The Journal of Computing Sciences in Colleges

Papers of the 35th Annual CCSC Southeastern Conference

January 28th-29th, 2022
Bob Jones University
Greenville, SC

Baochuan Lu, Editor
Southwest Baptist University

John Hunt, Regional Editor
Covenant College

Volume 37, Number 5 **October 2021**

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.

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:

Karina Assiter, President (2022), (802)387-7112, karinaassiter@landmark.edu.

Chris Healy, Vice President (2022), chris.healy@furman.edu, Computer Science Department, 3300 Poinsett Highway Greenville, SC 29613.

Baochuan Lu, Publications Chair (2021), (417)328-1676, blu@sbuniv.edu, Southwest Baptist University - Department of Computer and Information Sciences, 1600 University Ave., Bolivar, MO 65613.

Brian Hare, Treasurer (2020), (816)235-2362, hareb@umkc.edu, University of Missouri-Kansas City, School of Computing & Engineering, 450E Flarsheim Hall, 5110 Rockhill Rd., Kansas City MO 64110.

Cathy Bareiss, Membership Secretary (2022), cathy.bareiss@betheluniversity.edu, Department of Mathematical Engineering Sciences, 1001 Bethel Circle, Mishawaka, IN 46545.

Judy Mullins, Central Plains Representative (2023), Associate Treasurer, (816)390-4386, mullinsj@umkc.edu, UMKC, Retired.

Michael Flinn, Eastern Representative (2023), mflinn@frostburg.edu, Department of Computer Science Information Technologies, Frostburg State University, 101 Braddock Road, Frostburg, MD 21532.

David R. Naugler, Midsouth Representative(2022), (317) 456-2125, dnaugler@semo.edu, 5293 Green Hills Drive, Brownsburg IN 46112.

Grace Mirsky, Midwest Representative(2023), gmirsky@ben.edu, Mathematical and Computational Sciences, 5700 College Rd. Lisle, IL 60532.

Lawrence D'Antonio, Northeastern Representative (2022), (201)684-7714, ldant@ramapo.edu, Computer Science Department, Ramapo College of New Jersey, Mahwah, NJ 07430.

Shereen Khoja, Northwestern Representative(2021), shereen@pacificu.edu, Computer Science, 2043 College Way, Forest Grove, OR 97116.

Mohamed Lotfy, Rocky Mountain Representative (2022), Information Systems & Technology Department, College of Engineering & Technology, Utah Valley University, Orem, UT 84058.

Tina Johnson, South Central Representative (2021), (940)397-6201, tina.johnson@mwsu.edu, Dept. of Computer Science, Midwestern State University, 3410 Taft Boulevard, Wichita Falls, TX 76308.

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 (2023), (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:

Bin Peng, Associate Editor, (816) 584-6884, bin.peng@park.edu, Park University - Department of Computer Science and Information Systems, 8700 NW River Park Drive, Parkville, MO 64152.

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

Welcome to the 2022 CCSC Southeastern Conference

Welcome to the 35th Southeastern Regional Conference of the Consortium for Computing Sciences in Colleges. The CCSC:SE Regional Board welcomes you – in person, once again! – to Greenville, SC, the home of BJU. The conference is designed to promote a productive exchange of information among college personnel concerned with computer science education in the academic environment. It is intended for faculty as well as administrators of academic computing facilities, and it is also intended to be welcoming to student participants in a variety of special activities. We hope that you will find something to challenge and engage you at the conference!

The conference program is highlighted by a variety of sessions, such as engaging guest speakers, workshops, panels, student posters, faculty posters, a nifty assignment session and several sessions of high quality refereed papers. We received 14 papers this year of which 7 were accepted to be presented at the conference and included in the proceedings – an acceptance rate of 50%.

Two exciting activities are designed specifically for students – a research contest and an undergraduate programming competition, with prizes for the top finishers in each.

We especially would like to thank the faculty, staff, and students of BJU for their help in organizing this conference, especially under the challenging circumstances caused by the ongoing pandemic. Many thanks also to the CCSC Board, the CCSC:SE Regional Board, and to a wonderful Conference Committee, led by Conference Co-Chairs Dr. Ethan McGee and Dr. Bob Knisely. Thank you all so much for your time and energy.

We also need to send our deepest appreciation to our partners, sponsors, and vendors. Please take the time to go up to them and thank them for their contributions and support for computing sciences education – CCSC National Partners: Turing's Craft, Google for Education, GitHub, National Science Foundation, Codio, zyBooks, National Center for Women and Information Technology, Teradata University Network, Mercury Learning and Information, Mercy College. Sponsoring Organizations: CCSC, ACM-SIGCSE, Upsilon Pi Epsilon.

We could not have done this without many excellent submissions from authors, many insightful comments from reviewers, and the support from our editor Baochuan Lu. Thanks to all of you for helping to create such a great program.

We hope you enjoy the conference and your visit to BJU.

Kevin Treu, CCSC:SE Regional Board Chair, Furman University
John Hunt, Program Chair, Covenant College

2022 CCSC Southeastern Conference Steering Committee

Marietta Cameron	Local Arrangements Chair
Kevin Sanft	Local Publicity Chair
Marietta Cameron	Speakers Chair
Kevin Sanft	Vendors Chair
Marietta Cameron	Local Sponsors Chair
Andy Digh	Programming Contest Co-Director
Chris Healy	Programming Contest Co-Director
Chris Healy	Student Research Contest Director
Nadeem Abdul Hamid	Nifty Assignments Co-Chair
Steven Benzel	Nifty Assignments Co-Chair

Regional Board — 2022 CCSC Southeastern Region

Marietta Cameron	2020 Site Chair
Kevin Treu	Regional Board Chair
Kevin Treu	CCSC:SE Regional Representative
John Hunt	Treasurer
John Hunt	Program Co-Chair
Chris Alvin	Program Co-Chair
Stephen Carl	Publicity Chair
Jean French	Local Registrar
Richard Chapman	2019 Site Chair
Ethan McGee	2021 Site Chair
Jim Knisely	2021 Site Chair

Reviewers — 2022 CCSC Southeastern Conference

Ali, Farha	Lander University
Alvin, Chris	Furman University
Barlowe, Scott	Western Carolina University
Dasgupta, Anurag	Valdosta State University
Digh, Andy	Mercer University
Dumas, Joe	University of Tennessee at Chattanooga
Goddard, Wayne	Clemson University
Gupta, Gireesh	Belmont Abby College
Heggen, Scott	Berea College
Heinz, Adrian	Georgia Gwinnett College
Holiday, Mark	Western Carolina University
Hunt, John	Covenant College
Hutchings, Dugald	Elon University
Lee, Gilliean	Lander University
Lewis, Adam	Athens State University
Luginbuhl, David	Samford University
McManus, John	Randolph-Macon University
Myers, Dan	Rollins College
Napier, Nannette	Georgia Gwinnett College
Necaise, Rance	Randolph-Macon College
Plank, James	University of Tennessee
Rao, M. Padmaja	Francis Marion University
Roach, Jeff	East Tennessee State University
Roberson, Cindy	Georgia Gwinnett College
Snders, Ian	University of South Africa
Summet, Valerie	Rollins College
Works, Karen	Florida State University

MOCSIDE: an Open-source and Scalable Online IDE and Auto-Grader for Introductory Programming Courses*

Max Barlow, Ibraheem Cazalas,
Chase Robinson, Jonathan Cazalas
Department of Computer Science
Florida Southern College
Lakeland, FL 33801

{mbarlow,icazalas,srobinson6}@mocs.flsouthern.edu

jcazalas@flsouthern.edu.edu

Abstract

Programming is learned through practice, with said practice in introductory programming courses often translating to a prohibitively large number of assignments, increasing the grading workload for faculty and/or teaching assistants. In short, this is unsustainable. Several publishers and a few notable companies have provided meritable solutions, although most are plagued with problems including minimal problem sets, limited customization options, high cost, and even a disconnect with the pedagogical needs within academia. This paper presents a survey of the more popular solutions currently available, followed by a presentation of our newly-developed web application, MOCSIDE: open-source and scalable online IDE and auto-grader for computer science education.

1 Introduction

Colleges and universities are facing an unprecedented demand for computing degrees, and employment in computing and information technology fields is expected to grow by more than ten percent by 2029.[3, 2] Scaling introductory programming courses to meet enrollment demand has proved challenging on the pedagogical front, and while approaches toward the teaching and learning of programming has evolved throughout the years, the old adage of practice makes perfect has never rung more true. Programming requires practice, extensive practice, ideally where feedback is both frequent and timely. Coupling this need with booming enrollment numbers results in a prohibitively large grading load for faculty and/or teaching assistants that is simply unsustainable. The consequence is that faculty are forced to limit graded practice or may simply remove it altogether, leaving students disengaged and disenfranchised.

Our proposal to meet this challenge is two-pronged. We have developed an open-source and scalable online IDE and auto-grader web application, which provides a self-contained and full-service experience for students. We also propose partitioning assignments into two groups: larger, more meaningful assignments, each possibly requiring over five hundred lines of code and taking one or two weeks to complete, and problem sets comprised of very small, scaffolded problems, drawing from previous research [1, 19], where students practice granular concepts learned each week or even each class period. These smaller, scaffolded problem sets are inherently designed with the auto-grading goal in mind and are perfectly suited for the formative feedback needs of the learners.

Auto-grading systems are not novel; in fact, they began arriving on the scene as far back as 2007 [15, 11] and have evolved from primitive solutions merely testing against provided test cases or unit tests[5] to an analysis of the actual code itself [10] and most recently even research into the feedback provided, arguing the need for detailed, formative feedback.[9, 8] Formative feedback is paramount and perhaps best described by Shute as "information communicated to the learner that is intended to modify his or her thinking or behavior to improve learning."[18] Haldeman et. al. developed an auto-grading system that provides formative feedback based on a variety of logical errors in the student's submission.[9, 8] As successful as their system is, it is quite involved and most certainly hands-on, starting with the design of a knowledge map, the the design of the assignment itself and a test suite of cases, and even a trained classifier that categorizes errors based on the outcomes of the test cases. The results, however, were impressive and showed success even with medium-sized problems, problems where meaningful formative feedback is crucial.

Much of the research in auto-grading assumes a one-size-fits-all approach, with little discussion on the size and scope of the graded assignments. Auto-graders have proven immensely valuable when used with medium-sized problems; they provide little value, however, on much larger programming assignments, where the size and content scope is too unwieldy for automatic feedback.

And what of very small problems, where the objective is quite granular? A multi-layered formative feedback classifier may be unnecessary. By partitioning assignments into two groups, the larger, more meaningful assignments and the smaller, scaffolded problem sets, we are able to leverage auto-grading on the smaller problems while still assessing broader software engineering outcomes with the larger assignments. Further, the smaller, scaffolded problems facilitate a quite directed formative feedback without the need to plan for a multitude of errors and build hint classifiers, as the problem scope is limited to finely-grained learning outcomes. While we continue to assign larger, multifaceted projects in our introductory programming courses, our work focuses on the creation of these smaller, scaffolded problem sets and an open-source online IDE and auto-grading system to support them. The value proposition is clear for faculty and students. Faculty are able to assess learning objectives at a granular level and can pinpoint areas needing further clarification. Students get the frequent and timely feedback they need, with the feedback, by design, honed in on the granular learning outcome of the problem. The caveat is that weekly assignments increase in number from four or five medium-sized problems per week to upwards of twenty or even thirty small, scaffolded problems. Even at institutions with smaller class sizes of twenty students, this could translate to six hundred graded exercises per week, dictating the use of a modern, web-based auto-grader with an integrated IDE.

The value of our developed web application is that it's free to use, open source, includes scaffolded problem sets in both Java and Python, and provides a full-service, self-contained experience for students. Whereas many auto-grading solutions instruct students to download the problem specification, solve the program within an editor or IDE on their own computer, and then revisit the auto-grader to upload their solution, our open-source web application provides a full-service, self-contained experience for students and faculty alike, reminiscent of the more popular paid solutions, Repl.it and CodingRooms. Students merely navigate to the given problem set, select a particular problem, code the solution within the provided web-based IDE, test their code, and then submit and await immediate feedback, all within the same browser tab.

Our contributions can be summarized as follows:

- We provide a survey of the more popular auto-grading solutions for introductory programming courses.
- We provide an open-source, auto-grader web application, along with scaffolded problem sets in both the Java and Python programming languages.
- We provide a technical overview of the developed online IDE and auto-grader and demonstrate the efficacy of its use for introductory programming courses.

The remainder of the paper is organized as follows. Section 2 reviews related work on the use of auto-grading for small programming assignments as well as a survey of the more popular auto-grader options currently available. Section 3 details the technical requirements of the system, including an overview of the application development process. Results are discussed in Section 4, and we give concluding remarks in Section 5.

2 Related Work

Booming enrollment numbers made automatic grading a reality, and a variety of free and paid options were developed, each providing unique ways to assess student code and provide feedback. Three of the more prominent free auto-grading systems will be discussed below: Web-Cat, OK, and Autolab. Web-Cat is a modular grading system initially developed in 2008 as a means to simply submit assignments.[6] Using plugins, Web-Cat provides auto-grading functionality and has also been used in research to explore the gamification of feedback and the value of providing motivational hints. OK was developed at UC Berkeley and supports auto-grading via output comparison as well as human-authored code review and detailed analytics.[5] The collection of in-progress work, or snapshots, during the assignment is unique to this system and provides useful data for instructors to gauge assignment difficulty and common errors. AutoLab was developed by Carnegie Mellon and is used by several top universities, with University of Washington and Cornell University among them.[5, 14] By default, Autolab does not provide feedback, with students simply receiving a pass/fail response and left to cope with their incorrect answers and/or underlying misconceptions.[12] Addressing these concerns, Haldeman et. al. augmented both Web-Cat and Autolab by developing an auto-grading system that provides formative feedback based on a variety of logical errors in the student's submission.[9, 8]

On the paid front, there are many contenders including zyLabs, Repl.it, Codio, CodeHS, Coding Rooms, and more. As with their free counterparts, each has their merit. zyLabs and zyBooks are well known within academia and computer science, offering interactive, online curriculum content as well as auto-grading solutions. zyLabs was first introduced in 2016 and has grown significantly in five years, a testament to auto-grading needs of modern classrooms.[7] Similar to other publishers, zyLabs offers automatically-graded problem sets at the end of each module, with formative feedback offered when submissions result in a failure. While the provided problems are serviceable, most require between twenty to forty lines of code and map to multiple learning outcomes. Limiting zyLabs widespread adoption is the lack of smaller, scaffolded problem sets coupled with the inability to add custom problems.

Repl.it started out as an online IDE, allowing users to write and run code in over fifty languages. Their academic product, Classroom, was considered a game changer for many, with students and faculty praising its ease of use and collaborative coding environment.[17] Creating problems, test cases, and setting up the auto-grader was complete in a matter of clicks, and their collaborative coding functionality, also known as Repl.it "Multiplayer", was icing on the cake. Unfortunately, during the Fall 2020 semester, their Classroom product began experience major technical problems. What initially were down times of a few hours then became a few days, leaving students frustrated and faculty without answers. Unable to address these issues, Repl.it announced that it would be discontinued, while encouraging faculty to test their new Teams platform, which even now one year later, does not provide the ease-of-use and streamlined functionality of their previous Classroom product, although it does include several overall improvements. Truth be told, we owe gratitude to Repl.it, as this research was a byproduct of that Fall 2020 and the unknowns it presented.

Coding Rooms is one of the newer paid offerings and has grown significantly since the start of COVID. Like Repl.it's previous Classroom product, Coding Rooms offers a streamlined experience for faculty in the creation of courses, modules, problems, and the test cases needed for auto-grading. Anecdotally, Coding Rooms appears to constantly improve due to being very receptive of feedback and quick to implement suggested changes. Other notable paid solutions include Codio and CodeHS, both offering similar functionality to Coding Rooms. The aforementioned paid solutions provide a modern take on auto-grading, extensive customization options, and formative feedback, all within an easy-to-use, self-contained ecosystem. Their collective downside is simply cost. Further, the Fall 2020 experience detailed above was motivation enough to research options and look to develop an in-house and open-source solution.

3 Technical Requirements & Overview of System

3.1 Analysis of Requisite Features

Modeling our web application after Repl.it and CodingRooms certainly streamlined this stage of requirements analysis. While we haven't the luxury of the millions in venture capital [20], our application does contain a similar feature set to these paid services. Faculty needs include course creation, lab/module creation within those courses, the ability to create problems within the individual labs, and the ability to administer the course, including adding and removing students, editing grades, and more. Student needs include the ability to register for course, the ability to navigate through labs and problems within said labs, the ability to solve a given problem within the provided online IDE, the ability to test their solution against a provided set of pretest cases, and the ability to submit their solution.

3.2 Server Setup

MocsIDE runs on an unmanaged VPS with a fairly simple stack of technologies. Our operating system of choice is Ubuntu 20.04, which comes pre-installed with Apache. Setup consists of a few broad steps: Linux setup (users, firewall, ssh keys), Environment setup (software install, project clone, test compilation), and then publishing/securing our app with Certbot and Apache. Choices made early on in the process of Linux setup greatly affect our security later on, so we take care to quickly setup UFW (Ubuntu Default firewall) and user ssh keys to prevent unauthorized traffic. Environment setup can be an incredibly labor intensive step if we need to start a fresh install, so we use tools like Systemback and SQL dumps to restore more swiftly after a "fatal" error or mistake.

3.3 Reactive Web Application via Vue & Laravel

The requirements of dynamic rendering and reactivity dictated the use of a front-end framework, facilitating a more engaging, interactive platform for users. We opted to use Vue as our front-end framework, as it presents a balanced coding experience, has the smallest package size, is measurably faster than React and Angular, and has a growing community of developers creating custom packages and modules which can be used to support our unique requirements. Thus, our front-end was coded in HTML, CSS, and JavaScript, all within the Vue 3 framework, and we further included a CSS framework, Bootstrap, to aid in reactive design. Laravel was chosen as our back-end API, as it works seamlessly with Vue and is packed with helpful features and documentation. Laravel made implementing authentication and authorization a simple task and allowed us to create our own front-end interpretations of its functionality. Using the framework, we designed a multi-functional API that allowed our application to work with stored data in more dynamic and intuitive ways on top of regular CRUD functionality, as well as safely handling the grading process, among others. As Laravel is a PHP framework, it does come with the drawbacks of using PHP in a fashion unlike the multi-page PHP web pages of the past - namely a lack of native support interacting with the Docker SDK (needed for auto-grading), which was a non-issue until later in the development process as we strayed from pure Laravel.

3.4 Docker Containers

Arguably the greatest hurdle faced was in how to compile code, run it against a set of test cases, compare the outputs to the expected outputs, and then present the results back to the user. This was well beyond the scope of typical front and back end development and required extensive research. Individual virtual machines, or jailed boxes, were appealing but are known to be resource heavy. [16] We were also careful to not sacrifice the security of our server, as we were very wary of running malicious code. The solution to this problem was Docker. Docker, a very popular containerization program, allows for the creation of virtual environments for each individual user and is incredibly customizable with the ability to adjust the size and complexity of the docker's image. We created our own custom image, which enabled us to keep the size of the container small to ensure we could have multiple students running their code simultaneously. Our custom Docker has several languages installed and handles running and grading the code and then sends the results back to both the front and back end to be displayed and saved.

4 Results & Discussion

All components of the web application were completed and rigorously tested by faculty and students during the Spring 2021 and Summer 2021 semesters. Having been modeled after paid solutions used during the past three years, students appreciated having a similar user experience with course navigation, an embedded online IDE, and submission and feedback systems. And per-

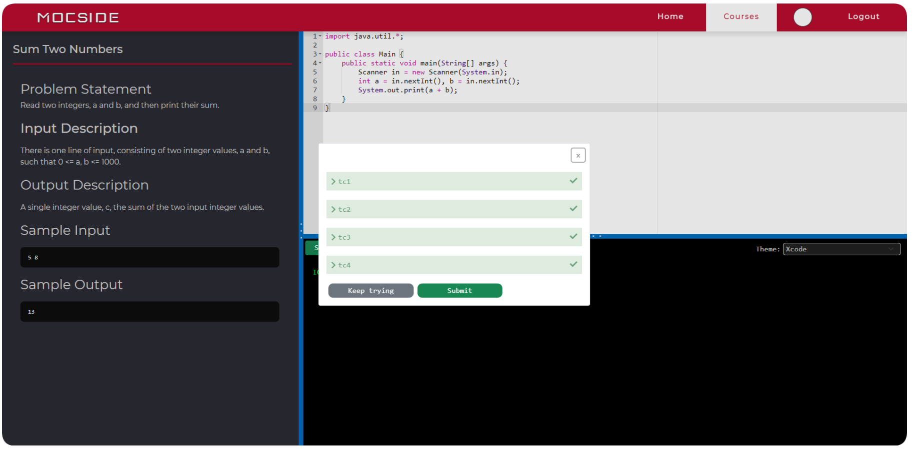

Figure 1: A student submission passing all test cases.

haps most appreciated was having all of that at no cost to them. Figure 1 shows a screenshot of a student submission having passed all test cases. The web application is open source and can be found here, `https://github.com/jcazalas/mocsidev1`. The included `readme` file provides the necessary documentation for server and database setup and details how to change the logo and color scheme if desired. The repository currently includes scaffolded problem sets for an introductory course taught using either Python or Java as well as for an object-oriented course using Java. Faculty can copy those sample courses, can modify their content, can create courses, modules, and problems, and can invite students to enroll in a course by providing a system-generated invitation link. We plan to improve the application by including code collaboration, gamification via live, anonymous scoreboards[13, 4], and an intelligent hint system for larger-sized problems. As improvements are implemented, new versions of the web application will be pushed to the above-linked git repository.

5 Conclusion

In this paper, we presented MOCSIDE, our open-source and scalable online IDE and auto-grader for introductory programming courses. Our web application provides students a full-service and self-contained learning experience, where they can navigate to the given problem set, select a particular problem, code the solution within the provided web-based IDE, test their code, and then submit and await immediate feedback, all within the same browser tab. Using the provided scaffolded programming assignments, faculty are able to assess learning objectives at a granular level, with the auto-grader providing frequent and timely formative feedback. Results indicate a positive user experience from students and instructors alike, with ease of use, cost savings, and being open source highlighted as key features.

