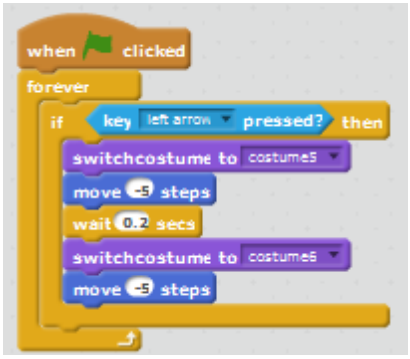
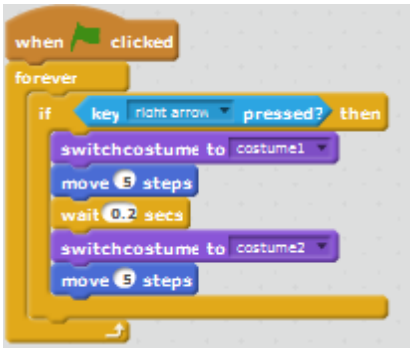
Simple Movement Techniques

Simple movement two costumes to simulate stepping, facing left

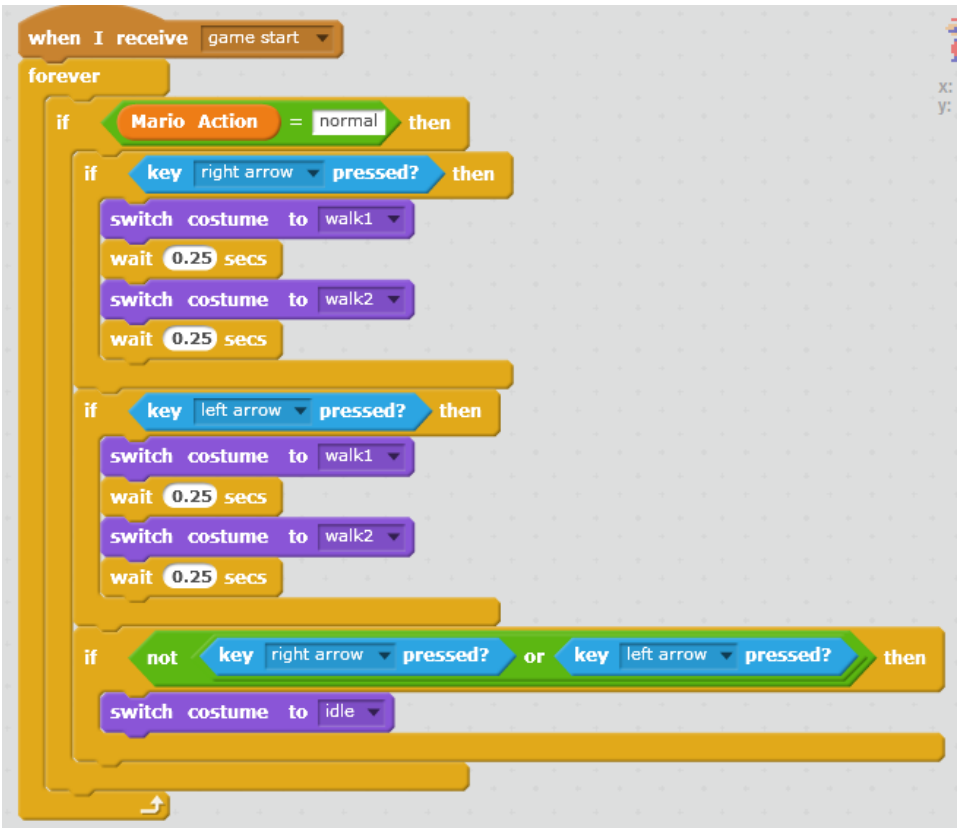


Simple movement two costumes to simulate stepping, facing right

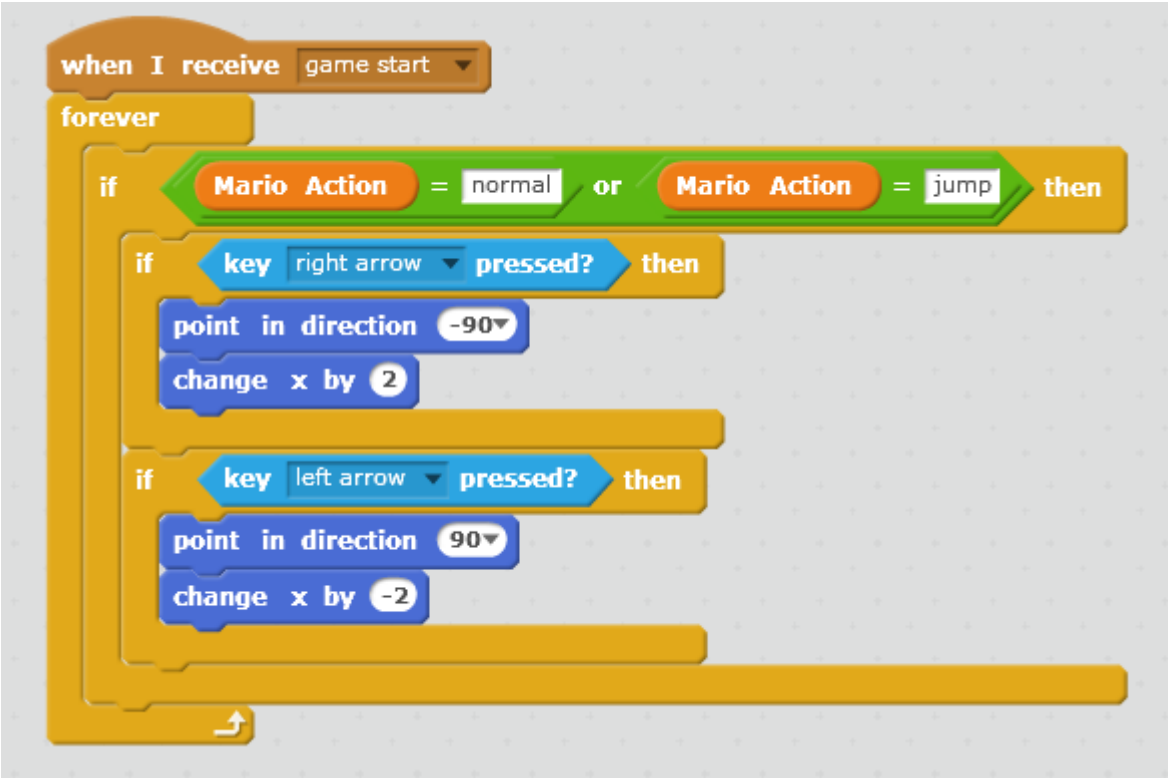


Or how about even simpler code? Does this code work as expected?

How about separating the Appearance from the movement



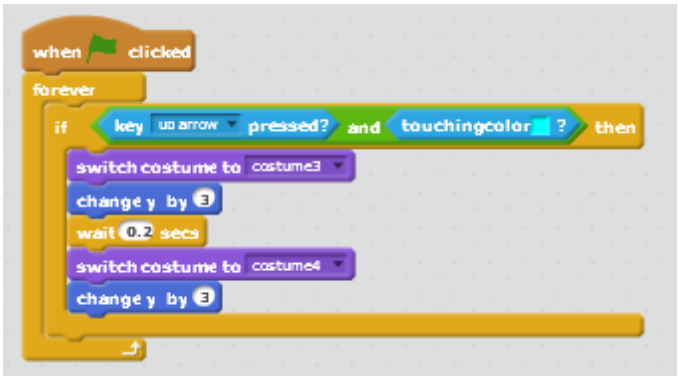
Control of movement on X-axis



And then maybe some gravity to keep Mario on the platform – note that this also keeps him on the platform (Red in this case)

