ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE

Paper ID #37157

Student Perceptions of Programming Instruction in a Makerspace vs Synchronous Remote Environment

James E. Lewis (Associate Professor)

James E. Lewis, Ph.D. is an Assistant Professor in the Department of Engineering Fundamentals in the J. B. Speed School of Engineering at the University of Louisville. His research interests include parallel and distributed computer systems, cryptography, engineering education, undergraduate retention and technology (Tablet PCs) used in the classroom.

Nicholas Hawkins (Assistant Professor)

Student Perceptions of Programming Instruction in a Makerspace vs Synchronous Online Environment

All first-year students at the J. B. Speed School of Engineering (SSoE) at the University of Louisville (UofL) are required to complete a two-course sequence. The purpose of the two-course sequence is to introduce incoming students to the fundamentals and profession of engineering. The first course in the sequence is titled *Engineering Methods, Tools, & Practice I* (ENGR 110) and primarily focuses on introduction to and practice with fundamental engineering skills. The second course *Engineering Methods, Tools, & Practice II* (ENGR 111) is a makerspace-based course primarily focused on application and integration of the fundamentals learned in ENGR 110. ENGR 111 includes a variety of fundamental skills in its instruction, one of which is programming. Therefore, all disciplines of SSoE engineering students are exposed to the basics and applications of programming through this course sequence.

Programming instruction in ENGR 111 is designed to include relevant software development skills that students might encounter in the engineering profession. The students have learned initial programming skills in their ENGR 110 course through the Python programming language. In ENGR 111, students practice programming skills learned in ENGR 110 on two different platforms: Arduino Microcontrollers (Arduino) and Programmable Logic Controllers (PLCs). In normal face-to-face semesters, students are put into teams of 3 to 4 and given modules to develop and practice these skills (two for Arduino, two for PLCs).

Due to the COVID-19 pandemic, ENGR 111 was augmented into a synchronous remote course to avoid close proximity and shared tools in the makerspace. Arduino programming instruction was performed using Tinkercad (tinkercad.com), a website that allows for Arduino programming and circuitry simulations. PLC instruction was performed utilizing a free online PLC simulator website, “PLCfiddle” [1].

At the end of each semester, students take a survey on their perceptions of the course. Included in this survey are questions pertaining to programming instruction. These questions assess student confidence in programming and platform preference. Results of these questions from Spring 2019 (a makerspace iteration) and Spring 2021 (a remote iteration) are compared in this paper.

1. Introduction

The J. B. Speed School of Engineering (SSoE) at the University of Louisville (UofL) has a required first-year two-course sequence that all aspiring engineers are required to complete regardless of their engineering major. The courses in this sequence are *Engineering Methods, Tools, & Practice I* (ENGR 110), and *Engineering Methods, Tools, & Practice II* (ENGR 111). The ENGR 110 course is the first in the sequence and is focused on introducing first-year students to the profession and fundamentals of engineering. The second course, *Engineering Methods, Tools, and Practice II* (ENGR 111) is normally a makerspace-based course primarily focused on application and integration of the fundamentals learned in ENGR 110. Fundamental skills identified for all majors were basic programming and circuitry. Programming fundamentals are introduced to the students in the ENGR 110 course using the programming language Python.

ENGR 111 follows with students applying their developed knowledge of programming, now integrated with circuitry.

Early in the creation of the two-course sequence, a decision was made to expose students to more than one type of programming environment. This decision was to help the students to understand that core programming concepts are the same regardless of the programming language used; i.e. a *for* loop logically works the same in all environments even though the syntax may differ. ENGR 111 also uses Arduinos and their associated Arduino Programming Language (APL). The Arduino Uno is the chosen microcontroller for the ENGR 111 course since it is an excellent tool to teach basic circuitry, programming, and the interaction of the two.

Scaffolded lessons are used in ENGR 111 to introduce the students to circuitry, programming on the Arduino, and interfacing between an Arduino and circuits. The lessons began with basic stand-alone circuits using breadboard, basic components, and wires. Programming the Arduino is next. The programming lessons focus on basic programming concepts and how to interact with an Arduino. These programming lessons also discuss the similarities and differences of the APL to Python. Following the "Programming lessons", there is a series of activities to help the students create circuit(s) and program(s) that interact with each other.