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, page 585–591, New York, NY, USA, 2019. Association for Computing Machinery.

[2] Bureau of Labor Statistics. Computer and information technology occupations, n.d. `https://www.bls.gov/ooh/computer-and-information-technology/home.htm`, Last accessed on 2021-8-2.

[3] Computing Research Association. The phenomenal growth of cs majors since 2006, n.d. `https://cra.org/data/generation-cs/phenomenal-growth-cs-majors-since-2006/`, Last accessed on 2021-8-2.

[4] Rafael G. de Pontes, Dalton D. S. Guerrero, and Jorge C. A. de Figueiredo. Analyzing gamification impact on a mastery learning introductory programming course. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, page 400–406, New York, NY, USA, 2019. Association for Computing Machinery.

[5] John DeNero, Sumukh Sridhara, Manuel Pérez-Quiñones, Aatish Nayak, and Ben Leong. Beyond autograding: Advances in student feedback platforms. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '17, page 651–652, New York, NY, USA, 2017. Association for Computing Machinery.

[6] Stephen H. Edwards and Manuel A. Perez-Quinones. Web-cat: Automatically grading programming assignments. In *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE '08, page 328, New York, NY, USA, 2008. Association for Computing Machinery.

[7] Chelsea Gordon, Roman Lysecky, and Frank Vahid. The rise of program auto-grading in introductory cs courses: A case study of zylabs. In *2021 ASEE Virtual Annual Conference Content Access*, Virtual Conference, July 2021. ASEE Conferences. `https://peer.asee.org/37887`.

[8] Georgiana Haldeman, Monica Babeş-Vroman, Andrew Tjang, and Thu D. Nguyen. Csf: Formative feedback in autograding. *ACM Trans. Comput. Educ.*, 21(3), May 2021.

[9] Georgiana Haldeman, Andrew Tjang, Monica Babeş-Vroman, Stephen Bartos, Jay Shah, Danielle Yucht, and Thu D. Nguyen. Providing meaningful feedback for autograding of programming assignments. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, page 278–283, New York, NY, USA, 2018. Association for Computing Machinery.

[10] Emlyn Hegarty-Kelly and Dr Aidan Mooney. Analysis of an automatic grading system within first year computer science programming modules. In *Computing Education Practice 2021*, CEP '21, page 17–20, New York, NY, USA, 2021. Association for Computing Machinery.

[11] Michael T. Helmick. Interface-based programming assignments and automatic grading of java programs. *SIGCSE Bull.*, 39(3):63–67, June 2007.

[12] Hieke Keuning, Johan Jeuring, and Bastiaan Heeren. Towards a systematic review of automated feedback generation for programming exercises. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '16, page 41–46, New York, NY, USA, 2016. Association for Computing Machinery.

[13] Celine Latulipe, N. Bruce Long, and Carlos E. Seminario. Structuring flipped classes with lightweight teams and gamification. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, SIGCSE '15, page 392–397, New York, NY, USA, 2015. Association for Computing Machinery.

[14] Dejan Milojicic. Autograding in the cloud: Interview with david o'hallaron. *IEEE Internet Computing*, 15(1):9–12, January 2011.

[15] Pete Nordquist. Providing accurate and timely feedback by automatically grading student programming labs. 23(2):16–23, December 2007.

[16] Matthew Peveler, Evan Maicus, and Barbara Cutler. Comparing jailed sandboxes vs containers within an autograding system. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, page 139–145, New York, NY, USA, 2019. Association for Computing Machinery.

[17] Md Mahmudur Rahman, Monir Sharker, and Roshan Paudel. Impact of infusing interactive and collaborative learning in teaching introductory programming in a dynamic class. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, page 1315, New York, NY, USA, 2020. Association for Computing Machinery.

[18] Valerie J. Shute. Focus on formative feedback. *Review of Educational Research*, 78(1):153–189, 2008.

[19] Angela A. Siegel, Tavis Bragg, Alexander Brodsky, and Eric Poitras. Exploring the use of auto-grading systems to improve the efficacy of feedback through small, scaffolded programming assignments. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 2*, ITiCSE '21, page 607–608, New York, NY, USA, 2021. Association for Computing Machinery.

[20] VentureBeat. Replit raises $20 million for collaborative browser-based coding, n.d. `https://venturebeat.com/2021/02/18/replit-raises-20-million-for-collaborative-browser-based-coding/`, Last accessed on 2021-8-2.

Monte Carlo Tree Search Applied to the Game Abalone*

Joshua Hutson, George Lamb, Chris Alvin
Department of Computer Science
Furman University
Greenville, SC 29613
{joshua.hutson, george.lamb, calvin†}@furman.edu

Abstract

We consider the perfect knowledge, abstract strategy game Abalone as a means of investigating AI search. In particular, we apply a Monte Carlo Tree Search technique to Abalone and assess its efficacy.

1 Introduction

In an artificial intelligence course, we sometimes find it difficult to define a domain and corresponding simple codebase for students to implement novel techniques. In this paper, we consider the abstract strategy game Abalone [3] as a means of exploring tree search. Some modern board games require fifty or more pages of rules, complex boards, and many pieces. By contrast, Abalone is a perfect knowledge game, in the vein of checkers, in which the rules can be described on an index card with a simple hexagonal board and two colors of pieces. Abalone is a game that has sold millions of copies worldwide [3], but is relatively unknown in the United States. Pedagogically it is therefore ideal to explore for a class interested in games, search, or strategy development in a competitive bot.

It is common for search tree techniques such as Minimax to be applied to games such as Chess, Checkers, and Go. While such search tree algorithms have been successful, even greater success has been achieved when combined with a Monte Carlo

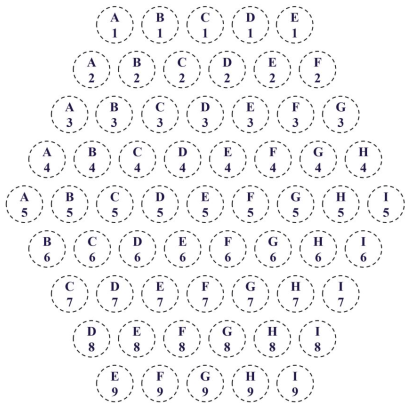

Figure 1: Coordinatized Abalone board.

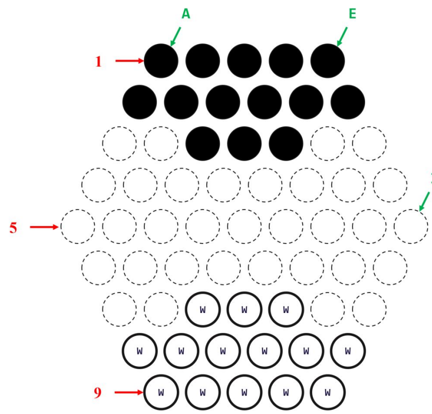

Figure 2: Starting state of Abalone.

technique. In this paper we apply the Monte Carlo Tree Search (MCTS) [4] technique to Abalone with moderate success. Our work is the result of two undergraduate summer research students and serves as a precursor to inclusion in an undergraduate AI course. In Section 2 we describe the game of Abalone. We then provide an introduction to the MCTS applied to Abalone in Section 3. We then empirically report on several MCTS players in Section 4.

2 Game Rules

Gameplay. The Abalone board is a hexagon consisting of 61 individual positions as shown in Figure 2. Each player begins with their own set of 14 black or white marbles positioned symmetrically at opposite ends of the board; black marbles are filled in, white marbles contain a `W`, and empty spaces are dashed. For clarity, we have labeled the horizontal rows from 1 through 9 and the diagonal ‘columns’ (diagonals, hereafter) from **A** through **I** as shown in Figure 1. This labeling will allow us to speak about particular cells. For example, at the beginning of the game black marbles populate spaces **A1** through **E1**, **A2** through **F2**, and **C3** through **E3**.

Play always begins with black having the first move. The players alternate turns until a player has pushed 6 opposing marbles off the board. Each turn consists of a player making a single *action*. We describe each of the five possible actions noting one definition. We define a **Line** to be a sequence of 2 or 3 marbles of the same color positioned linearly (i.e., all marbles fall on the same horizontal or diagonal); Figure 2 contains a line of 3 white marbles from **E7** through **G7**. According to this definition any two adjacent marbles of the same color form a Line. Therefore, every

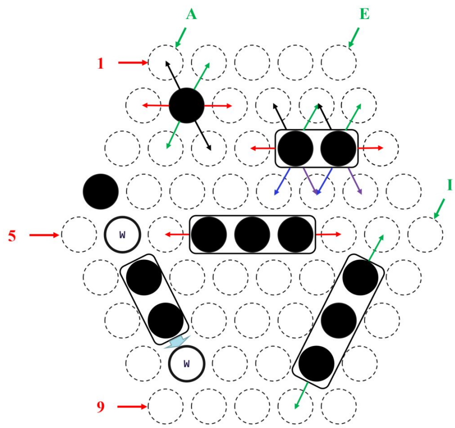

Figure 3: Configurations for possible player actions (not a realistic board state).

Line of length 3 can be broken down into two Lines consisting of 2 marbles. We limit Lines to length 3 as there is no greater utility for longer lines in the game as we will see below with *actions*.

Actions. During a turn, each player must take a single action (i.e., make a move). Although our nomenclature for player moves may differ from some Abalone game instruction nomenclature, our choices are for clarity. Independent of our definitions, we always adhere to proper game rule interpretations.

Simple. A player may move a single marble to any adjacent, valid, empty space. For example, in Figure 3 the black marble in **B2** may be moved to any of the six adjacent spaces (e.g., **A1**, **A2**, etc.).

Line shift. All marbles in a Line may be shifted one position linearly as long as the destination space is valid and empty. In Figure 3, the Line of length 2 in row 3 can be shifted from positions **E3** and **F3** either left or right as indicated by the red arrows. A left Line shift results in the 2 black marbles being moved into spaces **D3** and **E3**; a right Line shift results in the 2 black marbles being moved into spaces **F3** and **G3**. Shifts can apply to any Line; the Line of 3 black marbles in 'column' **H** may shift into positions **H7** through **H9** as indicated with an downward green arrow; similarly upward into spaces **H5** through **H7**.

Sidestep. Similar to a simple move, an entire Line may move as a unit in any 'open' direction. A Line may be moved laterally ('broadside') as a unit assuming all destination spaces are valid and empty. In order to avoid clutter in Figure 3, we focus our example on the horizontal Line of length 2 in row 3. This line may move as a unit in four directions: northeast (green), northwest (black), southwest (blue), or southeast (purple). For clarity, we differentiate Line shifts from side step moves since Line shifts only require one destination space be open and valid while side step moves require we verify all marbles can be moved to valid and empty spaces.

Push move. The goal of Abalone is to 'eject' six of your opponent's marbles off of the board by pushing. Pushing is accomplished by using numerical superiority ('Sumito') to force an opponent's marble(s) to move. In particular, a Line of length 3 can push a linearly aligned marble of the opponent (3-Push-1 Sumito) or a linearly aligned Line of an opponent of length 2 (3-Push-2 Sumito). Similarly, a line of length 2 can push a single opponent marble (2-Push-1 Sumito). In Figure 3, there is a 2-Push-1 Sumito of the black Line including **C6** and **D7** onto opponent's white marble at **E8** resulting in the black Line at **D7** and **E8** and the white marble onto **F9**. We can only perform a push when the pushing destination space is empty or is off the board. So the black Line of **C6** and **D7** cannot push the white marble at **B5** since **A4** is occupied.

3 Monte Carlo Tree Search

A Monte Carlo method refers to a class of algorithms that use random sampling to identify patterns. In our case, we applied the classic MCTS as a competitor in Abalone. Although many resources exist, we describe the MCTS algorithm to place our experimental parameters into context.

Algorithm 1 Monte Carlo Tree Search Algorithm

1: **function** MCTS($root$)
2:     $root$: Root of Tree
3:     EXPAND($root$)
4:     **while** RESOURCESREMAIN( ) **do**   ▷ Time or Iterations
5:         $\ell \leftarrow$ TRAVERSE($root$)   ▷ Identify a leaf node via best UCB score
6:         **if** ROLLEDOUT(ℓ) **then**   ▷ Previously rolled out node.
7:             EXPAND(ℓ)
8:             $\ell \leftarrow$ RANDOMCHILD(ℓ)
9:         $\mathsf{SF}_\ell \leftarrow$ ROLLOUT(ℓ)   ▷ Monte Carlo game simulation from ℓ
10:         BACKPROP(ℓ, SF_ℓ)   ▷ Back-propagate result from ℓ up to $root$
11:     **return** BESTCHILD($root$)

Algorithm. As described in Algorithm 1, MCTS is a technique defined by four core sub-routines: EXPAND, TRAVERSE, ROLLOUT, and backpropagation (BACKPROP). We consider each of these operations in turn.

Expansion. Given a node n in a search tree T, *expansion* refers to adding all children of n to T. With Abalone, each node in the tree maintains a game board layout. A move m (as described in Section 2) transitions a parent node $p \in T$ to a child node cT: $p \xrightarrow{m} c$. Expansion can be precarious with Abalone since there are 44 possible moves for the black player in Figure 2 as implemented on Line 3 of Algorithm 1.

Traversal. Each iteration of the MCTS algorithm must traverse the existing tree to identify a leaf node to either *expand* or *rollout*. Each node n in a tree T maintains two attributes: (1) a cumulative score (S) updated when n is included in a simulated gameplay path and (2) a count (N) of number of times n is visited. The decision about which child c of n to visit next is based on the largest value of the Upper Bound Confidence Bound 1 applied to Trees [4] (UCB1) for a parent and each of its children:

$$\mathsf{UCB1}\,(p,c) = \frac{c_S}{c_N} + E\sqrt{\frac{\ln p_N}{c_N}} \tag{1}$$

where E is an exploration parameter. Equation 1 consists of two expressions. The expression $\frac{c_S}{c_N}$ refers to the level of success previously found by traversing the particular edge between parent and child; hence, the colloquial name 'exploitation' expression. The second expression (bandit expression) $E\sqrt{\frac{\ln p_N}{c_N}}$ values new nodes to explore; UCB1 returns ∞ for a child that has not been explored (i.e., $c_N = 0$). We can tune the importance of exploration in a game by selecting E to be small in order to avoid a breadth-first exploration that may arise in games with many options for moves.

Rollout. Given a leaf node ℓ from traversal (Line 5 in Algorithm 1), we expand it if needed (Line 6 through Line 8), and simulate a game between two players who choose their moves randomly (Section 4 describes how our approach to rollouts is semi-random). When a rollout has reached its conclusion either by the game finishing or a cutoff parameter being reached, we return a score $\mathsf{SF}_\ell \in [-1, 1]$ indicating success or failure attributed to a game starting with ℓ for the root player's turn.

Backpropagation. For each node n in the path from ℓ to the root tree T on Line 10, we update n_S (total score of n), additively with $\mathsf{SF}_\ell(Board, P)$ where P is the current player, and increment the number of times n was visited (i.e., $\text{++}n_N$).

Iteration and Return. We continue to iterate on Line 4 through traversal, rollout, and backpropagation until we exhaust our search parameter(s): time, space, or some other cutoff (e.g. maximum number of rollouts, etc.). Last, on Line 11, we choose move m from $root \xrightarrow{m} c$ where c_S is the maximum among all children of $root$.

4 Experimental Analyses

The goal of our experiments is to explore the utility of the MCTS algorithm for Abalone.

Bot Players. There are several parameters that can be tuned with the MCTS algorithm. Our experiments attempt to demonstrate the goodness of a MCTS player compared to other naive players. As shown in Table 1, we had 7 different players compete against one another. We defined three Player Types that choose moves at random: (1) a player that randomly chooses one move out of all possible moves, (2) a player that prioritizes a random *Push* move (if one is not available it randomly chooses from

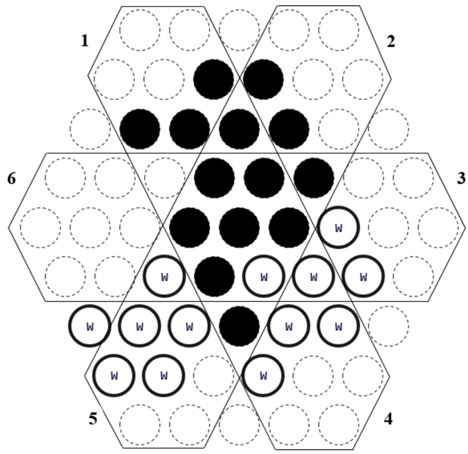

Figure 4: A sample penetration-based mid-game board state with regions indicated for dominance-based board-state scoring.

one of the other moves), and (3) a tiered player that chooses a random move from a category of moves in the order *Push*, *Line Shift*, *Sidestep*, and *Simple*.

Our final four players choose moves based on MCTS. Player Type 4 implements the MCTS loop condition (Line 4 of Algorithm 1) with a fixed number of iterations. Player Type 5 iterates a fixed amount of time.

For our remaining players, we recognized that early-game board states and mid-game board states are quite different. We define a *formation* to be a connected subgraph of marbles of a single color; for example, Figure 2 has two distinct formations. We also define *adjacent formations* as two formations, W and B, of opposing color marbles in which at least one marble of W is adjacent to one marble of B. Although it is difficult to identify when a mid-game board state has occurred, we relied on our experience with the game to define two criteria. We define *mid-game* as a board state in which (a) one marble has been pushed off the board or (b) one formation of marbles, B, penetrates through an opponent's marbles resulting in two formations, W_1 and W_2, both adjacent to B (see Figure 4 for an example). Thus, our last two hybrid players switch modes mid-game. Player Type 6 starts with an iteration-controlled MCTS and transitions to time-controlled; Player Type 7 took the opposite order of loop control (time-based MCTS followed by iteration).

MCTS Parameters. Players following a MCTS must define the exploration constant: E in Equation 1. Since Abalone defines an exponential search space, it is easy to explore vast shallow elements of the search tree. In order to favor exploitation over exploration we empirically identified $E = 0.05$ to provide a balance between depth (roughly 8 to 10 moves deep) and breadth (approximate maximum width of 15000).

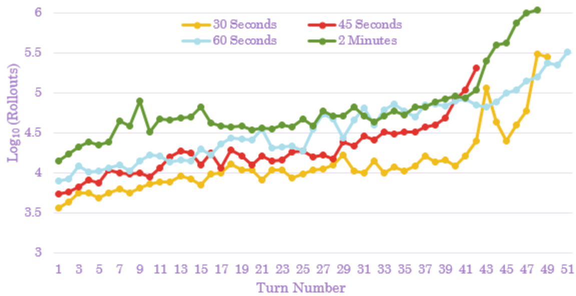

Figure 5: Scaled number of rollouts per turn for time-based iteration of MCTS.

For the rollout operation of MCTS we must identify a reasonable upper bound for the number of turns in a rollout-played game as well as what random Player (either Type 1, 2, or 3) will play rollout games for our MCTS Players (Type 4, 5, 6, and 7). Empirically, we observed strictly random rollout games completed after 750 moves; hence we set the maximum number of moves for rollout to be 800. Although this is excessive compared to a standard game which last about 100 turns, we wanted to ensure rollout games completed. To expedite the rollout process, instead of using strictly random players (Player Type 1), we instead used a random push mover (Player Type 2) for all MCTS Players. Our goal in this choice was to maintain as much randomness as possible in the spirit of a Monte Carlo method, but to expedite execution as well; 10000 iterations of MCTS with rollout using Player Type 2 was almost 8 times faster than using Player Type 1.

As described in Section 3 with Algorithm 1, it is common for iteration to be based on time. We executed Player Type 4 using 30, 45, 60, and 120 seconds per move with the number of resulting rollouts shown in Figure 5. We observe in Figure 5 that there was a significant difference between the number of rollouts in the early- and mid-game. Ultimately, we want as many rollouts as possible and thus the 30 and 45 second players were not able to make a strong, informed move. We observed that a 60-second turn gives satisfactory results while the 120-second did not make a notable, significant success in move selection (and win ratios) compared to 60-seconds. Hence, Player Types 5, 6, and 7 use 60-second turn bounds.

For Player Types 4, 6, and 7 using iteration-based MCTS we must identify a reasonable upper bound number of iterations. Comparing 1000, 5000, and 10000 iterations, we observed a reasonable balance between tree depth and execution time using 5000 iterations noting the Player took less than a minute toward the beginning of the game and less than five seconds by the end of the game.

Board-State Scoring. When rollout game play is stopped or it completes, we define $\mathsf{SF} : Board \times P \to [-1, 1]$ where -1 indicates a loss for a player P and 1 indicates a win. If the game is not complete, we weight two different methods for scoring a board equally, one method based on number of pieces and one based on piece placement so that players would seek avenues to win while attempting to maintain strong board positioning.

Piece-Based Scoring. It is a distinct advantage to have more marbles than your opponent. We defined an exponential piece scoring function (ps) emphasizing the difference in the number of remaining marbles ($|M|$) between players:

$$\mathsf{ps}\,(Board) \;=\; \mathsf{ms}\,(B) - \mathsf{ms}\,(W)\;;\; \mathsf{ms}\,(M) \;=\; \begin{cases} 2^{(14-|M|)/7} & 10 \leq |M| \leq 14 \\ 2^{(14-|M|+1)/7} & |M| = 9 \end{cases}.$$

We note the non-linearity of this function as to award higher scores for a player being closer to a win. In particular, we valued having 5 opponent marbles pushed off (9 remaining on board) to indicate a strategic advantage (allowing the player to be more aggressive); this accounts for dividing by 7 instead of 6 in `ms` since the fifth marble pushed off can be viewed as twice as important as all others. As an example, consider the case where a player has pushed off 5 opponent marbles compared to their opponent pushing off only 2 of theirs, we would calculate $\mathsf{ps}\,(Board) = 2^{6/7} - 2^{2/7} \approx 0.592$.

Dominance-Based Scoring. Our second scoring method considers board dominance of a player and it consists of two equally weighted parts. The first part splits the board into 7 hexagonal regions as shown in Figure 4 (region 7 is in the center of the board) and their corresponding center spaces in **B2**, **E2**, etc.; in this configuration, there are 6 interior and 6 exterior spaces on the board that are not within a region. A player earns a score of 1 for each region except for the center hexagon which receives a score of 2. Dominance of a region r is determined by the color marble that is situated on the center space of r. If no marble occupies the center, r is considered to be dominated by the player with the majority of marbles in r. In Figure 4, white dominates area 5 since it occupies the center and black dominates regions 1 and 2 since black has the majority of marbles. In Figure 4 white dominates 3 regions and black dominates 3, including the center. We would compute a dominance score for the black player as $\mathsf{dom}\,(B) = (4-3)/8 = 0.125$ where division normalizes the value to $[-1, 1]$.