CLIMBING

Climbing something Blue (eg ladder), using costumes to simulate movement



Climbing down something blue (eg Ladder) using costumes to simulate movement

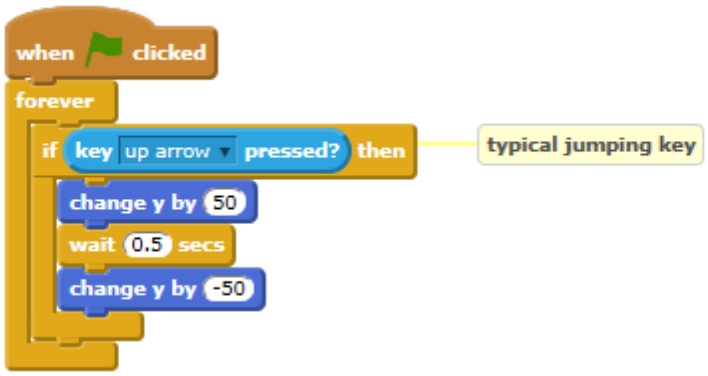
when clicked
forever
if key down arrow pressed? and touchingcolor ? then
switch costume to costume3
change y by -3
wait 0.2 secs
switch costume to costume4
change y by -3

Jumping Techniques

Jumping - Scratch Wiki
http://wiki.scratch.mit.edu/wiki/Jumping

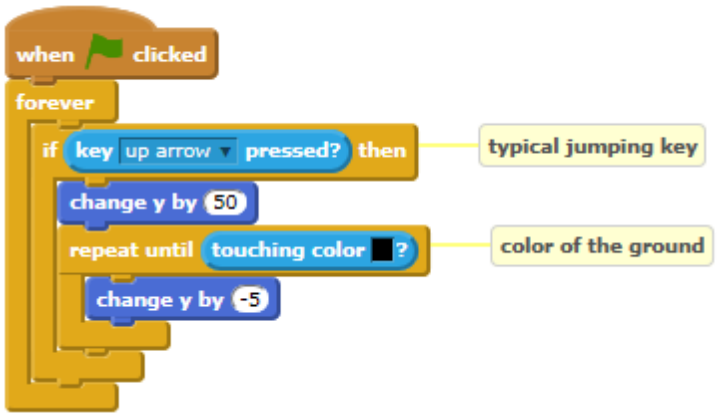
Simple jumping

The following is commonly used in animations and results in a sprite effectively teleporting upwards, and then downwards again.



Falling

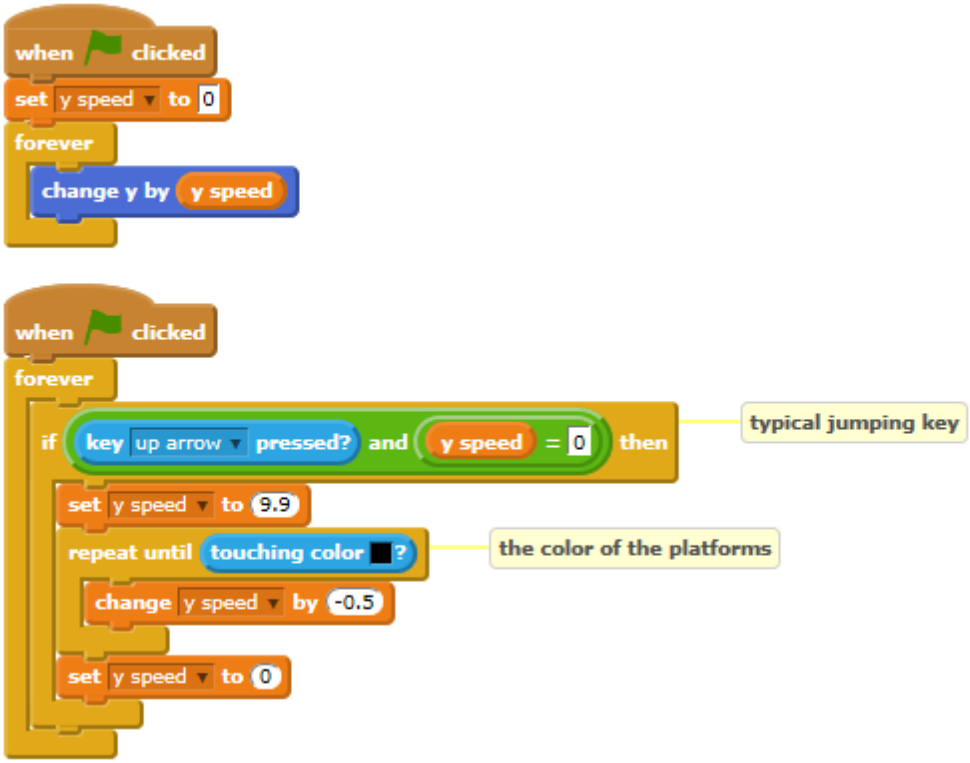
The following script "teleports" the sprite upwards, and then has it fall back down at a constant rate, until it lands on a platform



