Bekir Mugayitoglu

Independent Study

THE DIFFERENCES AND SMIILARITEIS BETWEEN SCRATCH AND ALICE

What are the differences and similarities between Scratch and Alice?

What are the challenges to use them in class environment?

How can we learn from their shortcomings to design better platforms?

How can we discover working patterns that might be applied to designing of

future programming platforms?

Summer 2014

Dr. Kush

The aim of this independence study is to investigate the two educational programming languages, (Carnegie Mellon University) CMU Alice and (Massachusetts institute of technology) MIT Scratch to lay the groundwork for my dissertation research. This study will benefit my research in three folds. I will be doing an extensive literature review to understand the basics of visual programming for education. I will then create taxonomies and use heuristics to discover the similarities and differences between the two platforms. This will then lead me to test my hypotheses.

Alice and Scratch are visual programming platforms that let novice programmers learn computer programing. If kids and adults are ready to tackle with programming, without prior experience, programmers can use Alice and Scratch two-keystone environments aim at them that will help them discover while entertaining programming techniques. My independent study is designed to build the conceptual framework of integrating technology in the classroom environment with visualize programming methods and techniques. Further, I want to begin to test the hypothesis that which programming language is for kids and which programming language is for adults to help to understand STEM topics. Also, Programming world is encouraged by Steve Jobs: “I think Everybody in this country should learn to program a computer, because it teaches you how to think”.

Null Hypothesis: There are not differences between Scratch and Alice?

Question 1: What are the differences and similarities between Scratch and Alice?

Question 2: What are the challenges to use them in class environment?

Question 3: How can we learn from their shortcomings to design better platforms?

Question 4: How can we discover working patterns that might be applied to designing of future programming platforms?

**What are the differences and similarities between Scratch and Alice?**

Thinking, playing, and learning are the occupational activities for kids to apply in their daily life. However, they don’t. We don’t mention about those things. What we do instead of those things is that we ask them numbers, grammar, and the French Revolution. The question is that why don’t we make them think, learn, and play?( Papert, 2005). Programming language is the potential solution to make them think, learn, and play. There are two different kind of programming languages such as text-based programming language and block-based programming language.

MIT Stratch and CMU Alice teach visualize and user-friendly educational programming languages before moving on to textualize and complex programming languages to deal with problem solving with algorithms. Starting with syntax, text-based programming language totally complex and if there is a bug it is difficult to figure out. In particular, novice programmers are able to put codes together and are familiar with programming. Programming makes both parts of brainwork (Cooper, Dann, & Pausch, 2000).

Even though they are educational programming languages, there are differences and similarities between them.MIT scratch works and the features of it to create animated stories, wishing cards, games, simulations, etc. It is based on colorful command blocks programming language to teach kids 8 or up to learn how to do programming with self-directed and collaborative way without prior knowledge experience. It is inspired by Logo and constructed the idea of Logo to transfer from physical environment to virtual website environment. The target of Scratch is specifically for kids and novice programmers to start with it and move to the text based programming languages such as C, Java, C++, etc. (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010).

MIT scratch is an authentic learning, intuitive and engaged programming language that two- dimensional engaged environment, interactive presentations, games, and animations are created by people to discuss and share their creative projects. Also, users contribute by sharing their projects, remixing the created project by others, adding as a favorite project and clicking to express their feeling by love it on other users projects (Velasquez, Fields, Olsen, Martin, Shepherd, Strommer, & Kafai, 2014).

****

**Figure 1. Screenshot from Scratch project created by me**

On the other hand, CMU Alice is a scripting, 3-D interactive, intuitive, visualize, and animation educational programming language tool for Windows built by Carnegie Mellon University. Functions of Alice built and inspired by Phyton programming language. Coders are able to take feedback immediately after building their animated projects to run it. Karel, The Robot educational programming language is the ancestor of CMU Alice to make it intuitive (Cooper, Dann, & Pausch, 2000).

The advantage of Alice is that students can take a visual feedback immediately. They see what they created. What they found is that students who is experiencing with Alice use recursive action which is easy to understand. The part of Alice is that event handling, control panels, list boxes, check boxes, and sliders. Alice is similar with widgets used in Java, not like MIT scratch like a drag- and-drop used in Visual Basic (Cooper, Dann, & Pausch, 2000).

****

**Figure 2. Screenshot from Scratch project created by me**

Scratch focuses on younger programmers to create projects through tinkering and self-directing. The target population of Alice is older people than Scratch. Scratch has a feature to draw pictures and record sounds (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010).

3-D graphical tool of Alice is more difficult to create and not easy-to-use than 2-D Graphical tool of Scratch programming language (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010).