The second part of dominance-based scoring considers *cohesion* of marble formations; tighter groupings are more valuable to prevent opponent Sumitos. If a player's marbles consist of multiple formations, the cohesion score is 0. For a player with a single formation, we compute the *spread* of the formation as the straight-line distance between the farthest nodes in a formation subgraph. In Figure 4, black consists of a single formation so we compute the distance between **F7** and **C2** (using Law of Cosines) as $\sqrt{(7-2)^2 + (6-3)^2 - 2\cdot(7-2)\cdot(6-3)\cos 60^\circ} = \sqrt{19}$; normalized to the maximum distance on the board (9), we have $\sqrt{19}/9 \approx 0.484$. Since we want formations with less spread, we take the complement: $1 - 0.484 = 0.516$.

As an example of the overall scoring computation, consider the board state in Figure 4 evaluating for black. For piece count we have $\mathsf{ps}\,(Board) = 2^{1/7} - 2^0 \approx 0.104$. Hence, $\mathsf{SF}(Board, B) = \frac{1}{2}\cdot\left[0.104 + \frac{1}{2}\cdot(0.125 + 0.516)\right] \approx 0.212$.

Analyses. We executed each of our Player Types against one another with win

Player 1 \ Player 2	Random Player All Move	Random Player Push Move	Random Player Tiered Move	Monte Carlo Tree Search	Time Based Tree Search	Hybrid Player Time First	Hybrid Player Move First
Random Player All Move	$\frac{819{,}472}{2{,}000{,}000} = .410$	$\frac{500}{2{,}000{,}000} = .00025$	$\frac{600}{2{,}000{,}000} = .0003$	$\frac{0}{50} = 0.0$	$\frac{0}{50} = 0.0$	$\frac{0}{50} = 0.0$	$\frac{0}{50} = 0.0$
Random Player Push Move	$\frac{1{,}999{,}980}{2{,}000{,}000} = .999$	$\frac{996{,}441}{2{,}000{,}000} = .498$	$\frac{848{,}064}{2{,}000{,}000} = .424$	$\frac{0}{50} = 0.0$	$\frac{0}{50} = 0.0$	$\frac{0}{50} = 0.0$	$\frac{0}{50} = 0.0$
Random Player Tiered Move	$\frac{1{,}999{,}990}{2{,}000{,}000} = .999$	$\frac{1{,}144{,}301}{2{,}000{,}000} = .572$	$\frac{996{,}631}{2{,}000{,}000} = .498$	$\frac{0}{50} = 0.0$	$\frac{0}{50} = 0.0$	$\frac{0}{50} = 0.0$	$\frac{0}{50} = 0.0$
Monte Carlo Tree Search	$\frac{50}{50} = 1.0$	$\frac{50}{50} = 1.0$	$\frac{50}{50} = 1.0$	$\frac{25}{50} = .5$	$\frac{23}{50} = .46$	$\frac{20}{50} = .4$	$\frac{25}{50} = .5$
Time Based Tree Search	$\frac{50}{50} = 1.0$	$\frac{50}{50} = 1.0$	$\frac{50}{50} = 1.0$	$\frac{21}{50} = .42$	$\frac{17}{50} = .34$	$\frac{19}{50} = .38$	$\frac{17}{50} = .34$
Hybrid Player Time First	$\frac{50}{50} = 1.0$	$\frac{50}{50} = 1.0$	$\frac{50}{50} = 1.0$	$\frac{23}{50} = .46$	$\frac{18}{50} = .36$	$\frac{21}{50} = .42$	$\frac{21}{50} = .42$
Hybrid Player Move First	$\frac{50}{50} = 1.0$	$\frac{50}{50} = 1.0$	$\frac{50}{50} = 1.0$	$\frac{22}{50} = .44$	$\frac{22}{50} = .44$	$\frac{20}{50} = .4$	$\frac{23}{50} = .46$

Table 1: Empirical win rates for first player (black) $\times$ second player (white) games.

rates reported in Table 1. As a measure of confidence in our implementation, we observe that the main diagonal of the upper left corner of the table indicates a fair, 50% win rate specifically over the large sample size of 2000000 games. Of note, we believe the lower win ratio for the Player Type 1 is a result of not finishing games in the number of moves allowed. We also observe that all MCTS Player Types 4, 5, 6, and 7 beat all random Players (Types 1, 2, and 3) 100% of the time. The remaining games involving MCTS Players seem to represent a relatively small sample size. However, a typical game between two MCTS Players such as two Player Type 5 can take approximately 2 hours to complete; hence, Table 1 represents weeks of computing time. Among the MCTS Player Types, there is no clear evidence of a 'best' player as all had win rates below 50%. As a result, while our sample size is small, these data suggest that the second (white) player has an advantage over the first (black) player.

5 Related Works

Chorus [1] considers different agents to play Abalone. While they conclude that the Alpha-Beta algorithm is superior to MCTS, their MCTS scoring approach is naïve in that they only use the number of pushed marbles as a method of scoring in a case where one player does not win outright. Whereas our scoring technique relies on state-based scoring of the board as much as it does counting pushed marbles by giving weight to the number of marbles pushed off thus reflecting an advantage of having more marbles on the board in the late game. Our approach does not consider an alpha-beta approach since our proposed goal is to investigate a more recent technique that has experienced some success with other games (e.g., Go using AlphaGo [2], Chess using an AlphaGo adaptation [6] in Leela Chess Zero, and more).

Ozcan and Hulagu [7] attempt to heuristically play Abalone through knowledge of the value of the center. Their Abalone player attempts to occupy the center of the board first to put their opponent in a disadvantageous position. We attempted something with our scoring method, but found that incentivizing the center of the board created a player that was afraid to push marbles off the board. Hence, we incentivized cohesion over acquiring and maintaining the center.

Our lack of a conclusion as to the best MCTS Player is similar to the findings of Kozelek [5] when applying MCTS to Arimaa, a game that is similar, yet more complex than Chess. He stated the MCTS "gave very poor performance" and suggested a static evaluation of the board at given points in time.

6 Conclusions

We considered the game Abalone as a venue for agent play. We presented the MCTS algorithm along with many different parameters for tuning the algorithm with respect to Abalone. We were not able to conclude a best MCTS Player for Abalone, but we were able to intuit that the second player has the advantage. There are many potential avenues for further consideration of MCTS with Abalone as an academic exercise; however, the search space is prohibitive.

References

[1] Pascal Chorus. Implementing a computer player for abalone using alpha-beta and monte-carlo search. Master's thesis, Maastricht University, June 2009.

[2] Michael C. Fu. Alphago and monte carlo tree search: The simulation optimization perspective. In *Winter Simulation Conference (WSC)*. IEEE, 2016.

[3] University Games. The game: Abalone, July 2021. https://www.universitygames.com/abalone.

[4] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006*. Springer Berlin Heidelberg, 2006.

[5] Tomáš Kozelek. Methods of mcts and the game arimaa. Master's thesis, Charles University in Prague, 2009.

[6] Gary Linscott. Leela chess zero, July 2021. https://lczero.org/.

[7] Ender Ozcan and Berk Hulagu. A simple intelligent agent for playing abalone game. Turkish Symposium on Artificial Intelligence and Neural Networks, 2004.

Introducing Algorithmic Bias Considerations in an Introductory CS Course*

Marguerite Doman and Chlotia Garrison
Computer Science and Quantitiatve Methods
Winthrop University
Rock Hill, SC 29788
{domanm,garrisonc@winthrop.edu

Abstract

With a growing focus on diversity, equity, and inclusion, the use of applications to make decisions is intended to help remove human bias from the decision-making process. However, the increasing number of examples of algorithm bias reveals the need to ensure the algorithms used in applications are not biased. A first step to creating unbiased algorithms is being aware that algorithms can contain bias. The goal of this research is to review the results of including a simple and non-threatening assignment into the requirements of an introductory CS1 class that helps make students and future programmers aware of algorithm bias.

1 Introduction

The ubiquitous use of computing applications to assist in decision-making and other cultural practices has made us more dependent on the credibility of the algorithms used in building those applications. That dependency and reliance on these applications may result in an implicit trust in the algorithm's outcomes. The "garbage-in, garbage-out" acronym is often changed for end-users to "garbage-in, gospel-out" [20]. Due to that trust and the prevalence of use, we must demand more rigor and less bias from computer applications.

The 'black box' nature of algorithms makes it difficult to assert whether the outcomes are correct, erroneous, or even biased. That is, the algorithm may slant to some bias. We define bias in algorithms as "computer systems that systematically and unfairly discriminate against certain individuals or groups of individuals in favor of others. A system discriminates unfairly if it denies an opportunity or a good or if it assigns an undesirable outcome to an individual or group of individuals on grounds that are unreasonable or inappropriate" [8]. To address the potential harm of biased algorithms in intelligent systems, IEEE launched an initiative to ensure designers and developers include the ethical implementation of autonomous and intelligent technologies [2]. ACM Public Policy Council stated Algorithmic Transparency and Accountability [16] which seeks to eliminate bias through a set of principles to be considered throughout the software development lifecycle, including deployment. Because people are resistant to algorithms making important decisions[2] programmers should intentionally seek to eliminate biased algorithms.

Algorithms can be written without bias, but it is imperative that the programmer is aware of the possibility to avoid unintentionally building applications that include pre-existing social biases. In order for students to recognize bias in algorithms or datasets, they must first be aware of the possibility for bias. We propose a way to introduce awareness of cultural diversity/bias in algorithms in introductory computer science classes. Introductory classes usually focus on programming design, logic, and syntax. These classes may also introduce the concepts of security and testing. We describe a non-threatening assignment to promote the awareness of cultural bias to students.

Learning Objectives of the assignment:

- Promote awareness of the potential for cultural bias in programming algorithms.
- Promote awareness of the individual and societal impacts of cultural bias in programming algorithms.
- Promote awareness of the need to identify and mitigate cultural bias during program design, implementation, and testing.
- Encourage questioning of the integrity and inclusiveness of algorithms.

We review the completed assignments and present a future study to assess the effectiveness of this intervention.

2 Background

Research has identified that bias exists in algorithms of various areas of computing. Social media such as Twitter, dating apps, and Facebook use some form of recommender systems that learn from data to predict behavioral patterns.

However, various biases have been modeled by developers in the algorithm to influence the pattern [3]. Bias has been found in application algorithms used in hiring [7, ?], credit scoring [10], facial recognition [6], and medical treatment [15]. Pedestrian detection algorithms used in mobile robotics, such as autonomous vehicles, can result in devastating results if bias is not considered. A study using INRIA/Caltech datasets for evaluation of pedestrian detection showed a clear bias with children and women having higher miss rates. That is, when presented with pictures of women and children pedestrians, the algorithm 'ignored' them [4].

In April 2016, IEEE Standards Association (IEEE-SA) launched the IEEE Global Initiative on Ethics for Autonomous and Intelligent Systems (A/IS) [17]. The initiative published "Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems[18] with the goal of beginning a discussion to align technology with values and principles that promote a more balanced view of human well-being in a given cultural context[12]. Also, a set of eleven groups were established to create the IEEE P70xx series of ethics standards and certification programs for A/IS. IEEE P7003 is the standard currently under development for Algorithmic Bias Considerations. The purpose of IEEE P7003 is to describe specific methodologies that assist in identifying and mitigating negative bias in algorithms. Where "negative bias infers the usage of overly subjective or uniformed data sets or information known to be inconsistent with legislation concerning certain protected characteristics (such as race, gender, sexuality, etc.); or with instances of bias against groups not necessarily protected explicitly by legislation, but otherwise diminishing stakeholder or user well-being and for which there are good reasons to be considered inappropriate" [1].

ACM Public Policy Council has issued a statement on Algorithmic Transparency and Accountability [16]. The council proposed that the opaqueness of some algorithms makes it difficult to assert whether outcomes are correct, erroneous, or biased. This policy identifies 7 principles for Algorithmic Transparency and Accounting. The first is "Awareness". Designers and programmers should be aware of possible biases in the algorithms, datasets, and testing mechanisms.

3 Assignment Description

We devised an assignment to introduce students to the idea of bias in algorithms. We wanted the assignment to be simple and non-threatening and one that would not take away from essential and required material. We chose to require students to write a reflective paper about algorithmic bias. The assignment was given the week of the midterm. No other submissions were required for that week; the week consisted of review, the paper, and the midterm.

The assignment was introduced with two different examples: **(1) An example of algorithmic bias in facial recognition applications:** Camera input tracking on computers was developed in the early 1990s. The motion-tracking software uses artificial intelligence algorithms to detect a face. Once it finds the face it can follow that face around within the video stream. As the technology became more affordable, more people began to use it. The problems of algorithmic bias in the device were found by customers as seen in the youTube video [22]. This shows a computer tracking camera following a white woman but failing to follow the face of her black colleague. The students were also given an article summarizing the result of testing face recognition algorithms by the US National Institute of Standards (NIST) [9]. Some powerful results of that study include:

- For one-to-one matching, most systems had a higher rate of false-positive matches for Asian and African American faces over Caucasian faces, sometimes by a factor of 10 or even 100. In other words, they were more likely to find a match when there wasn't one.
- This changed for face recognition algorithms developed in Asian countries, which produced very little difference in false-positives between Asian and Caucasian faces.
- Algorithms developed in the US were all consistently bad at matching Asian, African American, and Native American faces. Native Americans suffered the highest false-positive rates.
- One-to-many matching, systems had the worst false-positive rates for African American women, which puts this population at the risk of being falsely accused of a crime.

(2) An example of algorithmic bias in popular digital assistants:l Bilingual cultures are not incorporated in many algorithmic-user centric products. Personnel digital assistants may not be able to recognize certain dialects. The students are shown a video of his phone not being able to understand a man asking to create a reminder. The man's Scottish accent is not recognized by the device [14]. The students are given an article by Wong which shows that *"while a white American male has a 92% accuracy rate when it comes to being understood by a voice-enabled assistant, a white American female has a 79% accuracy rate. A mixed-race American woman only has a 69% chance of being understood"* [21].

Students are then presented with five different scenarios of algorithmic bias. Students were required to submit a reflection on only one of the scenarios, answering the following questions:

- Describe the problem

- How could this problem make someone feel? How did it make you feel?
- What problems does this demonstrate or promote through society?
- What harm is there?
- Without writing code, suggest a way to address the problem with this algorithm

Students' reflection was accepted as either a 2-page paper or a 2-3 minute video. The intent of the assignment was not to evaluate a composition or video but to have the student seriously think of the scenario and relate it to their experiences. A summary of the scenarios presented to the students follows:

> **Scenario 1. Online screening applications:** The recruitment algorithms sampled from historical data that overlooks underrepresented applicants
>
> **Scenario 2. Cultural Bias in Search Engine Algorithms:** Use two or three search engines to search one of your identifying characteristics (gender, race, height, job, birthplace, or some other aspect about yourself).
>
> **Scenario 3. Algorithms that require a change in your name:** Most of our applications that require names do not accommodate this cultural difference.
>
> **Scenario 4. Bias in criminal justice algorithms:** In our court system, judges use algorithms to help decide whether defendants will be detained or released while awaiting trial.
>
> **Scenario 5. Bias in training datasets:** Datasets used to test AI and other applications can affect the ethics of the algorithms tested.

The assignments were evaluated as to how the students used the reflective process. We considered descriptive reflection and dialogic reflection as described by Hatton and Smith [22]:

1. Descriptive: This is not reflection, but simply describes events that occurred with no attempt to describe 'why.'
2. Dialogic Reflection: Reflection as a personal dialogue involving questioning things, considering alternatives, etc. Examples include "I wonder..., what if..., perhaps..." types of statements.

4 Review of Assignment Submissions

The assignment was a requirement of an introductory CS course. Students in the class have a variety of programming experiences that include high school programming course(s), college programming course, no programming experience but math that includes some calculus. The data collected is from 2 semesters, Fall 2020 and Spring 2021. It should be noted that this assignment was given during the Covid Pandemic. The assignment was required of a total of 83 students as a part of the course. 60 students completed the assignment. Table 1 shows the number of students that completed each scenario with the most students completing scenario 2, Cultural Bias in a Search Engine.

Here is an example Table 1. Note that each table or figure **must** have a title as our style requirement.

Table 1: Total submitted assignments by scenario

	Fall 2021	Spring 2020	Total
Scenario 1	4	3	7
Scenario 2	14	9	23
Scenario 3	7	6	13
Scenario 4	2	1	3
Scenario 5	4	6	10

Scenarios were selected from researched articles and suggested from interviews with colleagues. Note that the actual wording for the assignment may come directly from the source. The source is cited in the assignment given.

4.1 Scenario 1: Online recruitment applications

This scenario was taken from article [19]. Actual wording of the assignment: *The Amazon recruitment algorithm revealed a similar trajectory when men were the benchmark for professional "fit," resulting in female applicants and their attributes being downgraded. These historical realities often find their way into the algorithm's development and execution, and they are exacerbated by the lack of diversity that exists within the computer and data science fields. Further, human biases can be reinforced and perpetuated without the user's knowledge. For example, African Americans who are primarily the target for high-interest credit card options might find themselves clicking on this type of ad without realizing that they will continue to receive such predatory online suggestions. In this and other cases, the algorithm may never accumulate counter-factual ad suggestions (e.g., lower-interest credit options) that the consumer could be eligible for and prefer. Thus, it is important for algorithm designers and operators to watch for such potential negative feedback loops that cause an algorithm to become increasingly biased over time.* Observations and selected student quotes

- "Being discriminated against, albeit unintentionally via an unfortunate algorithm, would still make anyone feel degraded."
- "People may see the harm of unintentional discrimination in such a scenario as this, but they may not see the effects that the discrimination might influence later."
- "Algorithms like this are very harmful and can make people feel less valuable."
- AI is not naturally biased but simply projects the biases of its programmers through its program.

Enhancements for the future: Some students read 'Amazon' and immediately thought of the recommender engine of the website. They reflected on how the recommender system is biased. The scenario can be updated to more clearly state that reflection on algorithms that screen data.

4.2 Scenario 2: Cultural Bias in Search Engine Algorithms

This scenario was suggested by the Assistant Dean for Diversity, Engagement, & Inclusive Excellence at Winthrop University. This was the most popular scenario selected. Actual wording of the assignment:*Use two or three search engines to search one of your identifying characteristics (gender, race, height, job, birthplace. . . . . .). What did you find? Are the suggested links what you would have picked for yourself? How does this make you feel? What problems does this search algorithm demonstrate or promote? How would you change the results?* Observations and selected student quotes: The submissions for this assignment were the most personal. While this may have been the 'easiest' assignment, students took it seriously. The option for a video allowed for very genuine responses.

- An African American woman looked up 'cute hair'. This resulted in a plethora of images of Caucasian women and hairstyles. Since her hair did not grow like that, she concluded that the message from the search engine was that it did not find her type of hair 'cute'.
- A young woman looked up 'bi-racial woman'. She was taken to sites about violence against biracial women and others that referenced "what are you?" question often asked of bi-racial persons. She opened up about her feelings and ended her video saying "This was definitely an interesting assignment, I never actually looked this up. It enlightened me."

Enhancements for the future: Searches for generic attributes, men, women, height, did not lead to interesting results. The assignment could change to suggest the student search for a 'unique' characteristic.

4.3 Scenario 3: Algorithms that require a change in your name

This scenario was suggested by a colleague who has dealt with the problem. Actual wording of the assignment: *Like many cultures, last names of Spanish origin are derived from family names, place names, descriptive names, or names of occupations. Spanish names, however, don't always follow a linear path. A person, for example, may have two last names, one from their mother and one from their father. Their children may take two last names from their father and one from their mother A woman may take only her mother's surname Most of our applications that require names do not accommodate this cultural difference. How do you think that makes people feel? Should we require them to change their name by adding a hyphen? Is that saying, "your name is a user error"? What does this do to someone or a culture? What harm is there?* Selected quotes from student responses:

- Students are more used to this problem. They talked more about how to fix it by explaining and requiring the human to adjust "It's just a computer and there is only so much it can do".
- "The problem that I see is that the algorithm is requiring some people to forego some identifying characteristic of who they are in order to give themselves a fair chance at some opportunity."
- "It is very unlikely that program errors like this are intended to exclude [polyonymous] (as this would definitely hurt both the PR and overall application acceptance rate of companies), but it still very much is an issue that needs to be addressed. As long as problems like these go unfixed, more and more similar issues will arise and cast companies with good intentions in a negative light."
- "I feel like it needs to be more understandable. Coming from a Hindu background, I don't have a middle name, and parents have different last names. I guess it is something that as a professional, you should consider this."

4.4 Scenario 4: Bias in criminal justice algorithms

This scenario was taken from articles [[19],[5], and [11]]. Actual wording of the assignment: *Northpointe, the firm that sells the algorithm's outputs, offers evidence to refute such claims and argues that wrong metrics are being used to assess fairness in the product. The algorithm assigns a risk score to a defendant's likelihood to commit a future offense, relying on the voluminous data available on arrest records, defendant demographics, and other variables. Compared to whites who were equally likely to re-offend, African Americans were more likely to be assigned a higher risk score, resulting in longer periods of* *detention while awaiting trial. The COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) algorithm, which is used by judges to predict whether defendants should be detained or released on bail pending trial, was found to be biased against African Americans, according to a report from ProPublica [[19],[5], and [11]]..* Selected quotes and paraphrased responses from student responses:

- "The best way to get through this is to fix up the algorithm within the program, so that it is fairly able to make its calculations no matter what the skin tone of the person. Maybe even just destroying the current and rewriting it all together from scratch, maybe with bringing in all different type of computer programmers."
- What I find interesting is whether the data is skewed or not skewed. Because data doesn't tell you everything.

