

This project explores how to create the popular Nintendo game of Donkey Kong using Scratch. This Sushi card will help you work through the different parts of the Game building on what you have learnt about Sprites, Movement, Sensing and Game Strategy.



For the Donkey Kong game, we need a few sprites to recreate the game...

- Donkey Kong
- Princess
- Mario
- Barrels
- Ladders
- Platforms
- Scores
- Heart (when Mario reaches the Princess)

To get started go to our Scratch GitHub repository at:

<https://github.com/CoderDojoDunLaoghaire/ScratchSushi>

In the SRC folder of this Sushi Pack you will find this Sushi card and a starter Project **Donkey Kong Base.sb2** project with some starter Sprites to get you going. There's a few basic things that need coding, but its up to you to decide how your version of this game will be coded!:

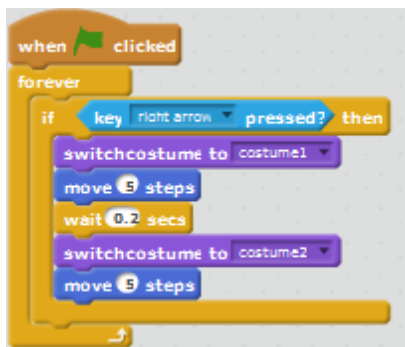
- Move left and right
- Move up and down ladders
- Jump Over barrels
- Die when touching Barrel or Donkey Kong
- Roll the barrels
- Score Points

1 Simple Movement Techniques

Simple movement using two costumes to simulate stepping, facing left



Simple movement using two costumes to simulate stepping, facing right



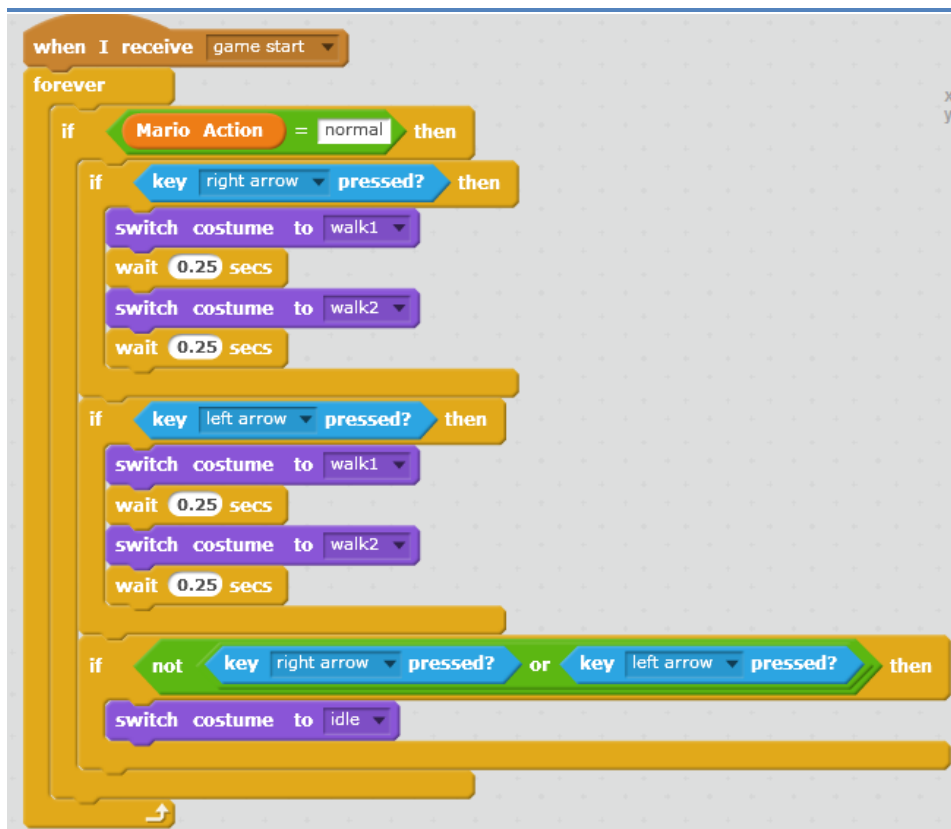
Or how about even simpler code? Does this code work as expected?



