Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #46619

SUCCESS: A Summer Camp Promoting On-campus Connections in Software Engineering

Dr. Kevin A Gary, Arizona State University

Dr. Gary is an Associate Professor in the School of Computing and Augmented Intelligence at the Ira A. Fulton Schools of Engineering at Arizona State University. His research interests are in agile and open-source software, software engineering in healthcare, and software engineering education. Presently he is focused on flow and quality metrics derived from agile research and applied to open-source software, and in identifying Regression Test Selection methods suitable for Agile and Lean software development.

He was a founding faculty member of the software engineering degree programs at ASU and developed the project-centric curricular implementation known as the Software Enterprise. He has served twice as program chair and led the program through multiple positive ABET accreditation visits. Kevin blends industry and academic experience to bring theoretically grounded, practice-oriented methods to the classroom.

Kevin is a member of ASEE, ACM, and IEEE.

Cecilia La Place, Arizona State University, Polytechnic Campus

Cecilia La Place is a fifth-year Ph.D. student at Arizona State University (ASU) studying Engineering Education Systems & Design. She has received her M.S./B.S. in Software Engineering through an accelerated program at ASU. She organizes, attends, and studies hackathons as informal learning environments that hold the potential to empower students of any and all backgrounds.

Vidya Rupak
Rakshilkumar Modi
Karthik Vaida

SUCCESS: A Summer Camp Promoting On-campus Connections in Software Engineering

Abstract

This paper presents the design, implementation, and results of a summer camp in software engineering created in response to declining student engagement in on-campus classes. COVID-19 accelerated the adoption of online education, not only in higher education but in elementary and high schools. Our faculty in Software Engineering anecdotally noticed a drop-off in student engagement in on-campus students, and an increase in local students opting for the online modality of our degree program. They hypothesized this was due to the onset of online education during the pandemic. To increase visibility and interest in software engineering, and advocate a return to in-person education, SUCCESS (A Survey of Computing, Coding, and Engineering Software Systems) summer camp was created for 6th-11th graders to reignite pathways in engineering.

The SUCCESS program is a one-week summer camp emphasizing a different theme each day - IoT, AI, Game development, AR/VR, and Security. The final day included a showcase for parents and friends to review the youth's work, and a certificate of completion was given during a camp graduation ceremony. The initial offering in 2024 enrolled 16 students from grades 6 through 11 who participated in technical activities, usually in pairs, in a fun, informal format. A pre-post custom survey instrument, based on prior summer camp studies, was given (with informed parental consent) with the results showing increased interest in software engineering, though the results related to on-campus connection were mixed. This paper will describe the camp, lessons learned, and present the results of our survey.

Introduction

Campus enrollment in the undergraduate software engineering degree program at Arizona State University has recently flatlined, and further, the typical student appears less engaged – less energy and less likely to interact, inquire, and express intellectual curiosity in front of their peers. Our open spaces for students are not as busy as they once were, and students tend to stay on campus only for class and are more willing to simply engage via remote learning mechanisms. Online remote engagement [7][9] is here to stay, but we believe re-energizing the in-person campus connection for junior high and high schoolers will lead to better outcomes in learning, engagement, retention, and most importantly student success. This age range is where key inflection points in professional pathways begin, and even the future remote learner can benefit by connecting directly with the on-campus experience.

The purpose of the SUCCESS program is to re-energize and reconnect students in grades 7-14 with in-person, hands-on activities in computing. The objectives were to:

1. Facilitate stronger identification of professional pathways in computing.
2. Facilitate stronger connection with the campus.

3. Educate those who may have a peripheral interest in computing as to the:
    a. Range of computing disciplines and professions.
    b. Real nature of computing. Our anecdotal observation is that present-day students are far more computer and technology literate as *users* of computer applications and technology, but have a surprisingly poor understanded of how computers work, are connected, and their information managed.
4. Diversify the pipeline of students interested in computing as a professional pathway through recruitment and inclusion of underrepresented populations.

In the following sections, we lay out the camp plan and rationale for said plan, discuss the facilitators' lessons learned, present the results and analysis of surveys given at the beginning and end of the camp, discuss related work, and conclude with our next steps.