Physically Accurate Jumping

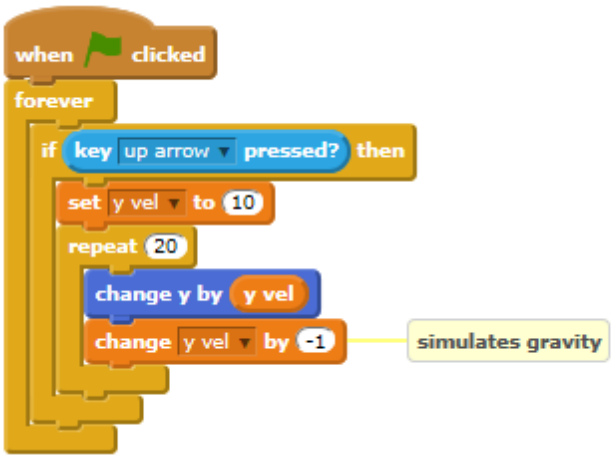
With Ground Detection

A more realistic effect for jumping commonly used in games is the following, using a variable to control the vertical speed of a sprite (simulating gravity) is this:



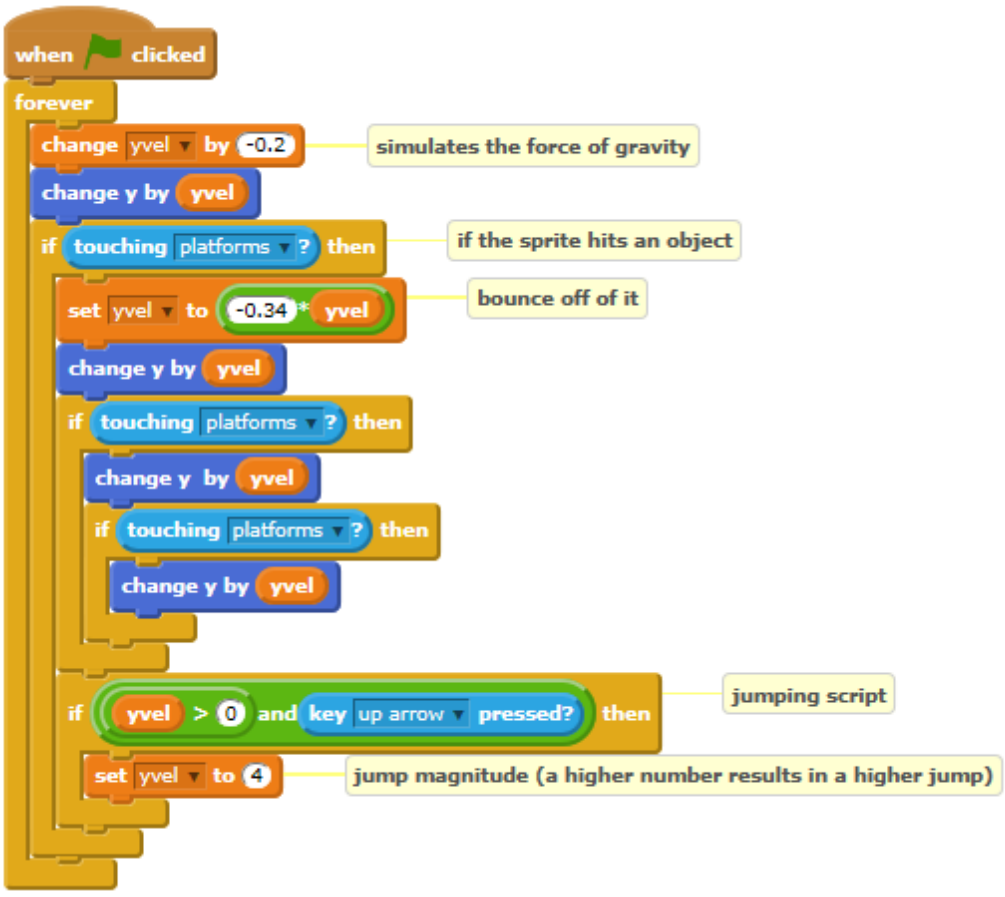
Without Ground Detection

If one does not need to sense the ground and simply want a realistic, gravitational jump, that will end at the starting point, the following script can replicate the jump:



Advanced Jumping

