

Game Programming Vocabulary

program -- a piece of software that performs some identifiable task, such as browsing the web, letting you manage photos, or providing a game to play.

user--the person using a program.

user interface (UI)--the part of the program that the user interacts with. Most programs today use menus, buttons, toolbars, etc. to provide a UI. In Scratch, our UI's will be generally be very simple and will rely on mouse moves and clicks and key presses.

code -- refers to the commands that a programmer writes to make a program. Code can be used to refer to both large and small amounts of programming commands. In Scratch, the word "code" will be used to refer to scripts.

programming -- basically, writing code, but can also mean other tasks involved in putting together a program.

event -- something that happens that a program may need to respond to, such as the user pressing the mouse button or some time limit having been reached. The script that responds to the event will start with a block like this:

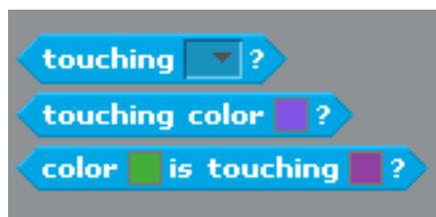


event-driven programming -- a style of programming that organizes the code around events. Scratch uses event-driven programming for everything it does.

script -- the name for a piece of code in Scratch that responds to some event. All scripts in Scratch start with a "hat" . In the world at large, script can mean any relatively short and simple piece of code.

message--a special type of event that one part of a program sends to another part

collision detection--the task of detecting when two objects in a game (or simulation) touch each other. Collision detection is an important and often difficult part of game design. In Scratch we use the **touching** blocks under **Sensing** to help with collision detection.



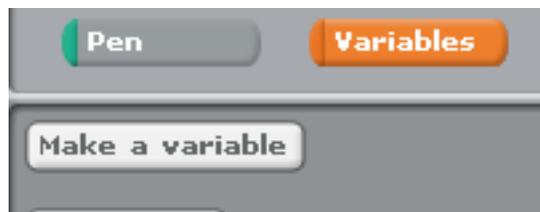
bug -- when a program doesn't work the way it should. The bug might cause the program to completely stop working, or it may cause the program to do something, just not the right thing.

debugging -- the act of finding bugs and fixing them. Debugging is a huge part of programming.

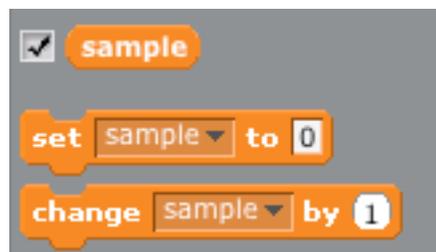
sprite -- an object in a program that includes both graphics and code. Scratch is built largely around sprites.

feature -- an identifiable characteristic of a program that is useful to the user. For example, a first-person shooter might allow you to host LAN parties with an easy set up screen. That would be a (big) feature. Other examples of features in games: saving any time; multiple languages; an advanced difficulty mode; etc.

variable -- a place in code to store information. We can create variables in Scratch by going to the Variables in the block menu and selecting Make a variable (see below).



Once you create a variable you get the following blocks to use with that variable:



Variables can be created that can be used anywhere in the program, or only by one particular sprite.

decision (or if statement) -- a script can make a decision about what to do next based on the current situation. This is usually done with the "if" blocks:



The block in the lozenge shaped hole at the top of the if block is the **test**. The test controls whether the code inside the if block gets run (or, in the case of "if else" it decides which of the two pieces of the script get run). In the decision below the sprites says "Hello" if (and only if) the sample variable has a value greater than 3.



test -- the part of a decision or loop that controls what happens. The outcome of a test will be either true or false. You can control the behavior of your script depending on which it is by using decisions and loops. See **decision** or **loop**.

loop -- a loop allows you to have part of script run repeatedly for some number of times. You can control how many times the loop runs either by specifying it directly (see left example below) or by having a test (see right example below). You can also have loops that just run forever until the program ends (not shown).



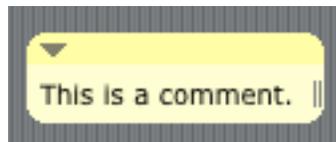
projectile -- a generic name for a missile, bullet, laser beam, acorn, basketball, or any other object that gets thrown, hurled, or shot as part of a game.

sandbox -- a program (or part of a program) used by the programmer to experiment and try out new things. A sandbox is used by the programmer(s) and is usually not shared with other people on a software team.

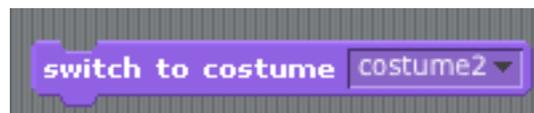
prototype -- a sample program with a limited set of features that is used to evaluate software to see how fun, challenging, and/or marketable it is. The purpose of a prototype is to demonstrate what the program might look like to people other than the programmers, so unlike a sandbox it is often shown to others on the software team or even people outside the team (like executives or people attending a trade show).

beta -- a version of a program that is almost finished and is being circulated for testing, marketing, etc. Unlike a prototype the beta version will have many features coded (perhaps all).

comment -- a note in the code to leave some information for the programmer (either for yourself for future reference or for other people reading your code). A comment is not executed by the computer; it is just a note for people. Comments in Scratch are created by right-clicking in the script editor and look like this:

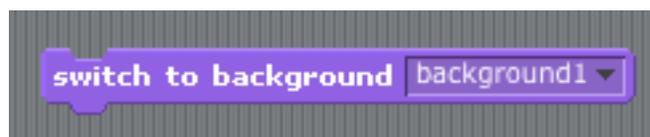


costume -- a graphic associated with a sprite. A sprite can have multiple costumes, only one of which is displayed at a time. “Costume” is a Scratch-specific term (other programming tools or communities might use a different word to mean the same thing). In Scratch, you can switch costumes with blocks like this...



stage -- In Scratch, the stage is behind all the sprites and can be used to manage aspects of the program that are shared across sprites or that are relevant to the display of a **background** (see below). The stage has its own scripts, but unlike a sprite the stage cannot have its own variables. The stage is specific to Scratch and unlike the word “sprite” is not in general use in the programming industry.

background -- a single graphical image associated with the stage. You can switch between backgrounds using blocks like this from the stage’s scripts:



animation -- animation is a technique that gives the illusion of movement by showing multiple images (called **frames** or **cells**) in rapid succession. In Scratch we can produce animations by having a sprite change quickly change costumes.

frame rate -- the speed at which successive frames are displayed by a game (or other software), usually measured as “frames per second” (FPS). A frame rate of roughly 26 FPS is required to have animation that seems more or less smooth to the human eye.

simulation -- a program that attempts to model some process in order to experiment or entertain. Simulations usually model something real although in the context of game programming what is “real” is more of a spectrum.

model-view architecture -- a way to design a piece of software that separates the user interface (the **view**) from the internal workings of the program (the **model**). The model itself contains both a **state**, which roughly speaking corresponds to the variables and their values, and the **logic**, which roughly speaking corresponds to the scripts or code.

client-server -- a technique for organizing software, related to but not exactly the same as model-view. The **server** (or **host**) has fundamental control of the game (or whatever) and generally holds most of the state. **Clients** connect to the server to play and each client has its own view of what’s going on. Games like MMORPG’s and multiplayer shooters rely heavily on this architecture.