Integration of Coding into Classrooms via Easiest Language to Learn

Logen Autry, Computer Science; Dr. Sonia Stephens, CECS

Research Statement:

The aim of this research is to clearly define a correct answer to what is the most approachable and easiest coding language to pick up as newcomers. The research will primarily focus on elementary grade students and how and when each language taught to them is picked up. The purpose of this knowledge is to inform educators on how to teach young students one of the key skills needed for overall technological literacy. With the current rate at which technology is being integrated within our lives, understanding how to teach it is the difference between our youth falling behind or keeping up with current logical standards. This keeping up will keep our colleges full and our country progressing rapidly through development.

Body of Knowledge:

Our current educational system lacks a taught foundation in coding. This lack of foundational knowledge keeps us behind internationally and will eventually lead to what can only be described as a severe STEM gap. Portlance states this best in his quote:

> compared to their international peers, US students lag behind in elementary and secondary math and science performance, are less likely to choose a STEM-related major in college and are more likely to eschew science and technology career fields even when they are qualified applicants. (Portlance 2015)

This is worrying as it creates a divide between us and the rest of the world that could end with our economy sputtering out. As Stated before, this lack of skilled workers is what defines a STEM gap. As the Center for Strategic and International Studies puts it "…STEM gap will negatively impact growth…The nation's economic and scientific leadership will fall behind unless we prepare students to become researchers in STEM…" (Athanasia 2022). The growth in this quote refers to our economy. We can already see the stalemate in both our economy and in STEM workers as with only 11% of U.S. adults holding a degree in some sort of stem field, the 30% of job openings in the U.S. goes severely unfilled (Athanasia 2022). This leads to the inevitable of firms outsourcing work to other countries, and in turn majorly hurting both our job market and economy.

However, there is a clear-cut way of preventing this dire problem from furthering. We solve this problem through the development of coding skills within our youth. Youth is the key word here as the sooner the better as proven by the quote "since students often decide to pursue a career in the sciences by eighth grade" (Mladenović et al., 2016). Our students learning the fundamentals of computer science helps widen their career opportunities and develop key concepts in navigating our digital landscape and solving STEM problems. This will also help to solve the STEM gap as we harbor more motivation within students who succeed in these programs to keep pursuing this knowledge and eventually enter the STEM fields.

The two languages I propose to study as prospective teaching languages are Scratch and Python. Python will be tested for its simplicity as a text-based programming language. Scratch will be tested for its ease of use for teaching concepts in computer science and its visual style. Both prove as worthy prospects as we move into research methods.

Research Methods:

Current studies done on these topics are either outdated or performed in other countries. However, the methods they used will be useful in defining my plan of action. The most common languages found within my literature review were the ones states above, Scratch and Python. This is most certainly for a reason as Mladenović states "These languages eliminate syntax error problems, so the student is able to focus on problem solving while programming becomes more attractive to student… positive affect on motivation for programming" (Mladenović et al., 2016). Each has their own benefits with Scratch providing a building block style of coding that allows for ease of use while python provides many libraries with plethora of functions to do any sort of task without having to fully understand the code behind it.