The following script includes velocity-based jumping and advanced landing. The sprite is constantly acted upon by the force of gravity and cannot pass through sprites. The sprite may appear to "bounce: slightly when "at rest":



Now add these techniques to your programme or use these in a sample project!

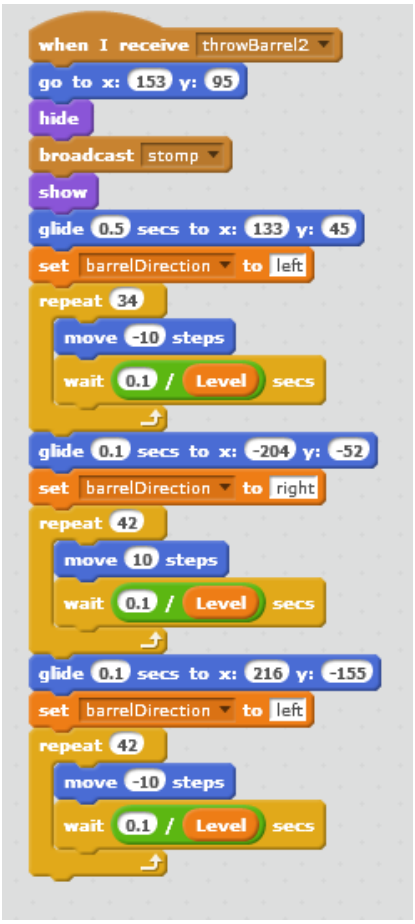
Barrel Movement

Jumping - Scratch Wiki
http://wiki.scratch.mit.edu/wiki/Jumping

Start Simple – drop a barrel from the top and roll along one level!

This Barrel stars at the top platform, falls to the bottom and rolls to the left at a set speed