Although the course structure of ENGR 111 is the antithesis of a remote pedagogical setting, course administrators decided to redesign the ENGR 111 experience as a remote delivery due to the reality of the Covid-19 pandemic. The definition of "remote delivery" institutionally is a course that meets online, and synchronously. The use of the makerspace was not feasible due to the close-proximity nature of numerous hands-on activities for as many as 96 students per class, and the provision of multiple shared tools amongst six different classes. The remote designation challenged instructors in retaining a heavy focus on teamwork, in addition to the active learning environment of the conventional course iteration. The following sections provide further detail in these efforts for ENGR 111 curriculum specific to programming.

2. Technology

There were several technology requirements identified as essential and practical in meeting the objectives for this course redesign. These included a computer, access to the internet, Microsoft (MS) Teams, and a Tinkercad [2] account. These requirements were deemed acceptable since:

1. all SSoE students are required to have a Tablet PC,
2. all Louisville students have access to the Microsoft suite of applications including MS Teams,
3. Tinkercad allows free account creation.

For the internet connection requirement, students have the option of free access across a very large area of campus property, and the instructors were confident that any student living off campus had internet access available.

2.1. Additional Tinkercad Details

Tinkercad is a free online collection of software tools provided by Autodesk. Many people are only aware of Tinkercad as an online 3D modeling platform, however in 2017 Autodesk merged its “123D Circuits” interface into Tinkercad [3] [4]. This makes Tinkercad an ideal platform to use for circuitry and Arduino programming. Therefore, the entire Tinkercad site provides powerful simulation software for a variety of purposes. Autodesk defines Tinkercad as a free online collection of software tools that help people all over the world think, create, and make [2]. The scope of this paper solely focuses on the electronics environment. This environment allows for design, programming, and simulation with Arduino boards and circuits.

Tinkercad’s simulation workspace (**Error! Reference source not found.**) for Arduino Circuits and programming has a component library that is based on existing Arduino kits.

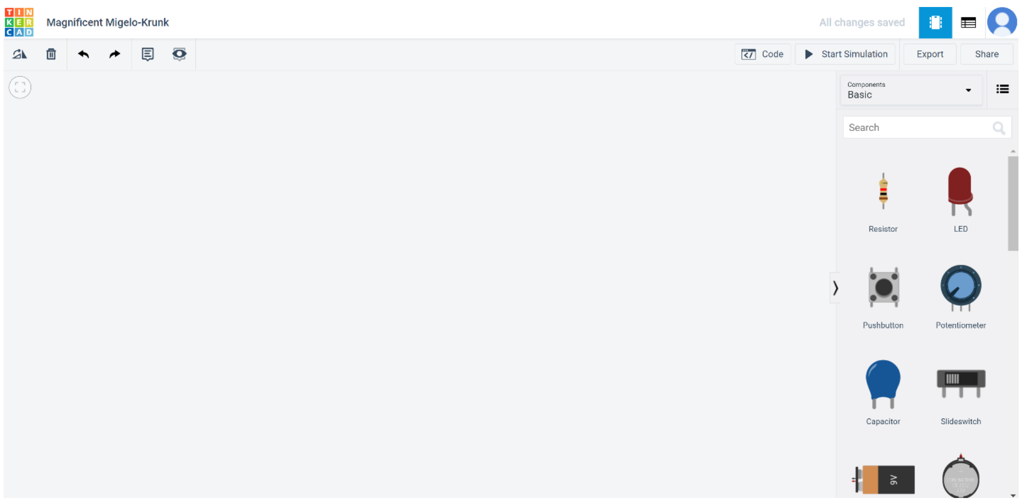

Figure 1: Tinkercad Simulation Workspace

There are many Arduino starting points inside of Tinkercad, one of the basic ones already wired to a breadboard is shown in Figure 2.