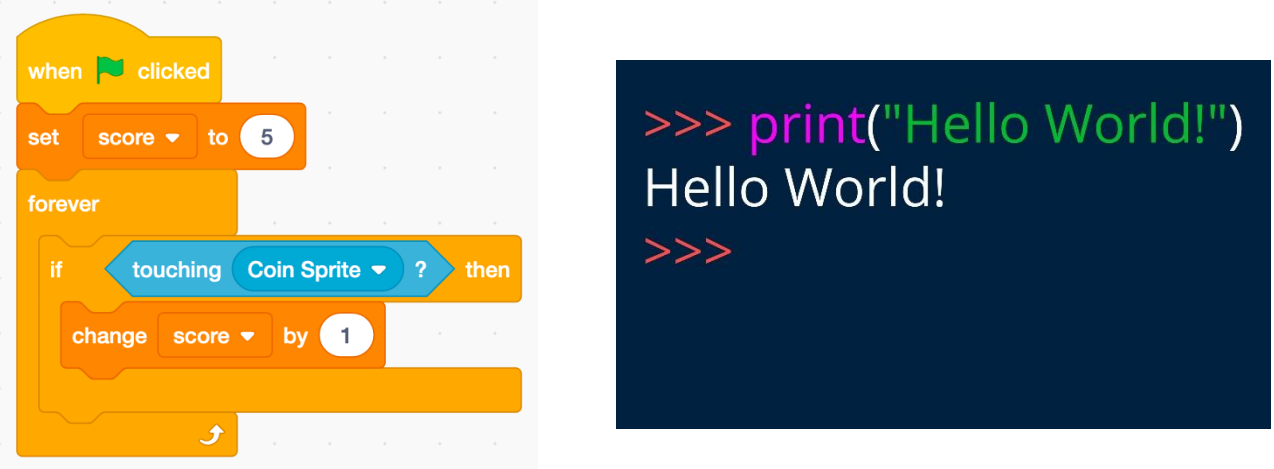


Figure 1(Left) demonstrates the block syntax provided by Scratch while Figure 1(Right) demonstrates on of the numerous functions provided by Python to simplify the coding experience.

First, I will implement a curriculum based on either python or scratch in order to test the ease of use and knowledge retention amongst students learning. There will be three python groups, three scratch groups, and three control groups in which they are taught nothing and then receive the same end of code exam. This will allow for easy comparison between python and scratch's ease of use using the control group as a baseline. This exam will test for knowledge of computer science principles and ability to apply these principles and will act as the main source of data for this study. These quick classes will be taught in either parts of math classes or electives and be carried out within local elementaries surrounding the University of Central Florida. The idea for an assessment and some of the questions comes from a similar study titled *Coding, Collaboration, and Computational Thinking* by Nicole Halmelburg. She provides a useful strategy for teaching through the usage of challenge packets and engaging content. One such challenge packet goes as follows:

> The goals for this unit (Appendix A) were for students to be able to properly use blockly code on Scratch to create a video game of their choosing, to collaborate with others to debug codes, and to be able to explain their use of the Engineering Design Process during each step of the video game coding process. (Hamelburg, 2019)

I will conduct the study using such engaging content to cement this knowledge and allow for the most accurate results on ease of use and retention within these coding languages. Lastly, I will determine ease of use through number of questions on average and overall class struggle through lessons (rated on a scale of 1-5 by me and teachers). Data from these reviews will help find a correlation between ease of use and knowledge retention which is important for determining which should be taught in schools.

IRB Statement:

This research study will require approval via the Institutional Review Board.

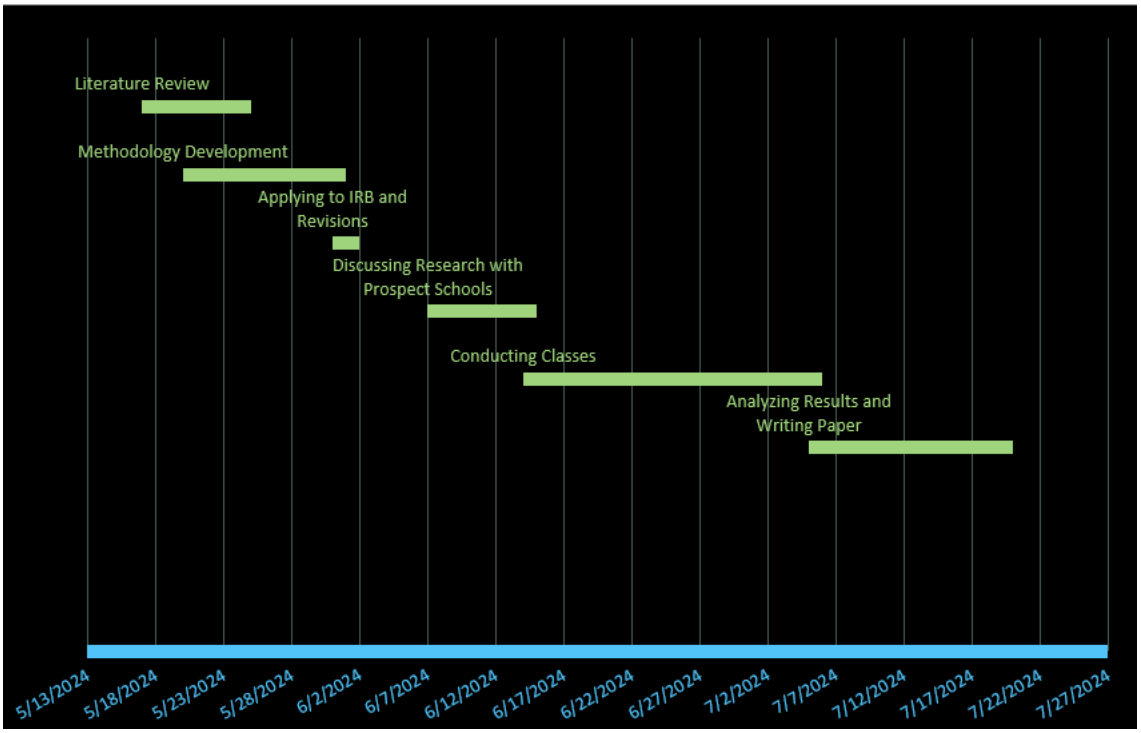
Expected Outcomes:

