0 which had some significant differences. All of the sensor interfaces were changed from having hard pins for connection in a breadboard to now having a female socket with anti-reverse cabling. Also some of the libraries used for the devices seem to have been changed or updated; students discovered problems with the original libraries not being compatible with V2.0 devices.

The new sensor kit also lacked the 4x4 keypad which was popular with the first group of students, so this item was purchased separately by the instructor and integrated into the course when using V2.0 kits. Yearly changes to LabVIEW and the upcoming transition to LabVIEW NXG also require an annual update of course content. The Arduino IDE also needs updated annually, although the newly launched Arduino Create browser based IDE and cloud system have made it easier to run Arduino on any computer [17].

The students complete detailed online course feedback surveys at the end of each semester. A total of 32 students have taken the course over 3 semester (12 students in SP17, 13 in SP18, and 7 in SP19) and 23 responded to the survey (9 students in SP17, 7 in SP18, and 7 in SP19) for a 72% response rate. The results in Table 7 show there is clearly an upward trajectory in the scores over the three semesters this course was offered which can be attributed to several factors. First, the instructor grew in confidence and experience using LabVIEW and Arduino. Second, The feedback from classes was heeded and modifications were made accordingly, removing or retooling assignments that were overly challenging (The LabVIEW GPA calculation went from calculating transcript GPA to a simpler semester GPA in SP19) or correcting for technology issues (removing certain sensors that did not function well like distance and humiture sensors and various Arduino shields). Third, more freedom was given for students to design their own projects and teamwork was encouraged more in class. Often students would help solve problems for each other if the instructor was occupied helping another student. This also built a sense of teamwork and group pride in the projects. The comments provided with the end of year survey seen in Table 8 confirm that this flipped classroom approach is viable and successful, as evidenced by the overwhelmingly positive comments about the course being both fun and educational.

From Strongly Agree = 5 to Strongly Disagree = 1	**SP17 (n=9)**	**SP18 (n=7)**	**SP19 (n=7)**
The subject matter in this course was presented in a well-organized manner	3.56	4.57	4.86
In general, the class time was used constructively	3.67	4.57	5.00
The instructor encouraged me to think more about the subject	4.56	4.57	5.00
The instructor created an atmosphere that made learning easier	4.00	4.71	5.00
Exams, assignments, or projects accurately reflected the course content	4.22	4.71	5.00
Assignments or projects helped me learn the material	4.56	4.71	5.00

Table 8: A selection of statements asked to students in the end of semester course evaluation.

One of the most popular projects was the Escape Room Challenge, which required students to design, build and demonstrate a puzzle that could be found in an escape room. One group built a system to encode pitches to the

Prompt: Assignments or projects helped me learn the material
The pre-labs was (sic) extremely helpful because it was an excellent introduction to the week's material.
Prompt: How could this course be improved to make the instruction more effective?
Don't make students struggle with the distance sensors, they just don't work right. (Instructor note: This lab will be modified)
Have the Escape Room project be the final project (Instructor note: this change will be made for the next course offering)
Some of the LabVIEW videos in the beginning of the course were difficult to follow along. Maybe do those assignments in class instead of for homework (Instructor note: these were identified, moved from pre-lab to lab for next year)
Prompt: What do you believe this instructor has done especially well in the teaching of this course?
The Arduino projects were helpful in building up my coding experience and the low-stress environment made learning a lot easier.
He assigns super interesting projects and they make you excited to do them, they don't feel like a chore.
It allows students to be creative and use real-world engineering practices to build and develop different mechatronic systems.
The class was well taught and (the instructor) did an excellent job tying concepts from projects together… The class was very fun and I learned a ton along the way.
Developing student's knowledge in multiple different forms of programming, both graphical and text based. I learned a lot!
Literally everything, wonderful course. Felt like I learned a lot of material and had tons of fun.

Table 9: Students' comment responses on the course evaluation

10 digits on a key pad and then play a random sequence of the pitches (Figure 2). The participants would then need to reproduce the sequence using the keypad. Every time the system was initialized, a new random sequence would be generated and held in memory for the participants to try to replicate. The pound "#" button would enter the last 6 digits pressed to be checked against the random sequence, and the star "*" button would replay the random sequence that was held in memory. There was also a reset button that would generate a new sequence and reset the system. When the 6 digit sequence of tones was successfully entered a congratulatory message would appear on the LCD screen and a triumphant sound sequence would play. If the entry was incorrect, a negative message would display and a dejected sequence of notes would indicate another wrong code has been entered. The system was very popular with the students in the class, and proved to be repeatedly challenging with each new note sequence generated.

4 Conclusions

A flipped classroom was successfully designed which allowed students to do out of class pre-laboratory work using an Arduino, Sensor Kit, and LabVIEW. The home lessons provided ample guidance (instructor and peer support were always a recourse for those that struggled) so that students came to class prepared with the basic skills needed to combine concepts in a more challenging project. Making the in-class projects structured as group assignments allowed students to help each other and solve many problems without instructor intervention. Groups would also voluntarily help other groups and share successful strategies for solutions to ensure everyone was able to get a fully functional (or nearly functional) implementation of the project. Students were very open

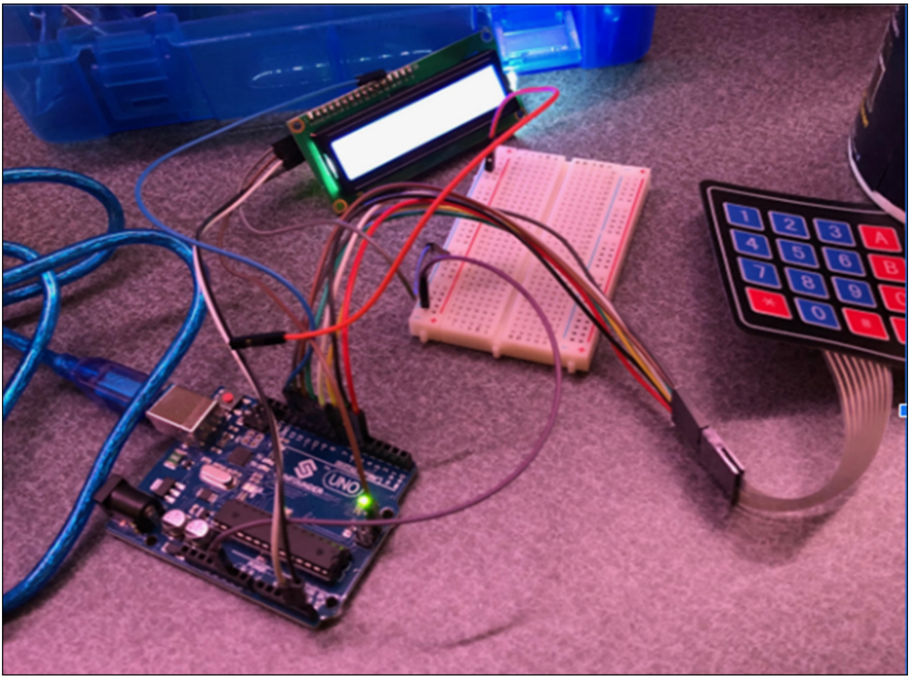


Figure 2: Escape room pitch sequence challenge (prior to speaker being added)

about which assignments were easy and manageable and which ones were nearly impossible to complete (in which case they were re-designed for the following course offering). Course survey data shows that learning objectives were met, and the students had a high degree of satisfaction with the course, finding it both enjoyable and educational. The students gained practical experience with system design and troubleshooting using hardware and software, which is a valuable lesson for anyone working in engineering or with computer based systems.

Based on the success and popularity of the escape room project, student feedback indicated that they would rather expand upon this project rather than do two smaller final projects. The expansion of the escape room project would entail building an enclosure for the system and making it completely self-contained in a modular format that could be integrated into a larger escape room sequence. For this to be a self-contained project, it may be shifted to Arduino only, rather than a LabVIEW/Arduino hybrid project. The students would then host a public escape room activity at the end of their semester to test their system and expose it to real-world use conditions which may inspire design change recommendations for the final report. By making the system modular and providing well-planned design criteria, the challenges created by each group each semester could be saved and combined in different variations for a portable escape room challenge for university events or outreach activities.

Recently, the electronics lab associated with this course was upgraded to have NI ELVIS III. The Signals and Sensors course is being modified to move away from the LINX add-in as an integration platform for the Arduino and LabVIEW. LINX will still be demonstrated and used for some activities, but using the ELVIS III as the DAQ system for in-class projects should reduce hardware and software complications since LabVIEW and ELVIS are both National Instruments products, whereas LINX as a bridge between the Arduino and LabVIEW can be buggy. In the redesigned version of the course, Arduino pre-labs will still be completed at home using the sensor kit, but in class, those sensors will be used with ELVIS and students will have to learn how to interpret the data from the sensor and calibrate it for ELVIS, instead of using an Arduino Library for configuration information. This will provide a more real world situation with problem solving related to integration and a deeper understanding of the signals created by transducers while teaching a framework for interpreting that information.

References

[1] J. L. Bishop and M. A. Verleger. The flipped classroom: A survey of the research. In *ASEE national conference proceedings 30(9)*, pages 1–18, 2013.

[2] S. J. DeLozier and M. G. Rhodes. Flipped classrooms: a review of key ideas and recommendations for practice. *Educational Psychology Review*, 29(1):141–151, 2017.

[3] C. F. Herreid and N. A. Schiller. Case studies and the flipped classroom. *Journal of College Science Teaching*, 42(5):62–66, 2013.

[4] M. Joordens and B. Horan. Making student experts within a flipped mechatronics class. In *25th Annual Conference of the Australasian Association for Engineering Education: Engineering the Knowledge Economy: Collaboration, Engagement & Employability*, page 615, 2014.

[5] A. Karabulut-Ilgu, N. Jaramillo Cherrez, and C. T. Jahren. A systematic review of research on the flipped learning method in engineering education. *British Journal of Educational Technology*, 49(3):398–411, 2018.

[6] B. Kerr. The flipped classroom in engineering education: A survey of the research. In *2015 International Conference on Interactive Collaborative Learning (ICL)*, pages 815–818. IEEE, 2015.

[7] G. S. Mason, T. R. Shuman, and K. E. Cook. Comparing the effectiveness of an inverted classroom to a traditional classroom in an upper-division engineering course. In *IEEE Transactions on Education, 56(4)*, pages 430–435, 2013.

[8] J. R. Q. Peña, J. Oliveira, M. Neto, and L. H. N. Rodrigues. Active methodologies in education of electronic instrumentation using virtual instrumentation platform based on Labview and Elvis II. In *Artigo–2018 IEEE Global Engineering Education Conference (EDUCON)*, pages 1696–1705, 2018.