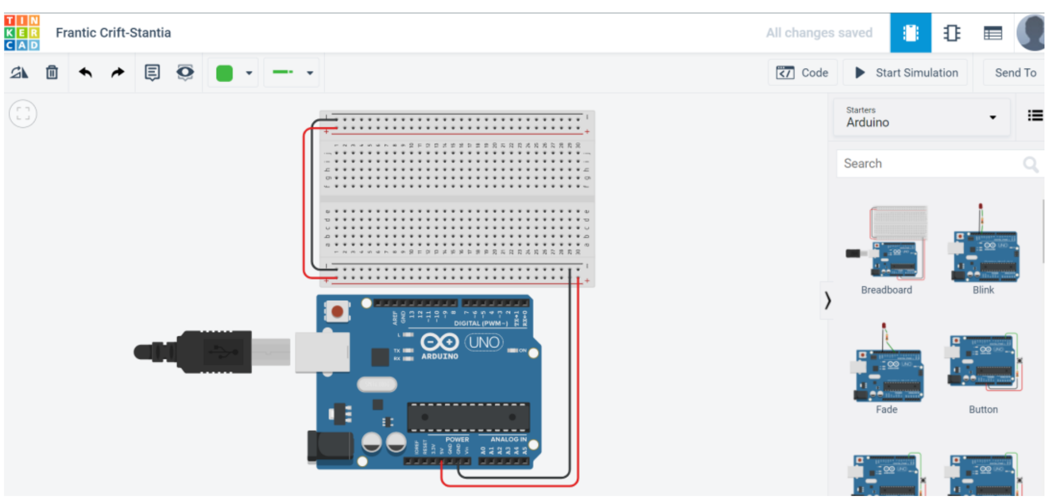

Figure 2: Tinkercad Arduino Starter

Selecting the “Code” button in the top right will open up an area for coding the Arduino. The code section allows for block coding (Figure 3), text+block coding, or just text coding (Figure 4).