4.5 Scenario5: Bias in training datasets

This scenario was taken from article [13]. MIT created a dataset library in 2008 open to the public to train various applications. It is, essentially, a huge collection of photos with labels describing what's in the pictures, all of which can be fed into neural networks to teach them to associate patterns in photos with descriptive labels. A problem occurred when the dataset became corrupted and was not monitored. As such it trained various applications before the error was caught. Actual wording of the assignment: *The training set, built by the university, has been used to teach machine-learning models to automatically identify and list the people and objects depicted in still images. For example, if you show one of these systems a photo of a park, it might tell you about the children, adults, pets, picnic spreads, grass, and trees present in the snap. Thanks to MIT's cavalier approach when assembling its training set, though, these systems may also label women as whores or bitches, and Black and Asian people with derogatory language. The database also contained close-up pictures of female genitalia labeled with the C-word.* Applications, websites, and other products relying on neural networks trained using MIT's dataset may therefore end up using these terms when analyzing photographs and camera footage. The problematic training library was created in 2008 to help produce advanced object-detection techniques. It is, essentially, a huge collection of photos with labels describing what's in the pictures, all of which can be fed into neural networks to teach them to associate patterns in photos with descriptive labels. So, when a trained neural network is shown a bike, it can accurately predict a bike is present in the snap. It's called Tiny Images because the pictures in the library are small enough for computer-vision algorithms in the late-2000s and early-2010s to digest [13]. Selected quotes and paraphrased responses from student responses:

- "This, of course, could create a negative or dejected feeling imposed on people. Humans certainly do not like it when others call each other derogatory terms, so having an AI do it would certainly not be any better. It could create some reinforcement that using these terms and their meaning could be constructed as socially acceptable. Given the current climate of trying to escape the previous environment of sexism and racism, this could cause prevention of achieving that goal soon."
- "It is disheartening to see people input derogatory items in the training set used for application that programmers work so hard to create. It is important to collaborate to manage and watch the input to these training sets."
- "The harm these incorrect and derogatory descriptions can cause is severe. ImageNet can misclassify people (especially minorities) as things they are not. On the severe end, this could lead to wrongful arrests which can have ripple effects across the economic and social status of minority groups. Less minor yet more widespread, this data set could cause inconveniences due to incorrect labeling in contactless shopping experiences (e.g. Amazon Go) and misidentify people in augmented reality applications."

5 Future Wiork

The goal of presenting this assignment to introductory-level students was simply to make them aware of the problem of algorithmic bias without taking too much time from the existing required syllabus. We are working on a survey to give the students after the course to understand if they think more about algorithmic bias. Have they noticed it in their daily work or have the checked for it in datasets given by the professors or in how they managed input in their assignments. Improve how the assignment is presented and how the students should approach a reflective paper. reflection. First-year students may not have written a paper like this. It is important to provide some additional direction. Update the questions the students should address:

- Describe the problem
- How could this problem make someone feel? How did it make you feel?
- How could this problem negatively affect an individual's life?
- What negative effects could this problem have on society?
- Without writing code, suggest a way to address the problem with this algorithm

Consider embedding algorithmic bias in other courses in the curriculum.

6 Conclusion

Ethics should be a core subject for the computer science field. It should begin at the earliest possible course and continue through the curriculum [2]. Ethics guide our understanding of what is right and affects our behavior. Algorithms can reflect the implicit bias and parochial cultural awareness of the humans writing or testing the algorithms. As such, they affect the behavior of the algorithm and may exhibit algorithmic bias. Awareness of this possibility is key in the effort to remove algorithmic bias.

7 References

References

[1] IEEE Standards Association. Project: P7003 - algorithmic bias considerations. `https://standards.ieee.org/project/7003.html`.

[2] Yochanan Bigman, Kai Chi Yam, Déborah Marciano, Scott Reynolds, and Kurt Gray. Threat of racial and economic inequality increases preference for algorithm decision-making. *Computers in Human Behavior*, 05 2021.

[3] Ludovico Boratto, Gianni Fenu, and Mirko Marras. *The Effect of Algorithmic Bias on Recommender Systems for Massive Open Online Courses*, pages 457–472. 04 2019.

[4] Martim Brandao. Age and gender bias in pedestrian detection algorithms. *arXiv e-prints*, page arXiv:1906.10490, June 2019.

[5] Tim Brennan, William Dieterich, and Beate Ehret. Evaluating the predictive validity of the compas risk and needs assessment system. *Criminal Justice and Behavior*, 36(1):21–40, 2009.

[6] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *FAT*, 2018.

[7] Jeffery Dastin. Amazon scraps secret ai recruiting tool that showed bias against women. *Reuters*, 2017.

[8] Batya Friedman and Helen Nissenbaum. Bias in computer systems. *ACM Trans. Inf. Syst.*, 14(3):330–347, July 1996.

[9] Karen Hao. A us government study confirms most face recognition systems are racist., 2019. Accessed on 2021-07-29.

[10] Bertrand Hassani. Societal bias reinforcement through machine learning: a credit scoring perspective. *AI and Ethics*, 1:1–9, 08 2021.

[11] J.Angwin, J.Larson, S.Mattu, and L.Kirchner. Machine bias. 2016. Accessed on 2021-07-29.

[12] Ansgar Koene, Liz Dowthwaite, and Suchana Seth. Ieee p7003™ standard for algorithmic bias considerations: Work in progress paper. In *Proceedings of the International Workshop on Software Fairness*, FairWare '18, page 38–41, New York, NY, USA, 2018. Association for Computing Machinery.

[13] K.Quach. Mit apologizes, permanently pulls offline huge dataset that taught ai systems to use racist, misogynistic slurs. *The Register*, 06 2020. Accessed on 2021-07-29.

[14] James McDonald. Siri cannot understand scottish folk., 2011. Accessed on 2021-07-29.

[15] Z. Obermeyer, Brian W. Powers, C. Vogeli, and S. Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366:447 – 453, 2019.

[16] ACM. Public Policy Council: Statement on Algorithmic Transparency and Accountability. (2017). `http://bit.ly/2n4RBjV`.

[17] The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. `https://ethicsinaction.ieee.org/`.

[18] The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. Ethically aligned design: A vision for prioritizing human well-being with autonomous and intelligent systems, version 2. ieee, 2017. `http://standards.ieee.org/develop/indconn/ec/autonomous_systems.html`.

[19] Lee Resnick and Barton. Algorithmic bias detection and mitigation: Best practices and polices to reduce consumer harms. *Brookings Institution*, 2019. Accessed on 2021-07-29.

[20] Judith Simon, Pak-Hang Wong, and Gernot Rieder. Algorithmic bias and the value sensitive design approach. 2020.

[21] Henry Wang. Siri Alexa and unconscious bias: the caase for designing fairer AI assestants. *Design Week*, 2020. Accessed on 2021-07-29.

[22] youtube Website. Hp computers are racist, 2009. Accessed on 2021-07-29.

Making Smart Platforms Smarter: Adding Third Party Applications to Home Automation Platforms*

Theppatorn Rhujittawiwat[1], Colin Anderson[1], David Keen[1]
Cort Miles[1], Csilla Farkas[1], Shiloh Smiles[2], Noah Wells[2]
Jessica Roginski[2], Slateon Frederick[2], Shankar Banik[2]
[1]University of South Carolina
Columbia, SC 29208
[2]The Citadel
Charleston, SC 29409

Abstract

In this paper, we propose an architecture to enable third party applications within the popular home automation environment. Our claim is that the functionality of these popular smart platforms can be improved by enabling third party applications, such as situation-aware decision making. We show how third-party applications may support home automation systems to respond to complex environmental changes. Our research team implemented a proof-of-concept prototype system to demonstrate how home automation applications can interact with logic reasoners to support dynamic system policies. Our implementation was built using the open source Home Assistant platform and the Protégé-OWL reasoner. Our theoretical results include threat modeling, development of use and misuse cases for smart home environment, support for data exchange between the home automation system and the reasoner, and the development of dynamic policy support.

1 Introduction

The development of technologies and infrastructure to support the Internet of Things (IoT) [12] impacts our everyday lives. Smart manufacturing, smart cities, and smart environments, built on these IoT technologies, are some of the examples of recent development efforts. Smart home environments [8, 2], where home owners can integrate and control multiple devices via wireless connections, have gained popularity during the last decade. Home Assistant [1], ioBroker [9], Domoticz [6], and openHAB [16] are some of the popular open source home automation systems.

Along with the development of smart systems, cybersecurity and privacy threats have emerged. Home Automation applications often lack the functionalities to protect sensitive data collected by the application. Moreover, they lack the capability to adopt the application policy based on changes in the environment. For example, during emergency the smart environment may want to respond differently than during normal operation. Consider the case when there is a medical emergency. In addition to the 911 alert, the smart home may want to initiate new actions, such as unlocking the entry doors to allow easy access for the first responders and contacting the home owner's medical provider. Clearly, such actions should not occur if there is no emergency. Current home automation systems do not support such advanced functionalities.

In this paper we aim to demonstrate the need to extend the current functionalities of home automation systems and to show how this can be achieved. We argue, that the best approach to support these advanced functionalities is by enabling third-party applications within the home automation environment. Such applications can support user-centric and dynamic requirements as well as address the specific safety, cybersecurity, and privacy needs of the home automation environment. Moreover, multiple home automation systems can integrate these third-party applications.

The main contributions of our work are: 1) We recommended an architecture to enhance current home automation technologies to meet the needs of end-users. Our approach is extensible and scalable and can be seamlessly integrated with the technologies used to develop IoT applications. 2) We have developed a flexible and scalable data exchange schema, using Extensible Markup Language (XML) [18], to facilitate secure and interoperable data sharing between the home automation system and the third party applications. 3) We have implemented a proof-of-concept system to support dynamic and situation-aware decision making for home automation. 4) We have shown that modeling potential threats against the home automation context may allow the discovery of new use cases and non-traditional functionality of the home automation environment. The uniqueness of our work is that we were able to control the behavior of the home automation system based on the situation decision derived by the Protégé reasoner. Such enhancement of smart platforms are critical in supporting complex requirements for owners with special needs, such as disabilities or aging in place.

The organization of the paper is as follows: In Section 2 we give an overview of the proposed system, the Home Assistant, and the Protégé reasoner. In Section 3 we describe the advantages of use and misuse cases and demonstrate their use via a specific example. In Section 4 we present the use of Protégé to derive situation-aware decisions. In Section 5 we discuss our approach for bi-directional data sharing between the Home Assistant and the Protégé reasoner. Finally, we conclude and discuss our ongoing work in Section 6.

2 System Architecture

Smart home automation systems aim to allow home owners to seamlessly integrate multiple devices. Typical functionalities provided by home automation systems include fire and carbon monoxide monitoring, remote control of lights, thermostat, and smart appliances; security systems and cameras; real-time alerts; and integration with smart platforms (e.g., Alexa, Siri, Google Home). Some of the home automation systems also support data analysis to help data-driven decision making. However, current home automation systems fall short in incorporating the achievements of the situation-aware decision making research or to allow easy plug-in third party applications.

2.1 System Overview

In this paper we present a local architecture for extending the popular home automation systems with third party applications. Figure 1 shows the system overview for our application. Note, in Section 6 we discuss our ongoing effort to extend this architecture to cloud computing platforms, such as Microsoft Azure [13], to provide real-time support.

The current implementation focuses on connecting Home Assistant, an open source home automation system, with a Protégé-OWL reasoner. The current proof-of-concept implementation is based on using local communication between Home Assistant and the reasoner. Device data from Home Assistant is converted to XML format and pushed to the reasoner. The result from the reasoner is sent back in XML format to be inserted to the Home Assistant database. Based on the received feedback from the reasoner, Home Assistant carries out the appropriate actions.

The advantage of decoupling the reasoner from the Home Assistant application is that our approach can also be used with other home automation software. The data exchange schema is based on data semantics and independent from local data models. The same schema can support other home automation platforms with mapping their data models to the data exchange format.

2.2 Home Assistant Overview

Home Assistant is a free and open source software platform that can be used for integrating IoT devices in a home network. It has a web user interface that can be used as a central system for controlling the devices in the home IoT network. The Home Assistant Operating System can be installed on various platforms. After the installation, the user creates an owner account with administrator privileges. Next, the Home Assistant shows the discovered devices in the network and provides the interface for integrating these devices. The user then can write automation rules based on triggering of an event (e.g., "Turn the porch light off at sunrise"), or based on user detection (e.g., "Turn the front door light on when user Alex enters the house").

2.3 Context and Situation-Aware Reasoning

Context-aware decision support has been studied extensively in the context of access control policies [5, 10, 14]. Weather, location, fire, computer network, and user security clearances are examples of contextual information. Context-Aware Access Control proposes a flexible and dynamic access control which can adapt to specific context characteristics. During the last decade, situation-aware security approaches were developed to address the dynamic and ad-hoc nature of IoT networks [19, 3, 17, 4]. These efforts aim to extend the security capabilities to identify situations and adopt the appropriate policy accordingly. Semantics-based data integration and ontological reasoning are the backbone of these approaches. Despite the achievements of the situational-aware security researchers, these technologies are not sufficiently incorporated in smart home environment. The closest to our work is by Chen et al. [3]. The authors address the specific needs in smart home environment, focusing of assisted living. Our primary focus on the security, safety, and privacy of smart homes, in particular, in the context of home automation systems.

3 Use Cases and Misuse Cases

Use cases are widely used to model functional requirements is software and system engineering. Uses cases allow the developers to analyze how the system is planned to be used and who are the intended actors. Actors may be human users of external systems or events. During the last couple of decades, the concept of misuse cases (or abuse cases) have been developed. Misuse cases allow the developers to to analyze negative scenarios and define how the system

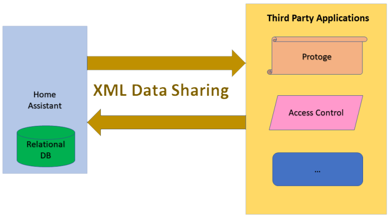

Figure 1: Smart Home Automation System Architecture

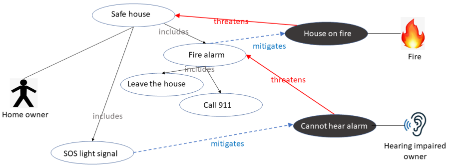

Figure 2: Home Assistant's response to fire in case of hearing impaired home owner

should behave under such circumstances (see Figure 2 for a representative example on misuse cases). Misuse cases impact the use cases and often lead to new functionality requirements. Modelling potential threats against the system has been shown to increase the security, safety, and reliability of such systems.

Understanding threats against the smart home environment is crucial. Most of the research and development efforts focused on the security and privacy threats against the underlying technologies and infrastructure (see [15] for an overview). More recently, researchers addressed the threat against smart home environment from the behavioral aspect of the malicious users, e.g., insiders' misuse of the technology [11].

Our aim is to discovers new use cases based on understanding both technical and behavioral threats. For example, consider the use case of providing fire safety for a smart home. The interleaved use case and misuse case diagrams are presented in Figure 2.

The main actors are the home owner (intended user); the house fire, and the hearing impairment of the home owner (malicious actors). The house fire threatens the safety of the house and the owners. The sound of the fire alarm mitigates the threat by alerting the home owner to the threat; so the owner could leave the house and call 911. However, if the home owner is hearing impaired, the alarm may go unnoticed, thus threatening the effect of the fire alarm. To mitigate this threat, we extended the home automation system to turn on the light and signal the Morse code for S.O.S. signal repeatedly. The three short, three long, and three short signal sequence is universally known to indicate a crisis situation. By projecting S.O.S., the home automation system would alert the home owner, as well as external observers for the crisis.

4 Protégé Reasoner

We implemented our dynamic policy reasoner using Protégé[7] and Semantic Web Rule Language (SWRL). Protégé is an open-source ontology tool for Web Ontology Language (OWL) development. It provide various functionalities such as creating, modifying, reasoning, querying, and visualizing an ontology. SWRL can enhance reasoning capabilities of OWL by providing a rule language.

Figure 3 shows a simplified temperature reasoner ontology. We used this simplified ontology to derive the current situation based on the observed temperature. Consider the following examples where the room temperature indicates normal, caution, or emergency situations.

SWRL rule 1. If the observed temperature at any given locations and time is between 70 to 80 degrees Fahrenheit, the temperature status is Normal.

$$Observation(?x) \wedge hasResult(?x,?y) \wedge hasTemperature(?y,?r) \wedge hasUnit(?y,"Fahrenheit") \\ \wedge\, swrlb:lessThan(?r,80) \wedge swrlb:greaterThanOrEqual(?r,70) \rightarrow \\ SystemStatus(CurrentTemperatureStatus) \wedge hasStatus(CurrentTemperatureStatus,Normal)$$

SWRL rule 2. If the observed temperature at any given locations and time is between 80 to 120 degrees Fahrenheit, the temperature status is Caution.

$$Observation(?x) \wedge hasResult(?x,?y) \wedge hasTemperature(?y,?r) \wedge hasUnit(?y,"Fahrenheit") \\ \wedge\, swrlb:lessThan(?r,120) \wedge swrlb:greaterThanOrEqual(?r,80) \rightarrow \\ SystemStatus(CurrentTemperatureStatus) \wedge hasStatus(CurrentTemperatureStatus,Caution)$$

SWRL rule 3. If the observed temperature at any given locations and time is more than 120 degrees Fahrenheit, the temperature status is Emergency.

$$Observation(?x) \wedge hasResult(?x,?y) \wedge hasTemperature(?y,?r) \wedge hasUnit(?y,"Fahrenheit") \\ \wedge\, swrlb:greaterThanOrEqual(?r,120) \rightarrow SystemStatus(CurrentTemperatureStatus) \\ \wedge\, hasStatus(CurrentTemperatureStatus,Emergency)$$

We use the temperature status (conclusions of the SWRL rules) to derive situational information. These conclusions trigger state changes in our finite-state machine in Figure 4. We derive four situations; normal, system malfunction, extreme heat, and fire.

Situation: Normal. The temperature status is normal. The system resumes normal operation and waits for the next observation.

Situation: Fire. If the temperature status changes rapidly from normal to emergency temperature. The system can derive that the fire occurs. The emergency procedures will be followed such as raising the alarm, calling the fire service, and activating water sprinklers. Note, that after reaching this situation state, the state will have to be changed manually by the system administrator to ensure that the emergency situation is resolved.

Situation: System Malfunction. If the temperature status changes slowly from normal to caution temperature. The system can derive that there

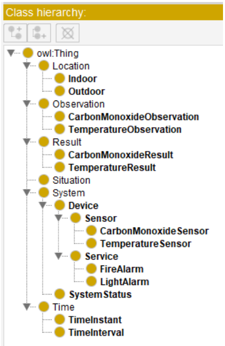

Figure 3: The Simplified Temperature Reasoner Ontology in the Protégé

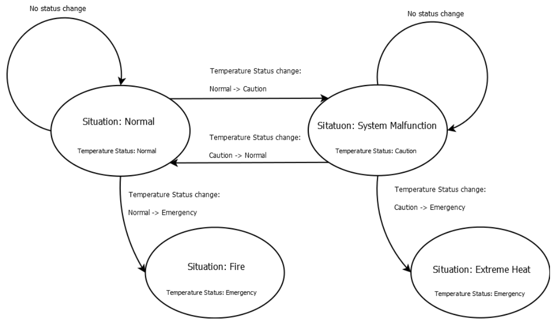

Figure 4: The Finite-State Machine for Situation Derivation

is a potential malfunction in the temperature control system. The maintenance should be requested to fix the issue. The system submits the alert and waits for the next observation to update the situation. The situation can return to normal after the malfunction is fixed and temperature status returns to normal.

Situation: Extreme Heat. If the temperature status changes slowly from caution to emergency temperature. The system can derive that there is a potential health hazard from built-up heat. To avoid heat stress, all people in the area should be evacuated. Note, that after reaching this situation state, the state will have to be changed manually by the system administrator to ensure that the emergency situation is resolved.

5 Home Assistant Data Model and Data Sharing

In this section we review the database schema supported by the Home Assistant and our proposed data sharing.

5.1 Home Assistant Data Model

Figure 5 shows the high-level overview of the database that stores the device data and control information of Home Assistant. Detailed information about the data structure and user support to query the database is available at the Home Assistant's website [1]. Our research team used the data model to answer the following two specific questions: 1) How to retrieve data that is needed for the Protégé decision making? 2) How to store the result from the reasoner such that the appropriate automation is carried out?

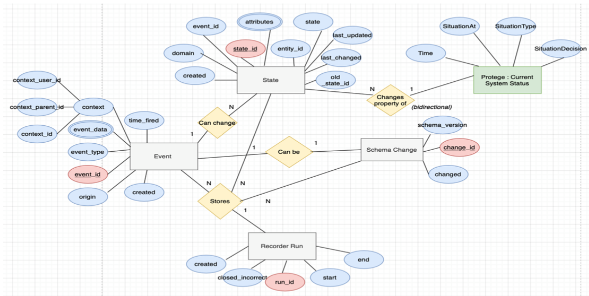

Figure 5: Home Assistant database Entity-Relational Data Model

The Home Assistant's database contains four tables: Events, States, Recorder Runs, and Schema Changes. The most important one is the events table. Everything that "happens" in Home Assistant is represented as an event. Each event has an unique ID, type, data, and other attributes such as context. The context is used to tie events and states together. Whenever an automation or user interaction triggers a new change, a new context is created. This new context is attached to all events and states that occur as the result of the change. States are representations of Entities in Home Assistant. Each entity is a certain function of a device. For example, a motion sensor is one device with three entities: motion, temperature, and light. Each entity has a state such as on/off associated with it. States have an unique state id, the event id that changed the state, and certain attributes that reveal extra state data about the entity. The value last changed is the last time the state changed, whereas last updated is the last time the state was checked, but not necessarily changed. A schema change is an event that changes the database schema. This can happen when an integration for a sensor platform is added. This allows certain data to be added and available. The recorder is responsible for storing all the data. Each time Home Assistant starts, the recorder starts a new recorder run. A run is finished when Home Assistant gracefully shuts down or is restarted.

5.2 Home Assistant and Protégé Data Sharing

Home Assistant supports built in functionalities as well as provides the capability to code additional functionalities. To facilitate data exchange between the Home Assistant platform and the reasoner, we developed an XML schema for data sharing. This approach allows to extend the data sharing for future applications as well as supported by most computing platforms. XML has been used to support data sharing among heterogeneous data sources. Figure 6 outlines how the data is extracted from the Home Assistant database, converted to the XML schema, and shared with the reasoner. Similarly, the decision reached by the reasoner is converted to the XML format, and incorporated in the Home Assistant database.

