Virtual App Development for Adolescents During COVID-19

Kaylah Mackroy
Morehouse College
Atlanta, Georgia, USA
kmackroy@scmail.spelman.edu

Kinnis Gosha
Morehouse College
Atlanta, Georgia, USA
kinnis.gosha@morehouse.edu

ABSTRACT

Minority students are not entering computing fields due to inadequate exposure in K-12 curricula. Online computing environments are effective at exposing more minority students to computing concepts before college. An HBCU hosted a virtual camp during COVID-19 to teach minority adolescent students the fundamentals of app development using MIT App Inventor, an app-development platform that allows its users to build fully functional apps for smartphones and tablets. The camp aimed to foster youth innovation and creativity through empowering students to create rather than simply use technology in their lives. Participants in the program showed an increase in wanting to pursue ongoing computing education.

CCS CONCEPTS

• **Social and professional topics** → **Race and ethnicity**; **Adolescents**; **K-12 education**; *Informal education*; *Computational thinking*.

KEYWORDS

Historical Black Colleges and Universities, COVID-19, Virtual Learning, Adolescents, Computer Science Education, Racial Minorities

ACM Reference Format:
Kaylah Mackroy and Kinnis Gosha. 2021. Virtual App Development for Adolescents During COVID-19. In *2021 ACM Southeast Conference (ACMSE 2021), April 15–17, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3409334.3452089

1 INTRODUCTION

Computer programming is one of the most valuable skill sets of the twenty-first century; however, computing education remains a rare commodity in K-12 curricula; this is especially true in schools with the highest percentages of minority students [10, 11]. Underrepresented students face structural and social barriers in access and exposure to Computer Science [6]. More avenues for access to Computer Science are needed for minority populations. Research shows online programs that focus on games design increase computational thinking and effectively introduce programming skills and concepts [7, 8]. This research reports on the Virtual App Inventor Camp, an HBCU (Historically Black College or University) led virtual summer program with a majority-minority cohort, that occurred amid the COVID-19 pandemic. This virtual camp utilized MIT App Inventor, an app-development platform that allows its users to build fully functioning apps for smartphones and tablets. This research further explores how adolescent students in an informal virtual classroom think about their daily interactions with apps, their ability to learn to program, and their possible trajectories as future app developers. The camp aimed to foster youth innovation and creativity through empowering students to create rather than use technology in their lives.

ACMSE 2021, April 15–17, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8068-3/21/04.
https://doi.org/10.1145/3409334.3452089

2 BACKGROUND

Students begin devising their career and educational plans in middle school; therefore, implementing computing concepts during or before middle school would direct more students into the field. [7, 8]. Exposing youth to computing concepts early on is essential in piquing student interest and a divisive tool to diversifying the field [7]. More minorities are not entering computing professions due to inadequate exposure to Computer Science (CS) education in K-12 curricula [1]. Experts in computer science education acknowledge skill development before college as the key to success in the field [2]. Minorities face disparities in attaining the skills necessary for success due to social and structural barriers such as; stereotypes, lack of role models, unwelcoming learning environments, inequitable school resources, lack of access to basic CS courses, and few opportunities for out-of-school computing activities [1, 6]. To shrink the diversity gap in CS, tech companies and educational institutions need to make computer science more engaging and accessible to underrepresented groups [1]. Black students are more interested in learning computer science than their white counterparts; Black students are also more confident in their ability to learn computer science [1, 11]. Therefore, the diversity gap is not due to a lack of interest but a lack of opportunity and access [1, 11].

Adolescents typically use technology more at home than in academic environments; however, with the onset of the current pandemic due to COVID-19, a growing reliance on technology in both education and social contexts is prevalent [3, 5, 9]. An effective way to introduce CS concepts in this environment is by utilizing programming platforms that are user-friendly and have the potential to lower the cognitive threshold for novice programmers[8]. In addition to user-friendly platforms, minority adolescents benefit from having models and mentors that look like them working in these computing environments. This representation substantially boosts their self-efficacy and sense of belonging [4, 8]. Unlike many other online coding programs, the Virtual App Inventor Camp employed HBCU near-peer Computer Science majors and affiliates with computing backgrounds to assist student learning. Near-peer instructors allow culturally relevant cues and further bolster the learning process while simultaneously confronting stereotypical

images and misconceptions of minorities in computing to create more engaging learning environments.

3 METHODS

The Virtual App Inventor Camp was a twelve-day program held from Monday through Friday from July 20th until July 31st. The camp consisted of two sessions: a morning session conducted from 9 am to 12 pm and an afternoon session conducted from 1 pm to 4 pm.

3.1 Participants

Applicants in grades six through eight within the United States were invited to apply. In total, the camp admitted 74 students. Ten participants either did not confirm their spots or did not complete the necessary steps required to participate. As the camp progressed, six more participants stopped attending. There were few occurrences (>5) where students were so late that class materials had to be emailed to them to remain on course with their class; to refrain from using class time. 24% of the students were female, and 76% were male. 95% of the students reported being Black, Afro Caribbean, or African American. The students' grade levels ranged from 5th to 10th grade. Student ages ranged from 10 to 16 years old. The average age was 12 years.