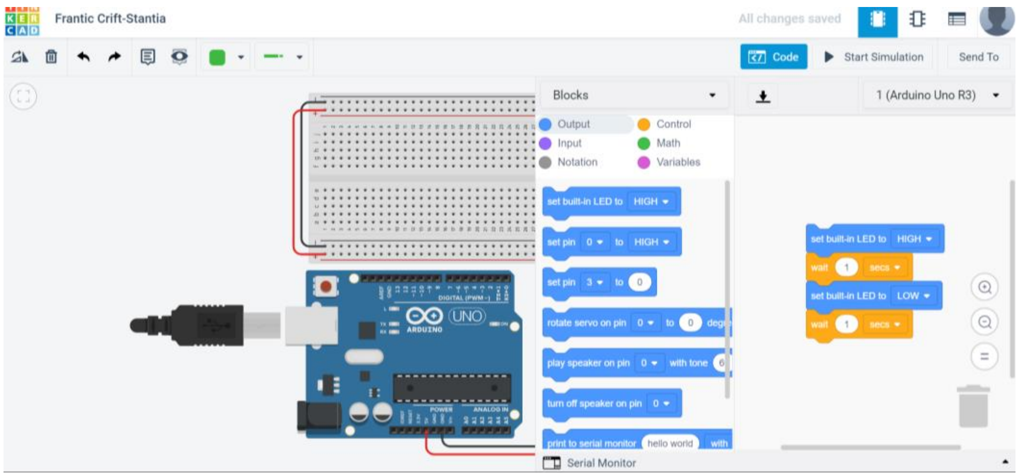

Figure 3: Arduino Block Coding in Tinkercad

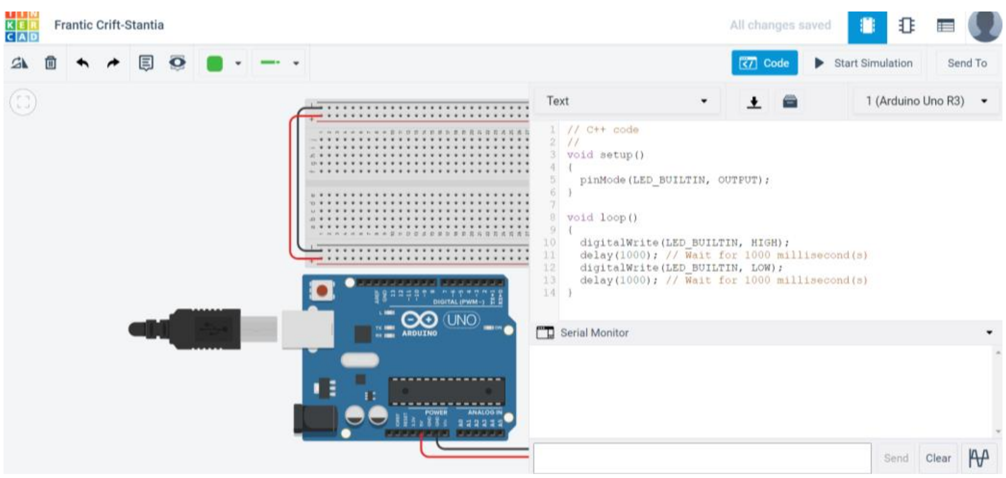

Figure 4: Arduino Text Coding in Tinkercad

2.2. Additional PLCfiddle Details

PLCfiddle is a free online software tool that provides a PLC simulation and ladder logic programming environment without any software to install [1]. It is an ideal environment for students to learn and try different aspects of ladder logic programming without the associated costs in purchasing and maintaining PLCs and the associated software needed. Figure 5 shows the landing page for the simulation. The tabs across the top allow for more advanced items to be added then the simple contacts shown. The math tab is shown in Figure 6 and everything is draggable in the browser window. After creating code, the “Other” tab allows you to save your project and access it by a created URL, like the following for URL example: https://www.plcfiddle.com/fiddles/fdd9aabb-102c-4f88-aebf-1ca34cd173c6

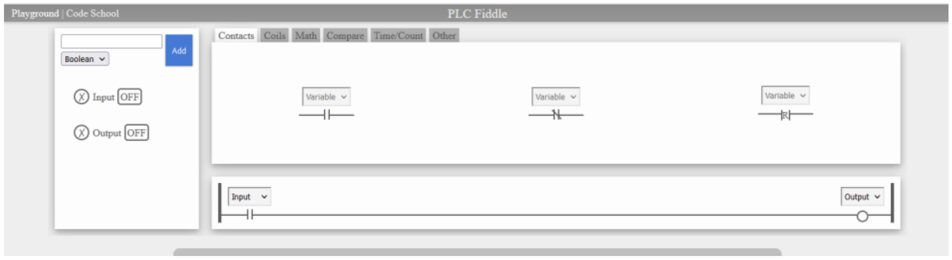

Figure 5: PLCfiddle Starting Page

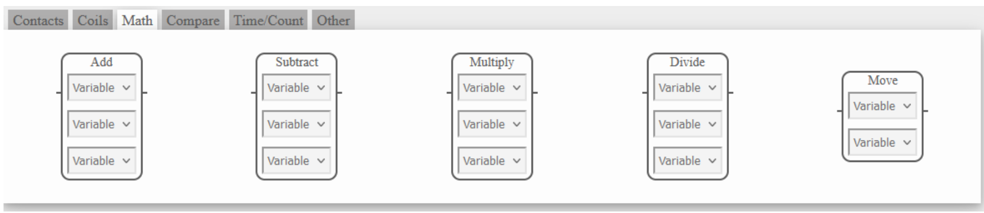

Figure 6: Ladder Logic Math Tab

3. Assignment Descriptions

For the (2021) remote iteration of ENGR 111, the Tinkercad environment was employed for multiple course topics including: “Introduction to Arduino Programming”, “Interacting with the Arduino”, and “Programming with Circuitry”. These classes were taught in pre-COVID iterations of the course utilizing a physical Arduino and circuitry components. However, in the transition from typical to remote instruction, these labs continued to include the same scaffolding of these programming activities. Regardless of iteration, these classes were performed by groups of 3-4 students working as a team.

The purpose of these programming assignments was to help students better understand the real-world applications of programming. These lessons start by building students’ skills in general programming with an Arduino: utilizing input and output (I/O) commands, applying Boolean logic, and defining basic looping structures. Then, students transition to applying these skills alongside the analog and digital I/O ports on the Arduino. This requires the use of conventional circuitry components such as light-emitting diodes, resistors, potentiometers, and pushbuttons to interact with the logic they build into their code.