[9] L. Svensson and T. Adawi. Designing and evaluating a flipped signals and systems course. In *European Conference on e-Learning*, page 584. Academic Conferences International Limited, July 2015.

[10] N. L. Toner and G. B. King. Restructuring an undergraduate mechatronic systems curriculum around the flipped classroom, projects, LabVIEW, and the myRIO. In *2016 American Control Conference (ACC)*, pages 7308–7314. IEEE, July 2016.

[11] Z. Zainuddin and S. H. Halili. Flipped classroom research and trends from different fields of study. *The international review of research in open and distributed learning*, 17(3), 2016.

[12] Ü. Çakiroglu and M. Öztürk. Flipped classroom with problem based activities: Exploring self-regulated learning in a programming language course. *Journal of Educational Technology & Society*, 20(1):337, 2017.

[13] `https://www.sunfounder.com/37-modules-sensor-kit-v2-0-for-arduino-1668.html`.

[14] `https://www.sunfounder.com/learn/category/Sensor-Kit-v2-0-for-Arduino.html`.

[15] `http://www.ladyada.net/learn/arduino/lesson4.html`.

[16] `http://www.ni.com/academic/students/learn-labview/`.

[17] `https://www.arduino.cc/en/Main/Create`.

A Case Study in Software Testing: Verification of a Face Identification Algorithm for Planar Graphs*

Drew Whitmire[1] *and Chris Alvin*[2]
[1]*Department of Computer Science and Information Systems*
Bradley University, Peoria, IL 61625
`drewwhitmire@gmail.com`
[2]*Department of Computer Science*
Furman University, Greenville, SC 29613
`chris.alvin@furman.edu`

Abstract

We describe a case study in software testing of legacy source code as part of a year-long, undergraduate capstone project. The solution for face identification of planar graphs is well-known. However, in practice, thoroughly testing an implementation of such an algorithm can be a challenge. We describe an algebraic method to facilitate automated testing of an implementation of a geometric face identification algorithm with the goal of exposing the underlying mathematical limitations of the implementation.

1 Organization

This paper describes a case study in software testing; it is thus important to bring the reader along on our journey of decision making, processes, and implemented procedures. Our overall goal was to integrate a legacy source code implementation of a solution to the face identification problem into a larger system. In §2 we begin by describing the face identification problem for planar graphs and its solution. In the next section (§3), we describe background discussion and rationale for limitation-focused tests of the legacy code. The last section (§4) introduces our method for limitation testing with planar graphs, formalizes geometric notions and our limiting parameters in §4.1, and describes an algorithm for constructing planar graphs with known facets in §4.2. Last, in §5 we empirically evaluate the limits of the legacy source code implementation.

2 The Face Identification Problem for Planar Graphs

A component of our software system involved implementations of several geometry processing algorithms. In this section we describe one such problem and its solution: face identification [3] of a planar graph [1]. Mathematically, the notion of a face is well known. We will equate the notion of a face with the graph theoretic concept of an elementary (or simple) cycle in an undirected planar graph. For example, the planar graph in Figure 1 has twelve faces consisting of six triangles ($f_1, f_2, f_3, f_4, f_8, f_{12}$) and six quadrilaterals ($f_5, f_6, f_7, f_9, f_{10}, f_{11}$).

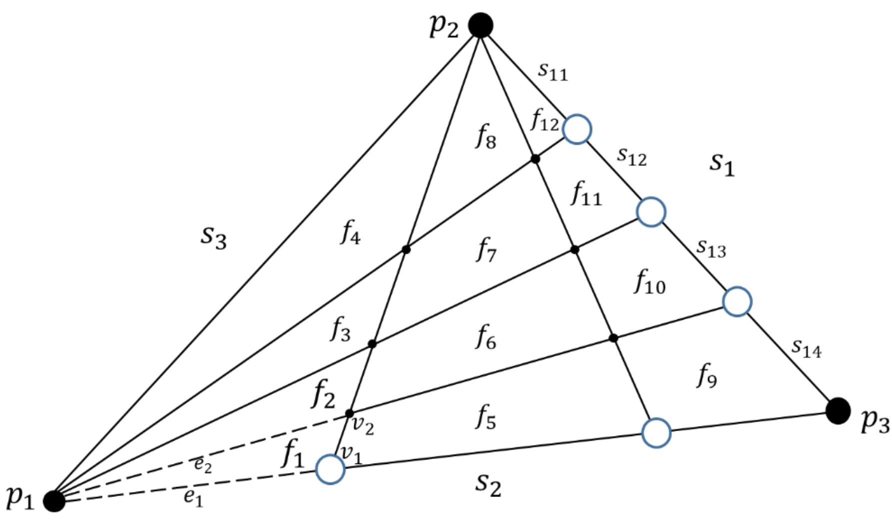


Figure 1: An Annotated, Triangular Planar Graph

An implementation of the geometric face identification algorithm we are attempting to verify is based on the algorithm mentioned in *Computational Geometry: Algorithms and Applications* [1] and described in detail in *The Minimal Cycle Basis for a Planar Graph* [2]. Given a planar graph represented as a set of coordinate pairs (embedded in the plane), the algorithm analyzes points ordered lexicographically from least to greatest. Based on the current least point, cycle identification greedily chooses an edge in the smallest counterclockwise order until the starting point is reached: thus, a simple cycle is achieved and a face is found. In Figure 1, this pass of the algorithm begins with point p_1 and continues counterclockwise to v_1 and then to v_2 until face f_1 is identified. The cycle identification procedure continues, but to avoid identifying the same cycle, we remove the first edge taken (e_1 in Figure 1). Hence, the next greedy, counter-clockwise cycle we would identify f_2 in Figure 1. This process continues until all simple cycles (faces) are identified.

3 Case Study Origin

We present a case study in software testing that arose from a software system refactoring project involving five senior undergraduate students in a year-long capstone course. While the overall project is not germane to this discussion, one sub-goal of the project was to leverage several thousand lines of Java source code modules implementing geometry processing algorithms including planar graph face identification. The main problem with the implementations was that it was not implemented using arbitrary-precision mathematics. Hence, a decision had to be made about how to integrate these implementations (including face identification) into the project. We summarize possible options with three questions and discuss each question in turn.

1. Should we start from scratch and re-implement the algorithms for our system?
2. Should we test the limits of the legacy implementations to ensure that they will work in the context of our overall project?
3. Should we leverage the algorithm implementations, but refactor to include arbitrary-precision mathematics as suggested by [2]?

Re-implementation. This option presents a path of least resistance for the students. Students are often eager to implement their own code thus maintaining complete ownership over the entire project from software architecture to detailed implementation. Implementing software from scratch has its own issues for even the best undergraduate students. However, it provides an authentic experience designing and building a component of a large software system from scratch: from requirements to implementation to testing.

To the advisor, a re-implementation avoids an important educational opportunity: working with legacy source code. Undergraduate programs often lack this type of authentic software engineering experience all too common in industry-based software development. As advisor of this project I was very much against a re-implementation simply because students would be avoiding a set of learning opportunities that would benefit them within months when they moved on to industry jobs.

Testing the limitations. There was an underlying question about the source code implementation of the geometry processing algorithms: are they robust enough as a component of our system on their own? This question was an interesting source of discussion for all parties involved because it exposes the reality of software engineering: under project constraints, does the software fit our needs even though it is not perfect? While an arbitrary-precision mathematics solution might be optimal, the fact is, our system may work sufficiently without the underlying arbitrary implementation in this module.

Refactoring the legacy code. This option seemed like a hybrid solution: leveraging as much legacy implementation as possible while increasing code robustness and reliability in the face of complex geometric input. From an educational perspective this seemed like a great solution in which students would be actively engaging with legacy code as well as working with Java classes such as BigDecimal [5] or other arbitrary-precision mathematics solutions.

The Decision. Exploring these questions was one of the most authentic activities with which the students engaged in this project. There was a gravity to their decision since it would guide their work for, possibly, months. We discussed the positive and negative aspects of each scenario and made an initial decision to test the limitations of the legacy code implementation.

In many ways, the entire group consistently came back to one simple question: will the code work as is? In order to answer that questions, students would have to write a quality set of unit tests as well as tests that would expose the mathematical limitations of the implementation. If those exposure tests indicate that the source code would work satisfactorily within the larger system, then we would be satisfied with the legacy code without intervention. However, if we were unsatisfied with the granularity offered by the legacy code, refactoring to include arbitrary-precision mathematics would still have to be verified with the same set of unit tests and exposure tests. Thus, as a backup we decided to use the hybrid approach knowing our work developing a test suite would not be in vain.

4 Parametric Limitation Testing

While [2] describes the face identification of planar graphs problem generally, our system requirements will not have to contend with all types of planar graphs. For example, while the legacy code implementation handles identifying filaments in planar graphs, our system will never encounter such cases. Therefore, our planar graph construction algorithms need not produce an overly complex structure. What is required is a means of increasing the number of faces in a planar graph to expose the limit of legacy code's lack of arbitrary-precision mathematics.

We begin our ‘algebraic’ approach for constructing a planar graph using a triangle T. We identify two sides of T and generate points on those two sides (open circles in Figure 1 on sides s_1 and s_2). We observe that adding n unique points between the endpoints of a line segment result in $n+1$ elementary segments; e.g., the three points along s_1 result in four sub-segments (s_{11} through s_{14}).

Our next step exhaustively connects each generated point to the vertex of the opposing side of T noting that each segment passes through the interior of T. In Figure 1, we create three such segments from points on s_1 to p_1 and two segments from points on s_2 to p_2. Along with the enclosure provided by the sides of the triangle, the result is a clearly identified set of 12 faces in Figure 1. These faces are a result of the number of constructed points we have added along each side: $(2+1)(3+1) = 12$.

Our planar graph construction procedure with n unique points along one side and m unique points along another result in $(n+1)(m+1)$ faces in the planar graph representation. Verification is thus achieved quantitatively by comparing the number of faces computed by the geometric algorithm compared to the expected number of faces constructed via our algebraic technique. Last, we evaluate our technique using increasingly complex tests by increasing the values of independent variables n and m.

4.1 Theoretical Foundations

In this paper, we assume all geometric elements are embedded in the Euclidean Plane (e.g., coordinates of all points are known). We leave ***point*** as an undefined intuitive concept and represent points using the standard (x,y) pairing. Two points define a ***line*** and a ***line segment*** is a finite length of a line defined between two endpoints. From there, we refer to the standard notion of a ***triangle*** as three non-collinear points (called vertices) connected by three line segments (called sides). We are interested in the smallest closed geometric object in a figure called *faces*. We define a ***face*** to be a Jordan curve [4]. We then use the standard definition of a ***planar graph*** as a graph embedded in the Euclidean plane without crossing edges.