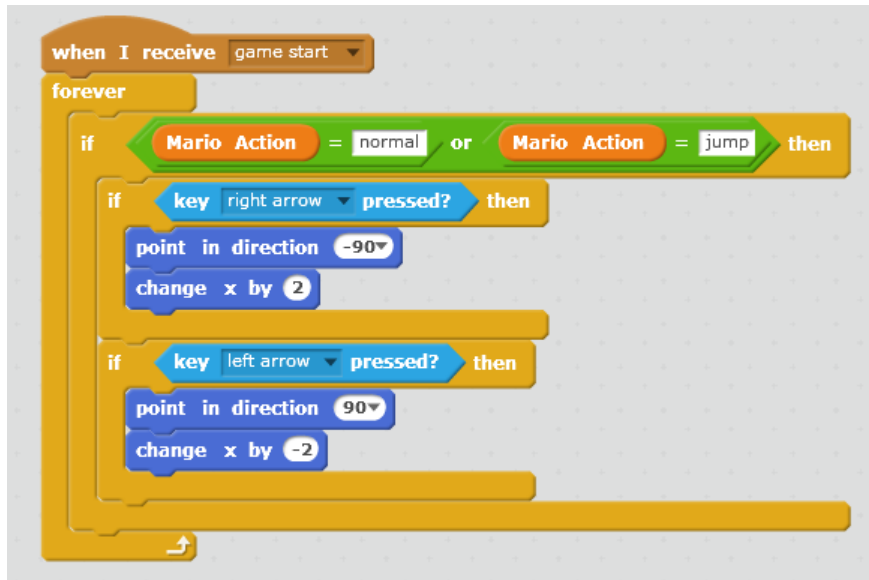
2 Movement and Appearance

As you start to add code to your project you will start to mix up code that deals with Movement and code that deals with Appearance. Having this in one block means the two become dependent upon each other and can often slow the program down. To avoid this, we can use the Parallelism feature of Scratch and separate this code, making the program work better, and the program easier to debug. It would also be a good idea to name your costumes and sprites so it makes your code easier to read!

For example, this code snippet deals solely with Appearance.



And this code controls the movement of Mario along the 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)