There were six camp African American instructors: four male undergraduate HBCU students majoring in Computer Science, and two female graduate students with computing backgrounds (at the time, near-peer female computing students were unavailable). The program coordinator was a female Ph.D. candidate in Human-Centered Computing.

3.2 Materials/Class Information

The camp required a laptop or tablet, a stable internet connection, a Gmail account with student access, a quiet work area, and a Zoom account to participate. Students created MIT App Inventor accounts on the first day of camp. Several Zoom functions were employed, including chat, screen-share, break-out rooms, and virtual backgrounds. Students utilized MIT App Inventor, a web-based programming environment that allows users to employ blocks-based visual programming (similar to Scratch) to design and build mobile apps. The camp curriculum consisted of seven cumulative units:

- `Unit 1. Hello, It's Me`: introduced students to the App Inventor platform. In this lesson, students created apps with images and voice recordings. Upon pressing a button, the recordings would play.
- `Unit 2. My Piano App`: students programmed a piano app with multiple buttons and corresponding musical notes. Students learned to abstract through the use of procedures: sequences of blocks grouped together.
- `Unit 3. Music Maker`: involved designing graphical user interfaces for a band app. For this app, students programmed multiple different instruments. Users could start and stop the music and record their music for playback.
- `Unit 4. Food chase`: students block coded a chase and catch food game. The user's character grew in size as it ate food objects. In this lesson, students learned learned how to animate objects on the screen and control movement through user interaction.
- `Unit 5. Make a Game`: Students applied previous concepts to program their own gaming app.
- `Unit 6. Two Button App`: Students learned about CloudDB, a component that allows users to share data and play over multiple devices. Students programmed a two-person game that can be used over different devices.
- `Unit 7. Sketch and Guess`: Students employed CloudDB in programming a doodle game. One user would draw on their device, and other players could see the drawing and guess what it was from their device.

Instructors administered online quizzes and Learning attitude surveys at the end of each lesson. These camp components assessed student attitudes towards computing and how well the students retained the information taught.

3.3 Procedure

The program director collaborated with the Marketing and Communication Offices at the host HBCU in designing and disseminating the Virtual App Inventor Camp flyer and other camp information via social media (Figure 1). The college's Office of Alumni Affairs advertised the camp to its alumni via email.

There was a two-step process application process. Assisted by their parents or guardians, students would complete the Zoom application to apply to the camp. Three need-based payment options were available $600 , $300 : half-tuition scholarships, and $0 : full-tuition scholarships.

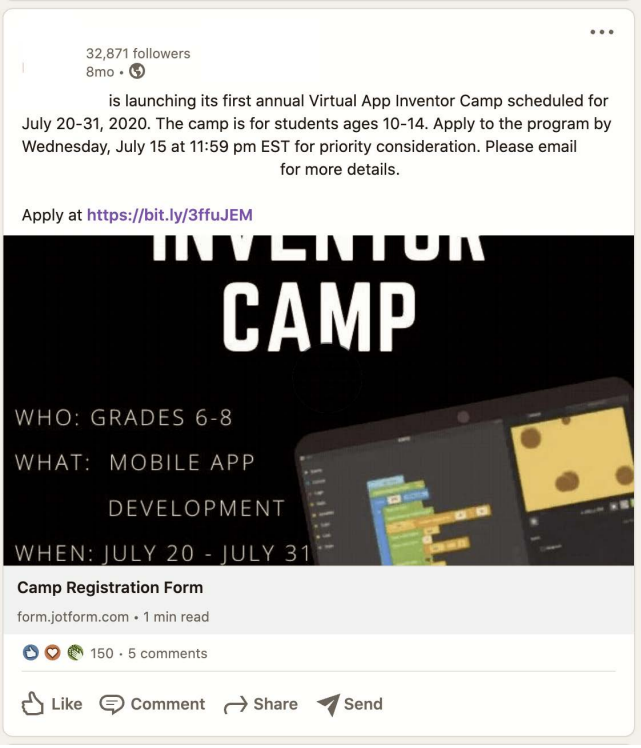

Figure 1: Social Media Outreach: Linkedin

Upon completing the application, students were accepted based on their application completeness and response to the student essay question. Accepted students were then required to complete the confirmation process that included payment, class time, device type, internet connection, and Zoom account information. Once complete, the students received their Zoom class information. The program coordinator separated the classes by device/operating system. This organizational strategy was set in anticipation of device specific issues.

All class files and links were sent via Zoom. At least one undergraduate instructor was assigned to every class. Instructors utilized the break-out rooms for students who needed additional help. The instructors utilized the Technical Support room to assist those with technical difficulties. Students were encouraged to practice the assignments and learn their devices on their own. Instructors allowed the class to teach students who struggled with concepts. Pre and post-surveys were administered to students and instructors, and on the final day, instructors conducted semi-structured interviews. Certificates of Completion were administered to students who completed the camp.