Our goal is an algebraic means of verifying the implementation of a geometric face identification algorithm [2]. The basis of this verification should (1) be a graph in which we know exactly how many faces are contained within it, (2) be efficiently constructed, and (3) have the ability to push the limits of the geometric face identification algorithm. We base our planar graph construction algorithm on the idea that the number of unique points added to two sides of a triangle result in an enumerable set of faces. We propose and prove the corresponding lemma.

Lemma 1. Let T be a triangle with sides s_1, s_2, and s_3 with respective opposing vertices p_1, p_2, and p_3. Suppose m distinct points lie on s_1, n distinct points lie on s_2, and segments are drawn from p_1 to the n points on s_1 and segments are drawn from p_2 to the m points on s_2. Then, T consists of $(n+1)(m+1)$ faces.

Proof. As a base case, we consider the case with no constructed points. Let n=0 and m=0. It is clear that with no segments constructed, T, itself, is the single face: $(0+1)(0+1) = 1$.

Now assume for $m \neq 0$ and $n \neq 0$ unique points, T has $(n+1)(m+1)$ faces. Consider m points on s_2 (e.g. $m = 2$ on s_2 in Figure 1); a similar argument would follow by considering n points on s_1. Adding another distinct point $p_{(m+1)}$ to s_2 results in a corresponding segment g being constructed from $s_{(m+1)}$ from p_2. Segment g cuts through each of the n segments connecting p_1 and points on s_1. Thus, adding $s_{(m+1)}$ and g results in $n+1$ new faces and thus $(n+1)(m+1)+n+1$ total faces. Algebraically, we see

$$(n+1)(m+1)+n+1 = nm+2n+m+2 = (n+1)(m+2).\blacksquare$$

We appreciate the fact that the number of faces of a planar graph can be found using Euler's Theorem for connected planar graphs: for a connected planar graph v+$f = 2$+e where v is the number of vertices, f is the number of faces, and e is the number of edges. However, the independent variables in our test refer to the number of points along two sides of a triangle. This configuration constructs planar graphs with discrete qualities facilitating analysis of any limitations in the legacy source code implementation.

4.2 Algebraic Planar Graph Construction

Our approach for constructing the algebraic planar graph is accomplished in two steps. We first generate a triangle along with all of its desired constituent interior segments called a *Segmented Triangle.* From the segmented triangle, we construct the planar graph.

Step 1: Construct the Segmented Triangle

Our goal in this step is to construct a segmented triangle as shown in Figure 3. This requires we represent our segmented triangle as an object containing a set of points and segments. As a basis for the segment triangle ST, we construct an initial triangle. The resulting triangle must minimally have three distinct, non-collinear vertices, but in practice and without loss of generality, we found non-obtuse triangles (e.g. isosceles or equilateral) work well. We then generate two sets of interior segments to ST from a randomly generated point and the opposing vertex as described in Algorithm 1.

```
1.  Procedure genSegments(s:Segment, p:Point, n ∈ ℤ+)
2.      S ← ∅ : Set < Segment >                // the set of generated segments
3.      for i ← 1 to n do
4.          t ← new Point(randomPointOn(s,M)) // generate unique point on s
5.          g ← new Segment(p,t)              // segment from p to t
6.          S ← S ∪ {g}
7.      return S
8.   Procedure BuildSegments(ST : SegmentedTriangle)
9.      // generate segments from a side to the opposing vertex (twice)
10.     ST.constructed_1 ← genSegment(ST.side_1,ST.vertex_1)
11.     ST.constructed_2 ← genSegment(ST.side_2,ST.vertex_2)
```

Algorithm 1: Construction of n Segments from Point p to Points on Segment s

As input, procedure `genSegments` in Algorithm 1 takes a segment s representing a side of a segment triangle, a point p opposite of s, and the desired number n of segments to generate. For each of the desired n points (Line 3 through Line 8), we perform a sequence of steps. On Line 4, we randomly generate a unique point t along s and construct the interior segment g on Line 6 and adding it to the accumulating set S (Line 7). We store both sets of constructed segments, S_1 and S_2, to ST (Line 6 and Line 7).

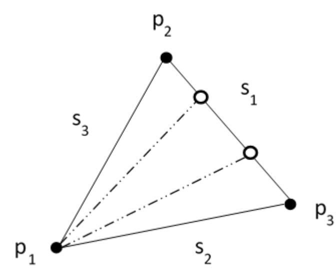


Figure 2: Result of `genSegments`$(s_1, p_1, 2)$ from Algorithm 1.

We invoke `BuildSegments` in Algorithm 1 with a basic, segmented triangle. Using vertex p_1, opposing side s_1, and $n = 2$ as input to `genSegments` would implicitly result in the segmented triangle with two interior segments as shown in Figure 2. Invoking `genSegments` again with vertex p_2, opposing side s_2, and $n = 2$ would then result in the segmented triangle shown in Figure 3. We observe a planar graph representation of the segmented triangle in Figure 3 would have $(2 + 1)(2 + 1) = 9$ faces.

Given a segmented triangle ST object with two sets of constructed seg-

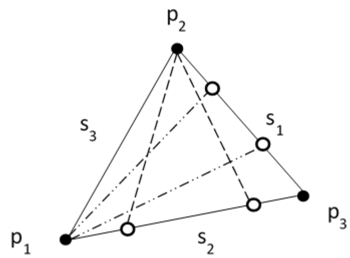


Figure 3: Result of `genSegments`(s_2, p_2,2) from Algorithm 1.

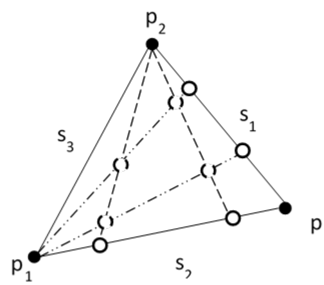


Figure 4: Result of Algorithm 2: Interior Intersection Points

```
Procedure intersections(S1 : Set&lt;Segment&gt;, S2 : Set&lt;Segment&gt;)
    M ← ∅ : Multimap<Object, Object>
    for each s1 in S1
        for each s2 in S2
            o ← new Point(intersection(s1, s2))
            M ← M ∪ {<o, s1>, <o, s2>}        // Add <point, segment> pairs
            M ← M ∪ {<s1, o>, <s2, o>}        // Add &lt;segment, point&gt; pairs
    return M
```

Algorithm 2: Exhaustive Computation of 'Interior' Segmented Triangle Points

ments, S_1 and S_2, exhaustively construct and add all points of intersection to ST:

$ST.points \leftarrow ST.points \cup \{intersection(s_1, s_2) | \forall s_1 \in S_1 \wedge \forall s_2 \in S_2\}$ as computed in Algorithm 2 and shown as four dashed points in Figure 4. For quick reference we use a *bi-associative multimap* which (1) maps a point in a segmented triangle ST to each segment it lies on in ST and (2) maps each segment in ST to each point in ST that lies on the segment. As an implementation note, this data structure is a wrapper class around two distinct multimaps: one maps points to segments and the other maps segments to points. We see the multimap data structure populated on Lines 6 and 7 in Algorithm 2.

Step 2: Constructing the Planar Graph Altogether, a segmented triangle (ST) now contains all constituent points and segments required for a corresponding planar graph (g). Planar graph construction first considers nodes then the edges.

Nodes. The nodes of the planar graph g consists of all points in the segmented triangle ST; there are eleven such points in Figure 4.

Edges. In order to add edges to the planar graph, we first define adjacency with respect to points on a segment. We define direct adjacent points to be the set of all smallest distance collinear points from a point p. If p is an endpoint of a segment s, s may contribute only a single point as being a directly adjacent point; e.g., vertex p_1 in Figure 4. If p is not an endpoint of a segment s, s will contribute two points as being directly adjacent since p is the middle point of three collinear points; e.g., any of the interior points shown in Figure 4.

Edges in the planar graph are then computed as follows. For each point p in the segmented triangle, we compute the set of direct adjacent points A_p. We then add an edge from p to all elements in A_p to g. This algorithm may add redundant edges to g, therefore, duplicate edges are ignored (our planar graph has no parallel edges). The resulting planar graph corresponds to Figure 4 and can be passed to the geometric face identification algorithm for verification.

5 Empirical Analysis

We first consider execution times of the algorithms and second we consider the limits of the geometric face identification algorithm. We implemented our algorithms in Java and executed on a laptop running Windows 10 Home build 14393.1066 with 8.00 GB of RAM and an Intel® Core™ i7-3630QM CPU running at 2.40 GHz.

Graph Generation. For limitation investigation and verification purposes we increased the number of generated points in the segmented triangle resulting in an increase in the number of faces in the corresponding planar graph. For simplicity, we generated planar graphs using $n = m$ number of points (according to Lemma 1).

We report the runtimes of our algorithms in Figure 5. For clarity, we note that the horizontal axis in Figure 5 is the Lemma 1 value for n (not the number of faces contained in the graph) and the vertical axis shows the time in seconds taken to (1) generate the planar graph and (2) analyze each planar graph containing n^2 faces. As shown in Figure 5, as n (hence, the number of faces in the planar graph) increases, our planar graph construction algorithm increases moderately via second-degree polynomial growth. That is, increase in time to construct the planar graph is nominal compared to the growth in time of geometric face identification. In fact, the average planar graph generation is 46% that of the time required to execute the geometric identification algorithm of that same graph. We conclude that our planar graph generation algorithms lead to efficient test-generation procedures to maximize use-case extremes of the geometric face identification algorithm.

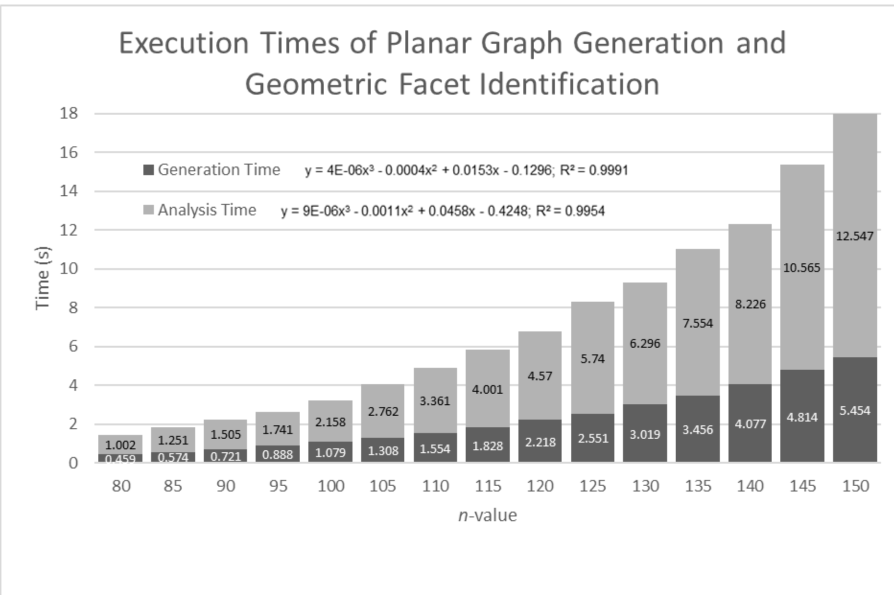


Figure 5: Time Analysis for Algebraic Verification of Face Identification in a Planar Graph

Epsilon and Face Calibration Using the Generated Planar Graph. We consider the impact of the number of points; that is, if we fix a triangle T and increase the number of points (n and m of Lemma 1). For simplicity, we again consider the case where $n = m$. As n grows without bound, the area of each individual face will tend to 0. That is, the measure of at least one angle in a triangular (or quadrilateral) face approaches 0 as $n \to \infty$. This scenario describes the upper bound of geometric identification algorithm since, computationally, there exists an $\epsilon > 0$ for which we cannot discriminate between an angle measure of $0°$ compared to $(0 + \epsilon)°$; similarly, detecting if two points are distinct within ϵ-distance.