The SUCCESS Camp

The inaugural SUCCESS camp took place July 9-12, 2024, as an on-campus experience. Participants were recruited through Arizona State University University's summer programs page, and ad hoc communications with local school districts and personal contacts. Camp development was funded by a corporate philanthropic grant from the State Farm Foundation, which included some funds for scholarships. Ultimately only one family applied for scholarship funding. Tuition was set at $325 for the 4-day session, which is comparatively less expensive than other summer camp programs offered by the University, particularly the Engineering College. 17 students ultimately signed up for the camp, though one dropped after one day due to a family emergency. 16 participants, grades 6 through 11, 3 females, attended the full camp. One female participant is deaf, the University provided a sign language interpreter to communicate with this participant. We requested parental and participant consent to take photographs and use these for future publicity, reports, and publications; no participants opted out.

Participants were dropped off by parents in the morning and picked up in the afternoon. The camp daily schedule ran from 9am to 4:30pm with an hour lunch and planned break times. Additional free time breaks were scheduled in which fun classroom games were played in team format to foster social interaction and to get the participants out of their seats. These breaks were supported by student volunteers from the campus outreach office. The technical staff supporting the participants was led by a faculty member and three graduate students, who were paid part-time to support the camp. University protocols for working with minors were followed, including background checks, fingerprinting, and mandated training.

Each day of the camp followed roughly the same schedule, with a half-day module dedicated for a particular computing topic. Seven modules were delivered during the technical sessions as described in the next section, with the final afternoon of the last day reserved for a showcase where participants could demonstrate their favorite projects to family and friends. Each day started with a setup/warm-up period and ended with a teardown/reflection, usually done with a Kahoot (www.kahoot.com) as these online quiz games were familiar to the participants. Each module was broken into 2 sessions divided by a break, with the first session usually involving "follow-me" style instruction, and the second on independent work.

The university classroom used for the camp has whiteboards on 3 walls, and large studio-style tables to allow campers to partner up in pairs. Most tables had 2 pairs of participants, or 4 per table. Campers were provided with university laptops and pre-provisioned accounts, and given USB flash drives to allow them to download artifacts and bring them home or move around to other machines during the camp. The entertainment theme of the camp was based on the popular television show “Stranger Things”. Marketing, technical activities, camp room decorations, and fun activities were designed around this theme. The camp even concluded with a certificate ceremony starring none other than Vecna himself!

SUCCESS Camp Technical Topics

In keeping with a “survey of computing” theme, the technical topics for the camp include embedded systems, game development, virtual reality, web development, artificial intelligence (AI), and cybersecurity.

The first day theme was “Systems and Networking”, and started with an embedded systems module on an Arduino UNO R4 Wi-Fi. These boards are easy to use, feature a 12x8 LED matrix, and connectivity via USB-C, Bluetooth, and WIFI. Initially participants worked with the Arduinos through a USB-C connection, starting with pre-packaged scripts (called “sketches” in the Arduino lingo) *Wink*, *SerialWithArduinoGraphics*, and *Game of Life*. After gaining confidence by successfully compiling and deploying these sketches, participants were given several challenges of increasing difficulty, ranging from enhancements to the loops in the Wink sketch, row/column traversal of the LED matrix, and creative forms on animations across the LEDs. Participants were also encouraged to experiment with any sketch that piqued their interest. The module instructional leader worked through examples at the podium that projected (split-screen) the Arduino IDE and the physical board (using a document camera) to monitors throughout the room, while the remaining staff provided troubleshooting support or simply perused the room to observe and discuss campers’ work and discuss their problem solving and creative processes. Given this was the first activity, there was more time spent in setup and troubleshooting compared to most other modules, but this was expected and even welcome as it presented opportunities to explain how and why things work in computing the way they do, and how one may systematically go about solving such problems.

After lunch on Day 1 a campus tour was scheduled and provided by the university outreach office on campus. During this tour campers were able to see the campus and visit multiple research labs, keeping with the objective to connect future students to the on-campus experience.

The afternoon module on day 1 started by having the participants work with programmable LED light strips (ALITOVE WS2811 RGB LED pixels). Initially the light strip was simply laid out on their tables and connected to the Arduino with male-to-male jumper ribbon cables to supply power, data, and ground (separate breadboards were not needed). Participants experimented with pre-built sketches to make the lights blink, glow, change colors, and animate, modifying loops, delays, and other control features to tweak the sketches. Then campers created the alphabet wall made famous in a season 1 episode of Stranger Things. To do this they laid out the letters on a

2x3’ foam poster board, attached the LED strip with pins to correspond with the letters, and connected the setup to the Arduino. In the sketch code they had to align letters to lights using arrays with a pre-built script (this exercise is described in several popular DIY websites). Initially they hardcoded strings to send to the wall, and then moved on to accepting user input on the console. They concluded the exercise by connecting to the Arduino via the Wi-Fi module, utilizing Node.js http server code pre-installed on the laptops to enter their strings via a web interface and connecting to a local router (not connected to the internet).