While the pedagogical content of each lab is analogous between the face-to-face and remote semesters, one major difference provided by the change to remote is the efficiency of exposure. In a typical environment, students in a group stand around a table sharing one set of equipment (i.e. one breadboard, one set of components, one Arduino, etc.). This means that during the programming instruction, students must share an Arduino, which often leads to one or more team members watching rather than programming the Arduino. With Tinkercad, each student creates an account and can program an Arduino individually while helping their team. This allows each student to approach programming task(s) with their own algorithm, providing more individual experience instead of relying on one teammate that has more programming experience. Student teams then work together to finalize a working program.

After completing the programming assignments, students were required to demonstrate the functionality of their programs. In the Spring 2021 semester, students were instructed to complete the programs themselves inside of Tinkercad while discussing with their teammates.

In addition to these Arduino programming assignments, students also received a tutorial in utilizing PLCs in a lab called "PLC Familiarization". In the Spring 2019 semester it was spread out over two class sessions, whereas in the Spring 2021 semester it was condensed into one. Both iterations of this instruction covered the same topics: the basics of ladder logic, timer circuits, and counter circuits. The desire was to help show students the ways in which PLCs are used in industrial applications. However, there is a large distinction between how students learned to program PLCs in the two course iterations. When in person, students interacted with a physical controller that had lights and buttons that triggered the ladder logic. During the remote iteration, students only saw what was simulated on PLCFiddle.

4. Comparison Results

As mentioned, learning modules pertaining to programming and circuitry covered the same objectives in Spring 2019 and Spring 2021, with the only difference being in the programming environment used (Arduino IDE vs Tinkercad). This allowed for a comparison of student performance to see if any impact was made by the utilization of Tinkercad in place of using a physical Arduino.

At the end of each semester, students were asked to complete a survey with questions spanning the entire curriculum in the course. Included in this survey were three questions regarding programming preference and confidence, as given below.

Q1: Which of the two programming methods used in ENGR 111 are you most comfortable interpreting?

- Arduino (line-based programming)
- Programmable Logic Controllers (PLCs, ladder-logic)

Q2: Express your opinion pertaining to the two programming methods in this course:

- I preferred learning PLC programming over Arduino programming.
- I preferred learning Arduino programming over PLC programming.
- I wish I had only received PLC programming instruction.
- I wish I had only received Arduino programming instruction.
- I am glad I learned both Arduino programming and PLC programming.

Q3: Rate your CURRENT confidence level in basic programming (using any language/software):

- Not confident at all
- Slightly confident
- Somewhat confident
- Very confident
- Extremely confident

The results for Q1 are shown in Figure 7.

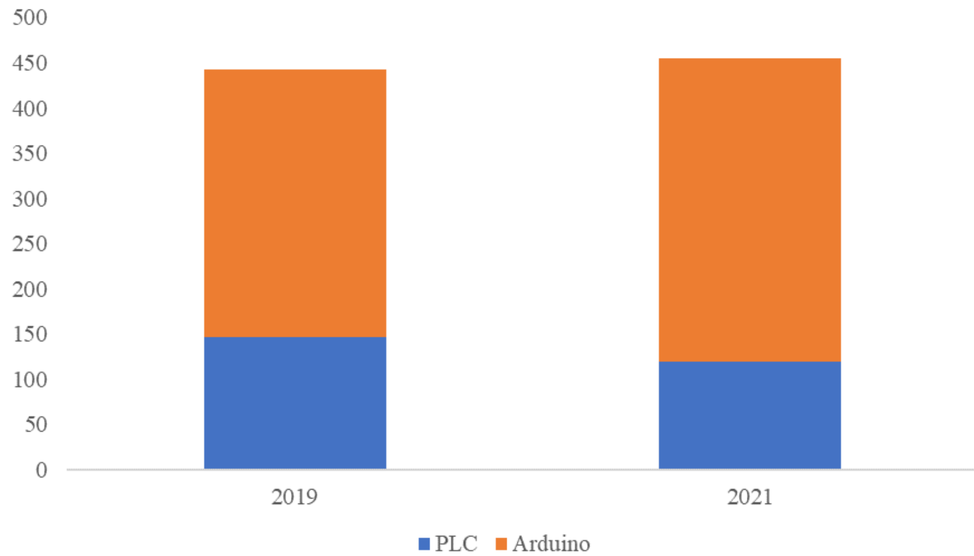

Figure 7: Survey results for Q1 regarding student comfort with programming platform.

