Context Report // CI504

IA-PENN GROUP
Melissa Burton
Ankit Agrawal
Becky Popelka

Product Description

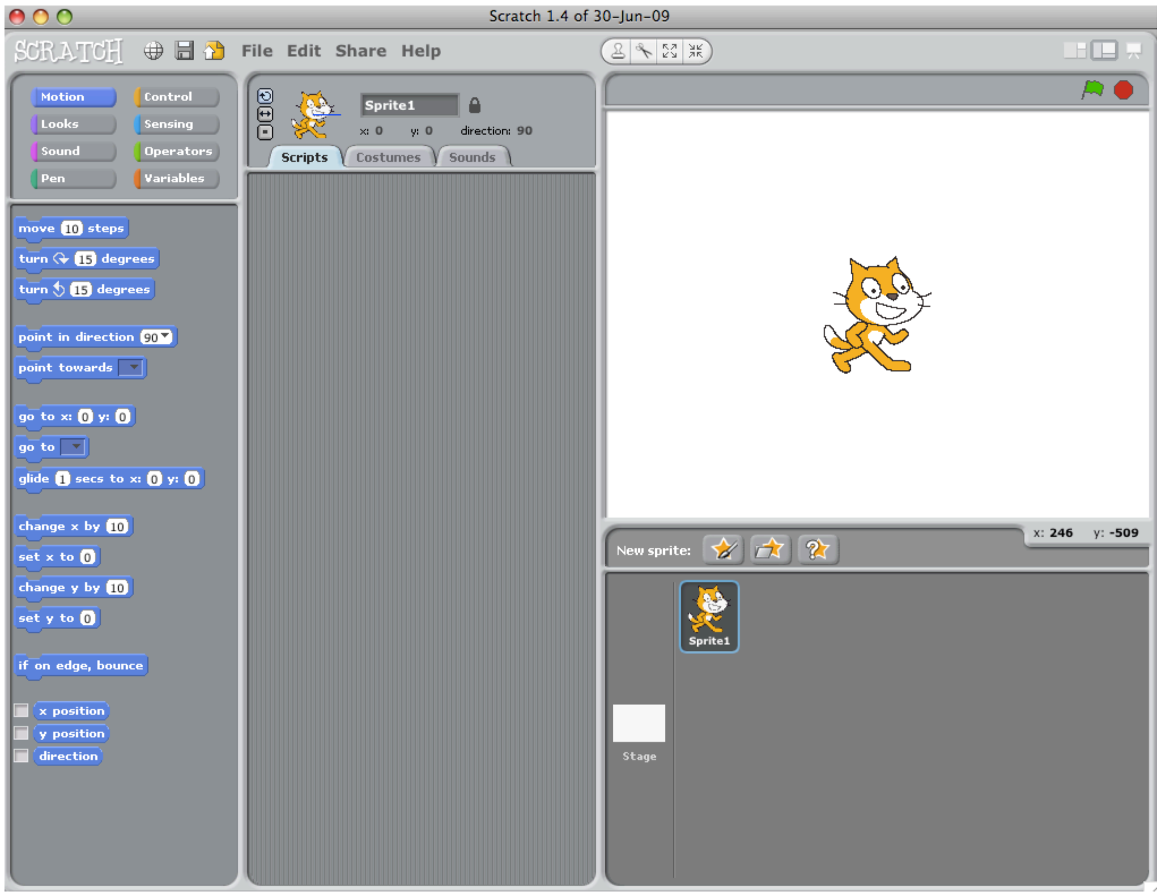
The product that the IA-Penn Group is evaluating is Scratch. Scratch is a programming language that is designed for everyone to use, from a novice, an expert, child, or an adult. Scratch is created by the Lifelong Kindergarten Group at the MIT Media Laboratory. We have included several screen shots of the user interface at the end of this section. The name Scratch is inspired by the turntablist technique of scratching. This name was chosen due to the nature of the language, where you can re-mix and re-use parts and create something new and innovative from these parts. Scratch differs from most programming languages because instead of relying on hand-written code that involves meticulous syntax checking and typing code, Scratch is visual and is done by dragging command blocks and assembling these blocks for media manipulation. This methodology of drop and dragging command blocks to substitute for typing code was inspired by concepts from EToys and LegoBlocks. Although Scratch can be used by anyone, Scratch was originally designed for novice programmers. Scratch is flexible and can be used to create simulations, animated stories, games, interactive greeting cards, and many other media intensive projects. Scratch is primarily considered an educational programming language since it allows users to explore and experiment with concepts that are used in more diverse and complex programming languages. These concepts that are used in Scratch include relative motion, absolute positioning, image transformations such as rotations or effects, recorded sounds, cell animation, music, and programmable pens. Additional functions include the use of arithmetic functions, comparisons, Boolean logic, conditionals such as if or if-else, loops such as repeat or forever, and event triggers such as allowing an event to happen if the Sprite touches an object.