3 Climbing

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



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



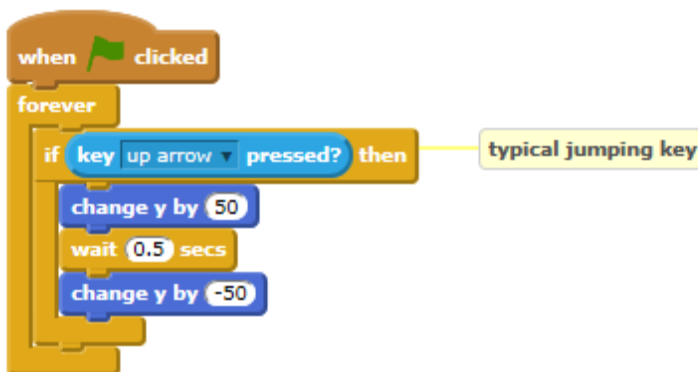
4 Jumping Techniques

Take a look at the Scratch Scratch Wiki for good examples of Jumping techniques

<http://wiki.scratch.mit.edu/wiki/Jumping>

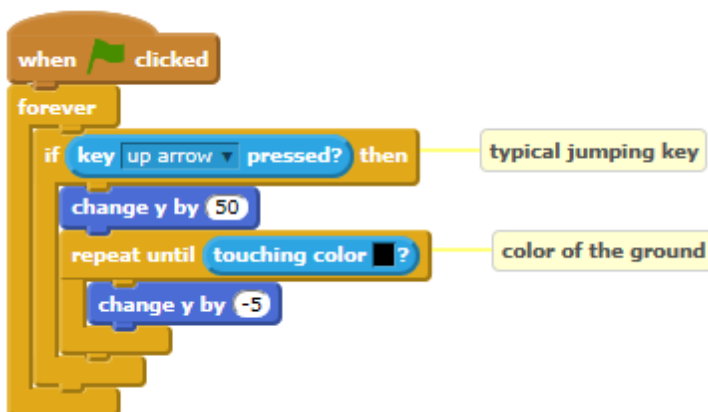
Simple jumping

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



Falling

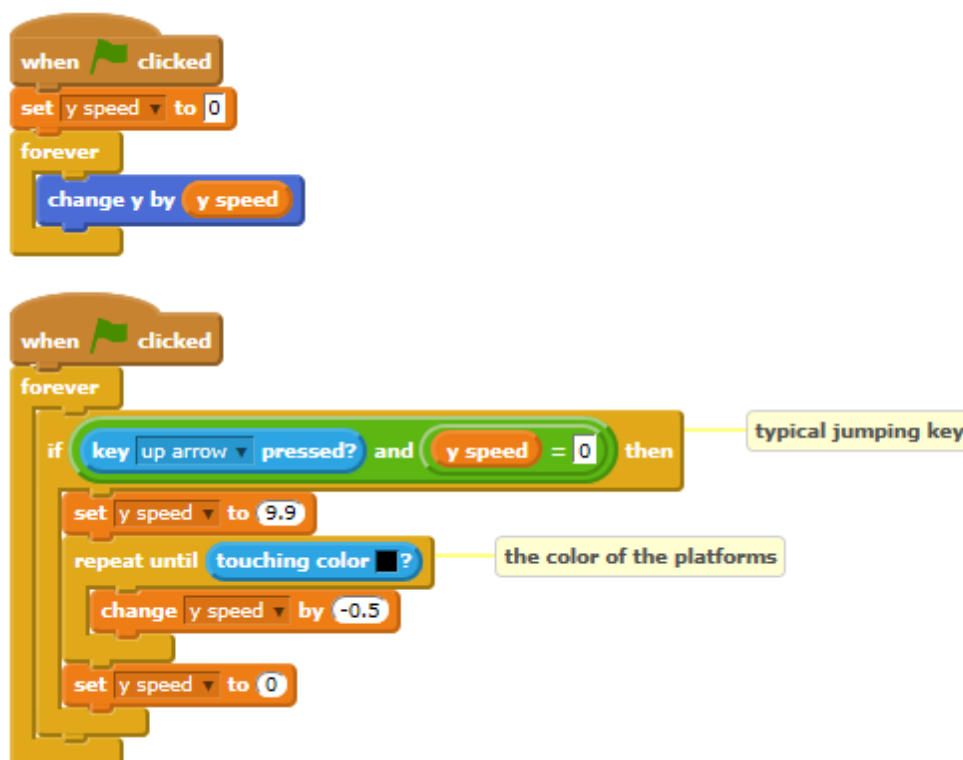
The following script "teleports" the sprite upwards, and then has it fall back down at a constant rate, until it lands on a platform



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:



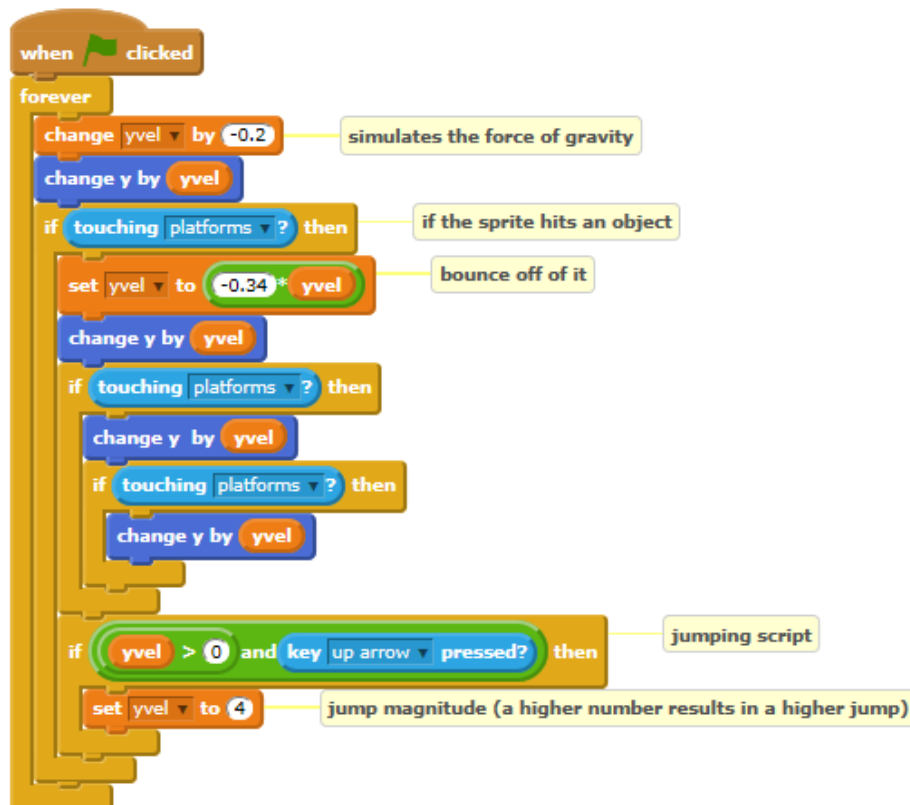
Without Ground Detection

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":



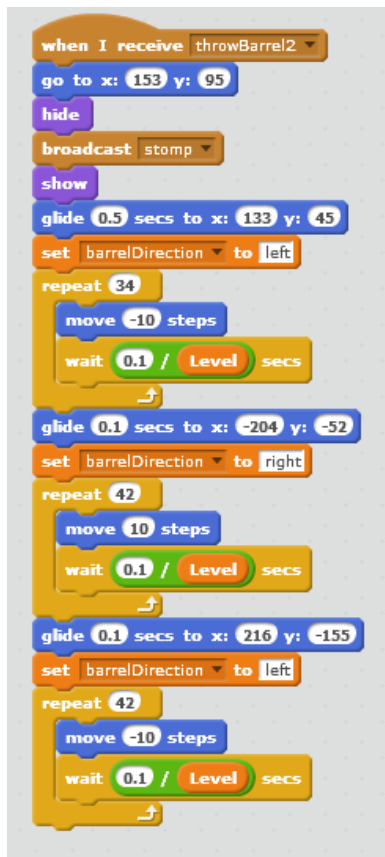
Now add these techniques to your program or use these in a sample project!

5 Barrel Movement

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



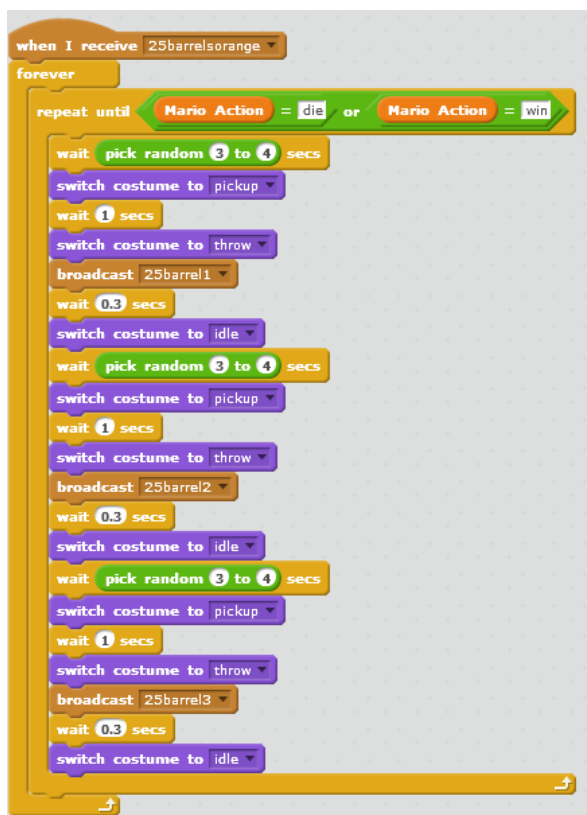
This Barrel starts at the top platform, falls to the bottom and rolls to the left at a set speed. Now make the barrel roll along all levels!