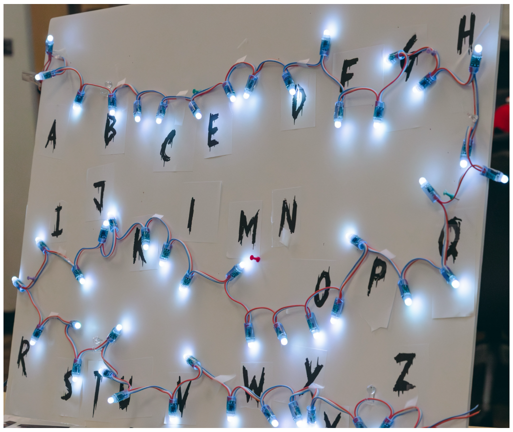

Figure 1. LED wall inspired by the TV show Stranger Things

To conclude the day, we made a simple guessing game, where we displayed the IP addresses of each Arduino Wi-Fi for all to see, and had campers send messages to others’ boards so they could decipher who had what IP address. This not only enabled discussion on networking, but also served as a lead-in to the next day activity on web development.

The day 2 theme was Web and AI. The morning session started with a recap of the prior afternoon session, and ensuring they had their setup back up and running. To introduce Web and AI concepts, the classic *Eliza* program [15] was presented and discussed. Eliza was an early instantiation of the Turing test, by presenting subjects with responses from a chatbot and a human and asking the subject to ascertain which was making the response. This initiated a lively discussion in the camp and set the context for the day.

Campers were given pre-built code in HTML5 and JavaScript (front-end) with the simple Node.js http server from the previous day to run locally on their camp-issued laptops. The front-end JavaScript contained an array of objects that corresponded to the keyword-answer-question structure of the Eliza program; the user input is matched with a keyword, and responds with a random selection of an answer based on that keyword followed by a random selection of a follow-on question. Participants could then customize the application by modifying the data structure or by tweaking HTML and CSS. This activity tied in data structures, how the web works (client/server and networking), and basic AI.

Figure 2. Eliza snapshot

The afternoon session introduced LLMs using ChatGPT. A two-server setup with provided code enabled access to the OpenAI’s API, with one server a modified version of the Eliza server from the morning, and the other a custom proxy server to enable access to the API with revealing API keys. The proxy server also pruned prompts to ChatGPT to ensure responses matched the format of non-ChatGPT version of the Eliza chatbot. Campers concluded the day by conducting the Turing test for themselves in their groups at each table.

Day 3 featured game development and virtual reality. To introduce basic concepts in game development, we used the free open-source Godot game engine (godotengine.org). Game development IDEs like Unity can be very complex and challenging, whereas Godot afforded a low learning curve entry point, while still support a multitude of features, ranging from 2D to 3D and XR games. Initially the entire camp worked together on enhancing a version of the popular "dino run" game (we started from https://github.com/russs123/dino_run_tut), a 2d strip animation game. In this game your dinosaur appears to be running left-to-right and has to jump over obstacles presented in its path. The running illusion is created by moving the background, ground, and obstacles are various speeds while the dinosaur sprite stays still but has an animation where its legs appear to be moving.


Figure 3. Godot dino run snapshot

Campers customized the game by introducing new resources (objects like the angry bird in the figure), mapping new collision zones, and modifying speeds and reaction to user input (manipulating how the dinosaur jumps). The concepts involved include algorithmic thinking, sprite animation, and collision detection. The exercise did prove challenging, as despite Godot being a relatively easy IDE to use, it was complex for first-time users. To conclude the session, campers connected this activity to the previous day's activity by exporting the game to a web application to run in a browser, a useful feature Godot provides.

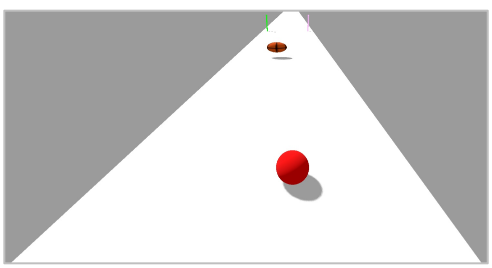
Figure 4. Godot 3D game

Game concepts were extended from 2D to 3D using a simple game created by the camp staff. This game modified the collision detection problem to 3D by manipulating a ball around objects on a moving platform. If the ball collided with the object or fell off the edges of the platform the player loses the game, if the ball makes it to the goal line the player wins the game, and starts a new level that introduces more obstacles and speed. This activity extended the collision detection concepts with visual perspective. Campers introduced new objects, changed perspective, and even manipulated the platform boundaries to create illusions.