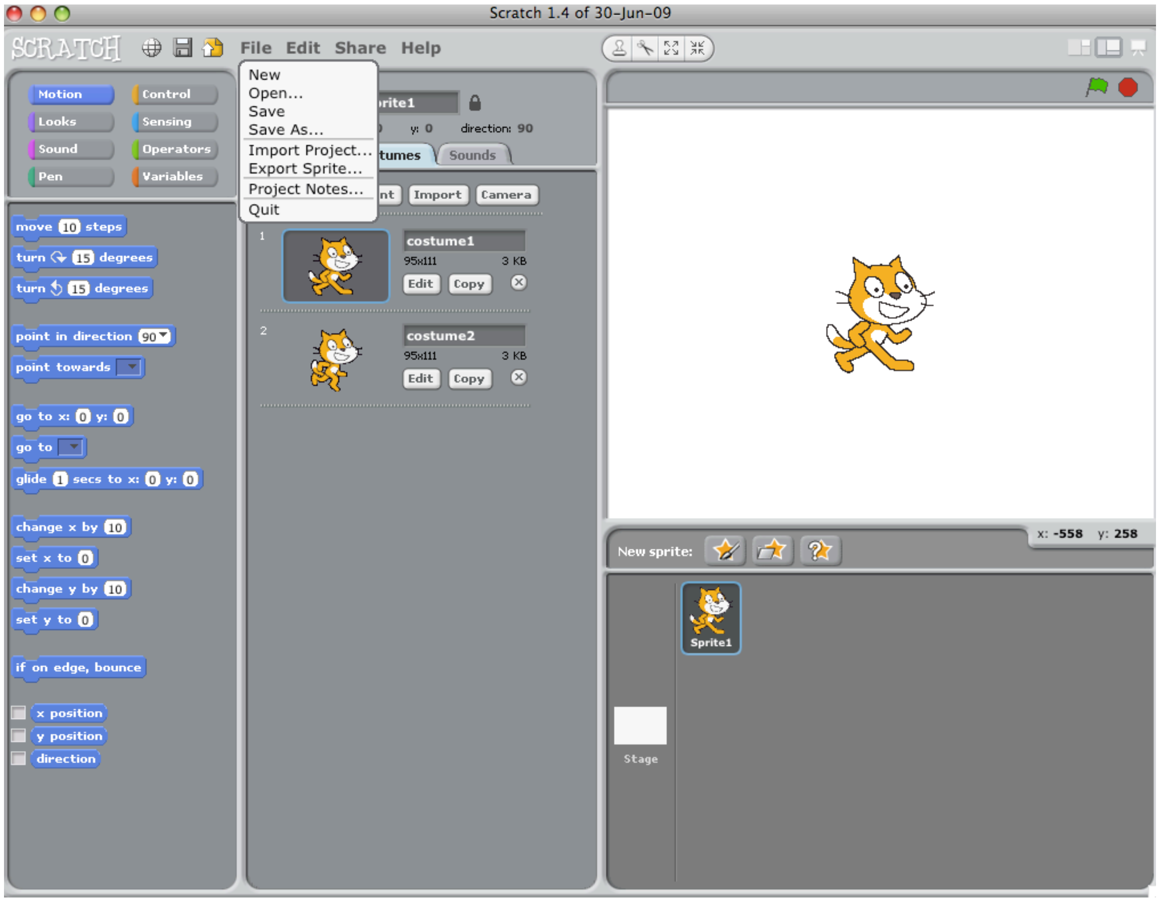
Scratch has a number of features but fortunately comes with a tutorial that provides lessons for individuals desiring to use Scratch. Due to the time constraints of the project, the IA-Penn Group will be evaluating the first lesson of the tutorial. The title of the first lesson is “Lesson 1: Introducing