With some investigation, the group was able to make a conservative conjecture based on a sample of planar graphs that our target software system required ϵ-precision between $1 \bullet 10^{-6} \leq \epsilon \leq 1 \bullet 10^{-4}$. It is important to note that the space of planar graphs in our target system was large and thus we could not definitively determine the required precision. The students thus argued for a two-pronged approach for software robustness and reliability: (1) determine a lower-bound ϵ threshold with our testing procedures described above and (2) add strategically placed lines of code to report warnings of possible precision issues (i.e. $\epsilon \leq 1 \bullet 10^{-6}$).

With our testing procedures, we empirically calibrated the epsilon value in the geometric face identification algorithm. As n increases, the faces within the triangle become increasingly smaller. When the epsilon value was too small ($\epsilon \leq 1 \bullet 10^{-7}$), the algorithm would fail due to an incalculable distance between two points. With proper epsilon calibration, we are able to generate a large number of very small faces (on the order of millions) without failure with a high success rate. Using the target system requirements as a guide, these cases where $\epsilon \leq 1 \bullet 10^{-7}$ were extreme and demonstrated to our group the viability of the legacy source code implementation without arbitrary-precision mathematics. In fact, our second prong of attack where we modified the code to emit precision-based warning messages was also illuminating: no warnings occurred in practice! As skeptical people, this left us feeling uneasy, but increasingly confident of our procedures with time.

Identifying these limitations using our parametric method for planar graph construction informed our testing of the overall system and thus mitigated false error detections. In some respects, these findings are indicative of a success story: developing a configurable method of testing informed us about the utility of legacy code and overall system limitations, but also offered insight into future system improvement by including arbitrary-precision mathematics.

6 Conclusion

We have presented a case study exploring the mathematical limitations of a legacy code implementation of a face identification algorithm for planar graphs. Our algorithm maintains tunable parameters facilitating increasingly complex planar graphs with faces of decreasing size. Experimentally, we demonstrated the efficacy of our planar graph construction algorithm and we feel our experience as a case study are important artifacts to share.

References

[1] M. de Berg, O. Cheong, M. Overmars, and M. van Kreveld. *Computational Geometry: Algorithms and Applications (third ed.)*. Springer-Verlag, Berlin/Heidelberg, Germany, 2008.

[2] D. Eberly. The minimal cycle basis for a planar graph. *Geometric Tools*, 2015. `http://www.geometrictools.com/Documentation/MinimalCycleBasis.pdf`.

[3] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, Berlin/Heidelberg, Germany, 1987.

[4] C. Jordan. *Cours D'analyse de L'Ecole Polytechnique*. Gauthier-Villar et fils, 1893.

[5] Oracle: Java standard documentation, May 2019. `https://docs.oracle.com/javase/10/docs/api/java/math/BigDecimal.html`.

[6] R. S. Pressman and B. R. Maxim. *Software Engineering: A Practitioner's Approach, 8th Edition*. McGraw-Hill, New York, NY, 2014.

Learning from Peers: Code Review in CS1*

John P. Dougherty
Department of Computer Science
Haverford College
Haverford, PA 19041
jd@haverford.edu

Abstract

We explore the use of code review in the process of inducing students to reflect upon not just getting the program right, but to appreciate the significance of proper design and implementation of a program that is clear and maintainable. In this paper, we report on how we use code review as a team-based active learning activity in CS1. We describe the methods used to conduct this review and include survey results of student perceptions of how this exercise impacted their understanding of the coding process.

1 Introduction

Computer science is more than learning to program. However, the ability to write clear well-designed and appropriately implemented code is appreciated in both academia and in industry. Software development consists of managing teams of programmers to implement code in such a way that all the distinct pieces work together. As demonstrated numerous times [2], this is not a trivial task. Code review is often used in industry as a way to ensure code is clear and well understood not just by the programmer but to those who might be using these programs in other routines.

CS1 at Haverford College provides a rigorous introduction to the field of computer science by requiring students to reflect on how data is represented and manipulated. To demonstrate this understanding, we expect students to complete a series of programming assignments, presently in Python. We have been using a code review exercise midway through this course in order to provide a mechanism for students to see the many different ways a problem can be solved, as well as to discover what features of a program provides a clear representation of an algorithm.

After a section looking at the relevant research in this area, we discuss the methodology used in order to provide this code review experience for the students in CS1. Results are presented from a survey completed by students after the code review exercise but well before the end of the term. Finally, we present our observations suggested by the survey results and how we plan to use this process in future courses, as well as thoughts about how this approach might be used by other computer science programs.

2 Relevant Research

Peer-based learning initiatives have been used in computer science education for quite some time [1]. They appear to be more effective in learning relative to standard lecture, mostly because they are a form of active learning that engage students to both listen and contribute, causing them to reflect on how they learn [5]. It is also a form of team-based learning in that groups work together to improve the design and implementation of code [4].

The practice of using peer review to teach has been seen in undergraduate education [8], with a few efforts leading to support tools [7]. In a few contexts, peer review has been used as an approach to help scale learning as class sizes have grown, including classes conducted online [3]. Furthermore, experience with code review provides some professional preparation as this approach is used often in industry (e.g., in a recent informal survey, 76.4% of software engineers report using code review [6]).

3 Methods

CS1 at Haverford College presently manifests in two of our courses. The first is entitled “Introduction to computer science” (cs105). This course presents the foundational concepts in computing, initially with a functional approach (e.g., recursion, immutable data), followed by imperative thinking where loops and mutability are introduced. The second CS1 course is entitled “Introduction to computer science and data structures” (cs107), and it provides an accelerated pace to introduce these concepts. In principle, the first part of cs107 covers all of cs105.

Code review has been used in each of these courses at Haverford College depending upon the choice of the instructor. The results presented here are obtained from the students who enrolled in cs105 in the fall of 2018.

Initially, students complete an assignment where they are given a circle and a rectangle and are asked to design an algorithm to determine if there is any overlap in two dimensions. We have found that this assignment visualizes well, and the solution is not trivial, providing a variety of approaches to solve. Visual test suites provide a way to see the various cases involved in this problem. In our experience, most students use a design by cases approach. However, with a few insights, the number of cases can be minimized.

Students complete, test and submit their code for this assignment within the first two weeks of the fall term. This assignment is scored and returned to the students directly with feedback from our grading team, which consists of undergraduate teaching assistants, and is supervised by a lab instructor. While students are completing the next assignment, we take these circle-rectangle-overlap programs from the first lab and strip away all identifying features. Even though we capped our course at thirty-nine students in the fall, this was a substantial amount of work.

Students are then partitioned into groups of approximately six each in such a way that they all are attending the same one-hour weekly lab period. Each student then receives a copy of the program from the first lab completed by another student in their group and are asked to review the code using the following rubric:

- Is the code readable? What makes the code readable?
- Does the code work as specified? How do you know?
- Does the algorithm implemented differ from your solution? If so, how?
- How might you solve this program now given your review of another approach?
- Any other feedback/thoughts?

Students are given a week to review this code, and then during their weekly lab to make a 5-minute presentation using the projected code to provide feedback using the above rubric as guidance.

We also asked students to fill out a quick evaluation of those doing the reviews. This feedback is collected and merged with the review of the lab instructor and the course instructor.

As this process completes, students are asked to complete a brief survey. The results of this survey are outlined in the next section.

4 Results

In this section, we present evidence of changes in students' perceptions of the coding process after this code review exercise in CS1. In each of the four figures, students were asked if they either strongly disagree (1), agree (2), neutral (3), agree (4), or strongly agree (5) with the statement above the figure. Also note that we made both positive and negative statements to maximize the likelihood that students read each statement. We also asked that each student explain their response after stating the degree of agreement to further ensure they were responding to the statement.

Responses were received from 28 students for a 72% response rate. Of these 28 who responded, 27 had completed the coding review exercise, implying that a single student did not complete the code review but did complete the survey. The survey was completed anonymously to encourage honesty.

Figure 1 and Figure 2 capture the clarity of the presentation of the exercise. Basically we wanted to know if students felt comfortable with the instructions given that this may have been the first time they ever experienced such an exercise in a coding class.

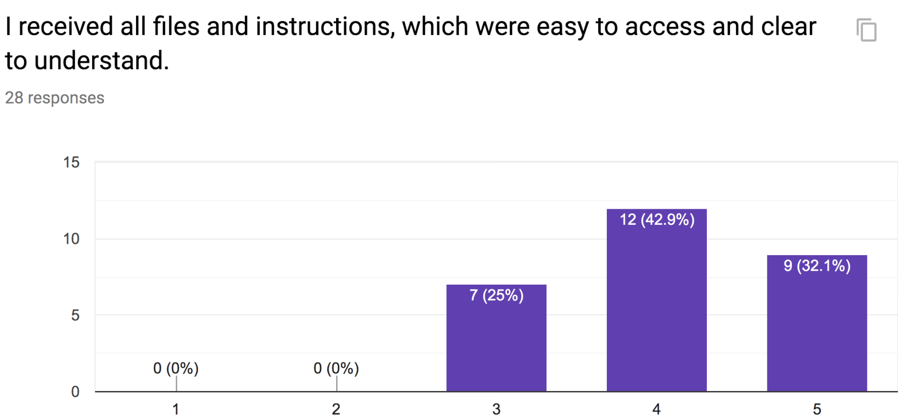


Figure 1: Clarity of the Instructions for the Activity

Qualitative responses regarding clarity were positive for the most part. There was one suggestion, "I wish we had talked about it in class beforehand." We found this a bit disappointing as we did demonstrate a sample code review in lecture.

