

How to Build a Space-based Game in Scratch

Workshop by Natalie Roberts
nar25@aber.ac.uk

Scripts Area

Block Menu

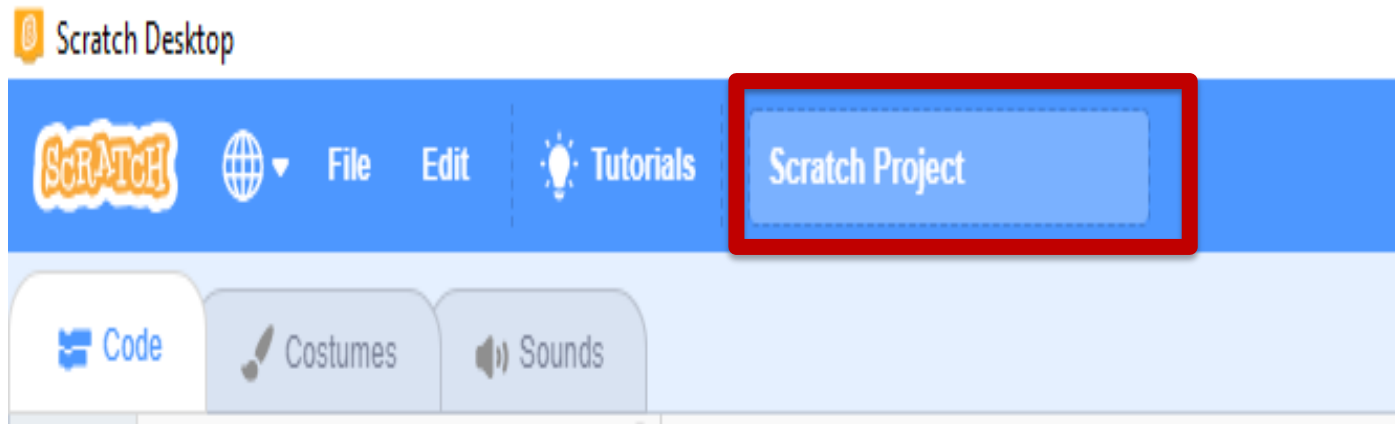
Stage



Sprite List

Stage Contents

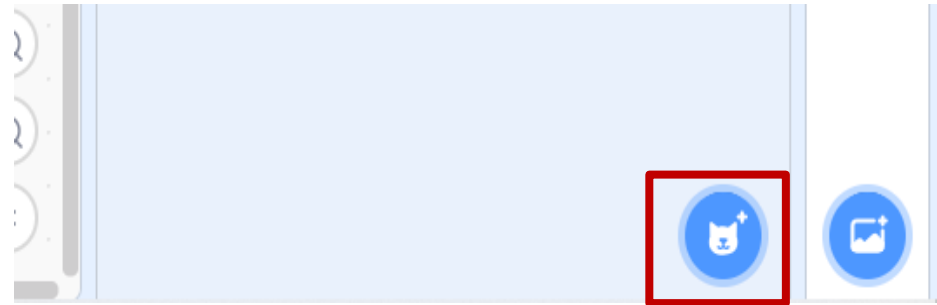
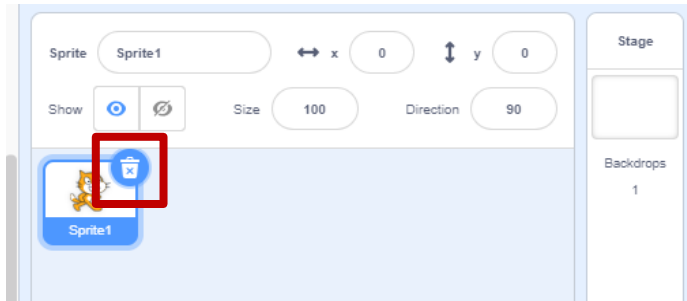
Think of a name for your project and type it in the box shown below



Waiting to move on? Start having a look through the block menu to get an idea of what instructions are available and where to find them

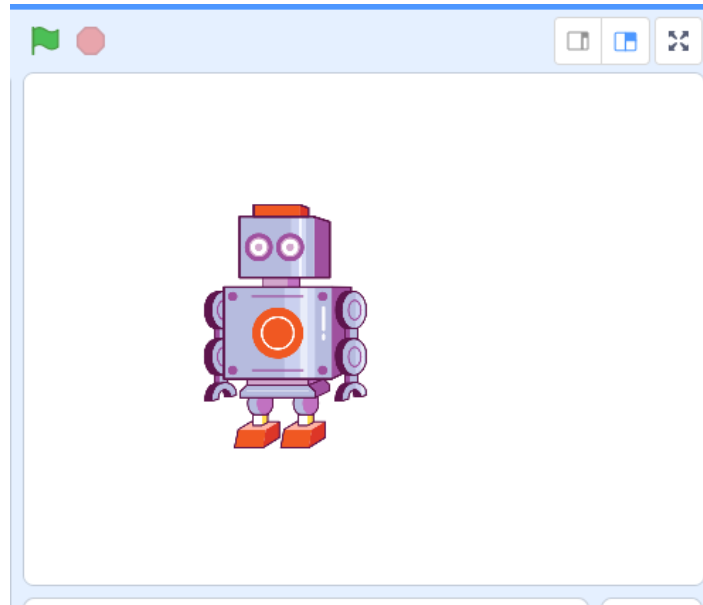
A cat is not the right character for this project so we shall change it to a robot.