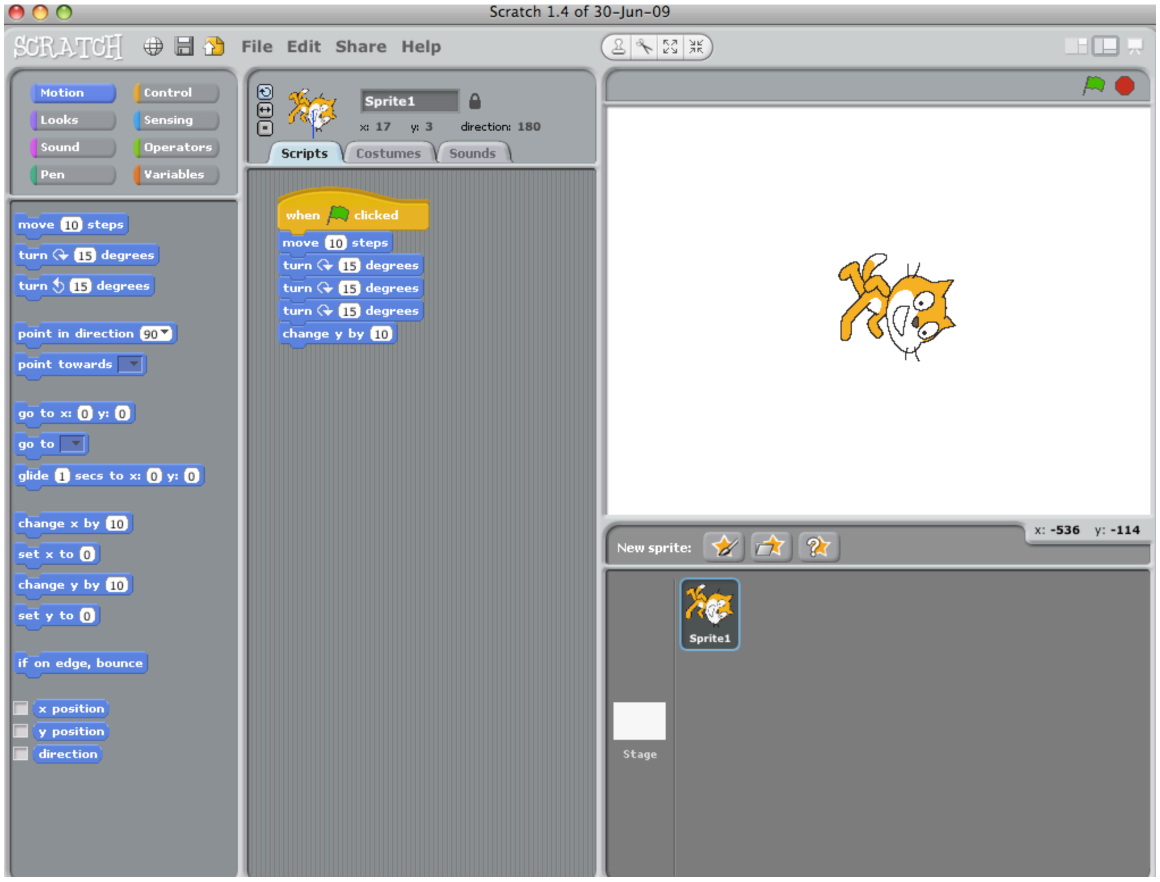
Scratch and Creating Sprite”. The purpose of this introductory lesson is to introduce the user to the program Scratch and start the user off by allowing them to create a sprite character. These Scratch lessons are generated by a company called Shall We Learn. This company has generated 15 lessons for using Scratch and also hosts other features such as Scratch Demos, quizzes, Scratch video lessons and some sample projects created in Scratch.

Although Scratch can be used in a classroom, museums, community centers or at home, Scratch is designed to be intuitive and self-directed, and therefore it is integral that any user be able to use the software with little or no instruction from a teacher or any other mentor.