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



The first barrel triggers the remaining barrels

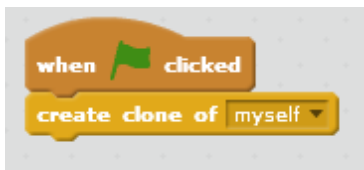


And then each Barrel Sprite called will have different paths through the platform

6 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 maintain your 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 it's 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:



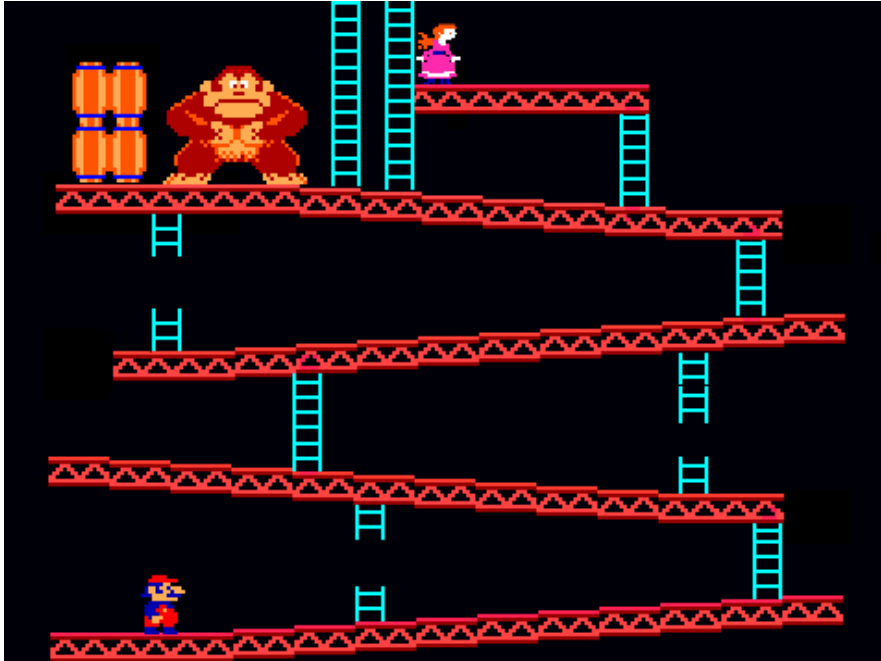
But of course you might need to make the operation of the clone dependent on variables to control the behaviour of the clone and you should have code that deletes the Clone, either on an event or when the function of the clone has been completed.



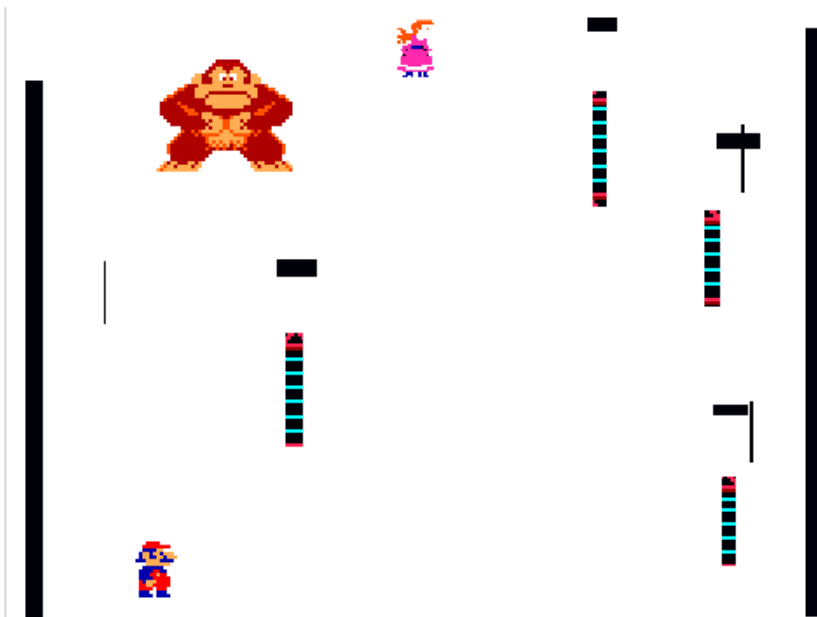
Because clones can be created in code, any variables used in the sprite code can either be associated only with this clone, or with all clones. This gives us some flexibility in controlling how code will work.