In maintaining the theme of connecting to a physical campus, camp participants were given a second tour on the university campus just before lunch break on day 3, this time from the CHART (Center for Human, AI, and Robot Teaming) lab. This lab conducts research in AI, robotics, cybersecurity and team dynamics, and had multiple interactive and fun exercises to spur curiosity and interest in the campers.

The afternoon of day 3 was the most challenging technically and to support for camp participants. The objective was to introduce virtual reality (VR), and to this end Meta Quest 2 headsets were purchased for each camper. Our intent was to evolve from 2D to 3D games and then into VR, as the amusement of the simple dino run game quickly wore off on our very experienced camp participants (most were male and expressed significant experience with video games). However, we discovered at a late date before the camp that not all of the university-issued laptops supported Unity 3D and the VR connection. Further, Unity is a complex environment that can easily frustrate the novice user, and we did not have enough time in the camp survey format to go into in-depth training on the environment. Therefore, we devised 3 parallel activities, one Godot OpenXR challenge, the second a Unity 3D challenge, and the third a Unity VR experience. We rotated groups between the 3 stations, utilizing a separate room across the hall with more powerful PC workstations for Unity and VR, and continuing the Godot activity in the main room.

Figure 5 shows screenshots of the Unity 3D and VR activities. On the left is a strip animation game similar to the Godot 2D game, but render in Unity with 3D perspectives. On the right is a VR "room" with a virtual alphabet wall, a prompt to answer on the wall, and objects that can be manipulated within the VR headset.

Figure 5. Unity 3D (left) and a virtual reality room

The jump from the Godot environment to the Unity environment, coupled with the complexity and setup challenges for Unity led to difficulties getting all campers to engage fully. However, most campers were very excited by the environments and the interactivity, and suggested more time be spent on Unity in the camp. While we may consider this in the future, we did point participants (and their parents) to another camp focused on Unity gaming at our university.

The 4th and final day of the camp we adapted capture the flag (CTF) challenges from Carnegie Mellon's picoCTF (picoctf.org) platform by building a custom web application around the challenges. These challenges did not seem to hold the interest of the campers as did the other modules, and we observed many of the campers going back to work on the LED light wall from day 1, and a few on the Godot game from day 3. As it was day 4 and we felt that perhaps we had saturated the participants with the broad range of computing activities over 3 days, we decided to switch after the mid-morning break and introduce an activity we held in reserve.

To tie the first 3 days together, as a final exercise campers were shown how to integrate the web server for Eliza on day 2 with the LED Wi-Fi wall from day 1, and utilize the virtual alphabet wall inside the VR application. In this scenario responses were spelled out on the LED wall from the OpenAI API. This required a local router to put all devices on the same network. This activity was a bit advanced, and not all campers were up to this challenge despite heavily scaffolded given code and multiple assistants available to help. This was fine, we allowed participants to return and explore whatever activity most interested them during the camp if they were not interested in completing this activity.

At the conclusion of the morning the camp staff re-played the Turing test game, this time using a large LED wall constructed in the back of the room, with a conversation happening alternating randomly between ChatGPT and a camp worker using the VR wall in the other room. Campers in their respective teams (the teams used for fun game break activities) had to guess who was actually answering the prompts given to the wall. This activity integrated the 3 previous days' work in fun ways.

The morning of the last day concluded the module-oriented half-day approach to the camp. The afternoon was reserved for wrap-up activities. After lunch, a post-survey was distributed, and a final fun (non-technical) game played with prizes awarded for each team. Then participants were given time to setup their favorite projects from the previous days to prepare for the showcase. This actually went fairly quickly as most had spent time in the morning with their favorite projects as discussed above. Prior to the showcase a final fun activity integrating the technical topics was a scavenger hunt. The scavenger hunt was devised by presenting encrypted clues delivered in various ways – some were hidden within their given 2D games (recall we provided the pre-built code for most activities), some by simple paper hidden beneath objects in the room, others were in plain sight in the room but only took on meaning in conjunction with other clues, and others delivered via the master LED wall in the back of the room. The clues were used together to resolve a riddle, and teams had to present their solutions to Vecna (hiding in the other room) to win the prize (gift cards to a local smoothie shop).

The showcase was an open time for campers to present their projects to parents, grandparents, siblings, and friends at their respective tables. The camp concluded with a certificate ceremony where the camp staff, including Vecna, presented certificates of completion to each participant.