Evidence in Figure 2 indicates that students were not surprised for the most part. Qualitative responses support this notion. There were a few comments that were not positive:

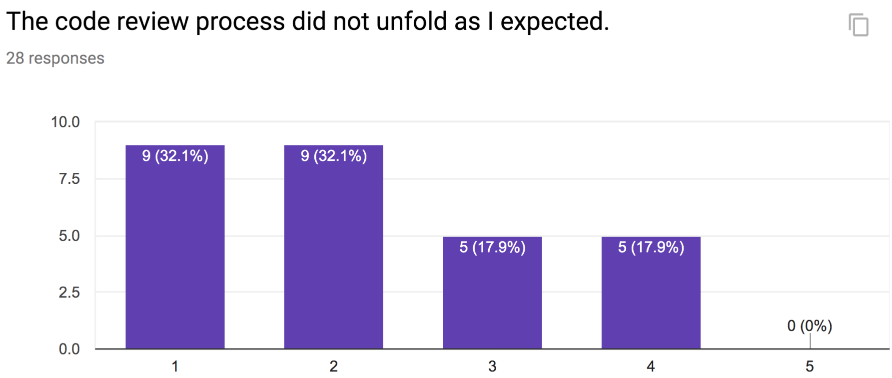


Figure 2: Expectations of the Activity

- I didn't realize that we were also going to explain why/how the code worked - I thought we were just going to say whether it worked and how well it was written
- My reviewer was not very well versed in programming, and was unable to properly critique my code.
- I couldn't get eclipse [sic] to work, so I felt like my review was subpar which really ruined my experience grading someone else.

Note this final comment was more a commentary of the tool used (Eclipse).

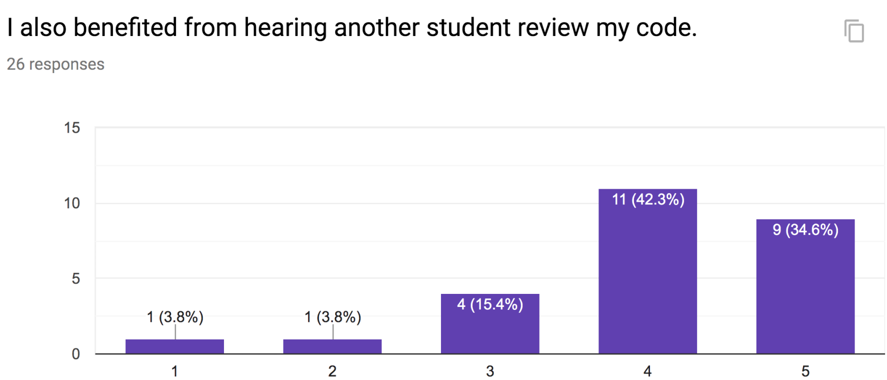


Figure 3: Perceived Benefit

Figure 3 gets at the perceived positive effect of hearing and watching another student comment about their code. Remember, the student did not know which student they were reviewing, but they did know that student was in the room (although often students would "out" themselves). Most people felt they benefited, but not all. From the qualitative comments, most were positive, and it was interesting to see unexpected observations such as below.

- I learned a lot from the other presentations all the same
- My reviewer was not well versed in program structure, and recommended that I define my helper function closer to where I used it in my program, i.e. inside my other function. I believe that lambda expressions are a bit too advanced for this course at this time, considering our code should be written so others in the course are able to review it.

This final comment was surprising as we do not introduce anything about anonymous functions directly in CS1. We suspect that students have experience with this and other advanced topics prior to the course.

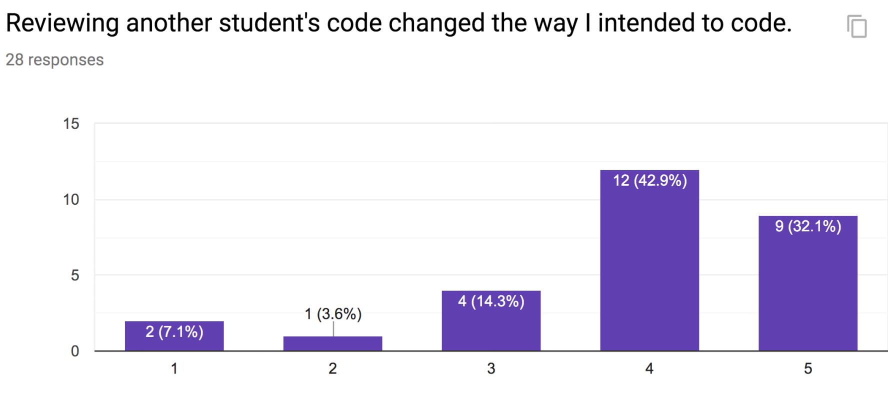


Figure 4: Perceived Impact on Coding

Figure 4 suggests that the majority of students believe that this exercise will impact the way they implement code in the future, which is the main goal. All comments in this area are included below (including the student who did not participate in the activity), and are telling of this range of responses.

- None
- It helped me see what was important to the code.
- I saw approaches I hadn't considered.
- The code i reviewed wasnt [sic] helpful as it had more errors then [sic] mine
- Including more explanations and examples became much more important to me, which I had otherwise struggled to understand in terms of the relevance.

- I plan on using words to define functions.
- I thought it was interesting to see all the different ways people approached the algorithm
- I realized there are many different ways to do the right code
- none
- Reviewing someone else's code made me consider how I format my code and has encouraged me to use more comments to explain my thinking
- It helped me realize how important comments are
- I agree but just because of the specific code I reviewed (didn't run) it was hard to know if the notes and portions of the code that were there were accurate.
- Showed me much insight into problem solving and coding my next algorithm
- Definitely going to comment more
- I didn't know python allows you to chain Boolean operators, very useful!
- I NEED TO WRITE COMMENTS!!!!!
- Yes, I am going to change the way I was doing coding.
- The person's code I had was very well developed so I think I can learn from how they did it
- I am going to write clearer code with more comments
- It depends on the code you are assigned. Sometimes it is the same or worse than yours. And sometimes it is way better. For me it was the same idea with almost the same cases. So I can't say it changed much.
- the code i reviewed was pretty bad and i don't believe i could take any of it and apply it to my future code
- Others had creative and new ideas.
- I learned how to make my code more obvious for other people to read. Also, new coding approaches.

Reading through these comments made it clear that not every activity works for every student, and it is useful to present material in more than one way. For example, we typically get that student's realizing importance of comments when they're trying to read someone else's code. At the same time, it is surprising that a student learned that you can chain binary operators together, a topic that we felt we had covered in lecture but clearly didn't get to the student.

Open responses elicited the following comments:

- In addition to the code I reviewed and the review of my personal code it was very interesting to hear the notes about everyone else's especially with the mast test suite.
- None
- We should do this more than once!

- It was a great step. If it could be done a little earlier, we can improve the rest of the codes.
- I don't know if this is very relevant. But I think we should be informed on the best code for the problems we are trying to solve. Because what we are doing so far is just attempting a better solution. I wonder if we would know what the best approach is at any point for time.
- My one suggestion would be to do the code review a week after that lab occurred.
- It was a good process

5 Conclusion and Future Work

From the results presented, we feel confident stating that this activity had either a positive impact on students' perceptions of how well they code, and sometimes had minimal impact. Also important, there is little evidence that suggests that students are negatively impacted by this activity.

We believe that code review provides an opportunity for students in the initial course to appreciate such features as clarity, documentation organization, documentation, and even the naming of variables.

The chosen assignment exploits visualizations as mentioned in the section on methods, especially during solution designs for novice computing students. We have not yet witnessed this code review exercise for students with any reported visual impairment.

We also need to be explicit that there is a substantial amount of administrative work to make the programs anonymous and ensure that each student gets a copy of code from someone in their group. The results of the survey support continuing to use this activity in our CS1 courses.

6 Acknowledgements

The author acknowledges Suzanne Lindell for supervising the code review process and David Wonnacott for proofreading this report.

References

[1] D. Badampudi, R. Britto, and M. Unterkalmsteiner. Modern code reviews - preliminary results of a systematic mapping study. In *Proceedings of the Evaluation and Assessment on Software Engineering*, EASE '19, pages 340–345, 2019.

[2] F.P. Brooks. Essence and accidents of software engineering. *IEEE Computer, 20(4)*, pages 10–19, 1987.

[3] C.E. Kulkarni, R. Socher, M.S. Bernstein, and S.R. Klemmer. Scaling short-answer grading by combining peer assessment with algorithmic scoring. In *Proceedings of the first ACM conference on Learning @ scale conference*, L@S '14, pages 99–108. ACM, 2014.

[4] P. Lasserre. Introduction to team-based learning. In Roelof Brouwer, Diana Cukierman, and George Tsiknis, editors, *Proceedings of the 14th Western Canadian Conference on Computing Education*, WCCCE '09, pages 77–78. ACM, 2009.

[5] M. Mani and Q. Mazumder. Incorporating metacognition into learning. In *Proceeding of the 44th ACM technical symposium on Computer science education*, SIGCSE '13, pages 53–58. ACM, 2013.

[6] Stack Overflow. Developer survey results, 2019. accessed online 18 August 2019 at `https://insights.stackoverflow.com/survey/2019#work-_-code-review`.

[7] J.G. Politz, S. Krishnamurthi, and K. Fisler. Captainteach: a platform for in-flow peer review of programming assignments. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*, ITiCSE '14, pages 332–332. ACM, 2014.

[8] D.A. Trytten. A design for team peer code review. In *Proceedings of the 36th SIGCSE technical symposium on Computer science education*, SIGCSE '05, pages 455–459. ACM, 2005.

The Role of Mentoring: How to Recruit and Retain Females and Minorities in STEM Degrees?*

Panel Discussion

Karen Paullet[1], *Diane Igoche*[1], *Natalya Bromall*[1]
Debra Nakama[2]
[1]*Robert Morris University*
{paullet,igoche,bromall}@rmu.edu
[1]*University of Hawaii, Maui*
debran@hawaii.edu

1 SUMMARY – Natalya Bromall

The Center for Cyber Safety and Education forecasts that there will be a 1.8 million worker shortage in information security professionals worldwide by 2020. This same study projects that in North America, there will be 265,000 more cybersecurity jobs than there are skilled workers in the field (Frost & Sullivan Center for Cyber Safety and Education, 2017).

Studies showing significant progress in students who recover from a grade decline show that personalized support and guidance can positively influence student retention. Transfer students from community colleges who enter four-year institutions encounter different barriers to persistence than traditional undergraduates. Creating accessible pathways for transfer students from community colleges requires institutions to have clear transfer processes and effective articulation agreements in place (Handel & Williams, 2012).