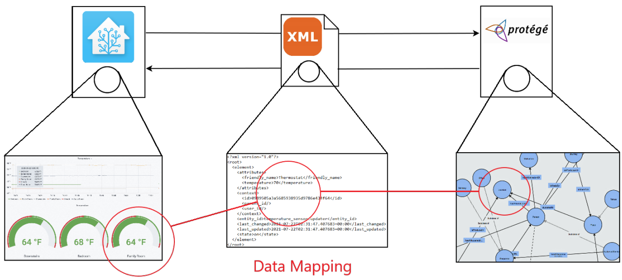

Figure 6: Communication between Home Assistant and Protégé via XML

To get data from Protégé to Home Assistant, we converted the CSV file that was exported by Protégé. We used basic bash commands to create a script that got the emergency status and the location of the emergency. To represent the decision data, a state was created within the Home Assistant system that contained the hazard status (normal/emergency), hazard location (bedroom, kitchen, etc.), and the hazard type (fire, burglar, etc.). To push the data, simple REST API calls were made. To monitor for emergency scenario, we implemented a Home Assistant script to monitor the state containing the Protégé decision. In case the hazard status changed to emergency, Home Assistant sat the emergency flag and ran the corresponding automation. In our fire emergency scenario, this automation is the repeated flashing of the S.O.S. light signals.

6 Conclusions and Future Work

In this paper, we proposed a local system architecture to support the extension of home automation systems with third party applications. We argued that such extensions would enhance the functionality of the home automation environment. In particular, we focused on supporting semantics-based data sharing and dynamic policy support. We proposed the use of XML data exchange format to allow interoperation and make our proposed approach applicable to a variety of home automation systems. We also demonstrated that our reasoner was able to derive situation-aware information from the device data. This information, when received by Home Assistant, lead to the desired activity, i.e., the signalling of the S.O.S. Morse code via smart light bulbs. The uniqueness of our work is that we were able to control the behavior of the home automation system based on the situation decision derived by the Protégé reasoner. Such enhancement of smart platforms are critical in supporting complex requirements for owners with disabilities.

The main limitation of our current results is that the current system is implemented locally. The communication between the home automation system and the Protégé reasoner is based on push/pull of the messages. This is clearly is not optimal for real-time support or to support multiple third party applications. Our ongoing work focuses on a cloud-based solution. Our initial design is to upload the device data to a cloud data-lake. This data-lake will supply the third party applications. Using the cloud-based solution will also enable the use of additional applications, such as public services and data analytics. Moreover, it will provide a centralized and consistent enforcement of data security and privacy needs.

Acknowledgement

This project was supported under the NCAE-C Cyber Curriculum and Research NCAE-C-003-2020 Program, Grant No. H98230-20-1-0367.

References

[1] Home Assistant. Open source home automation. `http://www.home-assistant.io/`, 2021.

[2] Sumathi Balakrishnan, Hemalata Vasudavan, and Raja Kumar Murugesan. Smart home technologies: A preliminary review. In *Proceedings of the 6th International Conference on Information Technology: IoT and Smart City*, ICIT 2018, page 120–127, New York, NY, USA, 2018. Association for Computing Machinery.

[3] Liming Chen, Chris Nugent, and Ahmad Al-Bashrawi. Semantic data management for situation-aware assistance in ambient assisted living. iiWAS '09, New York, NY, USA, 2009. Association for Computing Machinery.

[4] Bo Cheng, Ming Wang, Shuai Zhao, Zhongyi Zhai, Da Zhu, and Junliang Chen. Situation-aware dynamic service coordination in an iot environment. *IEEE/ACM Trans. Netw.*, 25(4):2082–2095, August 2017.

[5] Michael J Covington, Wende Long, Srividhya Srinivasan, Anind K Dev, Mustaque Ahamad, and Gregory D Abowd. Securing context-aware applications using environment roles. In *Proceedings of the sixth ACM symposium on Access control models and technologies*, pages 10–20, 2001.

[6] Domoticz. Home automation system. `http://www.domoticz.com/`, 2021.

[7] Stanford Center for Biomedical Informatics Research. Protégé. `https://protege.stanford.edu/`, 2021.

[8] Terence K.L. Hui, R. Simon Sherratt, and Daniel Daz Snchez. Major requirements for building smart homes in smart cities based on internet of things technologies. *Future Gener. Comput. Syst.*, 76(C):358–369, November 2017.

[9] ioBroker. Open source home automation platform. `https://www.iobroker.net/`, 2021.

[10] Devdatta Kulkarni and Anand Tripathi. Context-aware role-based access control in pervasive computing systems. In *Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 113–122, 2008.

[11] Roxanne Leitão. Anticipating smart home security and privacy threats with survivors of intimate partner abuse. In *Proceedings of the 2019 on Designing Interactive Systems Conference*, DIS '19, page 527–539, New York, NY, USA, 2019. Association for Computing Machinery.

[12] Zakaria Maamar. Internet of things: The way ahead. In *Proceedings of the 12th International Conference on Management of Digital EcoSystems*, MEDES '20, page 1, New York, NY, USA, 2020. Association for Computing Machinery.

[13] Microsoft. Ms azure. `https://azure.microsoft.com/`, 2021.

[14] Yaser Mowafi, Dhiah Abou-Tair, Tareq Aqarbeh, Marat Abilov, Viktor Dmitriyev, and Jorge Marx Gomez. A context-aware adaptive security framework for mobile applications. In *Proceedings of the 3rd International Conference on Context-Aware Systems and Applications*, ICCASA '14, page 147–153, Brussels, BEL, 2014. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[15] Gerard O'Driscoll. *Essential Guide to Smart Home Automation Safety & Security: Use Home Automation to Increase Your Families Safety Levels (Smart Home Automation Essential Guides) - Volume 3*. CreateSpace Independent Publishing Platform, North Charleston, SC, USA, 1st edition, 2015.

[16] openHAB. Vendor and technology agnostic open source automation software. `https://www.openhab.org/`, 2021.

[17] Mozhgan Tavakolifard. Situation-aware trust management. RecSys '09, New York, NY, USA, 2009. Association for Computing Machinery.

[18] World Wide Web Consortium (W3C). Extensible markup language. `https://www.w3.org/XML/`, 2021.

[19] Seung Yi. *Situation-Aware Security for Wireless Ad Hoc Networks*. PhD thesis, USA, 2005. AAI3199187.

Evaluating the Impact of the Pandemic on STEM Outreach Events Among Female Participants*

Rahaf Barakat, Cindy Robertson,
Rebecca Kalman, Caroline Hanson, Lorraine Jonassen
School of Science and Technology
Georgia Gwinnett College
Lawrenceville, GA 30043
{rbarakat, crobertson2, rkalman, chanson, ljonassen}@ggc.edu

Abstract

The COVID-19 pandemic has abruptly affected our lives in so many ways. This paper presents our observations and comparisons between virtual and onsite (face-to-face) implementations of a STEM outreach event that targets female minorities and low-income families. This STEM outreach event has been facilitated through the S3 (Super Saturday Series) faculty led committee at Georgia Gwinnett College (GGC). The aim is to determine if the online STEM event, which teaches programming, chemistry, and biology, can be as effective as the onsite event, that uses interactive, hands-on workshops to teach the same skills. Data analysis indicates no significant difference in event effectiveness; however, the paper discusses alarming enrollment and attendance issues and highlights important challenges related to the limitations of designing and delivering an online STEM workshop.

1 Introduction

Gender gaps continue to be high among the top sought-after, fastest growing and highest-paid jobs in STEM, especially in computer science and engineering

([4], [8], [11]). For example, in 2019, data from the Census Bureaus' American Community Survey (ACS) shows that women filled 48% of all U.S. jobs but held only 27% of STEM jobs [2]. Therefore, increasing the number of women in STEM becomes a mandate for U.S. higher education ([8], [10]). Unfortunately, despite the continued efforts to increase diversity in STEM, the gender gap persists in the U.S. and across the world. This gap is fueled by stereotypes that women lack the ability and talent to perform well in science and technology ([6], [9], [16]).

In recent years, there have been substantial efforts aimed at gaining a better understanding of the factors that drive women away from STEM. The literature documents multiple barriers for women in STEM including gender bias, a climate of intimidation, perception of STEM as an inclusive environment for men and women, and discrimination in a typically male-dominated workplace ([1], [3], [7], [11]). For example, an experimental work concluded that gender bias exposure produces a gap in STEM engagement as it decreases the interests and sense of belonging among women [7]. In addition, women often feel isolated due to the lack of female peers, mentors, and role models [11].

Research has also indicated that female representation in STEM can be enhanced through educational institutions and departments by offering formal and informal STEM interaction opportunities [18]. These forms of learning experiences promote positive attitudes and engage women in science and technology. In addition, the middle and high school years are a critical time when STEM exposure could influence careers for women in STEM [14]. For example, the Girls Who Code program at Columbia University School of Engineering aims to close the gender gap in technology and engineering by running programming classes during the school year for female participants [12].

However, the COVID-19 pandemic has brought significant disruption to education [17]. It led to learning losses, decrease in student engagement, and an increase in inequality [13]. For instance, alarming emerging evidence found learning loss to be higher among students from low-income families that have limited access to technology ([5], [15]). In the absence of intervention, this learning loss could lead to fewer opportunities for minorities.

In this paper, we provide a comparison between the onsite (face-to-face) and online STEM outreach implementations and their effectiveness on female participants. Our research questions can be categorized into two groups: enrollment/attendance questions and event effectiveness/engagement questions. In addition, we share our findings on how the pandemic influenced the STEM event's design and delivery as well as its demographics.

2 Background

Gerogia Gwinnett College (GGC) is located about 35 miles of Atlanta in Gwinnett County, the most ethnically diverse county in Georgia. As metro Atlanta's leading hub for technology, bio-science, and innovative companies, Gwinnett has a strong need for tech-savvy professionals. Unfortunately, many women shy away from science and computing careers (e.g., only 19% of the IT majors at GGC are female as of 2019). To reverse this trend, GGC started the Super Saturday Series (S3) STEM initiative to excite girls about STEM early on in their education. Informal learning experiences like camps and extracurricular STEM activities can play a pivotal role in encouraging middle and high school students, especially females and those from underrepresented groups, to choose STEM careers.

S3 was originally established in 2011 with funding from the AAUW (American Association of University Women). The goal of S3 is to offer year-long, low-fee STEM programs aimed at increasing the number of women active in STEM through middle and high school. The program features an all day workshop held on GGC's campus with interactive hands-on technology, chemistry and biology activities. The event also features female professional guest speakers, on-campus dining, and demonstrations. All workshops are prepared and led by GGC faculty from the School of Science and Technology and student volunteers. Participants are exposed to problem solving, the scientific method and critical thinking skills with a focus on building confidence.

3 Methodology

3.1 Program Structure

Onsite S3 event structure: A typical S3 event starts at 8am and ends at 4pm. The event has the following layout: check in, delivery of pre survey, an introduction to the program, a programming activity, a chemistry activity, lunch, a professional guest speaker, networking, a biology activity, delivery of post survey, and dismissal.

Online S3 event structure: In order to facilitate an online event, we designed separate workshops over a 3 week period to cover specific topics in STEM. Each workshop lasted between 1 to 2 hours and had the following layout: join online, delivery of pre survey, provide an overview of the workshop, guide the students through the workshop, and delivery of post survey.

3.2 Workshop Design

Onsite S3 workshop design: The STEM workshops are designed to be highly interactive and hands-on. For example, the programming workshops involve different touchable technologies like robots and manipulation devices. All devices are pre-configured and provided to participants during the workshop. Chemistry and biology activities are equally interactive. They are held in lab settings and involve a series of engaging science quests. All supplies are provided to participants at the time of the workshop.

Online S3 workshop design: All workshops had to be redesigned for online delivery and interaction. The programming workshops required participants to pre-install software in order to participate. The chemistry and biology workshops required a specific list of supplies.

3.3 Advertisement and Recruitment

Onsite S3 advertisement: Our biggest recruitment body is the local county school system, providing 93% of our participants. We reach out to principals and other STEM coordinators at local schools. We also advertise on our campus for students and faculty. We only charge $25, including lunch, to remain accessible to low-income families.

Online S3 advertisement: Our advertisement and recruitment efforts for the online event stayed the same during the pandemic. However, to encourage participation, we offered the event free of charge for the first time.

4 Results

Enrollment and attendance. Our first research question seeks to understand how enrollment varied before and during the pandemic. In this study, we differentiated between participants who enrolled and attended the event. We provide comparisons between event enrollment in spring and fall 2019 (onsite before the pandemic) and fall 2020 and spring 2021 workshops (online during the pandemic). As a side note, there was no spring 2020 event due to campus closure because of the COVID-19 pandemic.

Overall we saw a significant decline of 36% in S3 enrollment because of the pandemic. This was alarming as we had expected that the ease of online delivery and the free event would attract more participants. We further analyzed the number of enrolled vs attended participants. Historically, our onsite events have maintained an average of a 5% decrease between enrollment and attendance, as some participants do not show up on the day of the event. For our first online event in fall 2020, we noticed a decrease of 33% between enrollment and attendance. However, our spring 2021 event only had a decline of 9%.

We believe this is due to the fact that participants and their families started adjusting to online learning and had more access to technology.

Demographics. Our second research question seeks to understand the influence of the pandemic on event demographics. Figure 1a demonstrates diversity in onsite attendance that aligns with the S3 mission. Figure 1b shows a sudden shift in the online demographics to reflect a 51% decline of participation of women of color and a 33% decline of Hispanic/Latino participants.

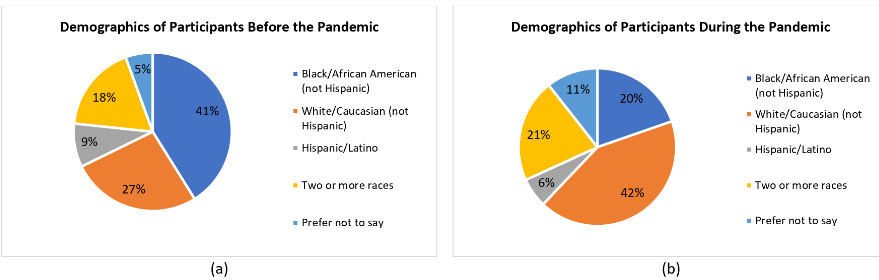

Figure 1: S3 demographics before(a) and during(b) the pandemic

Event effectiveness. Our third research question seeks to compare the effectiveness of our online verses onsite STEM outreach events. For that purpose, we administered pre and post surveys to participants. The surveys ask participants to rate themselves on a scale from strongly agree to strongly disagree in each of the STEM areas (programming, chemistry, and biology) over five categories: confidence in STEM abilities, perception of STEM, gender equality, interest in STEM, and family influence(pre)/motivation(post). Table 1 includes some sample questions from each category in the surveys.

We only consider the results of participants who completed both the pre and post surveys. Survey results from the onsite (spring and fall 2019) workshops are analyzed together, and the survey results from the online (fall 2020 and spring 2021) workshops are analyzed together.

Onsite S3 event. For the onsite event, 96 participants completed both the pre and post surveys and the results are shown in Figure 2. Four of the five categories were measured in both the pre and post surveys: confidence, perception, gender equality and interest. The results showed a 21% improvement in overall female confidence in STEM and a 44% improvement in their confidence in programming. Female perception of chemistry and biology stayed almost the same before and after the event. However, their perception of programming improved by 45%. While participants showed a high interest in STEM in the pre survey, their interests improved after the workshop, especially in programming which had an increase of 18%. Consistently, our participants showed that

Table 1: Sample Survey Questions

Category	Sample Questions
Confidence	I can get good grades in biology. I am confidant with my abilities in programming.
Perception	Programming is hard. Chemistry is boring.
Gender equality	Girls can do just as well as boys in programming. There are many females who are excellent in chemistry.
Interest	I intend to take courses related to biology in the future.
Family influence	My family encourages me to experiment with chemistry.
Motivation	Now that I have attended the workshop, I am motivated to learn more about programming.

they had a strong sense of their equal abilities in STEM when compared to their male peers.

We only measured family influence in the pre survey because that metric would not be affected by our workshop. Results suggested that our female participants felt most encouraged by their families to learn biology while least encouraged to learn programming. In addition, we only looked at motivation in the post survey to analyze if our workshop motivated students to pursue additional STEM learning. Almost 80% of the participants showed interest in learning more about biology, 78% showed interest in chemistry, and 76% expressed additional interest in programming after the event. In addition, among the 96 analyzed participants, 99% indicated that they would recommend this STEM event to others.

By comparing the pre and post survey of the S3 onsite event, we conclude that the event was effective in improving female participants' confidence, perception, and interest in STEM as well as maintaining their strong sense of gender equality.

Online S3 event. In this study, we identified 55 participants who participated in the STEM workshops (programming, chemistry, and biology) in fall 2020 and spring 2021. We analyzed their pre and post survey results for programming and chemistry. However, we did not report our results for the biology workshops because we did not have enough data. Our fall 2020 event

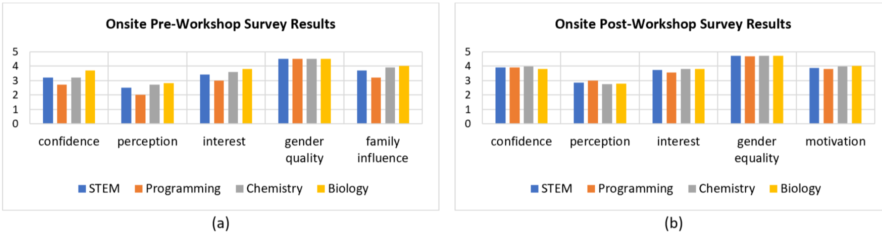

Figure 2: Onsite pre(a) and post(b) survey results over the five categories.

was cancelled and we only had 9 participants in spring 2021.

Programming workshops: We analyzed the results of 25 participants as shown in Figure 3a. The results suggest an improvement in confidence, perception, and interest in programming. However, in relation to gender equality, the female participants actually changed their mind after participating in the workshop. We saw a 6% decrease in the belief that girls can do just as well as boys in programming after attending the workshop. In addition to our previous 5 categories, we asked online participants about their perception of the online programming workshop in terms of enjoyment, engagement and benefit. Results are displayed in Figure 4a and illustrate an increase in all metrics.

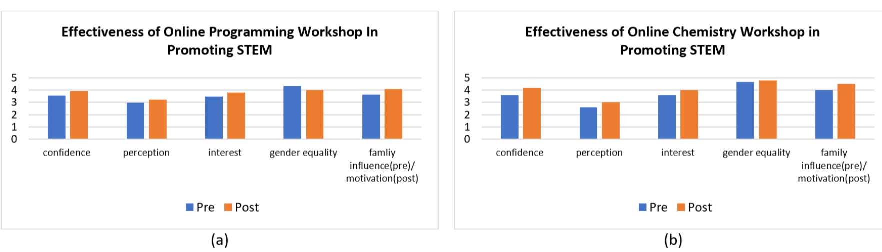

Figure 3: Effectiveness of the workshops during the pandemic.

Chemistry workshops: We analyzed the results of 21 participants as shown in Figure 3b. Results confirm the effectiveness of the workshop. Again, we asked additional questions regarding perception of enjoyment, engagement and benefit. Students showed consistent perception of enjoyment and engagement, but there was a slight decrease in benefit (5%) as shown in Figure 4b.

We compared the overall effectiveness of the event delivery format (onsite vs online), and saw no significant difference in confidence, perception and interest. However, we noticed a 15% difference in gender equality perception in the onsite verses online format of the programming workshops. This suggests that online

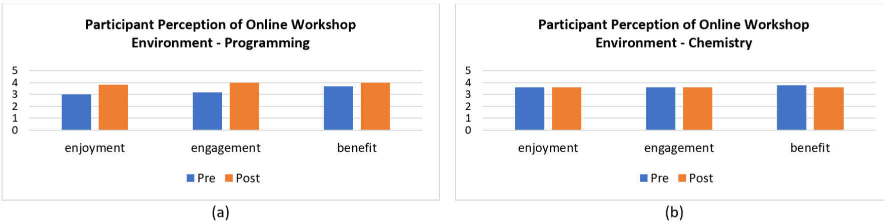

Figure 4: Participant perception of the workshops during the pandemic.

workshops negatively impact female confidence in their ability to program as compared to their male peers. 99% of the online participants indicated that they would recommend online workshops to their friends.

5 Discussion

While we found that online STEM events can be as effective as onsite, we are aware of limitations in our study. First, we acknowledge that our online study group is much smaller as compared to the onsite study group. Second, participants provided self-assessment data which could be under or overrated. Third, we did not ask specific questions about family income/parent education level in order to speculate about why our demographics shifted. However, this does not detract from the importance of our findings.

The main difficulties we faced during the pandemic were observed in enrollment/attendance and demographics and adjusting to online delivery.

Enrollment/attendance and demographics. Despite our efforts to recruit for the event, we were faced with our lowest ever attendance rate and a shift in our demographics during the pandemic. We believe this can be attributed to limited technology capacity among participants, screen fatigue, missing e-communications because of the overabundance of communications from schools and the change in the event's structure. However, attendance improved slightly in spring 2021 as people started to adjust to online learning.

Adjusting to online delivery. We observed different obstacles for the online programming and chemistry sessions. For the fall 2020 programming workshop, participants needed to pre-install required software on their computing devices. However, many were using school provided computers and Chrome books with limited privileges and technical capacity, not allowing them to configure their devices. We did learn from this experience and designed our spring 2021 workshop so that it did not require any software installations. However, in both cases, there were still technical difficulties in trouble shooting participants' code. We contribute these obstacles to the 6% decrease in the gender equality metric that girls can do just as well as boys in programming. On the other hand, the biggest challenge for chemistry sessions was designing experiments that the participants could do at home with household items that were safe and commercially available. Although packets were prepared for the participants to pick up in advance, several families said they could not come to campus to pick up those items, causing them either to not attend the workshop (affecting our attendance rate) or to just watch (contributing to the 5% decline in benefit).

