Engaging students with MIT scratch computer-based and Hopscotch ipad educational programming languages in the classroom to enhance their creativity

GDIT 711

SP 14: Distance Learning Global Learning Seminar

Duquesne University

By

Bekir Mugayitoglu

1. INTRODUCTION
	1. PROGRAMMING LANGUAGE

1.2.EDUCATIONAL PROGRAMMING LANGUAGE

1.2.1. MIT SCRACTH COMPUTER-BASED EDUCATIONAL PROGRAMMING LANGUAGE

1.2.2. HOPSCOTCH IPAD-BASED EDUCATIONAL PROGRAMMING LANGUAGE

1.3.) RESEARCH QUESTIONS

1.4.) RESEARCH HYPOTHESIS

2. SIGNIFANCE OF STUDY

3. METHODS SECTION

3.1. SAMPLE SIZE

3.2. POPULATION

3.3. DATA COLLECTION PROCEDURES

4. DATA ANALYSIS

5. CONCLUSIONS

REFERENCES

1. INTRODUCTION

* 1. PROGRAMMING LANGUAGE

Programming language is basically explaining to computer what you want to do. It is speaking the language of computers to develop students’ computational and critical thinking skills and show them how to create new things. Coding encourages and motivates students to emphasize them to think, create, and build by themselves. (Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S., 2011).

 Most people don’t spend time and work on programming language because the syntax looks complicated and too many unfamiliar symbols such as semicolons, parenthesis, bracelets, etc. If the coders mess up with one of these things while coding, there is a bug or glitches appears and it is a complicated process to fix the problem. Learners have stereotypes for programming is that it is such a difficult skill to try and learn.

New generation of today’s tech-savvy world recently is encouraged by Steve Jobs: “Everybody in this country should learn to program a computer, because it teaches you how to think” In the 21 century, technology has been changing rapidly. Digital natives have kept up with this technology to become the perfect users of it. There are a lot of computer programming languages like Java, C++, Objective C, JavaScript with text-based coding, and on the other hand also Educational Programming Languages like MIT scratch, Hopscotch, Daisy the Dinosaur, Hopscotch with block-based programming languages. There are two kinds of programming languages, text-based and block-based, which is considered as the main difference.

* 1. EDUCATIONAL PROGRAMMING LANGUAGE

Programming languages are not just for computer scientists or programmers, but also for doctors, fashion designers, soccer players, students, teachers, instructors, adults, fathers, mothers and kids. However, Barack Obama has a statement to encourage American youth, “Don’t just play on your phone, program it”.

Barack Obama and Steve Jobs know digital natives have the opportunity to play around Internet, play video games, and use Facebook, twitter, Instagram social medias to communicate with the world. Also, they want them not to justplay and also program for novice people to learn programming languages since programming not only for professionals anymore. Educational programming languages are for those people who are novice 8 to up years old to engage with block-based, user-friendly, and kid-friendly programming languages such as Scratch, Alice, Hackety-Hack, code monster, Daisy the dino, and Hopscotch. Educational programming language is “killing two birds with one stone”. Kids learn programming while having fun, entertaining while taking place in an environment that reduces anxiety and increase motivation.Also, Finding a solution with programming languages is the best motivation for students to be encouraged. (Buckley, S.B; Voskoglou, M.G.,2012).

1.2.1. MIT SCRACTH COMPUTER-BASED EDUCATIONAL PROGRAMMING LANGUAGE

Resnick et al. (2009) pointed out that the educational programming language is Scratch, established by MIT to teach basics of coding for beginners with interactive way to be taught Scratch got the idea from logo, but replaced typing code style to a drag-and-drop block-based technique to get inspired by logoblocks and etoys. Despite of text-based arcane coding, novice programmers use scratch with visual block-based and drag-and-drop style to create animation stories, games, interactive presentations, music videos, greetings. Feedbacks are positive by teachers, students, and instructors from all over the world about scratch as a teaching programming language tool. Scratch has three main benefits for programming language learners as block-based coding, simplify to minimize syntax errors, provide feedback about stacks of command blocks, and give immediate feedback for the project.

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008) reported that Scratch is one of the educational programming languages to teach novice programmers to indicate that coding is not difficult, coding is fun, coding is entertaining, and coding is letting you want to do. Kids are not just learning, they also have fun to have awesome creations, create new things and share with their friends. A scratch forms of an editing center and a number of movable sprites. Each object contains its own set of images, sounds, variables, and scripts. Coder can create stacks with coding to palette into command blocks. Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008) pointed out there are a number of control structures; conditionals like if, if-else blocks, loops like repeat, forever, repeat until blocks, and event triggers like when-key-pressed block. Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008) indicated that coders use loops and conditionals are using more. Variables, Boolean logic, and random numbers were less common using, but increasing over time. In addition, there are few commands, like square root and absolute value never appeared in the projects.

**Editing Area – Center**

The middle of the screen is called editing center. Coder drops the blocks of coding from the left to the right to build blocks of code.



Figure 1: Editing Center

http://scratch.mit.edu/projects/editor/?tip\_bar=getStarted

**Toolbox**

There are several scripts in toolbox, such as looks, sound, pen, data, events, control, sensing, operators, and more blocks to use. Each script has several blocks of coding to build programming to drag from here. For example, Motion scripts have move steps, turn degrees, point in direction etc. Looks scripts have show, hide, change color effect by, and set color blocks, etc.



Figure 2: Scripts

http://scratch.mit.edu/projects/editor/?tip\_bar=getStarted

**Loop – Repeat**

Loops block of code is c shape. Whatever you drag block in the c shape, it repeats same thing several times depending on the number in the blank circle next to the repeat text on the top.