2 Diane Igoche and Debra Nakama

Community colleges among other two-year colleges are playing a role in creating pathways for under-represented students to four-year institutions (Boswell, 2004; Cohen et al., 14; Wang et al, 2017). Students are presented with a realistic pathway through the provision of smaller class sizes, flexible admissions, and lower costs. These factors make it possible for more underrepresented students especially low-income and first generation college students to begin exploring a career in the STEM fields. Women, students from a low-income background and minorities typically choose two-year colleges because they are more conducive to a work-life balance.

Students from under-represented populations in STEM report an intent to transfer into STEM fields when there is a high level of field-oriented interaction (Wang, Chan, Soffa & Nachman, 2017), i.e. engagement in STEM field related activities during their educational career, interactive sessions about career options, and the availability of a mentor to guide and inform academic and post-graduation choices. Over 42% of job opportunities in the STEM fields require employees to possess bachelor's degree (cite), there is an evident gender, race and socioeconomic gap in the participation and placement rates of individuals in the STEM fields as shown in the table below. Broadening the participation of these groups will contribute to the decrease in skilled worker gaps in this STEM field.

3 Karen Paullet

Students, especially from underrepresented populations, benefit from strong connections with a faculty member who can serve as an engaged role model (Cuseo, 2017). A study conducted by Glass (2013) shows that a lack of mentors is an important factor in few girls studying cybersecurity in middle school, high school and at the collegiate level. If the mentor projected a stereotypical image of a "geek or nerd", girls were less like to believe in their success in the field. The success of role models for young women is needed to help girls succeed in cybersecurity.

LeClair, Shih, and Abraham (2014) indicate that while equal pay and advancement opportunities are important for the retention of women in cybersecurity, having mentors is more important. The mentor does not need to be female but needs to have an interest in the mentee's success. Cheryan, et. al (2011) found there to be no difference between male or female recruiters in the STEM field but found that women mentors are more effective for keeping women in the field.

It is the panelist's belief that if students are matched to faculty mentors upon acceptance to the University they will have a positive experience in their field of study. A dedicated studentfaculty mentoring relationship allows the scholars to have personal, accessible support for matters directly related the CIS field. At the beginning of each semester, the mentor coordinator will hold an orientation meeting for new scholars and mentors. Faculty mentors will be assigned to two mentors, a dedicated mentoring style has been shown to effectively support students' retention in STEM.

4 Acknowledgment

The material in this submission is based upon work supported by the National Science Foundation under Grant #1834083. Any opinion, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] S. Cheryan, J. O. Siy, M. Vichayapai, B.J. Drury, and S. Kim. Do female and male role models who embody stem stereotypes hinder women's anticipated success in STEM? *Social and Psychological and Personality Science, 2(6)*, pages 656–664, 2011.

[2] J. Frost and T. Sullivan. cybersecurity workforce shortage to reach 1.8 million as threats loom larger and stakes rise higher, 2017. `https://www.isc2.org/News-and-Events/Press-Room/Posts/2017/06/07/2017-06-07-Workforce-Shortage`.

[3] J. L. Glass, S. Sassler, Y. Levitte, and K.M. Michelmore. What's so special about STEM? a comparison of women's retention in STEM and professional occupations? *Social Forces: a Scientific Medium of Social Study and Interpretation, 92(2)*, pages 723–756, 2013.

[4] S. Handel and R. Williams. The promise of the transfer pathway. collegeboard advocacy and policy center, 2012. `https://securemedia.collegeboard.org/digitalServices/pdf/professionals/handel-williams-promise-of-thetransfer-pathway-2012.pdf`.

Projects Using METAL Highway Graphs with Traveler Data*

Nifty Idea

James D. Teresco
Department of Computer Science
Siena College
Loudonville, NY 12211
jteresco@siena.edu

1 Abstract

The Map-based Educational Tools for Algorithm Learning (METAL) project[1] provides graph data representing actual highway systems around the world, along with interactive, map-based algorithm visualizations that operate on that data [2]. The graphs, ranging in size from just a few vertices and edges to about a half million, can be used within our outside of METAL's visualization system. METAL's graph data [1] is derived from the Travel Mapping (TM) travel hobbyist project[2]. TM allows its users to track their cumulative travels and view statistics and maps. METAL's graph data was recently extended to include information about which edges in a graph have been traveled by which TM users. These new "traveled-format graphs" were first used as part of the final project in an Analysis of Algorithms class. Working in groups, students chose at least one algorithm to implement that could make use of the traveled-format graphs in some meaningful way. Algorithms chosen ranged from simple, such as ranking users by the number of segments or total distance traveled, to straightforward variations on algorithms we had studied such as finding a "most-traveled" route with a modified Dijkstra's algorithm or a convex hull of a given traveler's visited vertices, to algorithms we did not study at all in class such as A*. Each group presented their project in a session at our campus-wide Academic Showcase. This "Nifty Idea" presentation will give a brief overview

[1]http://courses.teresco.org/metal
[2]http://travelmapping.net/

of METAL, describe the new traveled-format graphs, present some of what students have done with the graphs in this first usage, and discuss how these graphs could be used in engaging examples for additional algorithms and data structures.

2 Additional Information

The intended audience is instructors in Data Structures and Algorithms courses who might be interested in using the traveled-format graphs and other METAL data and visualizations in their courses. All of METAL is freely available on the web, so attendees with their own electronic devices will be able to explore the data and visualization capabilities. A handout will be provided with a URL and a QR code to help attendees find METAL either during the presentation or when they return home. I intend to run the presentation and demos using my MacBook Pro. An HDMI connection would be ideal, but I can connect to a projector with DVI or even VGA if necessary.

References

[1] James D. Teresco. Improved highway data and map visualizations for teaching data structures and algorithms. In Jeffrey Cummings Li-Jen Lester, Neelima Bhatnagar and editors Jason Sharp, editors, *2017 Proc. EDSIG Conference*, volume 3, Austin, 2017. ISCAP (Information Systems & Computing Academic Professionals), `http://iscap.info`. ISSN 2473-3857, `http://proc.iscap.info/2017/pdf/4325.pdf`.

[2] James D. Teresco, Razieh Fathi, Lukasz Ziarek, MariaRose Bamundo, Arjol Pengu, and Clarice F. Tarbay. Map-based algorithm visualization with metal highway data. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, pages 550–555, New York, NY, USA, 2018. ACM.

Using Simulation Games to Teach About ERP*

Nifty Idea

Jennifer Paige Nightingale
School of Business
Slippery Rock University
Slippery Rock, PA 16057

Business systems, such as enterprise resource planning (ERP) or customer relationship planning (CRM) can be difficult for students to understand, especially if they have yet to experience an internship that would expose them to a business environment along with such systems. A simulation or strategy game, such as Lemonade Tycoon, can mimic different modules included in ERP and can be demonstrated in a lecture along with the concepts of CRM. This game allows students to manage their own business, while making them responsible for paying rent, marketing the business, hiring staff, purchasing assets and ingredients, manufacturing a product, and collecting customer feedback. These tasks dovetail with several modules included within an ERP such as Manufacturing, Financials, Human Resources, and Sales and Marketing. This discussion will illustrate how functional areas of business are interdependent and need one another for an organization to operate efficiently and effectively. Additionally, the discussion includes how customer feedback—as stored in a CRM and integrated with an ERP—can help a manager make decisions that will improve his/her operations. Finally, the discussion validates the importance of how information systems are critical to everyone within an organization, not just the IT staff. Examples of the in-class demo/lecture as well as assignments will be presented.

Using Git in an Android Mobile Application Development Course*

Nifty Idea

Amanda Porter
Mathematics and Computer Science
Edinboro University of Pennsylvania
Edinboro, PA 16444
aporter@edinboro.edu

There are many challenges when teaching a mobile application development course using Android Studio, the official integrated development environment (IDE) for the Android Operating System (OS). The difficulties of using Android Studio to teach a mobile application development course include continuous updates to the IDE, continuous updates to the Android OS and the management of projects. The only one of these difficulties that can be controlled by the instructor is the management of projects. Git, a free and open source version control system, is seamlessly integrated into Android Studio and is essential for working with mobile application projects in a classroom environment. Utilizing Git provides students a repository for their code base and collaboration tools for group projects. Git also helps streamline student project submission and the grading process of the Android Studio mobile application projects. Without the use of Git or another version control system, projects become unmanageable in Android Studio.

Broadening Student Perspectives with Secondary Book Pairings*

Faculty Poster

Lonnie Bowe
Department of Mathematics and Computer Science
Concord University
Princeton, WV 24739
lbowe@concord.edu

Computing students from small, rural colleges often feel a disconnect from the "real world" of industry software development. It is a common industry complaint that students are not prepared for the workforce when they graduate. For the past five years, I have required students to read and respond to secondary reading books. These books are separate from the primary textbook and targeted toward non-academic audiences. The intention is to provide exposure to topics that are important, but tangential to the material covered in the course. The books also provide alternate perspectives from my own, expose students to current practices, promote a lifelong reading habit, and improve writing skills. This poster provides text suggestions, and offers tips on how others can use the same process in their courses. There will be space on the poster for others to write in their own text suggestions.

Proposing A Simple Algorithm for Teaching Concepts of Cryptographic Data Validation Through Hand-Coding Projects*

Faculty Poster

Penn P. Wu[1] *and Phillip Chang*[2]
[1]*DeVry University*, pwu@devry.edu
[2]*Ayla Networks Inc.*, philchang88@gmail.com

Cryptography is a difficult topic to most students and instructors. It requires students to have a solid background in pure mathematics and number theory to understand the notations and processes of calculation, not to mention developing their own encryption and decryption algorithms.

This presentation proposes a simple algorithm that can generate a pair of asymmetric keys. While one key is kept (or saved) by a server or a validating device (such as RFID reader); the other is printed, encoded, burned, or saved on a paper, a chip, or a digital device. The term "cryptographic data validation" refers to the use of the proposed "cryptographic keys" to check the accuracy of a submitted data for acceptance, such as validating an event ticket. The proposed algorithm consists of two steps: key generation and validation. The "key generation" algorithm can generate a pair of asymmetric keys: one is kept (or saved) by a server; the other is printed, encoded, burned, or saved on a paper, a chip, or a device. The "validation" algorithm describes how the key sent by the "validate" can be validated for key-pair matching (or the accuracy). The term "validator" refers to a device that will perform an arithmetic calculation to match the key presented by the "validatee" with the key kept by the "validator".

The presenters will: (1) describe the core concepts and mathematic foundations that lead to the development of the proposed algorithm, (2) explain why it is "simpler" compared to the professionally developed algorithms, such as Triple DES, Blowfish, RSA, Diffie-Hellman, and Elliptic Curves, (3) discuss the pedagogies and instructional strategies that could engage students in hand-coding projects to explore concepts of cryptography, (4) demonstrate sample codes that illustrate how the proposed algorithm works, and (5) identify the weaknesses and threats of the proposed algorithm.