We believe the importance of this study shows the impact of the COVID-19 pandemic on STEM outreach among underrepresented female minorities. We acknowledge that participant engagement is higher for in-person sessions, especially for chemistry since they can be more hands-on when in a lab. However, should future need arise, our study shows that online outreach events can be effective if special consideration is taken for workshop design and delivery.

6 Future Work

We will continue to offer S3 in both formats. In addition, we would like to offer the workshops for both male and female participants to compare enrollment patterns and performance in STEM.

References

[1] Brittany Bloodhart, Meena M Balgopal, Anne Marie A Casper, Laura B Sample McMeeking, and Emily V Fischer. Outperforming yet undervalued: Undergraduate women in stem. *Plos one*, 15(6):e0234685, 2020.

[2] United States Census Bureau. Women making gains in stem occupations but still underrepresented, 2021.

[3] J McGrath Cohoon and William Aspray. *Women and information technology: Research on underrepresentation.* The MIT Press, 2006.

[4] Nilanjana Dasgupta and Jane G Stout. Girls and women in science, technology, engineering, and mathematics: Steming the tide and broadening participation in stem careers. *Policy Insights from the Behavioral and Brain Sciences*, 1(1):21–29, 2014.

[5] Emma García and Elaine Weiss. Covid-19 and student performance, equity, and us education policy: Lessons from pre-pandemic research to inform relief, recovery, and rebuilding. *Economic Policy Institute*, 2020.

[6] Jennifer LaCosse, Denise Sekaquaptewa, and Jill Bennett. Stem stereotypic attribution bias among women in an unwelcoming science setting. *Psychology of Women Quarterly*, 40(3):378–397, 2016.

[7] Corinne A Moss-Racusin, Christina Sanzari, Nava Caluori, and Helena Rabasco. Gender bias produces gender gaps in stem engagement. *Sex Roles*, 79(11):651–670, 2018.

[8] Ryan Noonan. Women in stem: 2017 update. esa issue brief# 06-17. *US Department of Commerce*, 2017.

[9] American Association of University Women. The stem gap: Women and girls in science, technology, engineering and math, 2020.

[10] Committee on STEM Education. ERIC, 2018.

[11] Maria Ong, Janet M Smith, and Lily T Ko. Counterspaces for women of color in stem higher education: Marginal and central spaces for persistence and success. *Journal of Research in Science Teaching*, 55(2):206–245, 2018.

[12] Columbia University Outreach at Columbia Engineering. Girls who code, 2020.

[13] L Pier, HJ Hough, M Christian, N Bookman, B Wilkenfeld, and R Miller. Covid-19 and the educational equity crisis: Evidence on learning loss from the core data collaborative [commentary]. *Policy Analysis for California Education. edpolicyinca. org/newsroom/covid-19-and-educational-equity-crisis*, 2021.

[14] Katherine Rainey, Melissa Dancy, Roslyn Mickelson, Elizabeth Stearns, and Stephanie Moller. Race and gender differences in how sense of belonging influences decisions to major in stem. *International journal of STEM education*, 5(1):1–14, 2018.

[15] James Gresham Robin Donnelly, Harry A. Patrinos. The impact of covid-19 on education – recommendations and opportunities for ukraine, 2021.

[16] Steven J Spencer, Christine Logel, and Paul G Davies. Stereotype threat. *Annual review of psychology*, 67:415–437, 2016.

[17] UNICEF et al. Covid 19 and school closures. *One year of education disruption. UNICEF*, 2021.

[18] Johanna Vennix, Perry den Brok, and Ruurd Taconis. Do outreach activities in secondary stem education motivate students and improve their attitudes towards stem? *International Journal of Science Education*, 40(11):1263–1283, 2018.

Experiential Learning and Job Experience: Opportunities for Cybersecurity Curriculum Support, Cyber Graduate Marketability, and Cyber Productivity *

Kenneth L. Shemroske and Kasra Pourang
Management and Information Sciences Department
University of Southern Indiana
Evansville, IN 47712
{k.shemroske,kpourang}@usi.edu

Abstract

The complex technical nature of cybersecurity jobs in the marketplace requires graduates of cybersecurity programs to have skills and experience beyond the classroom. Specifically, employers look for individuals who have industry–relevant experience. In this study, 152 educational institutions were researched for details about their cybersecurity programs. Experiential learning theory and workforce alignment frameworks were evaluated as tools for designing, implementing, and evaluating mechanisms which can address the experiential learning and job experience opportunities missing from cybersecurity programs. A detailed example of how a security operations center (SOC) is used to address experiential learning and job experience is evaluated using an approach combining Kolb's Experiential Learning Theory (ELT) and the National Institute of Standards and Technology's framework called the National Initiative for Cybersecurity Education (NICE). Finally, conclusions are drawn which give guidance for adding experiential learning and job experience mechanisms to cyber curriculum support where it may be lacking.

1 Introduction

Cybersecurity is a field which continues to grow in scope, stature, and importance. Workforce development coalitions have formed around the world in attempts to address the increasing gap between cyber skilled talent and industry demand [3]. While academia seeks to do its part to address the gap, opportunities present themselves which might further align academic programs with the requirements of industry.

This research paper presents a study of 152 universities, the Cybersecurity programs offered, and support for those programs outside of the classroom. While curriculum would be a basic requirement for all programs, this study looks beyond the classroom environment for opportunities that can give students 'hands-on' experience with the tools, techniques, and concepts of cybersecurity. Further, it looks for ways in which students might build industry–relevant experience prior to completion of their degree.

The work of cyber analysts proves to be complex and challenging. To this point, the need for exposure to 'real-world' situations in which to develop skills is critical and must be introduced early [2]. The research in this paper suggests there is ample opportunity to improve on this approach across academic programs through experiential learning mechanisms and job experience components.

The following sections of this paper review extant theory and literature on experiential learning and cybersecurity industry–relevant frameworks pertaining to job experience. Approaches to addressing this theory and practice that are in use by the university programs from our data set are then broken down and discussed. The use of a Security Operations Center (SOC) as one tool for both experiential learning and job experience components is discussed in further detail as an example from the University of this paper's authors. Experiential learning mechanisms are identified and categorized and the same is done for job experience mechanisms. Finally, conclusions are drawn, and some recommendations are made based on the findings of this research.

2 Literature and Theory

Experiential Learning: It is helpful, if not critical, to have a frame of reference and a framework for evaluating a support platform like experiential learning. The extant literature on experiential learning and theory was reviewed to find bases for analysis. The following are theories and guidance which were dominantly referenced or used as foundations for incremental theory.

Experiential Learning Theory (ELT) is defined as "the process of creating knowledge through experience" [9]. Experiential learning is also defined as "learning through experience, discovery, and exploration" [5]. ELT focuses on the "process" of learning rather than the "outcome" of learning [13]. ELT has six essential pillars [1, 8]:

1. Students make the decisions and take the responsibility of the results. Students must receive feedback on the effectiveness of their decisions.
2. Students are engaged in a dynamic learning environment. The learning process enables the students to think critically, be creative, and come up with new ideas.
3. Students learn to cooperate and work with each other. The learning process helps students understand disagreement and reflect on the critical decisions.
4. Students learn to adapt to different situations. The learning process engages students intellectually, emotionally, and socially.
5. Students learn how to react to different outcomes. The learning process provides opportunities for the students to succeed, fail, or face uncertainty and helps them react appropriately.
6. Students learn to produce their personal knowledge. The learning process is dynamic and enables the students to experience the knowledge they need to learn.

Students who engage with experiential learning go through a five–stage process. In each of the following stages, they understand how to self–reflect, learn, and produce new knowledge [8]:

Explore: In this stage, the learning environment engages students in hands–on projects. Students learn what to "do" and how to "do" it. *Reflect:* In this stage, students study the results of the Explore stage and try to find out what happened during the Explore stage. The learning environment enables the students to share their knowledge and feelings with other students. Analyze: In this stage, the learning environment enables the students to share how they succeeded, the steps they took to address possible challenges, and how they would use their knowledge in future situations. *Generalize:* In this stage, students find real–world examples of their newly acquired knowledge. Apply: Students apply their new knowledge to the similar situations. The learning environment enables students to discuss the steps they would take in a possible similar future scenario.

Roberts, Conner, and Lynn Jones expanded the Kolb's theory of experiential learning and developed a framework for engaging learners during study abroad programs [15]. They argue that the process of experiential learning needs to be applied at three stages:

Before an experience: Educators should prepare learners for the experience (pre–flection). At this stage, educators should focus on providing a safe learning environment, assess the learners' knowledge about the experience, and personalize the learning process for each learner. *During an experience:* Educators should design problem–solving activities and prepare the learners for reflection. At this stage, educators should personalize the activities to each learners' learning speed. *After an experience:* Educators guide the learners through reflection process and provide feedback. At this stage, educators design post–experience activities that motivates the learners to discuss what they learned.

Dewey's model of experiential learning has multiple stages that starts with the experience and ends with application [4]. Dewey believed that a "meaningful" and "concrete" experience is the result of an interaction between the learner and the learning environment [4]. Thus, he defined each step of his model as follows:

Concrete Experience: The interaction between the learner and the learning environment. *Reflection:* The action(s) to solve the problem provided by the learning environment. *Abstract Conceptualization:* Combining the newly generated knowledge from the experience with previous knowledge. *Application:* Applying the new knowledge to the situations similar to the experience.

Itin developed Dewey's model and proposed the Diamond Model in which the learner, the educator, and the learning environment interact with each other to create a meaningful experience [7]. The learning environment in Itin's Diamond Model facilitates the transfer of information between the learner and the educator in both ways. Itin believed that simply providing real–world examples cannot promote the process of experiential learning. The learning environment must be designed to engage the learners and the educators intellectually, socially, emotionally, and physically.

Job Experience: The importance of job experience and its relevance to the field of cybersecurity has been an increasingly urgent topic of conversation. While this is true of many applied sciences, both the complexity and detailed skills needed for cybersecurity jobs make appear to escalate the need. Additionally, many employers will not consider a candidate that doesn't have practice with the desired skills because they don't have the time or resources for the steep learning curve that comes with many cybersecurity positions [10]. As a matter of practicality, and to add value to any cybersecurity academic program, providing job experience as part of the program will give graduates opportunities that might not otherwise have in the job market.

A popular way to accomplish this task is to have students participate in internships at positions of their chosen field, in this case cybersecurity. This may be desirable in that it takes the responsibility off of the academic institution. A challenge can be the supply of internships available. As mentioned previously, employers tend to lose interest in untrained talent. This can be a hurdle that is difficult to overcome.

Providing work experience that is within the domain of the academic institution could be a solution to this limitation. This does not come without its own set of challenges, perhaps the largest of which would be assuring alignment between the provided activities and industry needs. Fortunately, there are frameworks that have been developed which can guide both curriculum and support activities to help with alignment.

The most widely referenced of these frameworks comes from the National Institute of Standards and Technology (`www.nist.gov`). NIST works to author and maintain a framework used by many academic institutions as guidance for workforce alignment. The framework is called the National Initiative for Cybersecurity Education (NICE) [12]. This framework identifies categories of cybersecurity activities that make up both overarching divisions and detailed specializations within the broad scope of the cyber industry. Further, this framework breaks down the skills, knowledge, and abilities needed to perform specific tasks that can fall under various job roles within the cybersecurity domain.

Other frameworks are widely recognized and may be leveraged for similar purposes. The International Standards Organization published ISO 27000 which defined risk, threats, and counteractivities for cybersecurity [14]. The (ISC)2 created a framework identifying ten domains for grouping the behavioral disciplines associated with cybersecurity activities [17]. These domains span architecture and access control to social, legal, and ethical aspects. The U.S. Department of Homeland Security created the Cyber Resilience Review (CRR) for assessing and categorizing cybersecurity capabilities specific to critical infrastructure [16]. The CRR also identifies ten units it defines as resource sectors. These focus primarily on management and training and range from asset management to situational awareness.

3 Data and Analysis

Data was collected on 152 schools which provide cybersecurity education. This sample of schools was derived through cybersecurity reference mechanisms which offered lists of schools providing curriculum (e.g., `sans.org`) [6] and popular media outlets which did the same (e.g., `usnews.com`) [11]. This was combined with various internet searches for similar lists from independent providers until enough repeated school names suggested reasonable saturation. Table 1 shows aggregate data regarding program offerings and other curriculum support across the sample.

While our analysis focused on the data in Table 1, some other numbers of interest include the total number of programs (associate, bachelor's, master's,

Table 1: Aggregate Cyber Program Data

Schools that provide a bachelor's degree in cybersecurity master's degree in cybersecurity PhD in cybersecurity associate degree in cybersecurity	Schools that provide a Schools that provide a Schools that provide an		
82	73	8	47
Schools that host a cybersecurity bootcamp undergraduate certificate in cybersecurity graduate certificate in cybersecurity learning activities/environments outside of the classroom	Schools that provide an Schools that provide a Schools that provide experiential		
25	61	41	33
Schools that provide/include industry experience mechanisms in their cybersecurity programs cybersecurity organizations cybersecurity competitions National Security Agency (NSA) Center of Academic Excellence in Cybersecurity designation	Schools that host/support student Schools that compete in Schools that have a		
15	50	38	56

and PhD) across all schools was 287. The total number of certificates offered across all schools was 211.

The data show strong support in both bachelor's and master's programs with 54% and 48% participation respectively. Associate programs were found in 31% of the sample. Some four–year schools did include associate degrees in their offerings, but most of the associate programs were provided by two–year schools in the sample. PhD programs appear to be largely neglected at only 5% participation across the sample.

Experiential Learning: There are two main areas of investigation in this paper. The first of these is experiential learning, already established as instrumental for engagement and retention of knowledge, especially in technical disciplines. For the purposes of our research, experiential learning is defined as any activities which transpire outside of the classroom, giving students the ability to work with cybersecurity technologies to accomplish some type of task. They should fit with one or more of the pillars suggested by Kolb's ELT. These tasks could range widely, from simple tools to in depth research to training for cyber competitions.

In our sample, 22% of the schools which provided cybersecurity curriculum had information about experiential learning environments, tools, or mechanisms that were available to students.

After evaluating the sample for content, we found experiential learning support was able to be categorized into six broad categories:

- **Centers for Cybersecurity Research and Analysis:** Serve as a hub for training, research, analysis, and programming in all areas of cybersecurity.
- **Cyber/Hacking Labs/Ranges:** Provide research opportunities and use hacking resources to teach students how to fight against malwares and other security disruptions.
- **Competitions:** Provide students with multiple challenging security scenarios to prepare the students for real–world security challenges.
- **Bootcamps:** Often held by a third party, bootcamps present the in–demand topics in a dense accelerated format to equip the students with the required skills for placement in the workforce.
- **Student Clubs:** Connect students to alumni and industry partners and deepen students' knowledge by holding speaker series and networking sessions.
- **Security Operations Centers (SOC):** When available to students, SOCs provide an opportunity for the students to work with information security teams. Students will be responsible for monitoring, detecting, and analyzing security issues.

Job Experience: The second main area of investigation in this paper is the incorporation of job experience activities while a student is working on their program. This also has been established as increasingly important to employers when evaluating candidates for open cyber positions. Providing opportunities to gain this experience prior to completion of a program not only makes students more marketable, but it establishes abilities which make them productive much sooner, if not immediately, upon hire.

In our sample, 10% of the schools which provided cybersecurity curriculum hosted or required some form of activity which would represent working experience for the student prior to completion of the program.

After evaluating the sample for content, we found job experience support was able to be categorized into five types of mechanisms:

- **Internships:** Designed as credit or non–credit courses, internships provide first–hand experiences for students to engage with real security challenges. Internships are often part–time, and students are responsible for finding a cybersecurity–related internship position.

- **Capstones:** Since not all students can find internship positions, capstones are designed and integrated into curriculum to provide all students with an opportunity to engage with industry partners and work on cybersecurity–related projects for a grade (instead of financial compensation).

- **Co–Ops:** Like internships, co–ops provide cybersecurity–related work for the students in prearranged agreements with the university. Co–ops are often full–time, and the students are not expected to take any courses during their co–op experience.

- **Project work:** When available, the NSA or other governmental/cyber consortium agencies offer cybersecurity projects to university partners. Students can work with agency staff as team members on live projects.

- **SOCs:** Some schools provide internship or co–op opportunities through SOCs which protect their campus infrastructure.

3.1 Using a Security Operations Center (SOC) for Experiential Learning

The authors of this paper hold positions at a university where a SOC has been built and managed by faculty and is run by students pursuing a cybersecurity certification within their curriculum. The students are paid as student workers with a job title of 'Security Analyst'. They serve roles helping to protect the university's information.

The primary function of a SOC is to employ resources which monitor the security posture of an organization, detecting, analyzing, and responding to security threats. However, the intentions behind building this SOC included using it as a tool for 'hands–on' experiential learning and to give students an opportunity to develop job experience prior to the completion of their program. Fortunately, this was a shared vision between faculty sponsors and the head of IT for the university.

Using ELT, the SOC represents an effective mechanism for experiential learning along each of the six pillars:

1. **Students make the decisions and take responsibility:** Daily operation of the SOC requires each student to process the information they have available to them and decide about what should be done. Because

of the close relationship the SOC analysts have with the IT department, this may mean turning activities over to them. But it may also mean the student analyst decides to handle the issue within their own boundaries.

2. **Students are engaged in a dynamic learning environment:** By nature, the cybersecurity realm is fraught with opportunities to test one's creativity and innovative spirit. Some threats may be well known, and tools are available to respond readily to them. Others are new, evolving, or simply little is known about them. This forces a cyber analyst to use creativity and innovation to keep the organization safe or at least minimize consequences from a threat.
3. **Students learn to cooperate and work with each other:** In the SOC environment analysts must work closely with each other to avoid duplicating effort or working against each other. In some cases, issues may be turned over from one analyst to another or one shift to another. Transfer of not just information, but knowledge is critical to effective solutions. Further, because some levels of responsibility lie with the IT group, cooperation extends outside the SOC.
4. **Students learn to adapt to different situations:** Here again, due to the diverse and ever–changing nature of cybersecurity as a function and as an academic discipline, student analysts are forced to learn new concepts, tools, and techniques while they are working to resolve or otherwise respond to incidents. This SOC is also unique in that it must function differently from a corporate SOC. The nature of the student worker positions means inherent turnover. This requires the student analysts to work with a steep learning curve. Additionally, trust issues, levels or responsibility, and augmented operations means responsibilities may change during their tenure in the position.
5. **Students learn how to react to different outcomes:** Because student worker positions come with an inherently high rate of turnover (i.e., students finish their degrees and move on), there is a carefully constructed separation of duties between the IT staff and the SOC analysts. While failures in the SOC positions are not necessarily desirable, the position is engineered such that they should not be mission critical. This means there are opportunities to learn from failures as well as successes without aggressive repercussions.
6. **Students learn to produce their personal knowledge:** There is always the hope that everything a student might need to learn about a discipline might be learned in the classroom. This is seldom seen in practice. Working in the SOC allows students the opportunity to reinforce, discover, or relearn concepts, tools, and techniques covering cyber operations.

3.2 Using a SOC for Gaining Industry Experience in Cybersecurity

As mentioned previously, a popular framework for aligning cyber workforce needs with educational mechanisms is provided by the National Institute of Standards and Technology in the form of its National Initiative for Cyber Education (NICE) [12].

The SOC aligns well with three of the main constructs of the NICE framework:

1. **Corresponding to a category:** The NICE framework defines workforce categories for the overarching themes related to various cybersecurity operations necessary for every organization. Our SOC analyst duties fit well into the 'Protect and Defend (PR)' category provided by the framework [12]. Identifying, analyzing, and mitigating threats stand as the key job responsibilities of the SOC analysts.
2. **Addressing specialty areas:** Work in the SOC allows students to focus on any one to all of three areas of specialization identified in the NICE framework. A variety of tools and logs give the analysts the resources need to perform Cyber Defense Analysis (CDA) to identify currently exploited or potentially exploitable breaches in defense. Levels of responsibility and corresponding access give analysts the ability to provide Incident Response (CIR) and mitigate active or potential threats. Finally, tools are available to the analysts which allow them to audit certain portions of network resources and perform Vulnerability Assessment and Management (VAM) tasks [12].
3. **Integrates knowledge, skills, and abilities into a defined work role:** The culmination of the above responsibilities along with many of the tasks, skills, knowledge points, and abilities identified in the NICE framework suggest the appropriate industry relatable role for our student cyber analysts as 'Systems Security Analyst' [12].

Using ELT as an evaluative tool and NICE as a workforce alignment framework, the SOC evaluates well as an extracurricular mechanism which serves well to give students experiential opportunities which translate to job experience in the cybersecurity industry.

4 Conclusions and Limitations

Conclusions: Our study identifies a gap in the completeness of academic programs in cybersecurity. This gap exists because industry demands, mixed with the complexity and technical nature of cybersecurity roles, suggest that experiential learning opportunities which translate well to job experience are critical to the graduate's success in the job market. Yet, a low percentage of available academic programs provide such experiences.

The analysis of cyber programs at 152 schools suggests there may be a need for some guidance in this area. Combining a theoretical approach to experiential learning with a framework for workforce alignment provides a powerful tool set for designing, implementing, and evaluating mechanisms which can support curriculum in these much–needed capacities. The example of the SOC at the authors' university shows how this may be done in such a way that a single mechanism may provide both experiential learning and job experience simultaneously.

The analysis in this paper included the categorization and description of both experiential learning and job experience mechanisms that are currently in use. This may be used on its own as direction for programs that are seeking guidance supporting their curriculum outside of the classroom.

Limitations: There were some possible limitations of this study that should be considered. While the internet searches and indexed resources that were used for this study's sample were not restricted in any way, the results were exclusively schools found in the United States. Extending this analysis to international programs might yield additional information for further analysis.

All information used in this study was publicly accessible information available through the internet. It is possible that some programs haven't published or updated information which could have been relevant to this study.

Finally, one data point that was captured concerned institutions that had a Center of Academic Excellence (CAE) designation from the National Security Agency and the consortium it uses to issue these designations. The designation means the institution met certain standards in their curriculum and/or research which corresponded to criteria from the consortium. Included in the members of the consortium is NIST and the NICE framework. This would suggest institutions with the certification are in alignment with workforce demands. It is possible that experiential learning and/or job experience components are incorporated in classroom activities but simply not published as such to the public.

References

[1] AEE. Association for experiential education. http://www.aee.org.