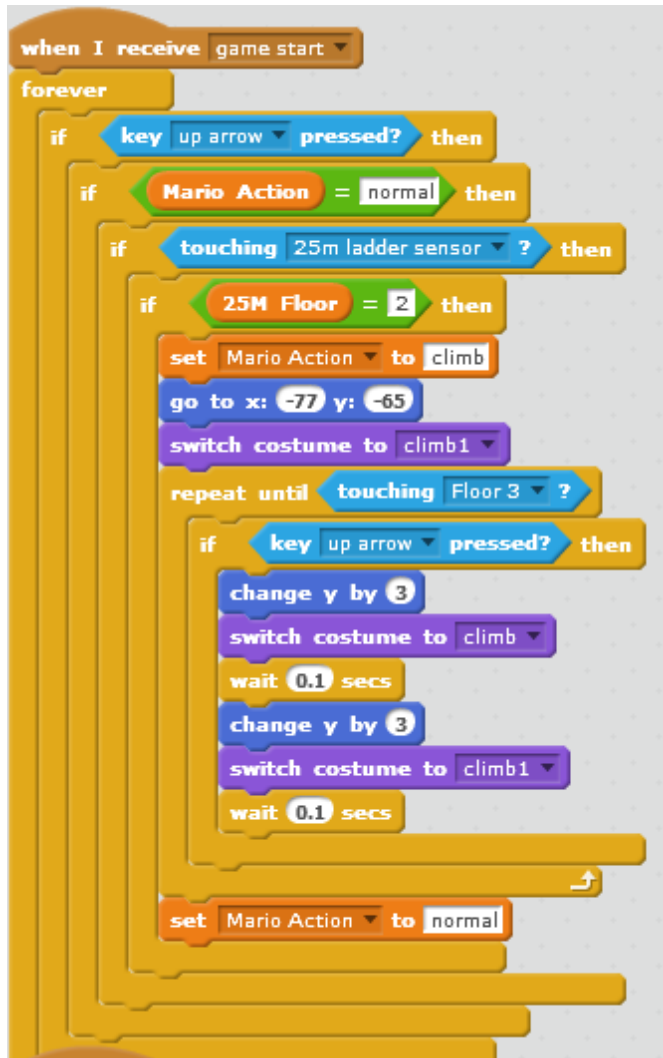


7 Sensing with Hidden Sprites



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

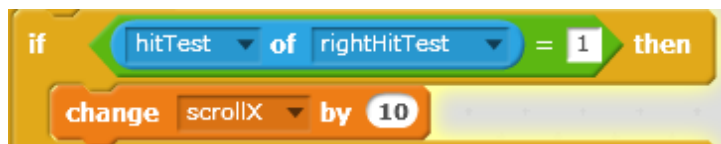
8 Sensing Block around Mario



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



9 Improving the Game

How do you think we can improve the game? Here are some suggestions:

- Use Cloning to reduce the number of sprites
- Keep score of the games and lives
- Randomize the possible barrel paths, avoiding ladders
- Add levels of Difficulty
- Add fireballs that chase Mario

This Sushi card was based on the Original Donkey Kong material developed by Simon Lewis can be found at:

Week 1 - Overview: <http://www.anseo.net/scratch-saturday-platform-games-1>

Week 2 - Moving Mario: <http://www.anseo.net/scratch-saturday-platform-games-2>

Week 3 - Platforms and Ladders: <http://www.anseo.net/scratch-saturday-platform-games-3>

Week 4 - Donkey Kong: <http://www.anseo.net/scratch-saturday-platform-games-6>

Week 5 - Barrels: <http://www.anseo.net/scratch-saturday-platform-games-7>

Week 6 - Reaching the Princess: <http://www.anseo.net/scratch-saturday-platform-games-4>

Week 7 - Adding the Heart: <http://www.anseo.net/scratch-saturday-platform-games-5>

Week 8 - Improvements <http://www.anseo.net/scratch-saturday-platform-games-10>