Teaching Cloud Computing in Colleges*

Faculty Poster

Weidong Liao and Osman Guzide
Department of Computer Science, Mathematics, and Engineering
Shepherd University, Shepherdstown, WV 25443

As a well-accepted computing paradigm, cloud computing not only impacts how IT professionals administer and manage computing infrastructure, but also changes the way software applications are developed, tested and deployed. Many industry, academia and government agencies have switched their IT infrastructure to cloud computing. It is therefore highly desirable that graduates from computing degree programs in colleges and universities are equipped with knowledge, concepts and skills of cloud computing.

Various practices have been conducted towards teaching cloud computing. Institutions have revised their current CS/IT courses to introduce cloud computing fundamentals. Other universities and colleges have been adding new courses dedicated to cloud computing. Since cloud computing platforms varies in their architecture and implementation, cloud computing standards such as NIST Cloud Computing Standards would be helpful, regardless of the fact that whether new courses are added or existing courses are revised.

A couple of methodologies have been adopted to help students gain practical cloud computing skills, for motivating student learning or making our students more employable. Cloud simulators, originally developed for cloud based research, could very well used in cloud computing education. Leading cloud service providers, such as Amazon, Microsoft, and Google are providing assistance for colleges and universities to teach cloud computing using their cloud platforms. The latter seems gaining more momentum as the knowledge and skills students gained could smoothly transmitted to experiences that industry is look for.

In this poster, we describe and discuss a variety of perspectives, opportunities and challenges in teaching cloud computing in colleges and smaller universities. Our practices, experiences and lessons learned in introducing cloud computing in our classrooms are also presented.

Preparing for the Inevitable: What to Do Before the Breach*

Faculty Poster

Elizabeth McGovern-Cole
Communications and Information Systems
Robert Morris University and Blue Bastion Security
canonsburg, PA 15317
elcst117@mail.rmu.edu

This research focuses on those best practices' organizations should adhere to for network health and security. No longer is it if an organization will be the victim of a cyber-attack, but when. There is a shift which began in 2019 of hackers moving their focus toward governments. Cities and counties across the United States are the new targets, while hospitals and other medical facilities are still considered worthwhile targets for hackers to launch their ransomware. In a review of the literature, there are specific processes all organizations should be cognizant of and work into their operational procedures. Organizations who provide security as a service (SECaaS) and offer Incident Response (IR) face not only finding and identification of an attack when engaged but also with the containment and eradication of an attack in an environment utterly unfamiliar to them. In many cases, the responding IR team learns the environment is unfamiliar to those individuals who maintain the environment on a day-to-day basis. This paper will discuss the best practices organizations should adopt to mitigate the impact of a breach and those tale-tale signs that there may be an impending ransomware attack.

A Framework for Teaching Cybersecurity on the Topic: Software Development - What Could Possibly Go Wrong?*

Faculty Poster

David Carlson
Computing and Information Systems Department
Saint Vincent College
Latrobe, PA 15650
david.carlson@stvincent.edu

Students learning software development sometimes try to access items off the end of an array, giving a buffer overflow, yet they do not realize this. Cybersecurity students may never have seen what such flawed code looks like or how it behaves. Prospective students and visitors may have heard that code can be flawed but usually do not know specifics.

This poster shows a way to remedy this situation by presenting cybersecurity examples in a customizable and easy-to-use framework. Students and visitors are guided through the examples by following directions on web pages that can be modified to provide the level of detail appropriate to the audience. Visitors may be given a general explanation and can see what happens when a flaw is triggered. They do not need to see detailed code.

Many cybersecurity and software development topics are difficult for beginning students to visualize until they experience them firsthand. These students can be shown the code for each example and can run and interact with it to see clearly what it does. Students could even be asked to identify security issues in the code before being told what flaws are present. Some instructors might also want to ask their students to fix these flaws, if time allows.

These materials are available on the web. This includes trying out the author's installation as well as downloading the materials for installation elsewhere. Current examples include two web applications and four standalone executables. The web applications illustrate cross-site scripting and SQL injection attacks. The executables show an ill-conditioned problem, integer overflow, stack overflow, and buffer overflow. These allow people to try out, in a safe setting, software that does not behave as it should. Along the way, students can learn secure coding principles such as the importance of filtering user input.

A Barometer for Marine Shipping Company: A Crucial but Struggling Industry*

Faculty Poster

H. Wei[1] *and M. Pinto*[2]
[1]*US Marine Merchant Academy (USMMA)*
Kings Point, NY 11024
hsinrong.wei@bcc.cuny.edu
[2]*Computer Systems Technology Department*
NYC College of Technology, CUNY
Brooklyn, NY 11201
mpinto@citytech.cuny.edu

Although it is responsible for carrying out ninety percentage of all world's trade flows, the international maritime shipping industry struggles to attract and maintain investors interested in their industry. The result is the industry relying solely on itself for growth and profitability. Stock price of publicly traded companies can serve as a barometer of a company's performance giving managers a tool to gauge and calibrate new investment. This paper describes a step-by-step process of creating a neural network to predict stock price for the shipping industry. This process aims to provide investor with a tool that can be used in deciding to invest or not in a shipping company. It is a hands-on example that can also be incorporated into college courses on Web services and/or mobile applications.

CIS Cyber Scholars: Developing a Faculty-Student Mentor Model*

Faculty Poster

Diane Igoche, Natalya Bromall, and Karen Paullet
School of Communications & Information Systems
Robert Morris University
Moon Township, PA 15108

The Computer Information Systems Department at Robert Morris University has established the CIS Cyber Scholars Program with funding from the National Science Foundation - Award #1834083. The program is designed to equip low-income, academically talented community college transfer students with a demonstrated financial need with the financial, academic and practical support to advance into STEM careers or graduate school.The objectives are to recruit 30 students from rural community colleges, graduate those who have been recruited with a STEM degree, help the student enter graduate school or a STEM career and to institutionalize effective practices to support transfer students and promote success in their academics and careers.

AbuSniff: An Automated Social Network Abuse Detection System*

Faculty Poster

Sajedul Talukder
Mathematics and Computer Science
Edinboro University,
Edinboro, Pennsylvania 16444
`stalukder@edinboro.edu`

In our research, we sought to develop an automated social network abuse detection system which is able to reduce the attack surface of its users, by reducing the number of, or isolating friends predicted to be perceived as potential attack vectors. This presents a substantial challenge as adversaries leverage social network friend relationships to collect sensitive data from users and target them with abuse that includes profile cloning, stalking, identity theft, fake news, cyberbullying, malware, and propaganda. We leverage these findings to develop AbuSniff (Abuse from Social Network Friends), a system that evaluates, predicts and protects users against perceived friend abuse by suggesting several personalized defensive actions for such friends. We began by developing the first ever mobile app questionnaire, that can detect perceived strangers and friend abusers. To replace the questionnaire, we then introduced mutual Facebook activity features that have statistically significant overall association with the AbuSniff decision and showed that they can train supervised learning algorithms to predict questionnaire responses using 10- fold cross validation. We trained our system with several supervised learning algorithms, including Random Forest (RF), Decision Trees (DT), SVM, PART, SimpleLogistic, MultiClassClassifier, K-Nearest Neighbors (KNN) and Naive Bayes and chose the best performing algorithm for predicting each of the questionnaire questions. Our approach provides a method to evaluate AbuSniff system through online experiments with participants recruited from the crowdsourcing site from 25 countries across 6 continents. Results showed that the predictive version of AbuSniff was highly accurate (F-Measure up-to 97.3%) in predicting strangers or abusive friends and participants agreed to take the AbuSniff suggested actions in 78% of the cases. When compared to a control app, AbuSniff significantly increased the participant self-reported willingness to reject invitations from strangers and abusers, their awareness of friend abuse implications and their perceived protection from friend abuse.

Integrating Fashion into Robotics for Broadening Participation*

Faculty Poster

Lily Liang[1], Jeffrey Enamorado[1], Yanxia Jia[2], Briana Wellman[1]
[1]Department of Computer Science and Information Technology
University of the District of Columbia
Washington, DC 20008
{lliang,Jeffrey.enamorado,briana.wellman}@udc.edu
[2]Department of Computer Science and Math
Arcadia University, Glenside, PA 19038
jiay@arcadia.edu

Innovations in computing require talents from diverse backgrounds. However, only one in five computer science degree recipients in 2015 are female, according to published data. The percentage of women in the computer science field has had little improvement when compared to decades ago. Research has shown that social identity plays a significant role in students' motivation to learn. To broaden the participation of computing, we integrated fashion into robotics and computer programming by designing and implementing fashion-bots.

The purpose of the integration is two-fold: 1) create robots with appearances more relatable to female students; 2) create robot programming experiences more meaningful and enjoyable to female students. Our goal is to improve female students' motivation in computing by increasing their sense of belonging. We experimented with different hardware designs to create low-cost fashion bots that are not only visually relatable but also adjustable to accommodate personal preferences. A demonstration was offered at a local library for a programming club of elementary-aged students. And we are currently developing a short introductory course on fashion-bots programming and building, to be offered at a high-school this Fall. Findings will be shared on the poster.

Our vision is to transform robotics and computing into a field with 1) cultures that female students can relate to; and 2) activities that allow them to express themselves.

Cyberstalking: What Are Students Doing Online?*

Faculty Poster

Karen Paullet[1] *and Adnan Chawdhry*[2]
[1]*Robert Morris University*
Moon Township, PA 15108
paullet@rmu.edu
[2]*California University of Pennsylvania*
California, PA 15419
chawdhry_a@calu.edu

Over the years, Internet usage has increased causing an impact on the number of online / harassing cyberstalking cases. The primary functions of the Internet are to communicate and research information allowing people to communicate locally or globally in business, education and their social life. The Internet has made it easy for people to compete, meet a companion, or communicate with people on the other side of the world using a mouse click. In 2018, according to the Internet World Stats Report, 320,059,368 people use the Internet in the United States; as a result, there is a concern for Internet safety (Internet World Stats, 2018).

Effect of Security Breaches on Stock Price*

Student Poster

Collin McClaine
Advisors: Amanda Holland-Minkley and Robert Dunn
Washington and Jefferson College
Washington, PA 15301
mcclainecm@jay.washjeff.edu

Modern times have begun to reveal the true cost data breaches have brought to companies. In my project, we are testing to see if there is an effect from security breaches on stock prices. Security breaches will be sampled from the years 2009 to 2018. We will be looking to see whether there is an increase in stock price, no change in the stock price, or a decrease in stock price, after the security breach occurred. This will be done using a linear regression through the statistical software STATA. Changes in stock price will be modeled with a Time Series Model over a large sample of companies that had significant security breaches. The breaches that occurred will be large scale security breaches of publicly traded companies and stock price of these companies will be analyzed in three different ways. These three ways include: before the breach had happened, when the breach occurred, and finally after the security breach occurred. Data will be drawn from the security breach dataset to ensure sufficient observations to minimize bias, give the model adequate explanatory power, and satisfactory explanation of the variation in stock price. The model will have several variables and these variables will be mainly financials of the companies that can affect stock price. We would also like to include a dummy variable, indicated by a 1(when there is a security breach present on that date or 0(when there is no security breach present). This may be able to help answer the question of whether the market can push companies to protect against security breaches.