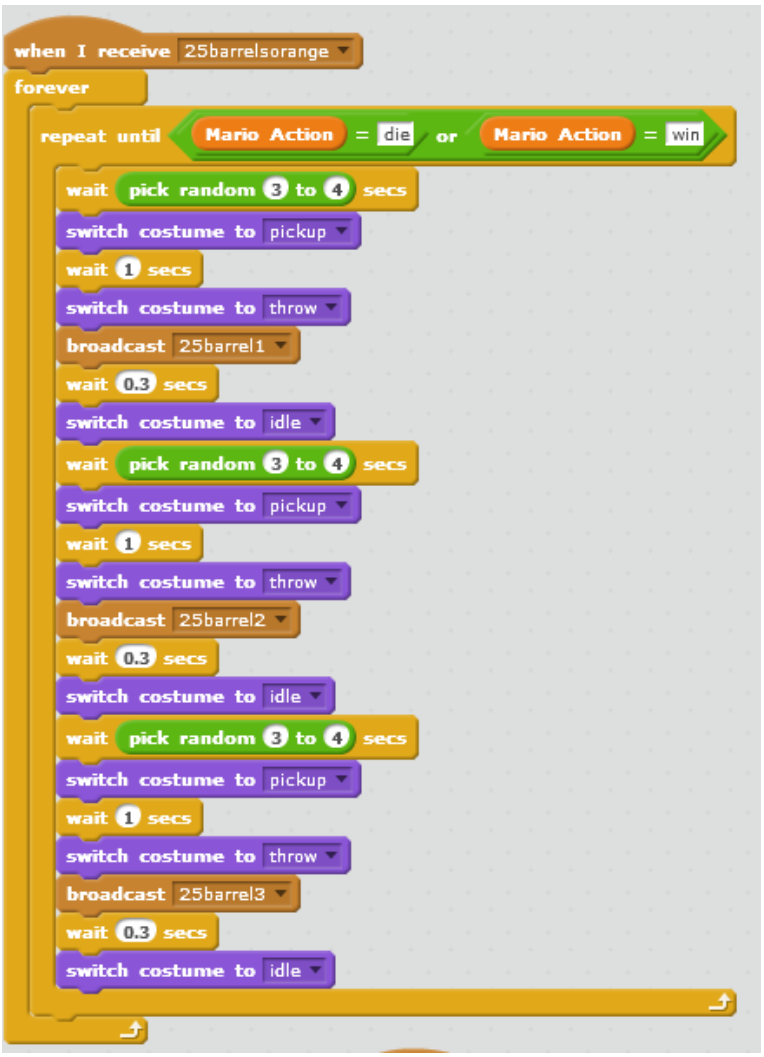
Now make the barrel roll along all levels!



We repeat the code across gliding between the levels – does this code work reliably?

And finally add some variation to Barrels

The first barrel triggers the remaining barrels

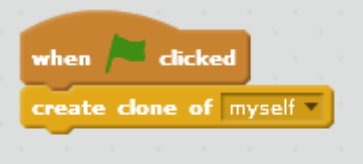


And then each Barrel Sprite called will have different paths through the platform

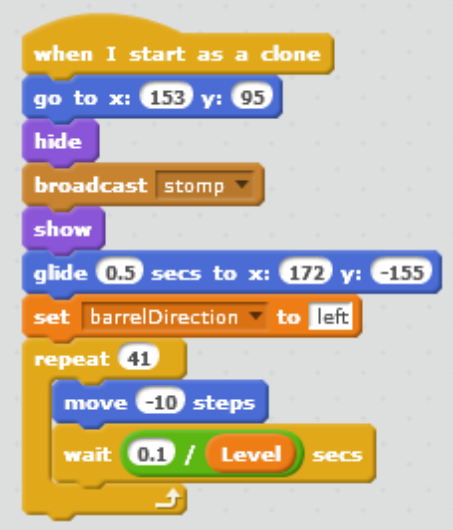
Using Cloning for Barrels

You may have found it easy to simply create a few barrels, with the intention to have different barrels travel different paths at different times. You may find it easy to create one, then duplicate it, then only make minor changes to it. That's easy to do, but you might find yourself improving your game and having to make the same changes to each of these sprites, making it difficult to maintenance you code. In some cases where you want lots of sprites that are really doing the same thing, it may be simply impractical to duplicate these. Cloning lets you create a copy of the sprite at runtime, and to control when the sprite is created and when its deleted. This can be useful in tower defense games, for example, for a wave of objects. Clones of a sprite will be the same as the original, or parent sprite, but as a separate instance. Clones inherit the parent's scripts, costumes, sounds, and properties, but can then be modified.