Figure 3. Four Degree Orientation – MIT Scratch educational programming language

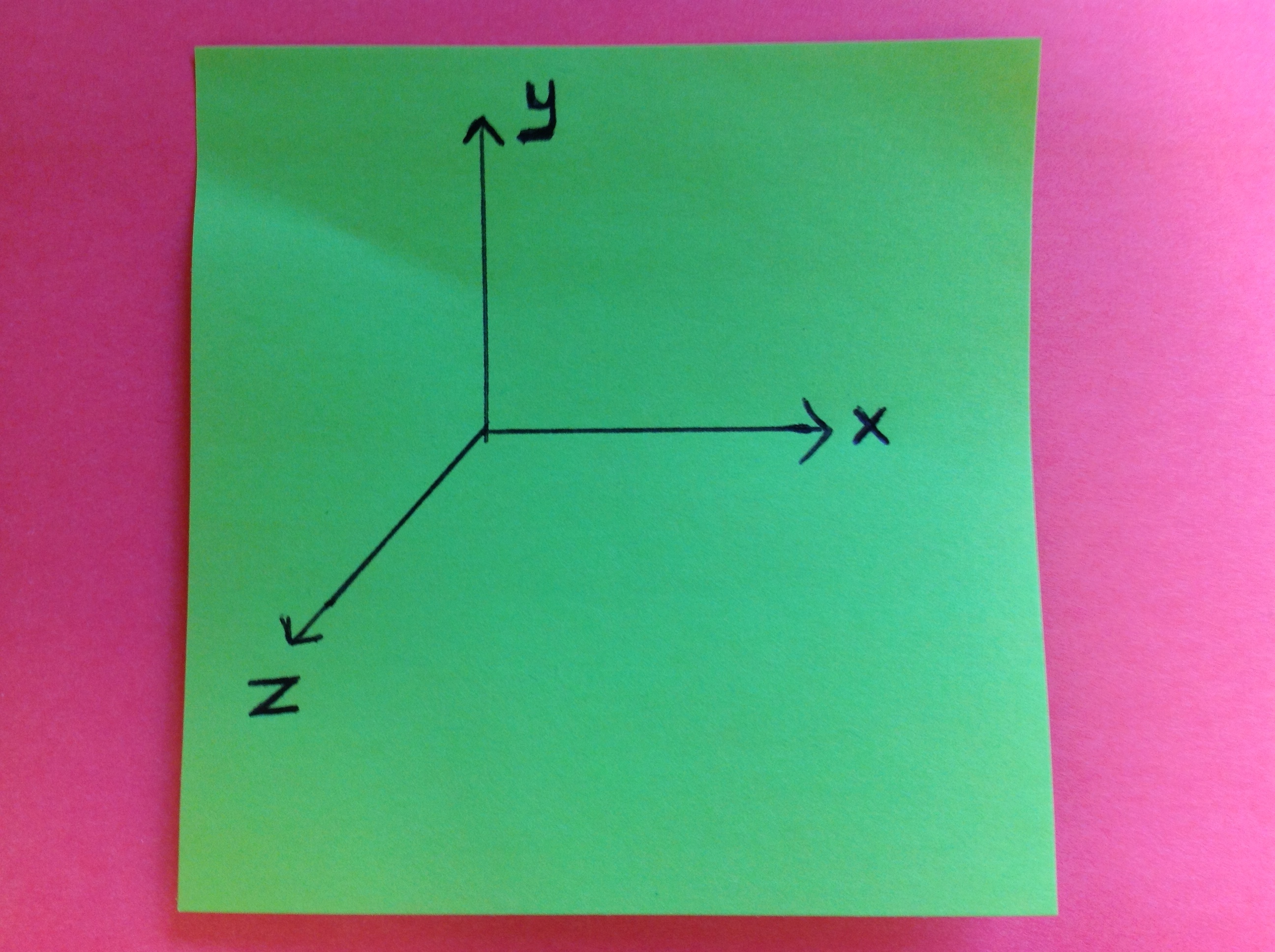


Figure 4. Six degree Orientation – CMU Alice educational programming language

However, there are similarities between MIT Scracth and CMU Alice that both of them are a gateway for text-based programming language such as Java that helps students to pass Advanced Placement Exam in Computer Science. (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010). MIT Scratch and CMU Alice are features to import media from somewhere CMU Alice and MIT scratch are introducing programming language to people who have no prior knowledge so that two of them have similar design goals. Both of them develop rich graphics and sound effects that let users create project that they are interested in (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010).

**What are the challenges to use them in class environment?**

There are numerous benefits of implementing educational programming language in classroom environment, but there are some challenges. Research has started in a long time ago to design programming languages and environments for people of all ages who have no prior programming language experiences (Kelleher, Pausch, & Kiesler, 2007). However, integrating in classroom is so significant to teach kids programming language. While introducing learners to programming language, they should be motivated by those three: 1) They are coming up with ideas. 2) They are expressing themselves. 3) They are getting into computer-based majors. (Kelleher, Pausch, & Kiesler, 2007).