Travelling Salesman Challenge for Algorithm Analysis Students*

Student Poster

Jesus Argel, Mohammed Batarfi, Brandon Belna, Kyle Burda, Isabel Ho Li, Tram Nguyen, John Sawyer, Hala ElAarag
Stetson University
Deland, FL 32723

One of the most important skills that students should master to succeed in the 21st century is the transfer skill. This is the skill that enables an individual to take what was learnt in one situation and apply it in another situation. Algorithm Analysis is a core component of the Computer Science Curricula. In this paper, we share our experience in using this course as a vehicle to enable students to master their transfer skill and hence ensure a deeper learning experience. The students were asked to participate in a challenge provided by a flight booking company to solve a harder version of the Traveling Salesman Problem. In this paper, we present the challenge as well as three different proposed solutions. We provide a comparison of the solutions based on their execution time and cost in various scenarios.

Analyzing Tweets for Cyber Security*

Student Poster

Kyle Murfitt, Eunbi Kim, Devon Smart, Ching-yu Huang
School of Computer Science
Kean University
Union, NJ 07083

We aim to perform sentiment analysis on Twitter tweets related to cybersecurity and compare it with our server's logfile. This project focuses on developing models and approaches to identify cybersecurity terminology and identify patterns of cyber threats in tweets. We will classify text and determine whether a tweet is of neutral, positive, or negative polarity. Tweets related to cybersecurity will comprise the data we utilize as the training set, which will help us study related terminologies and patterns concerning cybersecurity in social media.

Twitter is a popular social media platform where users create status messages called tweets. We are trying to capture the sentiments of tweets and understand what words strongly correlate with cybersecurity, identify the patterns of cyber threats, and uncover connections between the users who tweeted and their human expressions. Several tools and methods will be developed for natural language processing on the Twitter data. We focus specifically on automating the sentiment analysis techniques and processes. The tweets will be downloaded, processed with Python3, standard libraries in Python3, and Tweepy API. These tweets will be saved into the database MySQL. The unstructured Twitter data will be transformed into structured and clean data that includes username, text, friend and follower counts, and geo locations with Python3. Also, the unstructured server logfile data will be transformed into structured and clean data that include IP address, date time, and geo locations of failed login attempts to the server with PHP. We will get IP addresses from logfile then develop to get geo-location from the IP address using PHP and insert into database. We will compare this data to compare to the countries and geo location from twitter data, therefore we figured if there are any connection between logfile and twitter data. This data will be visualized with Google chart using JavaScript and PHP.

The project will research and develop methods and techniques to perform sentiment analysis on Twitter Tweets. It will encompass building the database of the related terminologies, summarizing the patterns of cyber threats, and identifying the connections between users who tweeted and retweeted, as well as their human expressions. Steps for this research will be Develop programs that can download the tweets automatically, Develop programs that can extract information from the tweets, clean and transform the data into the database MySQL, Design database schema and storage architecture that can store tweets and cybersecurity related information (log files) and the relationship between the data, and Develop integrated programs to perform sentiment analysis on the text, summarize the patterns of cyber threats, and identify the connections between users who are experts and cyber hackers.

This project will try to find patterns of cyber threats by developing a database of cybersecurity terminology using Twitter tweets. This project will provide an opportunity to apply data mining technology and research on social media and cybersecurity. The results of this project will have a broader impact on cybersecurity.

Teaching Parallel and Distributed Systems in the Cloud*

Student Poster

William Krzyzkowski
James Madison University
Harrisonburg, VA 22807
krzyzkwm@dukes.jmu.edu

This project explored the cost of creating and running a parallel cluster of machines in three common cloud providers. This research sheds light on not only the performance of highly parallelized programs running on a virtual cluster, it also shows how a school can run a class focused on parallel programming without investing in a costly hardware cluster.

Why Elliptic Curve Diffie-Hellman (ECDH) Is Replacing Diffie-Hellman (DH)*

Student Paper

Edmond Mbadu
Department of Computer Science
Chestnut Hill College
Philadelphia, PA 19118
mbadue@chc.edu

The Elliptic Curve Diffie-Hellman is an anonymous key agreement protocol that allows two parties, each having an elliptic-curve public-private key pair, to establish a shared secret over an insecure channel[1]. The key obtained will then be used to encrypt a message using a known symmetric cipher such as AES, Twofish, DES, etc. Motivation for this work comes from an interest in the application of *group theory* in cryptography. Real world applications of the ECDH are currently seen in secure internet protocols which are used by different web browsers such as Google Chrome, Mozilla Firefox and Safari. In this paper, we describe the ECDH and explore both its security and the reasons why it is replacing the Diffie-Hellman (DH) protocol.

References

[1] Elliptic-Curve Diffie-Hellman. Available https://en.wikipedia.org/wiki/Elliptic-_curve_Diffie-Hellman.

An Exploratory Deep Learning Approach to Mobile Malware Detection*

Student Paper

Jocelyn Diaz-Gonzalez, Alex V Mbaziira, and Michelle Liu
Marymount University
Arlington, VA 22207
{j0d88296,ambaziir,xliu}@marymount.edu

Traditional malware detection techniques have been widely adopted in various anti-malware programs for desktop as well as laptop computers. However, its counterparts in the arena of mobile devices gain less visibility and discussion. This paper investigates the use of deep learning to detect mobile malware. We use a deep neural network (DNN) implementation of deep learning called DeepLearning4J (DL4J) to generate our models for mobile malware detection. The results from our models when applied to the test sets are encouraging. The models detect mobile malware with accuracy rates ranging between 97% and 99% when applied to two types of datasets. Furthermore, the analysis illustrates that the effectiveness of DNN on detecting Android Malware increases when more layers are added to the classifier. This study demonstrates the practicability of using the DNN to continually learn from the past malware attacks and to detect and predict new types of mobile malware threats. This paper sheds light on a new direction of examining malware prevention, detection and prediction and motivates future direction of exploring new strains of mobile malware can be detected using machine learning

Detecting Personally Identifiable Information Violations Attempts in an Email Provider using Probability Density Function*

Student Paper

Ana H. Valentin
Marymount University
Arlington, VA 22207
`ahv22211@marymount.edu`

Data protection is an object of serious attention by enterprise security personnel to protect personally identifiable information in compliance with privacy laws from will potential serious consequences of noncompliance through an email provider. This study examines whether the Poisson and Gamma-Poisson Distributions explain the use and value of regular expressions through the detection of the probability of personally identifiable information violation attempt for data loss prevention.

The study describes the occurrences of personally identifiable information attempt violations when regular expressions were updated to neither diminishes nor increases the chance of another incidence and Poisson-Gamma model, and simultaneously describes the personally identifiable information attempt violation occurrence and intensity at once for a suitable model for zero inflated data which reduces personally identifiable information violation. attempts. The results of the study provide a quantitative methodology for organizations to use regular expressions and Gamma-Poisson Distribution for monitoring and controlling the incidence probability of personally identifiable information attempt violations on email providers over a period of time.

Hardware-Tampering Security Risks in the Supply Chain*

Student Paper

Zelalem Mengistu, Pipop Nuangpookka,
Phuong Nguyen, and Gregory Liu
School of Business and Technology (SBT)
Marymount University
Arlington, VA 22207
{zwm26248,pnuangpo,p0n75796,gliu}@marymount.edu

The topic of security risk on the global supply chain is a vast one, as it incorporates various sub topics which are hived under the bigger picture of supply chain. For this study, we focused on hardware-based security risks which are caused by implanting a tiny chip on the original motherboard architecture, during manufacturing or while in transit. This paper examined the various hardware attack detection methods- mainly on destructive and non-destructive methods of hardware-based error detection. The destructive detection methods is extremely difficult to implement as it often requires a physical presence to inspect the device during the manufacturing and/or transition process. Despite this fact, we tried to detect abnormal activities of hardware components through non-destructive method by building a custom code using JSensor - a high performance sensor network simulator developed with a Java programming language. To monitor the hardware, we set a scheduler to gather the required information (example: every one or two hours during off-peak hours) so as to identify similarities and differences of the resources used in the computer systems. Besides CPU loads, CPU speed/clock rate has also been retrieved by using JSensors and Oracle Java Standard. By default, the size of configuration file does not automatically change. We deliberately altered the size and run the JSensors code which was scheduled to run for every three seconds and we were able to detect it as JSensor flagged the alteration which we deliberately made. Therefore, we concluded that besides monitoring hardware sensors for suspicious activities, checking an important file whose size should remain unchanged is an effective method of monitoring critical systems

within a given organization. The paper also identifies the major hardware-based attack vectors on the supply chain, targeting various organizations. We concluded by making suggestions on how hardware-based supply chain risks could be mitigated and/or eliminated through future efforts.

The cyber supply chain risk is a real-world practice oriented and problem-centered challenge which governments and organizations are currently facing. In effect, the security of the global technology supply chain has been compromised, even if consumers and most companies didn't know about it (Robertson & Riley, 2018). This trend, if it has not been mitigated to the soonest, will become an Achilles hill and ultimately jeopardize the eco-system of the global supply chain.

Always Fresh: Tracking Food Expiry with Mobile App*

Student Paper

Kazi Zunnurhain, Karan Nayee, Rahul Patel, and Parth Patel
Department of Computer Science
Kean University
Union, NJ 07083
{kzunnurh,nayeeka,patelra2}@kean.edu
parth199042@gmail.com

Modern world can not be think of without the existence of mobile and mobile apps. Mobile app has the widest varieties in terms of services to the users. Due to its portability and convenience any tasks can be accomplished with minimal efforts from the user. Many iOS Mobile Applications such as Fridgely, Expiry and Fresh Box let the users manually input and track the food items from its inventory to help less wastage of food items. Our project implements a similar idea where the user will be able to access the information regarding the existing food items in the inventory. But the difference is our app can scan the barcode of the purchased item and store the details in our database immediately. We have accomplished more flexibility and automation for users' perspective. To implement this project, we used a third party app builder called "Appery" along with several plugins from Cordova. With the use of this application, users will achieve the privilege to a full scale tracking of food items and the receipts including the barcode values. For an instance, a user is desiring to see list of items purchased previously, to make sure nothing is expired. The application will provide a list of all the items with item description and the remaining days till expired. With just few clicks the user will accomplish the task and on top of that the notification feature will make it even simpler for the user to identify the expired items.