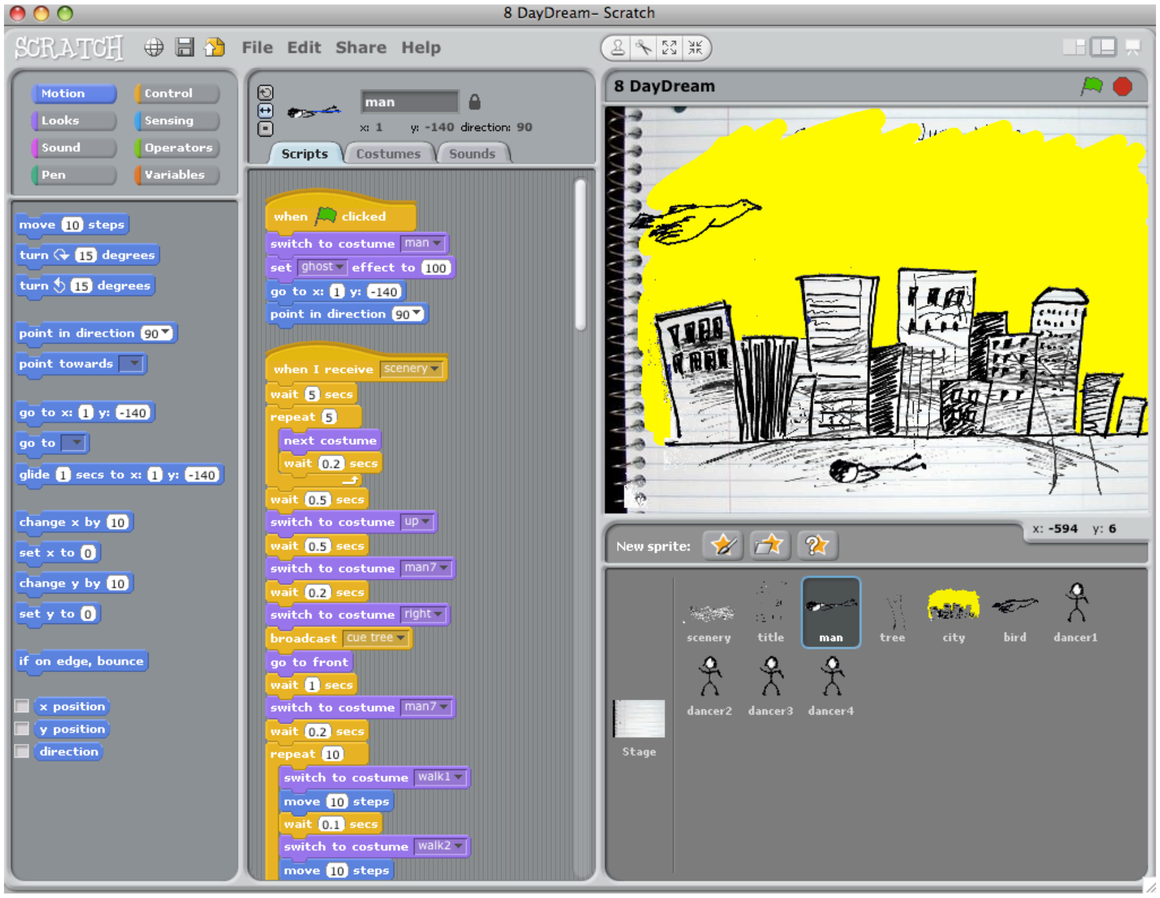
Scratch is open-source and is available for anyone to download, distribute and use for free. Scratch can be used on most operating systems such as Windows, MAC OS X or Linux computers. Scratch is available to download at http://scratch.mit.edu/ . The expectations of the Scratch Designers are that users would be able to reference lessons or tutorials and successfully use the Scratch program with no teacher instruction. The goals of the IA-Penn Group are to evaluate the usability aspects of the lesson by observing test participants work through the lesson.







```
when [flag] clicked
move 10 steps
turn 15 degrees
turn 15 degrees
turn 15 degrees
change y by 10
```



```
when  clicked
switch to costume man
set ghost effect to 100
go to x: 1 y: -140
point in direction 90

when I receive scenery
wait 5 secs
repeat 5
    next costume
    wait 0.2 secs
wait 0.5 secs
switch to costume up
wait 0.5 secs
switch to costume man7
wait 0.2 secs
switch to costume right
broadcast cue tree
go to front
wait 1 secs
switch to costume man7
wait 0.2 secs
repeat 10
    switch to costume walk1
    move 10 steps
    wait 0.1 secs
    switch to costume walk2
    move 10 steps
```

Description of Target Audience

According to the introduction of the Shall We Learn PDF tutorial: *Scratch Programming, Lesson 1: Introducing Scratch and Creating Sprite*:

> Scratch is a programming language for all, even for kids. In fact, Scratch, unlike all other programming languages, is designed first and foremost for kids. Because it's designed for kids, it's very easy to learn and use. They can create animations like never before. For older kids or teens, they can create single-level or multi-levels Scratch games.
>
> But Scratch is not just for kids or teens. Teachers and adults can use scratch to create effective education tools such as math quiz, physics simulation and educational videos.

