
SCRATCHING THE SURFACE

Cobi Nguyen
cnguyen32@email.arizona.edu

Zachary Wiegand
zwiegand@email.arizona.edu

April 13, 2020

ABSTRACT

1 Introduction

2 History

Scratch is a visual block-based programming language with the goal of making programming more accessible to children. Scratch gets its name from DJs and their technique of combining different vinyl records called "scratching". Scratch is developed by the MIT Media Lab with the first prototype appearing in 2002 and officially launching on January 8, 2007. Currently on version 3.0 scratch is implemented using JavaScript while versions used Squeak and ActionScript. Currently Scratch is alive and well with over 40 million website visits, as seen from the analytical data on their website, and version 3.0, the newest version, coming out on January 2, 2019.

3 Control Structures

Scratch makes learning and accessing its control structures fairly simple because Scratch was created to introduce people into the world of coding. On the left side of the web page where the different sections of code blocks can be found, there is a whole section dedicated to control structures. Within this section the developer can find different forms of iteration, conditions, wait statements, and even an interesting cloning mechanic.

In Scratch there are three given versions of iteration for the programmer to use not including recursion. These three versions of iteration are the repeat loop, the repeat until, and the forever loop. While having different names they are equivalent to control structures that an experienced coder would have already come into contact with. The repeat loop is equivalent to a for loop so it repeats its code block a certain number of times. The repeat until is effectively a while loop which means it will repeat until a conditional is satisfied. Finally, the forever loop does exactly what it sounds like; the forever loop will repeat its code block indefinitely or until the code is stopped somewhere else.

This is also a great opportunity to mention options to stop code being run early. Scratch gives programmers a 'stop' function. The stop function can specify if **all** code needs to halt, that single block of code is to stop running, or if all the code exclusive to that sprite should stop. This function gives further control to the developer by allowing them to decide when a script stops running. Further control of when code should be ran comes from two provided wait functions. The first wait function acts like a sleep, it pauses the script at that statement for the specified number of seconds. The second wait function has a little more complexity. The programmer can pause the script in that sprite at the wait statement until the specified condition is met. This second wait statement can come into use if a programmer wants an object to pause until interacted with by the user or another object.

Within the control section the developer also has the standard conditional statements that an experienced coder would come to expect. Those conditionals are if statements and if else statements. One might notice there is no if else-if statements which means if a developer wants to create branching conditional checks they would have to create a somewhat robust version of a switch case. That being said Scratch also does not have a formal version of switch cases so a developer would have to use multiple if blocks in succession and make sure that if one of those if blocks gets ran that the others are also not ran.

A unique type of control structure for Scratch is three functions for cloning sprites. The first cloning function is 'create clone (sprite)'. This is what is called to create the clone of either the sprite calling the function, or create the clone of another sprite. The second function is 'when I start as a clone'. Say a clone has just been made of sprite A, in sprite A code following this second function will execute. After a clone has finished doing its job, the program may want the clone to go away, that is what the last clone function is for. The third clone function given to programmers is 'delete this clone'. This function simply removes that clone completely from the program.

A great example of cloning can be found within the creative program for milestone 2, '99 Bottles'. The way the program was designed the end goal was to have 99 soda bottle sprites on the screen that could run code when clicked on by the user. It would have been unreasonable to create 99 individual sprite objects manually. Therefore one single soda bottle sprite was made and then was recursively cloned to fill the screen with 99 bottles.

4 Data Types

Due to its nature of accessibility to new programmers Scratch does not have any data types that seem out of the ordinary to the experienced programmer. The three data types that Scratch has are numbers, strings, and booleans.

Numbers in Scratch refers to both integers and decimal values. Scratch does not see a difference between the two. There are ways to manipulate number variables within Scratch such as flooring, converting a number to the nearest whole number lower than the decimal, ceiling, converting a number to the nearest whole number larger than the decimal, and even taking the square root of your number variable.

String is a series of characters of letters, numbers, characters, and basically everything you can type. Strings are useful for holding any information that is readable for humans.

Booleans are a variable type that only allows two values, true and false. By using booleans a programmer can test conditions and then proceed to tell the program what to do based on the result of those tests.

Even though that the amount of different data types is fairly short the variables within Scratch also have different types. The user has access to global, local, and cloud variables. By default all variables are created as global and are able to be accessed by any part of the program. Local variables are more personable in the way that only their owner is allowed to change them. However, other pieces of the program may be able to look at the local variable. Lastly, cloud variables are variables stored on the cloud which allows the variable to affect multiple instances of an open project and gets saved when the project is closed.

5 Subprograms

6 Summary

References

- [1] Al Sweigart. In *Scratch Programming Playground: Learn to Program by Making Cool Games*. No Starch Press, 2016.
- [2] MIT Scratch Team. Scratch 3.0. https://en.scratch-wiki.info/wiki/Scratch_3.0. Accessed: 2020-3-26.
- [3] MIT Scratch Team. Scratch for developers. <https://scratch.mit.edu/developers>. Accessed: 2020-3-26.

[1] [3] [2]