A clone can be as easy as:



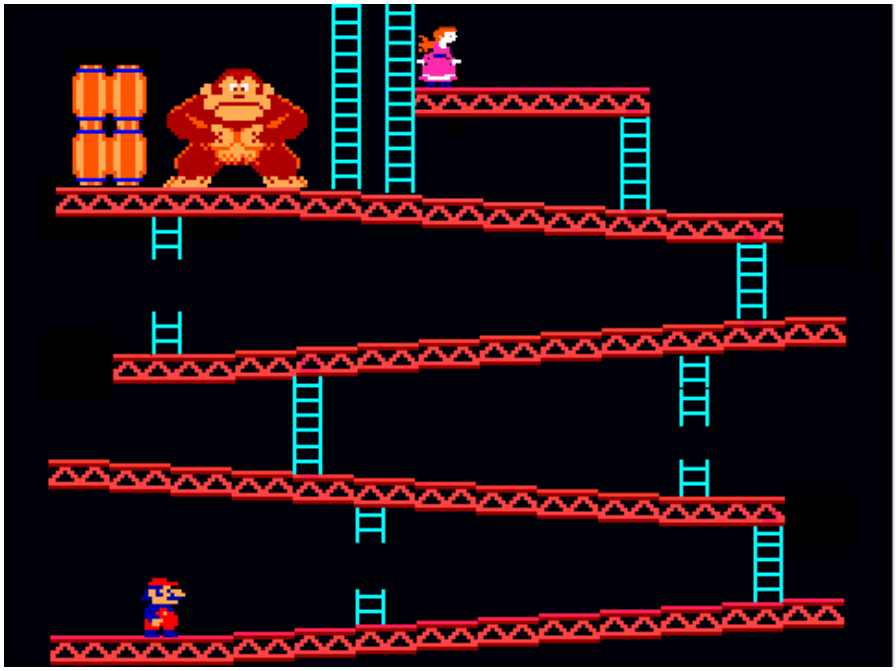
And associated specific code when the Clone Starts:



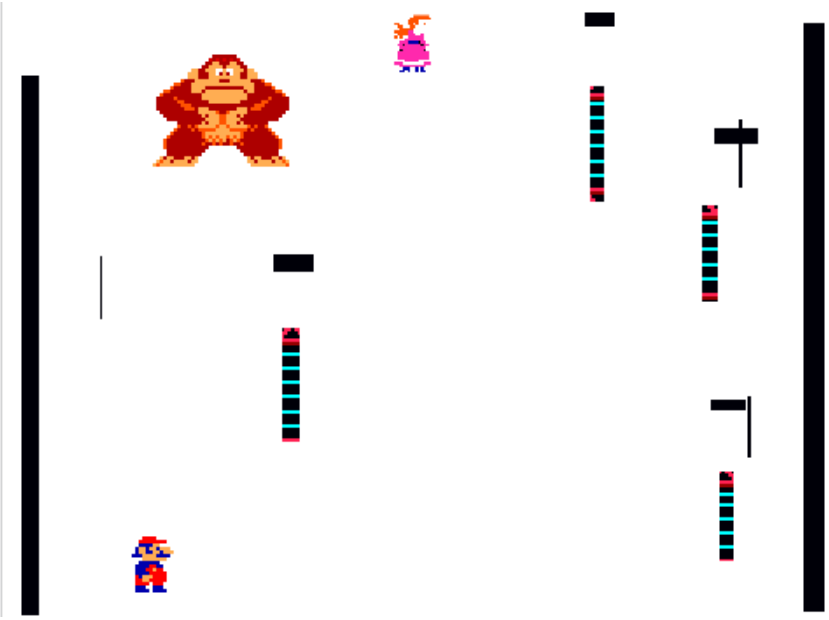
But of course you might need to make the operation of the clone dependent on variables to control the behaviour of the clone and you should have code that deletes the Clone, either on an event or when the function of the clone has been completed.