[2] R. Beveridge. Effectiveness of increasing realism into cybersecurity training. *International Journal of Cyber Research and Education (IJCRE)*, 2(1):40–54, 2020.

[3] T. Coulson, M. Mason, and V. Nestler. Cyber capability planning and the need for an expanded cybersecurity workforce. *Communications of the IIMA*, 16(2):2, 2018.

[4] J. Dewey. *Experience and Education*. Collier, New York, 1938.

[5] Northern Illinois University Center for Innovative Teaching and Learning. Experiential learning. in instructional guide for university faculty and teaching assistants. https://www.niu.edu/citl/resources/guides/instructional-guide.

[6] SANS Institute. https://www.sans.org.

[7] C. Itin. Reasserting the philosophy of experiential education as a vehicle for change in the 21st century. *Journal of Experiential Education*, 22:91–98, 1999.

[8] A. Y. Kolb and D. A. Kolb. Learning styles and learning spaces: Enhancing experiential learning in higher education. *Academy of Management Learning and Education*, 4(2):193–212, 2005.

[9] D. A. Kolb. *Experiential learning: Experience as the source of learning and development*. Prentice–Hall, New Jersey, 1984.

[10] Brian Krebs. Thinking of a cybersecurity career? read this. https://krebsonsecurity.com/2020/07/thinking-of-a-cybersecurity-career-read-this/.

[11] U.S. News and World Report. http://www.usnews.com.

[12] NIST. National institute of standards and technology. https://www.nist.gov.

[13] University of California Davis (UC Davis). 5–step experiential learning cycle definitions. http://www.experientiallearning.ucdavis.edu/module1/el1_40-5stepdefinitions.pdf.

[14] K. Prislan and I. Bernik. Risk management with iso 27000 standards in information security. *Inf. Secur*, 2010.

[15] T. G. Roberts, N. W. Conner, and B. Lynn Jones. An experiential learning framework for engaging learners during study abroad experiences. *North American Colleges and Teachers of Agriculture*, Special Issue:28–35, 2013.

[16] R. Sabillon, V. Cavaller, and J. Cano. National cyber security strategies: global trends in cyberspace. *International Journal of Computer Science and Software Engineering*, 5(5):67, 2016.

[17] M. Theoharidou, E. Stougiannou, and D. Gritzalis. A cbk for information security and critical infrastructure protection. *IFIP World Conference on Information Security Education*, pages 49–56, 2007.

Autonomous UAV Navigation in a GPS-Denied Outdoor Environment Using Discontinuous Visual Contact With Fiducial Markers*

Auburn REU on SMART UAVs 2021 Technical Report#CSSE21-01

Nicholas Dunn[1] and Zachary Schallenberger[2]
with Saad Biaz[3] and Richard Chapman[3]
[1]Computer Science & Mathematics
Columbus State University, Columbus, GA 31907
dunn_nicholas2@columbusstate.edu
[2]Computer Science
Montana State University, Bozeman, MT 59717
z.schallenberger@student.montana.edu
[3]Computer Science and Software Engineering
Auburn University, Auburn, AL 36849
{biazsaa, chapmro}@auburn.edu

Abstract

Autonomous navigation of Unmanned Aerial Vehicles (UAVs) has become an increasingly popular area of study as technology has advanced over the years. One technology that has been developed for use in the autonomous navigation of UAVs is fiducial markers. Fiducial markers are a type of physical tag that can be placed in an environment to assist with UAV localization, navigation, and landing. A UAV can identify and scan a fiducial marker with an onboard camera and take appropriate action based on the data received from the marker. Previous research using only fiducial markers and onboard sensors for UAV navigation in a GPS-denied environment requires the camera to have constant visual contact with at least one marker. This paper explores the use of fiducial markers for outdoor UAV navigation in a GPS-denied environment without the restriction of constant visual contact with a marker. Physical implementation and formal methods are used to study the performance and overall viability of this solution.

1 Introduction

1.1 Context and Terminology

An **Unmanned Aerial Vehicle** (UAV) is an aircraft vehicle that does not have any pilots or passengers on board. It is often informally referred to as a drone. A UAV can be controlled in many different ways but this paper exclusively explores the autonomous control of these vehicles. A **fiducial marker** is a known shape or pattern that is typically printed onto a flat surface. Similar to a QR code, a fiducial marker can provide data via camera recognition. The data obtained from a fiducial marker can take the form of pixel coordinates to track the four corners and center of a fiducial marker and provide a unique identification number associated with each fiducial marker. **AprilTag** is a type of fiducial marker system widely used with UAVs. AprilTags have demonstrated consistent performance (Kalaitzakis *et al.* [2]) in the past and are the fiducial marker family that was chosen for this research paper. The detection software for AprilTags can compute the 3D positioning, orientation, and identification of the markers from an adequate distance relative to their size. AprilTags can be decoded to provide navigational data to a UAV as a set of predefined instructions assigned to the corresponding AprilTag ID. A **GPS-denied environment** is an environment in which GPS technologies are unable to function due to exterior circumstances. **Autonomous navigation** is the execution of some plan or path without any human input or control along the way. Autonomous navigation can be paired with computer vision software to assist the program in its decision-making process. One computer vision library is **OpenCV** (Open Source Computer Vision Library): an open-source software library that provides access to tools essential for computer vision and machine learning. OpenCV is used in conjunction with the AprilTag software to assist a UAV in properly visualizing and decoding an AprilTag. **Pose** refers to the position and orientation of an object. Specific to this research, pose describes the translation of a UAV relative to a marker or particular coordinate system in terms of pitch, roll, and yaw. A "**dead zone**" is a term used in this paper to describe the period in which a UAV is autonomously flying without the detection of a fiducial marker or using any other form of a visual or GPS-based navigation system.

1.2 Problem

GPS-based navigation is an incredibly common form of navigation that many devices rely on. UAVs typically rely on GPS to fulfill autonomously driven missions. However, GPS-based technology faces many different risks as a chosen form of navigation. GPS is susceptible to spoofing and environmental interference. Furthermore, GPS does not function well in indoor environments and cannot be relied on for precision navigating. Due to the potential vulnerabilities of GPS-based navigation, there is a need for research into other forms of navigation for autonomously flown UAVs.

1.3 Solution

This research paper explores the idea of a form of visual-based navigation that centers around fiducial markers. The research analyzes how a UAV could autonomously navigate a predefined course using fiducial markers as a form of instruction in a GPS-denied environment while relying on autonomous flying algorithms in between discontinuous visual contact "dead zones" of the fiducial markers. This solution will mitigate any concern of a failure of onboard GPS-based navigation of a UAV.

1.4 Evaluation of Methods

This paper will be evaluating the research through a few key metrics:

1. The research will measure how well a UAV can read an AprilTag through a program using OpenCV to process a live feed of a camera attached to the UAV and return data showing the detection areas of a fiducial marker. The research will measure the success by conducting tests that involve flying the drone at arbitrary locations with an AprilTag within view of the camera and measuring the success rate at which it can detect a marker.

2. The research will measure how well a UAV can decode a fiducial marker as a set of instructions. This will be done by setting up a test involving multiple fiducial markers placed at a variety of distances away from a camera. The measurement of success will be determined by the UAV's ability to complete a set of instructions received from decoding a fiducial marker's unique identification number.

3. The research will measure how well a UAV can navigate between fiducial markers using autonomous navigation by evaluating the performance of a UAV on courses where it will be required to navigate through substantial "dead space" between AprilTags relying solely on autonomous navigation.

2 Related Work

Previous research has studied the use of fiducial markers for assisting UAVs in localization, navigation, and landing in both GPS-enabled and GPS-denied environments. Some of the methods used have incorporated other technologies as well, such as accelerometers, magnetometers, and gyroscopes. Both indoor and outdoor environments have been used for different applications. Different types of fiducial markers have also been tested.

Kalaitzakis *et al.* [2] performed a survey of different fiducial markers and conducted studies comparing four of the most widely used fiducial marker families. The markers that were chosen for the study were ARTag, AprilTag, ArUco, and STag. A comparison of the accuracy of detection, detection rate, and computational cost of each marker was conducted. Experiments with the markers used different lighting conditions, camera angles, and distances between the camera and the markers. Each marker had strengths and weaknesses, but the AprilTag and STag markers had the overall best performance in the study.

Indoor environments provide unique navigational challenges for UAVs due to the interior features of buildings. Nahangi *et al.* [4] used fiducial markers for automated UAV localization of a GPS-denied indoor construction environment. Their experiment used a Parrot Bebop 2 drone with an onboard camera to detect AprilTag fiducial markers. The drone also used an accelerometer to assist with localization and navigation. The AprilTags were linked to previously known coordinates in the building information model (BIM), and the UAVs were localized using the information from the AprilTags and their position relative to the onboard camera. Previously computed information could then be used to successfully navigate the environment; however, the UAV was required to detect at least one AprilTag at all times.

A study conducted by Bacik *et al.* [1] used ArUco fiducial markers to assist with UAV navigation. The researchers used a Parrot AR Drone with two cameras to detect the markers. They also used an ultrasound, pressure altimeter, gyroscope, accelerometer, and magnetometer. Their method of navigation required the $(n\text{-}1)^{\mathrm{th}}$ marker to be detected by the camera at the same time the n^{th} marker was detected. They then calculated the pose of the n^{th} marker with respect to the $(n\text{-}1)^{\mathrm{th}}$ marker and used that information to keep track of the path of the UAV. A fuzzy control algorithm used the calculated pose and trajectory of the UAV to navigate it from one marker to the next. The UAV achieved a fully autonomous flight in an empty warehouse using only marker-based navigation combined with a fuzzy control algorithm. However, this is another method that requires at least one marker to be detected at all times.

Lamberti *et al.* [3] combined ARTag fiducial markers with image feature matching to navigate UAVs. This method of navigation used image matching to estimate the pose of the UAV between markers. It then used markers to determine the true position of the UAV and to correct any drift that occurred. The marker-less pose estimation of the UAV was performed by frame-to-frame comparisons of the features in the images captured. After an initial image was captured, each additional image underwent a feature analysis comparison against the initial image and pose estimation of the UAV. A distance threshold between the UAV and the initial image was set and after the threshold was passed, a new reference image was obtained, in which each additional image was compared for pose estimation. A control algorithm was implemented to maintain the pose of the UAV. Whenever a fiducial marker was detected, a marker-based pose estimation algorithm was executed. These algorithms corrected the drift errors that accumulated during the marker-less navigation. The researchers concluded that this approach could maintain an acceptable level of drifting errors while navigating UAVs.

Other studies have focused on using fiducial markers solely for pose estimation. Seng *et al.* [6] used fiducial markers to estimate the pose of a Parrot AR Drone in a GPS-denied environment. Their experiments were conducted in an indoor room that had fiducial markers placed on a wall. The ARToolKit software library was used to help compute the pose of the UAV in relation to the detected markers. At least one marker had to be detectable to estimate the pose of the UAV. However, the accuracy of pose estimation increased as the number of markers detected increased. This was a result of computing the average pose of the UAV based on the poses of all detected markers.

Although different technologies and environments have been used for autonomous UAV navigation, the methods that only use fiducial markers and onboard sensors require continuous detection of at least one marker by an onboard camera. The purpose of this paper is to explore autonomous UAV navigation using only fiducial markers and onboard sensors, without the requirement of continuous detection of a marker in an outdoor, GPS-denied environment.

3 Methodology

To develop a system for autonomous navigation of UAVs using only fiducial markers and onboard sensors, a suitable fiducial marker system, AprilTag, was selected. After selecting a fiducial marker system, a Raspberry Pi 3B with a Raspberry Pi Camera Module v2 was obtained and a Python program was developed to detect the AprilTags. Next, two markers were printed, and experiments were conducted to detect them with the camera. The next step was to select a UAV to use for the research: the Parrot AR.Drone 2.0. After selecting the drone, several Python programs were developed to control the drone's movements, onboard sensors, and cameras. Several flights were then successfully executed using the programs. Next, the Raspberry Pi was mounted onto the hull of the UAV and the camera was positioned towards the front of the UAV at a downward-facing angle. Python programs were later developed and tested to align the UAV with a detected fiducial marker. A program was developed to navigate the UAV from one marker to the next using only onboard sensors to minimize drift errors. Then, the previously computed position of each marker was stored as data with the previous marker which enabled the UAV to adjust its heading to navigate towards the next marker.

3.1 Fiducial Markers

To select a fiducial marker family, some of the most widely used packages in autonomous UAV navigation were investigated and it was decided to use AprilTags. Kalaitzakis *et al.* [2] performed a comparison of the ARTag, AprilTag, ArUco, and STag markers and found that AprilTag and STag had the overall best performance in their study. Additionally, AprilTags were also found to be used in several other papers involving UAV navigation, such as the research done by Nahangi *et al.* [4]

AprilTags were developed and introduced by Edwin Olson [5] to improve upon other markers that existed at that time. It was noted that AprilTags provide "a fast and robust line detection system, a stronger digital coding system, and greater robustness to occlusion, warping, and lens distortion." [5] Since then, AprilTag has undergone several generations of development. The AprilTag fiducial marker system offers several different families of tags to choose from based on the application. The AprilTag software is also open-source and readily accessible.

3.2 Unmanned Aerial Vehicle

The UAV selected for this research was the Parrot AR.Drone 2.0. The Parrot AR.Drone 2.0 is a quadcopter with a protective hull. It comes equipped with both forward and downward-facing cameras as well as several sensors, including an ultrasound altimeter, three-axis accelerometer, magnetometer, and a gyroscope. There is also a rechargeable battery capable of twelve minutes of flight time. The UAV generates its own Wi-Fi network allowing for a multitude of devices to connect and control the drone. The Parrot AR.Drone 2.0 provides a software development kit (SDK) for developers and has been used in many different studies involving UAVs, including research done by Bacik *et al.* [1], Lamberti *et al.* [3], and Seng *et al.* [6]. The application programming interface (API) enables developers to access and control the cameras, sensors, and movement of the drone. Other APIs have also been developed for the Parrot AR.Drone 2.0, such as PS-Drone. For this research, the PS-Drone API was chosen due to the functionality it provides and its ease of use. PS-Drone provides a simple library of commands that can be used in a Python program to control the movements of the drone and acquire valuable data from its sensors.

3.3 Autonomous Navigation

Autonomous navigation of the UAV was achieved using a combination of its onboard sensors and AprilTag fiducial markers. The position of each AprilTag with respect to the previous was computed and stored as positional data. At initial startup, the drone increased its altitude to a height greater than one meter to expand the camera's field of view. After adjusting its altitude, the UAV proceeded to fly forward while using the Raspberry Pi camera to scan for an AprilTag in the environment. When a new marker was detected, the data associated with the marker was used to acquire the positional data of the next marker. The UAV then used that data to adjust its heading to navigate in a straight-line path towards the next marker. Once navigation began, the drone used only its onboard sensors to minimize drift errors until the next marker was detected by the camera. When the next marker was detected, the data associated with the marker was once again used to acquire the positional data of the next marker. This process was repeated until the drone fully navigated the desired path.

3.4 Wireless Control

All the programs implemented in the system were written in Python and executed on the Raspberry Pi 3B using the PS-Drone API to communicate with the Parrot AR.Drone 2.0 on its Wi-Fi network. Secure Shell (SSH) was used to enable control of the Raspberry Pi from a remote computer connected to the drone's Wi-Fi network. The Wi-Fi network provided by the drone functioned as a bridge between the Raspberry Pi and the remote computer. The remote computer was used to execute all of the programs on the Raspberry Pi.

3.5 Indoor Experiments

The majority of the tests were performed on an indoor course to improve the algorithms. To begin, programs were developed to detect AprilTags from images captured by the Raspberry Pi Camera. A program was developed using OpenCV to help visualize the position of the center of the tag with respect to the center of the camera. This program overlays an 8x8 grid on the image and highlights grid-squares that detect an AprilTag. The program also displays a straight line from the center of the view to the center of the AprilTag. Next, programs were implemented using the PS-Drone API to access and control the sensors, cameras, and movement of the Parrot AR.Drone 2.0. Tests were conducted to acquire real-time sensor data for the altitude and orientation of the drone as well as accessing its front and ground cameras. The drone's battery performance and Wi-Fi link quality were also evaluated through tests developed in Python. An external Adafruit LSM303DLHC triple-axis accelerometer/magnetometer was also incorporated into the study as a supplement for the drone's onboard sensors. The LSM303DLHC was wired to the GPIO pins on the Raspberry Pi and tests were performed to acquire real-time sensor data while manually manipulating the drone. Programs were then implemented on the Raspberry Pi using the PS-Drone API to demonstrate motion control of the drone. Commands were executed to control the drone's altitude, position, orientation, and speed. Override controls were also implemented to enable remote initiation of a safe landing sequence when necessary.

After demonstrating successful control of the drone's movements and sensors, the main flight program for the system was developed. The Raspberry Pi, camera, and external accelerometer/magnetometer were attached to the drone. A flight course for the UAV was then created by taping two AprilTags to the floor at different locations in the lab. A compass was used to determine the angle of the second marker relative to the first. This angle was stored in the program with data for the first marker as positional data for the second marker. The program was executed to test the system. After many tests, the code was refined. Programs were also developed to read in and plot the yaw angle and magnetometer values of the drone during both real and simulated flights. Simulated flights were accomplished by manually manipulating the drone through the course.

3.6 Outdoor Experiments

An outdoor environment was chosen to conduct the final tests of the system. AprilTags were taped to the top of two separate boxes placed apart. The boxes were positioned in a large grassy area to create an outdoor flight course for the UAV similar to the indoor course. The flight program was then executed to test the system and the program was, again, refined.

4 Results

4.1 Marker Detection and Decoding

Marker detection and decoding capabilities were initially evaluated before mounting the hardware to the UAV. The position of the camera was manipulated at different angles to verify that the Python program could successfully detect and decode AprilTags. Marker detection and decoding performed quite well

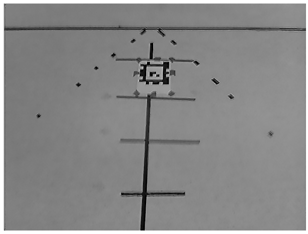

Figure 1: Picture taken of detected AprilTag during successful indoor flight

even while the camera was in motion. The camera was able to successfully detect and decode the AprilTags under all circumstances if they were within the camera's field of view.

After the Raspberry Pi and Raspberry Pi Camera Module v2 were attached to the UAV, marker detection and decoding were evaluated again. Since the Raspberry Pi was attached to the UAV, the pictures taken by the camera were saved to the Raspberry Pi for processing later. A VNC viewer was also implemented by a remote computer to access the graphical interface of the Raspberry Pi and obtain a live feed from the camera during flights. In-flight pictures were taken while the UAV was navigating towards the markers. Similar to the initial marker detection and decoding experiments, the camera was successfully able to detect and decode the AprilTags if they were fully within the camera's field of view. Marker detection and decoding were found to be quite robust even during active flight.

4.2 Hardware Performance

4.2.1 UAV Hardware

The evaluation of the performance of the UAV's hardware began by testing the onboard sensors, cameras, and battery life under simulated flight conditions before any external hardware was attached to the UAV. A Python program was implemented to access and display the values of the UAV's pitch, roll, yaw, altitude, remaining battery life, and Wi-Fi link quality. The UAV's front and bottom cameras were also accessed with a Python program that displayed a live feed from the cameras onto a computer monitor. The UAV was manually manipulated while executing the programs. The UAV seemed to calibrate its sensors based on its initial pose during a manual reset or after reconnecting its battery. Thus, the pitch, roll, yaw, altitude, and magnetometer values were relatively consistent if the pose of the UAV was consistent during a manual reset or while reconnecting its battery. The yaw angle and magnetometer values were also successfully acquired during a simulated flight through an indoor course. Battery performance was also found to be stable while manually manipulating the UAV and experienced only small decreases in percentage with each execution of a program. However, the UAV's cameras did not perform well. Both the forward and downward-facing cameras were accessed, and a live feed was displayed on a computer monitor, yet the lag produced by the cameras was significant enough to render them unusable for this research. Additional time spent researching different methods of accessing and manipulating the cameras may have yielded better results. However, the Raspberry Pi camera proved to be sufficient for this research. Wi-Fi link quality was also evaluated and found to produce adequate results.

Although the UAV's onboard sensors displayed consistent and stable performance while simulating flight, in-flight performance was found to be inconsistent and unstable. After mounting the hardware to the UAV, the UAV's sensors and battery life were evaluated during flight. Of particular importance to this research was the yaw angle and magnetometer values, which seemed to be entirely arbitrary after being mounted to the drone. One possible explanation for this phenomenon may be interference from the Raspberry Pi and its lithium-ion battery power supply that was connected via the GPIO pins. Another possible explanation could be interference caused by other nearby electronic devices in the lab. The UAV's battery also suffered significant decreases in performance during real flight conditions. The specifications state that the battery is capable of twelve minutes of flight time, however, only about half of the stated capability was realized. Battery performance increased after obtaining new batteries but remained inadequate for flight after falling below 40%. Wi-Fi link quality was stable during flight, although the UAV occasionally experience instances of unresponsiveness to remote commands issued to initiate a safe landing sequence.

Structural components of the UAV were also indirectly tested for resiliency during flight. Many flights were conducted during the research; some were stable, while others were not. The structural integrity of the components of the UAV was maintained during stable flights. However, a substantial amount of the minor crashes experienced by the UAV produced significant deformation to the stainless steel shafts connecting the rotor wings to the UAV and rendered the UAV flightless until the shafts were replaced as the deformity caused substantial drift. While replacing the stainless steel shafts, the E-clips holding the shafts in place were also easily damaged and had to be replaced often. The other structural components of the UAV displayed adequate resiliency.

4.2.2 External Hardware

Performance of the Raspberry Pi Camera Module v2 and external Adafruit LSM303DLHC triple-axis accelerometer/magnetometer was initially evaluated

Figure 2: Parrot AR.Drone 2.0 with Raspberry Pi hardware mounted to the hull

before mounting the hardware to the UAV. A Python program was implemented to access and display the values obtained from the LSM303DLHC as well as a live feed from the camera onto a computer monitor. The values from the LSM303DLHC were successfully acquired and produced consistent results while manually manipulating the LSM303DLHC. The camera displayed adequate performance when evaluated by visually inspecting its live feed for lag time, field of view, and distance of AprilTag detection while manually manipulating the camera.