Figure 6. Camp activity and certificate ceremony

Evaluation and Analysis

To evaluate the impact of the camp against the objectives, pre- and post-surveys were administered with IRB approval from Arizona State University and parental/participant consent.

Campers were instructed they could skip any questions they did not want to answer, and could stop the survey and not turn it in at any time. The pre-survey was introduced at the end of a welcome/icebreaker activity on the morning of day 1 of the camp, and the post-survey was administered on the last afternoon of camp, before the demonstration showcase. Participants were given a randomly generated 5-character alphanumeric string pre-printed on their paper pre-surveys, and asked to copy down (by sending themselves a text message) the identifier so they could use it on the post-survey so we could align answers.

The survey instruments are of our own design, but influenced by prior works on engineering summer camps [3][10][12]. Some of the pre- and post-survey questions are the same or target the same objective, while others are unique to that instrument. The surveys are shown in Table 1 below, where we have taken the liberty of aligning the rows according to how questions aligned in the pre- and post-surveys. Likert scale questions ranged from 1-5 with 1 being Strongly Agree and 5 Strongly Disagree (lower averages mean greater agreement), and the Type column shows the average of the responses in the Type columns. Several free response and Word questions are given as well as discussed below, and there were multipart questions as well which are also discussed below in Table 2. Before presenting and discussing the results, we acknowledge the limitations of these survey instruments. The sample size (n=16) is too small for statistical significance, there is a selection bias as the population is clearly not a random sample, and there may be a variety of answers particularly in the free response questions as the age and grade range of the participants may affect understanding and richness of responses.

Table 1. Survey Questions

Pre-survey question	Type	Post-survey question	Type
1. In your own words please describe your interest in computing and in attending this camp	FR	*1. In your own words please describe your experience attending this camp. What parts did you like or not like*	FR
2. In your own words describe any outside experience/schoolwork you have had related to programming computers	FR	*2. What kinds of things would you suggest to make the camp more fun and interesting for next year*	FR
3. Does your school teach a computer science or programming class (circle yes, no or I don't know, and use the space below to indicate if you have taken that class or plan to take that class in the future)?	Y/N and FR		
4. I am excited to be at camp today	2.13	*3. I am glad I attended this camp*	1.93
		4. I would tell my friends to attend this camp next year	2.36
		5. Computers and programming are interesting	1.79
5. I can see myself going to college to study some sort of computing	2.73	*6. I can see myself going to college to study some sort of computing*	2.64
6. I can see myself pursuing a career with some sort of computing	2.67	*7. I can see myself pursuing a career with some sort of computing*	2.79
7. I spend some of my own personal time trying to learn more about computers and/or programming	2.6	*8. I plan to spend some of my own personal time trying to learn more about computers and/or programming*	2.29
8. I have one or more personal models (people) in computing I look up to	2.33		
9. I have one or more personal models in engineering and/or science (excluding the models in question 8) I look up to	2.33		

10. I have an older family member (parent or sibling) who works in, or studies, computing	2.53		
11. In order to be at good engineering, I have to be good at math and science	1.6	*9. In order to be at good engineering, I have to be good at math and science*	1.79
12. In order to be at good engineering, I have to be good at art and design	2.4	*10. In order to be at good engineering, I have to be good at art and design*	2.5
13. In the space below, write individual words (as many as you like) that, to you, describes a computer scientist	Word	*11. In the space below, write individual words (as many as you like) that, to you, describes a computer scientist*	Word

Reviewing Table 1, first we can see in the pre-survey (questions 8-10) that campers were between "Agree" and "Neutral" when it came to identifying mentors and role models in computing or engineering. Question 10 on a family member in computing had a high standard deviation (1.15) more likely due to the poor wording (in retrospect) of a Likert scale question (it is more of a Yes/No).

For the common questions, the results varied. Pre Q4 and post Q3 indicate a reasonable increase in enthusiasm, at least for the camp experience, with 4 campers giving higher agreement. Pre Q7 / post Q8 shows a significant difference in interest in computing, corroborating Q4/Q3. Pre Q5 / post Q6 indicates a slight increase interest in going to college for computing (2 stronger, 1 weaker), but is contradicted by Q6/Q7 indicating a slight decrease in a career in computing (1 weaker). Again, we caution against over-interpreting the results in a small sample size. However, there is a clear argument to be made that the camp experience enhanced interest in computing.