Delete the cat from the **Sprite List** and then click on the add sprite icon.



Scroll through the collection and select 'Retro Robot' from the list

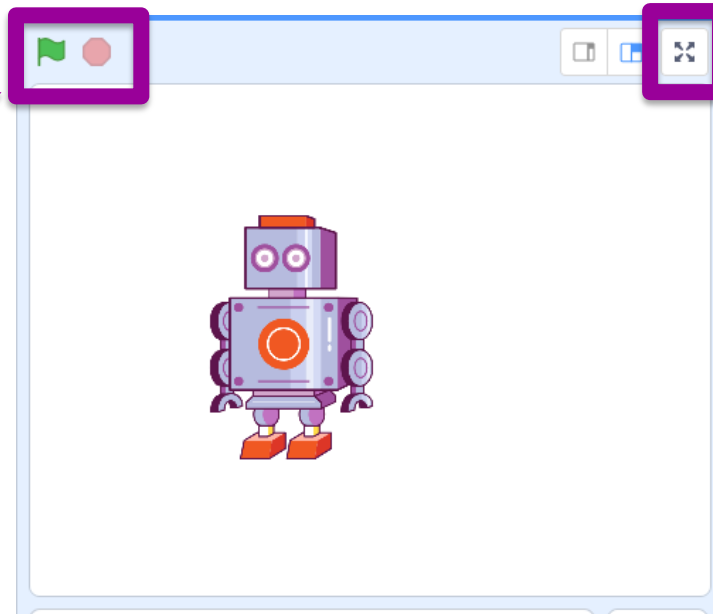
Your **stage** should now look something like this:



Waiting to move on? Check out the “costumes” tab above the blocks list and choose which robot you prefer

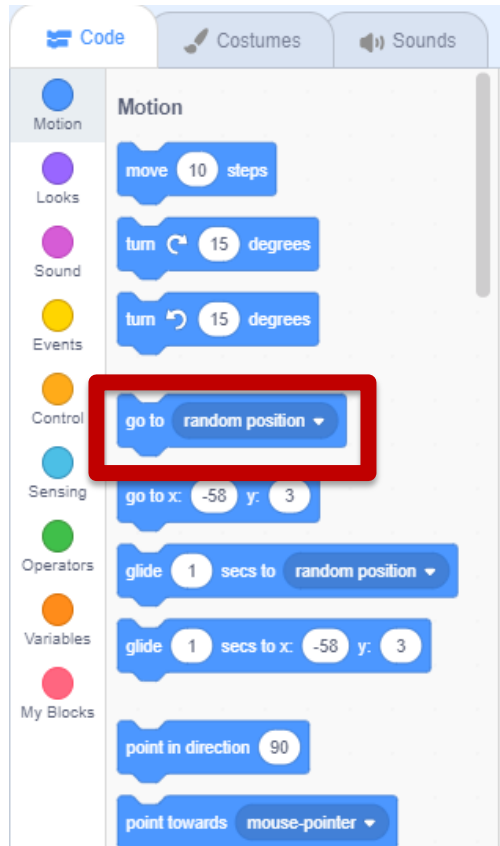
How to use the stage

The green flag will start your program running in the stage. The red circle will stop the program



This will expand the stage to full screen size

Waiting to move on? Check out the “costumes” tab above the blocks list and choose which robot you prefer

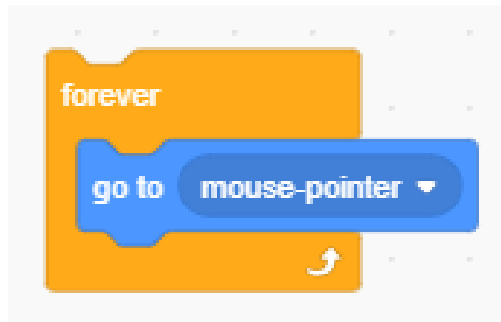


We want the robot to be our main character and move it with the mouse around the screen.