Figure 3: Repeat block of code

http://scratch.mit.edu/projects/editor/?tip\_bar=getStarted

**If statement**

If statement block of code is c shape, whatever you drag block in the c shape, it does several times depend on the number in the blank circle between if and next. For example, If I put ahead block in the C shape, the object will go ahead if the object doesn’t touch the corner.



Figure 4: If then block of code

http://scratch.mit.edu/projects/editor/?tip\_bar=getStarted

**If-else statement**

If else statement is making a decision between two things, For example, if my object is not turning right, go straight and if my object is turning right, not go straight.

****

 Figure 5: IF then else block of code

http://scratch.mit.edu/projects/editor/?tip\_bar=getStarted

1.2.2. HOPSCOTCH IPAD-BASED EDUCATIONAL PROGRAMMING LANGUAGE

Hopscotch for ipad makes coding kid-friendly. It is user-friendly, drag-and-drop, and block-based ipad educational programming language. Hopscotch is created specifically aimed at girls’ age of 8 to up them to learn how to do programming. There are not enough articles about Hopscotch because it just created recently.



Figure 6: Hopscotch block of codes screen shot



Figure 7: Hopscotch editing center

1.3. RESEARCH QUESTIONS

- Does teaching of educational programming language improve student problem solving skills?

- Do people who likes math tent to learn educational programming language?

- Do people use move block of codes more than random block of code?

1.4. RESEARCH HYPOTHESIS

Null Hypothesis

* Teaching of programming language doesn’t improve student problem solving skills
* Kids who likes math don’t tent to learn educational programming language
* People doesn’t use move block of codes more than random block of code

Hypothesis

 - Teaching of programming language improve student problem solving skills

 - Kids who likes math tent to learn educational programming language

 - Kids use move block of codes more than random block of code

2. SIGNIFICANCE OF STUDY

Educational programming languages like MIT scratch and Hopscotch broke the stereotypes so that kids do coding to increase their problem solving, critical thinking, computational thinking, and math skills.

3. METHODS SECTION

Using a MANOVA analysis, I will randomly assign kids to two groups.

Kids who do programming will experimental group and kids who don’t do programming the control group. I will let experimental and control group take a pre test.

I will host a 1-hour tutorial session to teach the basics of educational programming language in the classroom. After 1 week, students will work on their project individually and submit it to me by email. Each of them put their names on it. After collecting data, experimental and control group will take post-test to find out is there any difference between control group and experimental group. Also, give them survey questions to find out their opinions about Hopscotch.

3.1. SAMPLE SIZE

The sample size found out which G\*Power, Effect size is 0.3, alpha err prob 0.05, and power is 0.8. After calculating the G\*Power the results indicates that sample size needed 278 students. There are more 600 students I need for my research study.



3.2. POPULATION

My target population is in this study freshmen student who are prospective teachers for K-12 grades in education department at one of the University in Virginia. The sample size found out which G\*Power, Effect size is 0.3, alpha err prob 0.05, and power is 0.8. After calculating the G\*Power the results indicates that sample size needed 278 students. (Appendix A). There are more than what is needed in this study that there are 300 students in the experimental group and 300 students in the control group.

3.3. DATA COLLECTION PROCEDURES

I connected to mention about my study with University in Virginia Tech to collect data. After get approved by the president by University, I will be soliciting student volunteers from that University. In order to measure their problem solving skills they took a pre and post assessment and experimental group do coding with programming language.. Experimental group does programming and Control group doesn’t do programming. After experimental group do programming, I let them both group to take a post-assessment. After collecting data, it is ready to run analysis.

4. DATA ANALYSIS

I will conduct data on SPSS to analyze about my research.

5. CONCLUSIONS

People don’t do programming language since they don’t even try and assume programming language is so difficult to learn. However, educational programming language encouraged kids to learn and break all the stereotypes to be coding.

You don’t have to be smart to do ipad or computer programming languages. In particular, kid-friendly, user-friendly and educational programming languages such as Hopscotch, Game salad, MIT scratch, Alice are welcoming people who are interested in.

 Programmers increase kids’ critical thinking, problem solving, and decision-making skills. We already use those coding activities in our daily life, so it is time to transfer what we do in the programming environment to be creative and create new programs.

**References**

Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011). Recognizing computational thinking patterns. *In Proceedings of the 42nd ACM Technical Symposium on Computer Science Education.* New York: ACM.

Buckley, S.B; Voskoglou, M.G. (2012). Problem solving and computers in a learning environment. *Egyptian Computer Science Journal 36*(4). 28-46.

Goral, T. (2011, January). Take II Tablets. *University Business*, 46-49.

Grover, S. & Pea, R.D. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*. *42*(1), 38-43.

## DeLoura, M. & Paris, R. *"Don’t Just Play on Your Phone, Program It"*. Retrieved 09 12, 2013 from http://www.whitehouse.gov/blog/2013/12/09/don-t-just-play-your-phone-program-it

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by

choice: Urban youth learning programming with scratch. *In Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*, Portland, OR, USA, March 12-15, 2008. SIGCSE ‘08 (pp. 367–371). New York: ACM.

Tickle, G. (n.d.). *Code.org Says “Hello, World” to Get Everybody Coding*. Retrieved 1 22, 2013, from http://www.geekosystem.com/code-dot-org-launch/

Utting, I., Cooper, S., Kolling, M., Maloney, J. and Resnick, M. (2010)