Performance of the Raspberry Pi Camera Module v2 and external Adafruit LSM303DLHC triple-axis accelerometer/magnetometer was reevaluated after mounting the hardware to the UAV. Pictures acquired from the camera and saved to the Raspberry Pi during UAV flight revealed adequate camera performance, whereas the data acquired from the LSM303DLHC revealed compromised performance. The erratic behavior of the LSM303DLHC is thought to be a consequence of the instability of the UAV during flight.

4.3 UAV Flight Performance

Similar to the performance evaluations of other components in the system, the performance of the UAV's flying capabilities was evaluated both before and after mounting hardware to the UAV. Successful flight was achieved by implementing a Python program that used the PS-Drone API to manipulate the altitude, position, orientation, and landing of the UAV. The UAV remained satisfactorily stable during flight and generally corrected most of the unsolicited positional drift that occurred due to momentum after the execution of each command.

After demonstrating satisfactory flight capabilities, the hardware was mounted to the UAV, and flight performance was re-evaluated using the same Python program. Flight performance suffered from significant drift while hovering after executing commands to manipulate the pose of the UAV. This resulted in an inability of the UAV to achieve adequate altitude. Due to a loss in performance, a manual reset of the UAV was performed. After the manual reset, flight capabilities were partially restored, however, they remained inferior to the previously achieved levels of performance. Although inferior to previous levels, the flight capabilities were deemed to be sufficient for the research.

4.4 Autonomous UAV Navigation

After each component of the system was tested, an indoor course was constructed. A Python program was developed to test the entire system and validate the proposed solution. Initial testing was conducted inside the lab. Through many trial runs and algorithm refinement, successful autonomous UAV navigation was partially achieved. The decrease in flight capabilities that was noted after attaching the hardware to the UAV was fully realized while navigating the course. Unwanted positional drift after executing commands to manipulate the pose of the UAV was magnified to the extent that the UAV could no longer correct for. The significance of the drift ultimately rendered the UAV incapable of fully navigating the course, however, partial course navigation was still achieved. After initiating takeoff, the UAV successfully increased its altitude and proceeded forward while taking pictures of the environment with the Raspberry Pi camera. Once the first AprilTag was detected, the UAV ceased forward motion, decoded the AprilTag, and used the data associated with the AprilTag's unique ID to acquire the previously computed positional data of the next AprilTag. The UAV then adjusted its yaw angle to navigate in a path towards the next AprilTag. However, this is the step that produced significant drift. Even a rotation of just 70° produced enough drift to render the UAV incapable of proceeding through the course. After conducting indoor flights, a similar course was then constructed outside to test the system. Just as with the indoor flights, the UAV was only able to achieve partial navigation of the course due to drift.

5 Conclusion

Technological advancements over the years have enabled the widespread use of UAVs in many different sectors. As the number of applications for UAVs has dramatically increased, so has the need for fully autonomous navigational capabilities. Many UAVs have achieved autonomous navigation using GPS and other technologies. Another technology developed to assist in autonomous robotic navigation is fiducial markers. Fiducial markers are particularly useful in GPS-denied environments, as they can be placed in an environment and serve as a physical waypoint for a UAV. Due to the limitations of GPS, prior research has been conducted studying different methods of autonomous UAV navigation in GPS-denied environments. However, previous research using

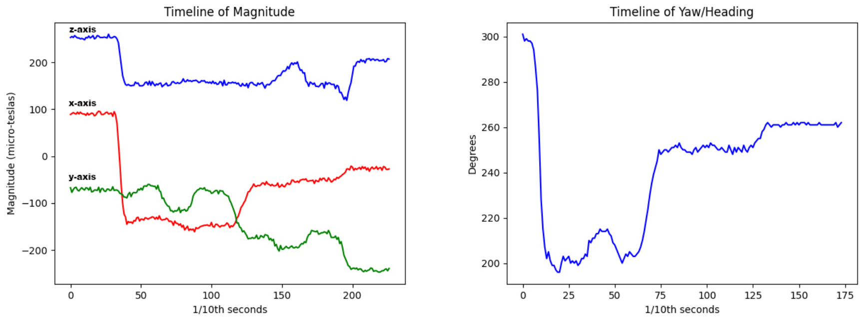

Figure 3: Recorded magnitude data of drone during a simulated indoor course run

only fiducial markers and onboard sensors for autonomous UAV navigation required an onboard camera to have constant visual contact with at least one fiducial marker. In this paper, autonomous UAV navigation in a GPS-denied outdoor environment using discontinuous visual contact with fiducial markers was explored.

The research conducted for this paper uses several different technologies including the AprilTag fiducial marker system, a Parrot AR.Drone 2.0, an Adafruit LSM303DLHC triple-axis accelerometer/magnetometer, a Raspberry Pi 3B, and a Raspberry Pi Camera Module v2. Many different software packages and technologies were used including Python, OpenCV, PS-Drone, and the software for the AprilTag fiducial marker system. These technologies were evaluated both independently and in combination with each other, and both indoor and outdoor courses were constructed to evaluate the viability of the explored solutions. Although autonomous UAV navigation was only partially achieved due to substantial UAV drift during flight, the results are promising. The technologies implemented generally met or exceeded performance and interoperability requirements. With additional research and experimentation, the performance issues with the magnetometers can likely be mitigated. Likewise, additional algorithms could be developed to minimize UAV drift, or a modern UAV with enhanced resiliency and advanced drift control mechanisms could be used with the proposed system to achieve fully autonomous UAV flight.

References

[1] Jan Bacik, Brennan Cain, Frantisek Durovsky, Pavol Fedor, and Daniela Perdukova. Autonomous flying with quadrocopter using fuzzy control and aruco markers. *Intel Service Robotics*, 10:185–194, 2017. DOI: 10.1007/s11370-017-0219-8.

[2] Michail Kalaitzakis, Brennan Cain, Sabrina Carroll, Anand Ambrosi, Camden Whitehead, and Nikolaos Vitzilaios. Fiducial markers for pose estimation. *Journal of Intelligent & Robotic Systems*, 101(71), 2021. DOI: 10.1007/s10846-020-01307-9.

[3] Fabrizio Lamberti, Andrea Sanna, Gianluca Paravati, Paolo Montuschi, Valentina Gatteschi, and Claudio Demartini. Mixed marker-based/markerless visual odometry system for mobile robots. *International Journal of Advanced Robotic Systems*, 10(5), 2013. DOI: 10.5772/56577.

[4] Mohammad Nahangi, Adam Heins, Brenda McCabe, and Angela Schoellig. Automated localization of uavs in gps-denied indoor construction environments using fiducial markers. *2018 Proceedings of the 35th ISARC, Berlin, Germany*, pages 88–94, 2018. ISBN: 978-3-00-060855-1.

[5] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, 2011. DOI: 10.1109/ICRA.2011.5979561.

[6] Lee Kian Seng, Mark Ovinis, Nagarajan, Ralph Seulin, and Olivier Morel. Vision-based state estimation of an unmanned aerial vehicle. *Trends in Bioinformatics*, 10(1):11–19, 2017. DOI: 10.3923/tb.2017.11.19.

CS Materials, A System to Assess and Align your Courses to National Standards*

Conference Workshop

Erik Saule[1], *Kalpathi Subramanian*[1], *Jamie Payton*[2]
[1]*Computer Science*
The University of North Carolina at Charlotte
[2]*Computer and Information Sciences*
Temple University
{esaule,krs}@uncc.edu, payton@temple.edu

Providing high quality learning experiences to undergraduate computer science students can help promote student retention and success in the major. Towards that goal, instructors are increasingly called upon to implement promising teaching strategies, develop engaging materials, and implement inclusive pedagogical techniques. Program administrators are tasked with assessment and accreditation of degree programs, requiring instructors to demonstrate alignment to national standards. However, few instructors have formal training in pedagogy, and lack awareness of evidence-based strategies for creating engaging materials with clearly stated learning outcomes and aligned assessments. In this workshop, instructors will be introduced to `https://cs-materials.herokuapp.com/` (CS Materials), a software infrastructure for assessing course materials with respect to national standards, aligning/comparing a course with other similar courses, as well as its role in the institution's curriculum. The workshop will provide hands-on experience to instructors to classify their own courses against accepted curriculum guidelines. Workshop attendees will will learn to input learning materials, perform topic coverage analysis, compare their course with other similar courses, and search for new materials for use in their classes. Participants will also learn how to participate in CS Materials project, to contribute and benefit from the project, and an opportunity to discuss curriculum related issues they face in the classroom.

Designing Specifications Grading Systems

Conference Workshop

Dan S. Myers
Department of Mathematics and Computer Science
Rollins College
Winter Park, FL
dmyers@rollins.edu

Specifications grading is an approach to assessment that does away with most of the elements of traditional grading. In a pure specs system, there are no percentages, no points, and no partial credit. Instead, work is assessed using only two categories: it is either *satisfactory* and receives full credit, or *unsatisfactory* and receives no credit.

Every assignment comes with a set of specifications — such as a rubric or autograded tests — that clearly outline what students must do to receive credit for a submission. Many systems also give students a mechanism to revise and resubmit unsatisfactory work to bring it up to the specifications. Students still receive a standard A-F letter grade at the end of the course. However, unlike a traditional system, the final grade isn't determined using a weighted average or numerical scale. Instead, the final grade is determined by completing a bundle of assignments that demonstrates a certain level of success at meeting the goals of the class. Because it focuses on affirming competence rather than ranking performance, many authors have argued that specifications grading is both more realistic and more equitable than traditional points-based grading systems [1, 2, 3, 4, 5, 6].

This workshop will present an interactive introduction to specifications grading, drawing upon my experience using it in more than a dozen computer science classes at Rollins College over the past three years. The target participants are computer science instructors who are considering adopting specs grading or interested in learning more about it. In particular, faculty will leave the workshop with multiple concrete examples for designing their own grading systems, as well as an awareness of the trade-offs between the different approaches. Interactive portions of the workshop will allow participants to work individually or in small groups to design a new grading system for one of their own classes. Topics will include:

- Why do we grade the way we do? A brief history of grades and arguments for new grading methods
- The specs grading philosophy
- Individual work and discussion: choose one of your own classes and describe its students and goals
- Models for specs grading: contracts, points-based systems, and hybrids
- Individual work and discussion: thinking about different models for your own class
- Implementation details: mapping learning outcomes to assignments and specifications
- Individual work and discussion: creating a specs-graded assignment
- Common challenges: complexity, retakes, work on the boundary, concerns about grade inflation
- Other approaches: labor-based grading, mastery grading, and ungrading
- Questions and discussion

References

[1] Andrew Berns. Scored out of 10: Experiences with binary grading across the curriculum. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 1152–1157, 2020.

[2] Susan D Blum and Alfie Kohn. *Ungrading: Why rating students undermines learning (and what to do instead)*. West Virginia University Press, 2020.

[3] Joe Feldman. *Grading for equity: What it is, why it matters, and how it can transform schools and classrooms*. Corwin Press, 2018.

[4] Asao B Inoue. *Labor-based grading contracts: Building equity and inclusion in the compassionate writing classroom*. WAC Clearinghouse, 2019.

[5] Linda B Nilson. *Specifications grading: Restoring rigor, motivating students, and saving faculty time*. Stylus Publishing, LLC, 2015.

[6] Christian Roberson. Techniques for using specifications grading in computer science. *Journal of Computing Sciences in Colleges*, 33(6):192–193, 2018.

Real-World Data, Games, Applications and Visualizations in Early CS Courses Using BRIDGES*

Conference Workshop

Kalpathi Subramanian[1], Erik Saule[1], Jamie Payton[2]
[1] *Computer Science*
The University of North Carolina at Charlotte
[2] *Computer and Information Sciences*
Temple University
{krs,esaule}@uncc.edu, payton@temple.edu

CS enrollments have seen dramatic increases in recent years, however, engaging all CS CS majors in early foundational courses is critical to their retention and long-term success. Grounding Computer Science concepts in reality by solving important real-world problems and relevant applications are a key to increasing students' motivation in computing. This workshop provides instructors with a hands-on introduction to http://bridgesuncc.github.io**BRIDGES**, a software toolkit for programming assignments in early computer science courses (CS1, CS2, data structures, and algorithm analysis). BRIDGES provides capabilities for creating more engaging programming assignments, through (1) easy access to real-world data, spanning domains such as social networks, science, government, movie, music, and literature, (2) *visualizations* of the data or data structures, (3) an easy to use API for creation of games, and, (4) algorithm benchmarking. Using BRIDGES in data structures, algorithms, and other courses have shown better student outcomes in follow-on courses, when compared to students from other sections of the same course. BRIDGES has impacted over 2000 students across 10 institutions since its inception 5 years ago. A repository of `http://bridgesuncc.github.io/newassignments.html` (**BRIDGES assignments**) is now maintained for BRIDGES users. Workshop attendees will engage in hands-on experience with BRIDGES, play with example BRIDGES programs and datasets, and will have the opportunity to discuss how BRIDGES can be used in their own courses.

Effective Pedagogical Practices in the Computer Science Classroom*

Conference Workshop

Chris Healy (moderator)[1], *Andy D. Digh*[2]
Paula Gabbert[1], *Michael Verdicchhio*[2]
[1]*Furman University*
Greenville, SC
{chris.healy, paula.gabbert}@furman.edu
[2]*Mercer University*
Macon, GA
digh_ad@mercer.edu
[3]*The Citadel*
Charleston, SC
mv@citadel.edu

This panel of presenters, with over 80 years of teaching instruction between them, will share with attendees a variety of actual tools, toys, and/or visuals that have helped students understand concepts in the programming sequence, i.e. CS 1, CS 2, CS 7 (algorithms) classes. These are based on moments in the classroom when students have reported that new or difficult concepts were made easier to learn.

1 Background

The book *Models of Teaching* provides evidence from educational theorist David Ausubel that any "new ideas can be usefully learned and retained if they can be related to already available concepts or propositions that provide ideational anchors." In addition, the book further illustrates that "when we help students acquire information and skills, we also teach them how to learn as well as how to think critically, compare, and apply new knowledge" [4]. This offers an important reminder that when providing additional connections or hooks for students in our course, it makes the course outcomes even stronger. As David Gooblar notes in *The Missing Course: Everything They Never Taught You About College Teaching*, "the more our brains work while acquiring knowledge, the better they are at retaining it" [3]. In addition, any time you use schemas to help a student "think about something they have previously learned or experienced," you help them transfer it from working memory into long-term memory, which is their "seat of understanding" [2].

2 Panel Structure

After brief introductions, each of the four panelists will have up to 15 minutes to focus on several effective pedagogical practices they have used to engage students and enhance student learning. Actual demonstrations involving visual, auditory, and tactile tools will be given. Most importantly, they will highlight having fun in the classroom while teaching computer science. In the last 30 minutes, audience members will be encouraged to comment on any similar times when they have watched "the light come on" with their own students in the classroom. They will be able to share any interesting or innovative methods of their own "to simplify and clarify complex topics" as well as ensuring that what is said will easily be understood or remembered [1].

3 Estimated Panel Time

90 minutes

References

[1] Ken Bain. *What the Best College Teachers Do.* Harvard University Press, 2004.

[2] Nicholas Carr. *The Shallows: What the Internet is Doing To Our Brains.* W.W. Norton & Company Publishers, 2020.

[3] David Gooblar. *The Missing Course: Everything They Never Taught You About College Teaching.* Harvard University Press, 2019.

[4] Bruce Joyce. *Models of Teaching.* Allyn and Bacon Publishers, 1992.

Counting Tiles

Generating a synthetic dataset for unsupervised learning*

Nifty Assignment

Steven Benzel
Department of Mathematics
University of North Georgia
`sbenzel@acm.org`

About

When studying algorithms for cluster analysis it is useful to have a variety of datasets for students to experiment with, ranging from low dimensionality to high dimensionality. High dimensional datasets with many elements can be quite large and problematic to store and disseminate. An alternate is to have students generate synthetic datasets for themselves.

Construction

A visually straightforward dataset can be generated where an element consists of a random number of square tiles randomly placed on a grid and labels given by the number of tiles placed. In particular, given the 4-tuple of positive integers (N, m, n, k) with $nk \leq m$ we can create a dataset of N elements where each element is an $m \times m$ grid upon which up to k $n \times n$ tiles are disjointly placed. The dataset will be a subset of $\mathbb{R}^{m^2}$ for gray-scale tiles and $\mathbb{R}^{3m^2}$ for color tiles. Construction of the dataset using numpy arrays in Python is straightforward. For example, 5 random elements where $m = 200$, $n = 16$, and $k = 5$ are given in figure 1.

For m much larger than nk the euclidean distance between 2 elements is somewhat independent of m. Interestingly, the dataset can be correctly clustered using a single linear function but popular algorithms such as DBSCAN[1] or K-Means[2] do tend to have trouble.

The dataset can also be used for supervised learning if required and elements can be created as needed and fed directly into a neural network for

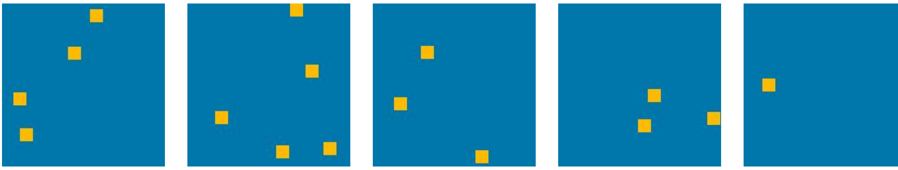

Figure 1: Five random elements

training without the need for storage. The dataset can be generalized in many ways including differing tile sizes, shapes, and colors, along with differing backgrounds.

References

1. M. Ester, H.-P. Kriegel, J Sander, X. Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. in KDD, vol. 96, pp. 226–231, 1996.

2. S. P. Lloyd, Least Squares Quantization in PCM, IEEE Transactions on Information theory 28 (2) 129-137, 1982.

Secure Programming App*

Nifty Assignment

Karen Works
Department of Computer Science
Florida State University
Panama City, FL
keworks@fsu.edu

About

Over the last decade, security vulnerabilities in applications have increased at an alarming rate[1,2]. It is essential that software developers are trained in the skills required to understand and implement the principles of secure coding.

This set of assignments is for an undergraduate secure programming course in Python. Upon completion of the assignments, students build a secure small scale real-world application that incorporates the principles of secure coding including cryptography, network security, and data protection.

Assignments

In this set of assignments students incrementally build a flask website with a SQLite3 back-end database using Python. Over the course of the assignments, the students are required to secure personal data when received by the website and stored in the database, and secure all messages sent and received between users and their "boss." To make the assignment interesting to the students it is based upon secret agents using a website to post secrets to share with other agents and view secrets posted by other secret agents.

Outline of the module long set of assignments: (See Figure 1)

1. SQLite3 Database- Write a Python script that creates a table to store login and user information and adds records to this table
2. Basic Flask Website- Using Python and the Flask library, create a Flask website with the following pages: Home, Add a User, List Users, and Results.

3. Role Based Access Control- Add role-based access control to the web site using a login screen and the security level of the user logged in.
4. Secure Data- Secure the personal data sent to and from the website as well as the data stored into the database.
5. Send/Receive Secure Messages- Send and receive secure messages between two different processes.
6. Send/Receive Authenticated Messages- Send, receive, and authenticate messages between two different processes.

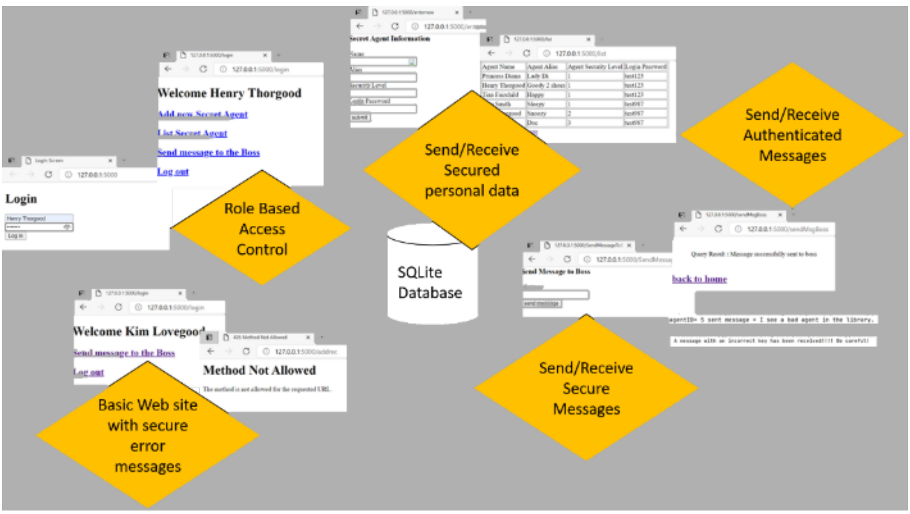

Figure 1: Assignment Flow

References

1. Department of Homeland Security, *ICS-CERT: Industrial Control Systems - Computer Emergency Response Team,* 09 2020. https://us-cert.cisa.gov/ics
2. MITRE-Corporation, *Common Weaknesses Enumeration,* 2019. https://cwe.mitre.org/

Metadata

Summary	• Create application in PyCharm • Implement the following: SQLite3 Database (using sqllite3 library), Basic Flask Website (using flask library), Role Based Access Control (using sqllite3 and flask libraries), Secure Data (using Crypto, Cryptodome, or Cryptodomex library), Send/Receive Secure Messages (using sockerserver, socket, and sys libraries), Send/Receive Authenticated Messages (using Crypto, Cryptodome, or Cryptodomex library)
Topics	Website development, creating and using a database in an app, principles of secure coding including cryptography, network security, and data protection
Audience	Students who have completed CS1
Difficulty	Medium – Depends upon the programming ability of the student
Strengths	• **Applies security algorithms and concepts to an application:** Supports tactile hand-on learning • **Implementation of security algorithms is not required:** Supported by Python libraries • **Adaptability to different programming abilities of students:** Python is an easy programming language for students with a background in another programming language to pick up
Weaknesses	Students who miss one portion of the assignment can fall behind
Dependencies	IDE of choice (PyCharm) and version of Python
Variants	• Adapt to trigger message to be sent to other users based upon content or critical level • Adapt to track conversations threads with security so that users can only see the portions of the thread that they have permissions to see