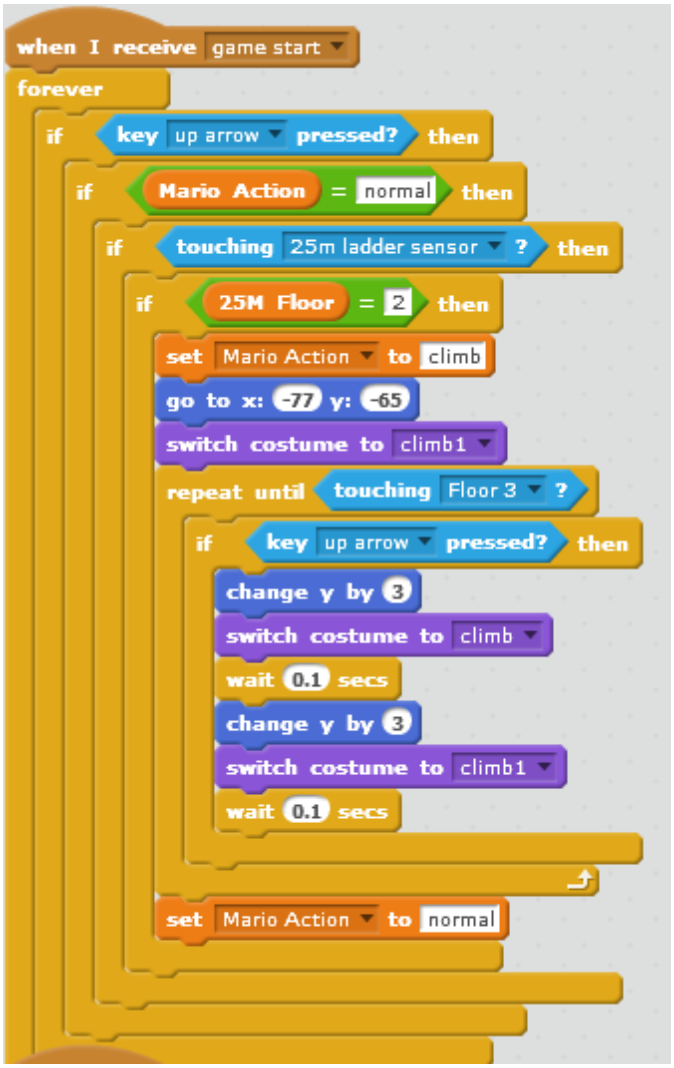
Because clones can be create in code, any variables used in the sprite code can either be associated only with this clone, or with all clones. This gives us some flexibility in controlling how code will work.

Sensing with Hidden Sprites



Here's what the Platform could look like, now if we change/delete the background we see a few hidden sprites:



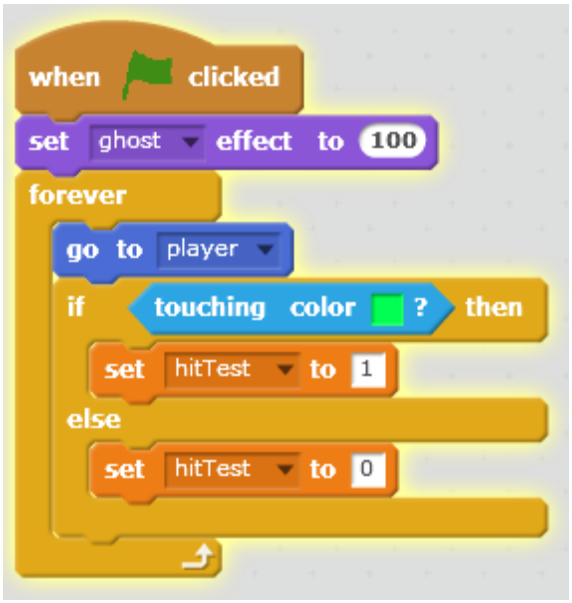


Where some of these “invisible” sprites act as sensors which set the floor level we are on and control which code is being executed....

Or to detect if Mario reached the Princess

Sensing Block around Mario

Four sprites with a sensing function that can be set to surround Mario and act as the sensing function using a local variable called “hitTest”



Now we can check if the sprite has touched something and then effect a move of the platform by changing the X or Y co-ordinates for the Platform/Background sprites

Improvements to the Game

- Randomizing the possible barrel paths, including avoiding ladders?
- Can we use Cloning to reduce the number of sprites?
- Add Scoring
- Add Lives
- Add Level's of Difficulty
- Add fireball's to chase Mario

To Be Added

Code for Princess