Children don’t get the point why they learn math and they don’t create their own formula for their projects. However, they can do that with computer. Children are getting familiar, knowledgeable, articulated, and sophisticated about improving formal systems. They are interacting with themselves and doing hands-on activity by thinking ( Papert, 2005)

If teachers don’t let them learn by self-directing and hands-on activity, they won’t be able to brainstorm and express themselves. Technology is not only a way to develop children, but also influence and control them to apply for their projects so that they can be confident, sophisticated, and knowledgeable. Dewey, Montessori, and Piaget indicate that children learn by hands-on activities and brainstorming about what to do ( Papert, 2005). Students are not strong problem solvers even they took algebra and pre-calculus so that they can increase the ability of solving problems by coding. However, they have to know how to do programming. This tool is provided for students to teach abilities and strategies, and techniques to make them programmers. The positive results of CMU Alice will be known when students go to the college and it will be compared with non-programmers who didn’t use CMU Alice (Cooper, Dann, & Pausch, 2000).

Computational thinking is the most beneficial source to give children priceless power to invent and carry out projects with technological devices using through programming language ( Papert, 2005). Experience is related to intellectual growth so that creating environment for them to be involved in experiences increases their intuitions and concepts for handling with thinking, learning, creating, and playing ( Papert, 2005).The process of learning is that child is involving it, brainstorming, coming up with the ideas, and experiencing with output. After getting results, share with others to criticize what he came up with ( Papert, 2005).

**How can we learn from their shortcomings to design better platforms?**

There are some shortcomings of MIT Scratch and CMU Alice such as CMU Alice graphic concepts, the notion of state, programming, and programming language concerns, and event-driven programming language (Cooper, Dann, & Pausch, 2000). Also, CMU Alice doesn’t have tutoring section to scaffold learners’ knowledge. When they encounter bugs or glitches, they might be frustrated. The disadvantages of Alice are whether students really want to know egocentric orientation 3-D graphics and animation. Maybe, 3-D graphics and animation might be difficult to understand and imagine what is going on. On the other hand, 2-D graphics and animation can be implemented very well for young novice learners since there are four directions instead of 6 directions. Also, it is time consuming to explain them basic objects such as rotation and color. The most important thing is that the spatial relationship of objects to one another and in order of objects. Another disadvantage of Alice is because of same functions of if statement with python. It makes sense to Phyton, but it doesn’t make sense to Alice. The positive result from students is that they had self-confidence after acquiring programming skills (Cooper, Dann, & Pausch, 2000).

**How can we discover working patterns that might be applied to designing of future programming platforms?**

We are not trying to reinvent the wheel; there are working patterns we can apply to designing of future programming platforms. MIT Scratch and CMU Alice have toolbox section, it is clearly and user-friendly created block of codes. They are colorful and bright for kids to be attracted and appealing. Also, editing center is the biggest part of interface that it is relaxing center to work since programmers spend most of their time in editing center. Both of them are computer-based programming language and they have scripts in their toolbox section such as motion, looks, events, control, etc. When user clicked on one of them, block of codes are appearing so that programmer picks what they need. This working pattern is beneficial for users not to scroll down because there are not too many blocks of code to overwhelm users. Otherwise, they might be frustrated and discouraged to keep on programming.

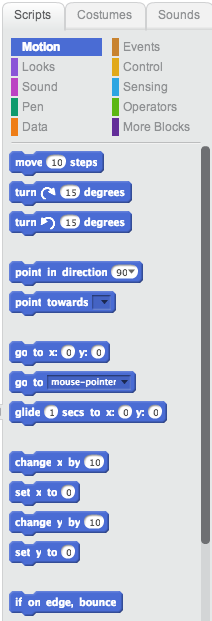


Figure 5. Screenshot from Scratch block of codes



Figure 6. Screenshot from Alice block of codes

Dr. Kush and I met weekly to discuss about CMU Alice and MIT Scratch and I went for workshop to Massachusetts institute of technology (MIT) at 17TH of May in Boston. Also, I will attend the conference at Massachusetts institute of technology (MIT) between 6th of August to 9th of August. I created with Dr. Kush to design some animations with MIT scratch and CMU Alice with him after coming back from Workshop. I signed up new account on MIT scratch to share my projects with scratch members. The final deliverable for this three-credit independent study is a paper that : 1) Increased the literature review on MIT scratch, CMU Alice, and programming language 2) developed my programming techniques to learn more 3) provided an opportunity to advance my thoughts toward my dissertation.

**References:**

Cooper S., Dann W., Pausch R.: Alice: A 3-D Tool for Introductory Programming Concepts. *Journal of Computing Sciences in Colleges*, 15, 5 (2000), 107-116.

Kelleher, C., Pausch, R., & Kiesler, S. (2007). Storytelling Alice motivates middle school girls to learn computer programming. *In Proceedings of the ACM Conference on Human Factors in Computing Systems.*1455–1464.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education (TOCE)*, vol. 10, no. 4 (November 2010).

Papert, S. (2005). Teaching children thinking. *Journal of structural learning*,*4*(3), 219-230.

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/

Velasquez, N. F., Fields, D. A., Olsen, D., Martin, T., Shepherd, M. C., Strommer, A., & Kafai, Y. B. (2014). Novice Programmers Talking about Projects: What Automated Text Analysis Reveals about Online Scratch Users’ Comments. 47th Annual Hawaii International Conference on System Sciences