The last 2 common questions (Q11/Q9 and Q12/Q10) show a decrease in campers' perception of needing to be good in math/science or art/design. We somewhat expected these to go in opposite directions but were surprised they both showed a post-survey decrease. It isn't clear why this is, we hypothesize that the camp activities may have been over-scripted with not enough time to go deeper into the problem-solving or creative design aspects of each topic. We do note that being good at math and science rates more strongly both pre and post, which may align with the general perception, but suggests those who tilt toward design thinking may think of computing as too rote and less creative; a perception that should be addressed.

Regarding the free response questions, pre-survey question 1-3 asked about their interest in the camp, prior coding experience, and whether they knew if there was a class at their school. 10 of 16 campers indicated there was class at school, 11 indicated prior coding experience, ranging from game scripting and "drag-and-drop" IDEs (e.g. Scratch) and 9 expressed a genuine interest in the camp and only 1 indicated s/he was made to attend camp by a parent. Game development was the most popular motivation but there were a variety of responses, with a couple including AI. The word association question (Q13/Q11) largely did not yield valuable insights; many respondents did not follow the instructions and wrote complete sentences (poor question wording), and few completed the question on the post-survey, perhaps due to fatigue at that point. Some responses were interesting however, with *artistic, creative, problem-solver, persistent,* and *creative* among the interesting words. It may be more interesting to do this question as an interactive exercise at the end of the next camp.

We were particularly keen to learn what the youths thought about the camp (Q1 of the post-survey). The comments were overwhelmingly positive; all 13 written comments (3 did not answer) expressed a positive opinion, with *fun*, *liked,* or *enjoyed* mentioned on all 13. The few negative comments included the time the camp took up to not liking the campus tours, though these were isolated responses. Only 6 campers responded to Q2 on the post-survey asking for suggestions for improving the camp; 3 indicated more game development, 2 wanted more break time, and 1 asked for it to be more interactive.

Each survey also had multi-part questions at the end. For the pre-survey we asked: "*14. Below are topics you may or may not have been exposed to during your schoolwork, or from other activities outside school including your own. Put an X in the place best describing your skill level. For example, I play Angry Birds but I am not very good at it! Answer relative to your peers in your grade."* The topics were:

a. *Angry Birds (example, pre-marked)*
b. *Programming (any programming language)*
c. *MacOS Administration (I know how to work my computer better than my peers)*
d. *Windows Administration (I know how to work my computer better than my peers)*
e. *Creating social media content (videos, stories)*
f. *Mobile smartphone (I know how to work it and its settings better than my peers)*
g. *Video Gaming (playing them)*
h. *Unix (Linux)*

These topics were rated on a Likert scale (1-5) from "Not at all" to "Expert", so higher scores indicated greater pre-existing competence. This question was more for us to get a feel computing skill level at the start of camp. The main things we learned were that participants were more familiar with Windows than Mac or Unix (we asked about Unix to get a sense of potentially advanced computer skills as familiarity with Unix is not something the typical middle schooler is exposed to at school or at home), and video gaming was the most prevalent skill (3.6/5).

The post-survey had two multi-topic questions, the first (Q12) was "*Below are activities from the camp. For each Topic, answer the question 'I had fun doing this topic's activities' by placing an X in the closest matching column.*" While the second (Q13) was "*Below are activities from the camp. For each Topic, answer the question 'I learned a lot about this topic' by placing an X in the closest matching column.*" For both questions the choices for each were Likert (1-5) scale ranging from Strongly Disagree to Strongly Agree. The topics were the same with the results for each question given below in Table 2.

Table 2. Post-survey responses to module topics with respect to fun and learning

Topic	Q12 Avg	Q12 StD	Q13 Avg	Q13 StD
a. Programming the Arduino Matrix	3.71	0.59	4.07	1.1
b. Programming the Arduino with the string of lights	3.79	0.86	3.86	1.19
c. Understanding how the Web works	3.11	0.81	3.71	1.03
d. Working with the Eliza web application	3.36	0.72	3.71	0.96
e. Learning about Artificial Intelligence (AI)	3.61	0.54	3.64	0.97
f. Learning about game development (Godot, Unity)	4	0.93	3.9	1
g. Learning about Virtual Reality	4	0.53	3.6	1.2
h. Security and Capture-the-Flag	3.2	0.4	2.5	0.8
i. Campus tours and guest talks	3.38	1.15	3.9	1.1