Students are typically more comfortable with Arduino programming than PLC programming as evident in the data for both cohorts. However, the ratio of students that are more comfortable with Arduino programming increased from 66.8% in 2019 to 73.6% in 2021.

The results for Q2 are shown in Figure 8.

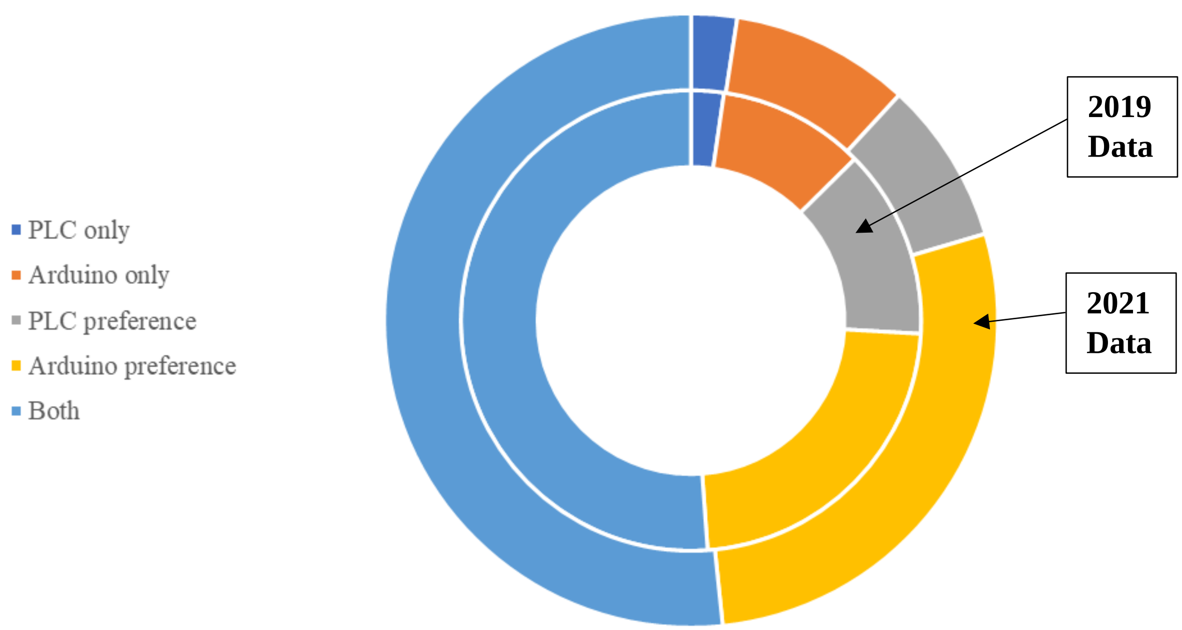

Figure 8: Survey results for Q2 regarding student preference with programming platforms (inside circle is 2019 data, outside circle is 2021 data).

These results clearly show that student preference has not changed much between the two cohorts. The only shift that has occurred is a slight increase in students that preferred learning Arduino programming over PLC from 22.9% in 2019 to 27.9% in 2021. Accordingly, preference for PLC over Arduino programming declined from 13.3% in 2019 to 8.6% in 2021.

The results for Q3 are shown in Figure 9.