Through the research I have conducted so far, I feel that scratch may have a higher ease of use for younger audiences while those in 4th grade and up may find python to be more enjoyable. This is best understood through the very purpose of scratch within the quote “Scratch is a free programming language developed by MIT that makes it easy to create interactive stories, animations, games, music, and art” (Tsikalakis et al., 2023). Scratch is used to easily create games and art however I feel the childish nature may not appeal to the older students which might be where python becomes easier to both use and retain information on for them. These older students I hypothesize will feel more motivation to understand coding through pythons more robust libraries allowing for them to explore concepts more intriguing to their individuality. Dr. Mladenović further pushes this point through the quote “Python meets all requirements for mini-language and is indicated as one of the most appropriate starting textual programming languages” (Mladenović et al., 2016). Its reputation as the most appropriate starting language in the textual field is what will set it apart from scratch and possibly appeal more to the older students.

Once I have my data collected, I will proceed to write a research paper detailing my findings using graphs, statistics, and the exam results collected. I will make sure to provide any cases of outliers and hypothesis for such. At the end of my paper, I will write my hypothesis for what will occur if we implement such language into the U.S. school system and an estimated benefit. I will also provide implications for further studies.

I hope to provide insight on how to teach such languages to students and what language should be taught to each age group. This information should help to develop a curriculum which can be integrated to further our youths understanding of computer science and grow our population of STEM enthusiasts and future employees. In turn this will greatly improve our job market and economy. I hopefully will improve the lives of those students surrounding our UCF community as well and maybe even harbor interest in STEM amongst them.

Timeline:


Literature Review
Methodology Development
Applying to IRB and Revisions
Discussing Research with Prospect Schools
Conducting Classes
Analyzing Results and Writing Paper
5/13/2024
5/18/2024
5/23/2024
5/28/2024
6/2/2024
6/7/2024
6/12/2024
6/17/2024
6/22/2024
6/27/2024
7/2/2024
7/7/2024
7/12/2024
7/17/2024
7/22/2024
7/27/2024

Works Cited

Athanasia, G. (2022, April 1). The U.S. Should Strengthen STEM Education to Remain Globally Competitive. *Center for Strategic & International Studies*. https://www.csis.org/blogs/perspectives-innovation/us-should-strengthen-stem-education-remain-globally-competitive#:~:text=With%20not%20enough%20workers%20in,the%20long%20term%20as%20well.

Hamelburg, N. (2019). *Coding, Collaboration, and Computational Thinking* (dissertation). Proquest.

Mladenović, M., Krpan, D., & Mladenović, S. (2016, July 6). *Introducing programming to elementary students novices by using game development in Python and Scratch*. EDULEARN16 Proceedings. https://library.iated.org/view/MLADENOVIC2016INT

Portelance, D. J., Strawhacker, A. L., & Bers, M. U. (2015, August 5). *Constructing the SCRATCHJR programming language in the early childhood classroom - international journal of technology and design education*. SpringerLink. https://link.springer.com/article/10.1007/s10798-015-9325-0#Abs1

Tsikalakis, G., Neofotistou, Eleftheria., Kontogiannakis, Emmanouil., Pau, A.-M., Caglayan, E. M., Fernandez-Canez, R., & Perez-Accino, M. (2023, October 11). *Sustainable development as an education tool for primary school students through scratch code*. E3S Web of Conferences. https://www.e3s-conferences.org/articles/e3sconf/abs/2023/73/e3sconf_iced2023_06007/e3sconf_iced2023_06007.html