Table 2 shows more interest and learning in the day 1 (Systems and Networking with the Arduinos) and day 3 (Game development and VR) though the reasons are probably very different. Only one camper indicated prior knowledge of an Arduino yet the results indicate interest (Q12) and the most learning. Further, most showed off their LED walls during the showcase while only a few showed their games. We hypothesize the immediate feedback afforded by the LED matrix on the board, and the LED light wall itself created a highly interactive tactile experience that the participants responded to. The game development module was most likely of interest as it was an expressed skill and pre-camp interest as discussed above. Given the number of video gamers in attendance, we observed that the nature of the interactions between campers was more competitive and at times confrontational compared to day 1. The security module was not well-received, most likely due to the day 4 fatigue discussed in the previous section. Day 2 on web development and AI was not as well-received but still adequate. These topics were most likely too advanced in how they were presented and the frustration with the setup, and less interactive after following on from the day 1 activities. Finally, the last topic (i) related to campus tours showed mixed results, but a wide degree of variability as evidenced by the relatively high deviations. We will need to review campus engagement activities more closely in the future to meet the goal of connecting young students to campus.

Related Work

Computing pathways remain a critical goal of educators and researchers in the wake of continuously advancing technology despite trends of unemployment and inflation in the US. Enrollments have been fluctuating, and according to Ford and Rokooei (2024) [5], the various impacts of recent events (COVID-19, unemployment, and inflation) may continue to affect university enrollments negatively without purposeful action. Additionally, impacts from COVID suggest community colleges will be a more versatile venue for students to pursue education and transfer into universities [2]. Degrees with racial and gender disparities will be subject to these effects, making early exposure to computing integral to broadening participation in computing at higher academic levels [1]. Engineering and computing have pursued methods such as workshops, after-school clubs, and, most commonly, summer camps [4][6].

K12 engineering camps continue to hold promise by promoting interest in a specific STEM field or STEM more broadly, thus potentially altering students' university pathways toward pursuing STEM majors and careers. Camps often have two goals: promote interest in a degree path and promote interest in applying to the camp's school [3][10][11][12][13]. Focusing on computing education and pathways, computing education literature leverages summer camps to teach and increase interest in computing, often to specific groups of people such as women and "different ethnic and socioeconomic backgrounds" in the US [6]. Computing camps also serve to increase chances for equitable learning [13], particularly when "Native American, Black, and Hispanic students [are] less likely to attend a school that teaches CS" [14].

K12 engineering camps are frequently structured, week-long summer programs evaluated by a pre-post survey. Some camps offer financial support to help underserved people attend the camp [12], while other camps may target specific demographics such as girls in middle school [1], high schoolers [3][11] and middle schoolers [10]. Camp sizes are typically small classroom sizes (i.e., 24), aiming to create a balance of instructors to students [12]. Additionally, camps are run by faculty at the university and supported by undergraduate [10] and graduate students [12] as mentors. Camp hosts aim to create a balanced diet of topics [11], rely on the previous day's learnings to build into the next day [12], and create a student showcase at the end of the camp for families [10]. Throughout the week, various activities blend into the program, from breaks and games to hands-on learning activities and tours [3][10][11][12]. Camps often use pre-post surveys to evaluate interest and knowledge in engineering, as well as collect feedback for future iterations of camps [3][10][11][12].

Conclusions and Future Work

The SUCCESS summer camp was motivated through the observations of faculty noticing the impact COVID [7] seemed to have on our students, and also our institution. Online delivery became necessary during the pandemic and now survives as a regular alternative modality, both in universities and in K12 education. At Arizona State University we noticed that 30-40% of our large online student cohort actually lives within a small commuting distance of campus. There are reasons for this of course as the personal situations of each student is unique, yet the overall percentage we find quite surprising. Online education is here to stay, is appropriate for many learners and personal situations, and our institution is heavily invested in this modality. But we maintain that on-campus in-person experiences are more engaging, both with the technology but more importantly with each other. SUCCESS was an attempt to reconnect potential future college students with the on-campus experience.

The SUCCESS summer camp at Arizona State University was designed as a breadth-first, survey-oriented introduction to the subfields of computing, with a goal to connect middle and high school students to campus life. This was challenging as the connective tissue between computing disciples is thin and getting thinner by the year. More computing professionals specialize in a subfield, and computing students are indoctrinated to this specialization through upper division electives in their degree programs. The common lower division in computer science (CS1, CS2, Data Structures and Algorithms) emphasizes competency development at the expense of giving students a survey of the field. Many try to address this, with some success, through a first-year project experience. The SUCCESS camps designed a thread through the camp experience based on pop culture (Stranger Things) but also the way the technical modules connected to one another.