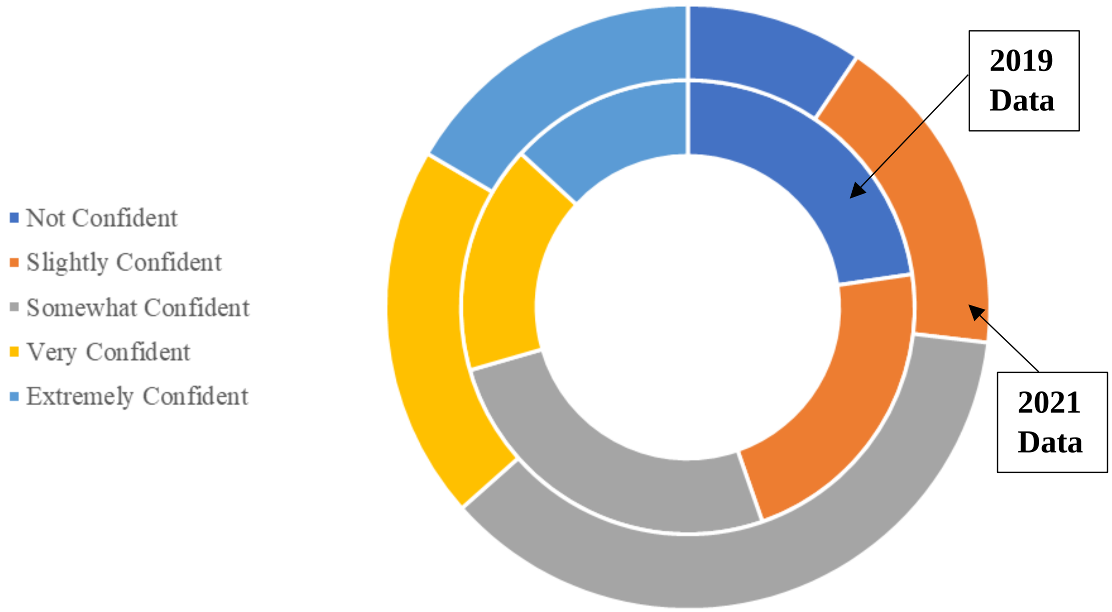

Figure 9: Survey results for Q3 regarding student confidence with programming (inside circle is 2019 data, outside circle is 2021 data).

These results indicate that student confidence in programming has improved overall. There is significant decline in the "Not Confident" selection (a 57% decrease) as well as large gains in both the "Very Confident" (26% increase) and "Extremely Confident" (29% increase) selection.

5. Conclusion

The ENGR 111 course at the J. B. Speed School of Engineering (SSoE) at the University of Louisville is typically a laboratory-based, hands-on course taught in a makerspace setting. Due to the COVID-19 pandemic, the course was modified to remote instruction for the Spring 2021 semester. The instructors maintained the inclusion of programming with circuitry in the remote setting utilizing Tinkercad and PLCfiddle.

Results indicate that student preferences regarding programming methods have largely remained constant between the two course iterations, with overall programming confidence improving when moving to remote instruction. Students likely maintained their comfort with and preference for Arduino programming due to their exposure to similar programming language in previous schooling, most notably ENGR 111's prerequisite ENGR 110, in which Python programming is taught.

The slight decrease in student opinion regarding PLCs is possibly due to a lack of physical interaction with the devices in remote instruction. During the 2019 course, students coded PLCs while having the controller on their tables. Students could press buttons to see how their ladder logic would react in real time. In 2021 instruction shifted to using a web client which removed this physical interaction, which appeared to be comparatively unpopular.

Overall, student perception of programming did not become worse upon transitioning to remote instruction.

References

[1] "PLCFiddle," 31 01 2022. [Online]. Available: https://www.plcfiddle.com/.

[2] "Tinkercad by Autodesk," [Online]. Available: http://tinkercad.com. [Accessed 27 01 2022].

[3] T. Makerspace, "AutoDesk 123D Circuits," [Online]. Available: https://makerspace.tulane.edu/index.php/AutoDesk_123D_Circuits. [Accessed 3 March 2021].

[4] Autodesk, "Autodesk 123D apps," [Online]. Available: https://www.autodesk.ca/en/solutions/123d-apps. [Accessed 3 March 2021].

[5] S. Fitzgerald, M. Shiloh and T. Igoe, Arduino Projects Book, Torino: Arduino LLC, 2012, p. 171.