4 RESULTS

At the beginning of the camp, approximately 90% of students responded "yes" when asked if they wanted to pursue a college education. The other 10% answered "maybe". About 21% of students were unfamiliar with what a computer scientist was at the beginning of the camp; however, 93% were interested in pursuing a computer science degree. 97% of students said they would be interested in receiving a coding certificate. Instructors conducted semi-structured interviews to determine what the students liked and disliked about the camp. Results indicate that students thought highly of the Virtual App Inventor Camp. Recurring responses included the students wanting to learn and discuss new skills each day instead of continuing the previous day's lesson and not dedicating class time to students who had fallen behind. Instructors administered quizzes and learning attitudes surveys at the end of each unit. The average assessment scores for units 1-4 are as follows: Unit 1. Hello, It's Me: 2.62/4 with 44 survey responses, Unit 2. My Piano App: 0.39/3 with 44 survey responses, Unit 3. Music Maker: 1.62/2 with 43 survey responses, and Unit 4. Food Chase: 2.44/4 with 25 survey responses.

After completing the camp, about 86% of students still wanted to pursue a college education. The remaining students answered maybe. No one answered no. After this course, about 92% of students knew what a computer scientist was, and 96% of students that took the post-survey said they were interested in pursuing a degree in computer science, a 3% increase from the start of the camp. The same percentage, 97% of students were interested in receiving a coding certificate as in the pre-survey. The results of this study include all student entries despite attendance.

5 DISCUSSION

Results from this program indicate a desire from students to want to pursue ongoing computing education. This program did not demonstrate the effectiveness of the students at retaining the information taught. Further actions dedicated to developing this program include: effectively monitoring students and convincing them to continue with the program. The ability to provide devices to students would greatly benefit programs like this one (geared towards underrepresented populations) so that requiring technology would not be as big of a hindrance. Future program efforts include having a course that will offer class credit as a motivational incentive.

6 LIMITATIONS

The virtual environment imposed several limitations: inadequate student attentiveness, uncontrollable work environments and technical issues, and absences throughout the camp. A few of the students utilized different devices during the camp; this further disrupted the classes by offsetting the organizational structure. Students also reported having to share devices with members in their household.These issues caused these students to fall behind. There were 58 responses to the student pre-survey and 24 responses to the student post-survey. The instructors planned to teach seven units; however, each class finished on different lessons. The instructors gauged their class pace and adjusted accordingly.

7 CONCLUSION

This camp cannot flatten every structural and social barrier minority adolescents face in equitable access to computer science education. The camp experienced many trials. Many of its students did not complete class requirements. Many students stopped attending class altogether. Teaching amid the COVID-19 Pandemic posed many challenges alone. The Virtual App Inventor camp was mere exposure to computing for many of its students. More programs like this are needed to continue to prepare young minds to go beyond the monotonous use of technology. The Virtual App Inventor Camp has shown that many of its students want to pursue ongoing computing education despite the short introduction. Further research is necessary to highlight the outcomes of underrepresented students exposed to computing at young ages.

REFERENCES

[1] 2016. Diversity Gaps in Computer Science. https://news.gallup.com/reports/196331/diversity-gaps-computer-science.aspx.
[2] 2020. Computer Science Before College. https://www.computerscience.org/resources/computer-science-before-college.
[3] B. Barron and Y. Kafai. 2006. Clubs, Homes, and Online Communities as Contexts for Engaging Youth in Technology Fluency Building Activities. *ICLS 2006 - International Conference of the Learning Sciences, Proceedings* 2.
[4] A. Egalite and B. Kisida. 2018. The Effects of Teacher Match on Students' Academic Perceptions and Attitudes. *Educational Evaluation and Policy Analysis* 40, 1 (Mar 2018), 59–81. https://doi.org/10.3102/0162373717714056.
[5] K. Goldschmidt. 2020. The COVID-19 Pandemic: Technology Use to Support the Wellbeing of Children. *Journal of Pediatric Nursing* (2020).
[6] S. Gretter, A. Yadav, P. Sands, and S. Hambrusch. 2019. Equitable Learning Environments in K-12 Computing: Teachers' Views on Barriers to Diversity. 19, 3, Article 24 (Jan. 2019), 16 pages. https://doi.org/10.1145/3282939.
[7] S. Grover, R. Pea, and S. Cooper. 2016. Factors Influencing Computer Science Learning in Middle School. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. 552–557.
[8] M. Lee. 2019. Increasing Minority Youths' Participation in Computing through Near-peer Mentorship. *Journal of Computing Sciences in Colleges* 35, 3 (2019).
[9] D. Levin and S. Arafeh. 2003. (01 2003), 1002–1007.
[10] A. Ochsner and G. Anton. 2014. Talking with Kids on Game Design, Computer Programming, and Taking Over the World with Dragons. *Unpublished Manuscript* (2014).
[11] J. Wang and S. Moghadam. 2017. Diversity Barriers in K-12 Computer Science Education: Structural and Social. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 615–620. https://doi.org/10.1145/3017680.3017734.