Results from the camp were mixed but overall positive. While the number of participants was too small in the initial offering in July 2024, survey results do demonstrate that campers had fun, learned about computing, and tended to think positively about continuing personal study in the field of computing. The camp was challenging to deliver for technical and logistical reasons, and some modules clearly did not have enough interest and interactivity for the participants. The camp is scheduled to return in summer of 2025, and armed with this first experience we expect to address the deficiencies and continue connection students to the on-campus experience.

Acknowledgements

The SUCCESS camp development was supported by a generous corporate philanthropy grant from the State Farm Community Foundation.

References

1. Bhounsule, P., & Lima, C. (2022). Coding Camp for Middle School Girls Helps Improve Awareness of Science/Engineering Careers and Pathways. *2022 ASEE Illinois-Indiana Section Conference Proceedings*, 42127. https://doi.org/10.18260/1-2--42127
2. Bulman, G., & Fairlie, R. (2022). The Impact of COVID-19 on Community College Enrollment and Student Success: Evidence from California Administrative Data. *Education Finance and Policy*, *17*(4), 745–764. https://doi.org/10.1162/edfp_a_00384
3. Chen, K., Schlemer, L., Smith, H., & Fredeen, T. (2011). Evolving a Summer Engineering Camp through Assessment. *2011 ASEE Annual Conference & Exposition Proceedings*, 22.658.1-22.658.15. https://doi.org/10.18260/1-2--17939
4. Clarke-Midura, J., Sun, C., Pantic, K., Poole, F., & Allan, V. (2019). Using Informed Design in Informal Computer Science Programs to Increase Youths' Interest, Self-efficacy, and Perceptions of Parental Support. *ACM Transactions on Computing Education*, *19*(4), 1–24. https://doi.org/10.1145/3319445
5. Ford, G., & Rokooei, S. (2024). University Student Enrollments during the next Decade. *2024 South East Section Meeting Proceedings*, 45578. https://doi.org/10.18260/1-2--45578
6. Gardner, T., Leonard, H. C., Waite, J., & Sentance, S. (2022). What do We Know about Computing Education for K-12 in Non-formal Settings? A Systematic Literature Review of Recent Research. *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1*, 264–281. https://doi.org/10.1145/3501385.3543960
7. Gary, K. The Impact of COVID-19 in an Online and On-campus Software Engineering Program. *2024 ASEE Pacific-Southwest Section Conference Proceedings* (ASEE-PSW). https://doi.org/10.18260/1-2—46066.
8. Gary, K. A., Acuna, R., Mehlhase, A., Heinrichs, R., & Sohoni, S. (2020). Scaling to meet the online demand in software engineering. *International Journal on Innovations in Online Education*, *4*(1).
9. Gary, K., Sohoni, S., & Lindquist, T. It's Not What You Think: Lessons Learned Developing an Online Software Engineering Program. *2017 IEEE Conference on Software Engineering Education and Training (CSEET).* https://doi.org/10.1109/CSEET.2017.45.
10. Haugh, A., Lang, O., Thomas, A., Monson, D., & Besser, D. (2016). Assessing The Effectiveness of an Engineering Summer Day Camp. *2016 ASEE Annual Conference & Exposition Proceedings*, 26311. https://doi.org/10.18260/p.26311
11. Kunberger, T., Csavina, K., & Zidek, L. (2013). *Engineering Summer Camp for High School Students from Underserved Communities*.
12. Ling, M., Kelliher, M., Broyles, J., Solnosky, R., Park, H. Y., Li, L., Alkhatatbeh, B., & Ouma, L. (2023). Energizing High School Students Toward Building Design: A Summer Camp Experience. *2023 ASEE Annual Conference & Exposition Proceedings*, 43275. https://doi.org/10.18260/1-2--43275
13. Miller, J., Raghavachary, S., & Goodney, A. (2018). Benefits of Exposing K-12 Students to Computer Science through Summer Camp Programs. *2018 IEEE Frontiers in Education Conference (FIE)*, 1–5. https://doi.org/10.1109/FIE.2018.8659101
14. Ottenbreit-Leftwich, A. T., Dunton, S., Fletcher, C., Childs, J., Jeon, M., Biggers, M., DeLyser, L. A., Goodhue, J., Richardson, D., Peterfreund, A., Guzdial, M., Adrion, R., Ericson, B., Fall, R., & Abramenka, V. (2022). How to change a state: Broadening participation in K-12 computer science education. *Policy Futures in Education*, 14782103221123363. https://doi.org/10.1177/14782103221123363.
15. Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36-45.