Based on this information it can be assumed that the main end user group of the Scratch software product are children and teenagers and a secondary end user group of teachers and adults. The visual design of the user interface of the software appears to cater to a target audience of children approximately ages seven to eleven years. The design is not likely to appeal to teenagers, although according to the Shall We Learn document the software is aimed at children and teenagers. Trying to appeal to too broad a target audience is often a flaw of educational websites and digital products; it is typically not possible to create instructional materials that appeal to both children and teenagers.

When evaluating the appropriateness of an instructional material designed for a main target audience of children, both in terms of usability and aesthetics it is important to realize that children comprise a group of people at different developmental stages. Co-authors Amy Bruckman and Alisa Bandlow from the Georgia Institute of Technology who contributed chapter 22 in *The Human-Computer Interaction Handbook* look to Piaget's 1970 research to understand child development. This research defined four distinct stages of human development (2003). The first stage is Sensorimotor, beginning at birth and lasting until two years of age. During the Sensorimotor stage children use their sensory perception to build knowledge and thinking skills. The second stage is Preoperational, beginning at two and lasting until around seven years of age. Preoperational children have fleeting attention spans, are able to hold only one item at a time in their memory, are often not able to comprehend abstract ideas and cannot empathize with others. Most preoperational children are not independent readers and are not able to use keyboards. The next stage of development is the Concrete Operational stage lasting from ages seven through eleven. Schneider elaborates on the characteristics of children at this level of development stating, "we see children maturing on the brink of adult cognitive abilities, Though they cannot formulate hypothesis, and though abstract concepts such as range of numbers are often still difficult, they are able to group like items and categorize" (Bruckman & Bandlow, 2003). The final Piagetian stage of development is the Formal Operational, beginning at age eleven and continuing into adulthood. At this point children's cognition is similar to that of an adult (Bruckman & Bandlow, 2003).

There are several prerequisite skills needed to operate the Scratch software and work through the Shall We Learn tutorial. The user must be able to: read independently, operate a mouse, operate a keyboard, understand basic software menu functionality, interpret pictorial icons, draw simple figures or images, and have an understanding of the basic concepts and terminology used in the program such as "edit", "costumes" and "controls". Based on the above descriptions of child development stages it is safe to assume that while the main end user group for the Scratch software is "kids", in order to work with the software the users must be at least at the Concrete Operational stage or beyond.

Scratch allows users to create very simple animations or more complex creations such as digital games and therefore users of different ages and skill levels may find the program valuable. However as mentioned before, the visual design of the product caters to younger users. This is not likely to deter adult users, such as the secondary end user group of teachers mentioned in the Shall We Learn tutorial. Teenagers and Pre-teens, on the other hand, are a highly image conscious user group and are likely to shy away from anything they believe appears childish.

Although the main target audience for Scratch is children, initial use of the program and working through the Shall We Learn PDF tutorial will almost certainly require adult assistance for concrete-operational stage children (ages 7-11). Children younger than 7 years would struggle to use the program at all, while most children 11 and older would be able to work through the tutorial independently. Concrete-operational children would eventually be able to work with the program independently, but will require adult assistance as they are learning the program.

It is expected that users (both adults and children) will approach this instructional material with the attitude that learning about and working with the Scratch software will be a fun experience. The author of the Shall We Learn tutorial states the following in regards to working with Scratch:

> Since I knew about Scratch from a coworker, I have been using Scratch, teaching Scratch, and now writing on Scratch. As you can tell, I just cannot get enough of Scratch. I have two school-age boys and I’ve been looking for ways to quickly create games and animations to help them learn. Scratch is what I’ve been looking for and more. It’s just a tool so awesome, so fun, and so easy to use and master, that I am sure, once you start, you will be just like me: Can’t Scratch Enough!

The author’s description of their experience using the Scratch software creates an expectation of a fun experience for the user.

Preliminary Evaluation

Scratch is a very interesting product to study and use. It has both strengths and weaknesses. It is very easy to learn and can be used by anyone. Scratch is said to be designed for anyone from novice to expert, but we feel that the level of detail and available options are a bit too simple and low in number to get a prolonged serious attention from an expert programmer.

One of the more noticeable negative attributes of the product is its user interface. It is too colorful to be an interface for a serious tool. The colors create the impression that it is designed primarily for kids, but at the same time font size is too small and some of the menu items are difficult to locate. Additionally, a lot of very commonly used features are missing. For example, if you want to create the side view of an animated character there is no option to simply rotate the created character as desired and take snaps to create different angled views. Instead the user will have to do a lot of erasing and redrawing to achieve the same effect. Another bad example of user interface design can be attributed to the 'change color' functionality. If the user wants to change the color of his sprite he has to drag and drop a change color tab from looks onto the scripts window and then change the number in small box according to need. Such design concept does not seem to favor either the younger users or expert ones as it is confusing and counterintuitive. Small font size along with a less than ideal color scheme makes the whole thing look very unappealing.

Talking about basic limitations on grain level, one can only undo five steps while editing a sprite. Similarly zoom is restricted to 5X. A lot of usability issues bring forward the lack of efforts in designing the tool and give the impression that is just a prototype instead of a serious attempt to build something widely usable and productive.

Quite a few options are not easily understandable and we had to rely on the help function to explain many of the options. One such feature is the option to set the costume center at the bottom left of paint editor which places a dragable cross on the editor window.

A couple of innovative ideas that we liked include the ability to take pictures and record voice options embedded in the tool itself. Clicking on 'camera' tab under costumes, takes a snap and creates a layer with that snap on it. Similarly audio can be easily embedded without the hassle of recording separately and then importing.

To improve the product we would like to suggest a lot of basic changes which do not require much effort. The ultimate objective of this tool/programming language is to provide the ease of use and to make it productive for both non-programmers and programmers.

Looking at it from a novice or kid's perspective, a lot of quick buttons should be included. There can be different menu items which pop up when the mouse is right-clicked at different locations. For example, if a user wants to change a sprite, there can be a pop up menu which can be controlled by right-clicking on the sprite. This will be very helpful as most of the novice users try pressing mouse buttons randomly in attempt to do what they are not able to find an option, which may be hidden somewhere deep in some menu. Other than that quick help should be embedded for functionality tabs that are not widely known. This can be formatted by putting a small question mark next to tab, which on clicking, displays help window for it. This will not only eliminate the need of training before using the tool but also make it self-explanatory and exciting to play with. A few places where this can be done are next to green flag tab and red circle on the top of preview window. The scissor icon on the top, which is actually used as a delete key, looks more like an option to cut something.

As the tool is also supposed to be used by instructors and expert programmers, there should be more professional look to the user interface. As a first thought designing a slightly different interface with more exhaustive menu options like those in other contemporary tools is not a bad idea. Two interfaces and the ability to switch between them is a good choice with a little overhead as it is difficult to capture the interest of kids as well as experts with the same interface.

Overall on first sight tool gives the impression that is it software created for children, however not all aspects of the design are kid or novice user friendly. Although a lot of functionalities do make the product easy to use in some respects, some of the very basic functionalities are missing. We would suggest designers to create two slightly different switchable interfaces to cover wide range of users. As a matter of fact, there can also be an option to customize the interface. This way expert user cannot only configure it as per their need, but instructors can also change it depending on the requirements of students for improved and more efficient learning.

Intended Outcomes

Scratch is designed to make programming engaging, motivating, and informative for everyone. The website for Scratch clearly states that the program is supposed to be usable by novice programmers and is designed to be enjoyable to use. Specifically with lesson 1, the user should be able to know what Scratch is, what the intended objectives of Scratch are, and how to successfully make a Sprite.

Cognitive Objectives:

Start the Scratch program successfully

Operate various tools of the Scratch program such as the “Ellipse Tool”, “Erase Tool”, “Stamp Tool”, “Fill Tool”, “Select Tool”, “Eyedropper Tool”, and “Line Tool”

Understand and operate the various modes of the tools

Create a sprite with the Scratch program

Create multiple views of the Sprite, such as the back, left side, and right side viewpoints of the Sprite.

Identify the Sprite Editor and be able to read the editor.

Save the project file, open and run the project again.

Affective Objectives:

Become more comfortable working with computer programming languages using the turntablist technique

Gain a sense of accomplishment in working with technology to create an interesting and aesthetically pleasing animated character

Enjoy using Scratch to create an animated sprite

Psychomotor Objectives:

Work and control and finger muscles required for mouse and keyboard operation

Improve/ Increase hand-eye coordination required for working with and controlling program tools and functionalities

References:

Bruckman, Amy & Bandlow, Alisa. (2003). Human- Computer Interaction for Kids. In Julie Jacko & Andrew Spears (Eds.), *The Human Computer Interaction Handbook* (pp. 429-440). Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

Lifelong Kindergarten Group. (2011) Scratch. Retrieved from http://scratch.mit.edu/

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y.(2009). Scratch: Programming for All. *Communications of the ACM*.

Shall We Learn (2011). *Scratch Programming – Lesson 1: Introducing Scratch and Creating Sprite*. Retrieved from http://shallwelearn.com/