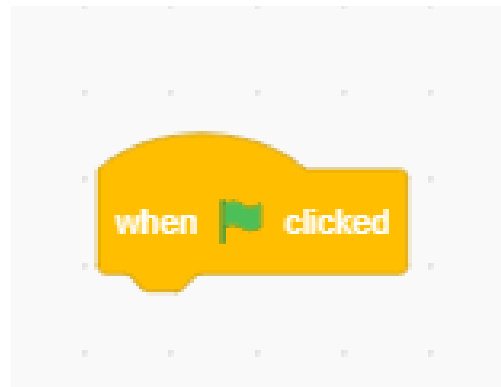
Find the block highlighted in the **block list**, click on the drop-down menu and select 'mouse-pointer'

Now drag that block into the **scripts area**. This is our first line of programming for the game.

Waiting to move on? What other ways could we use for moving the player character in a game?

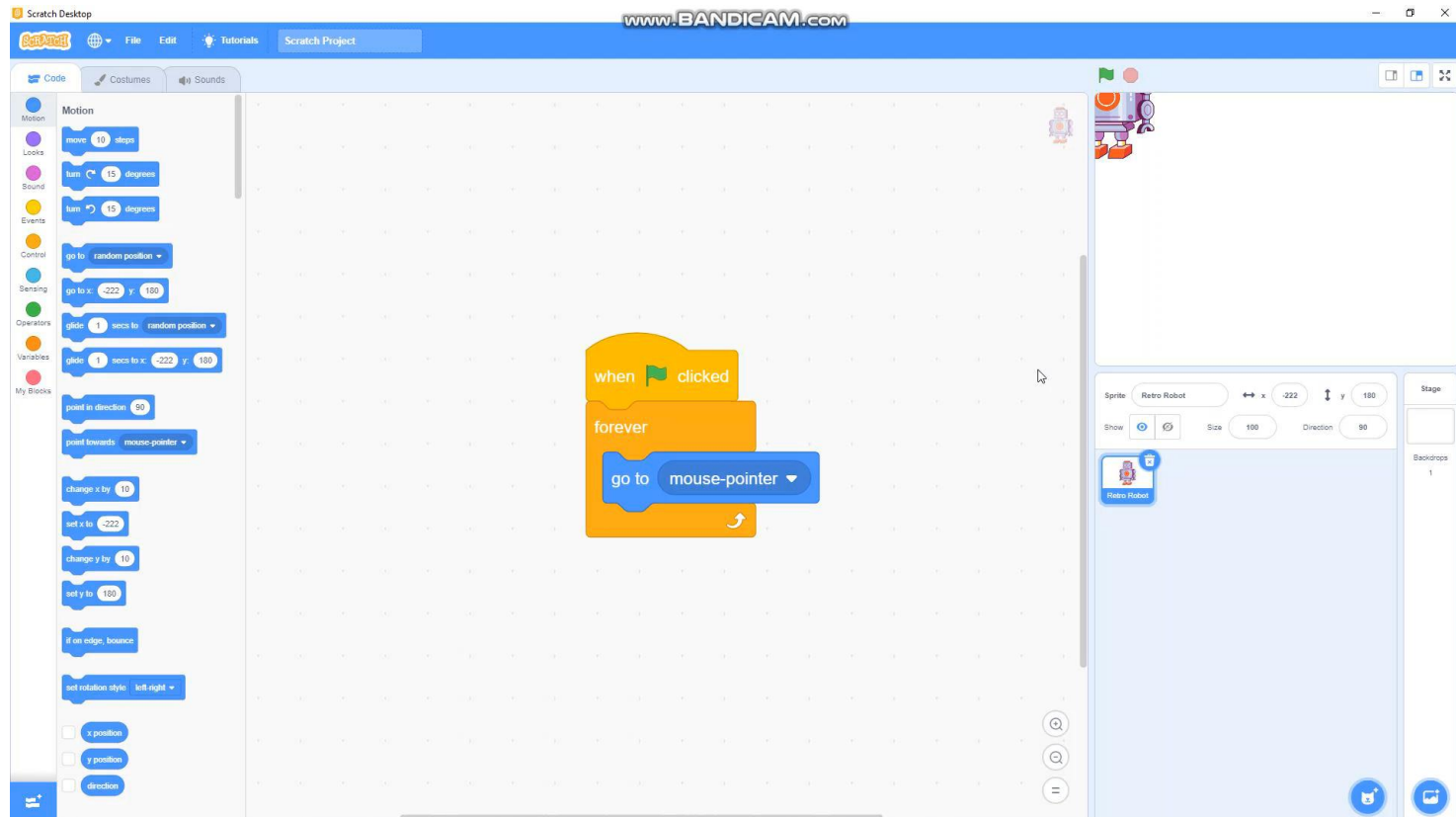


So the program knows to always have the robot following our mouse movements we need to put it in a *forever loop*. This means that it will keep doing this forever, or until we end the program

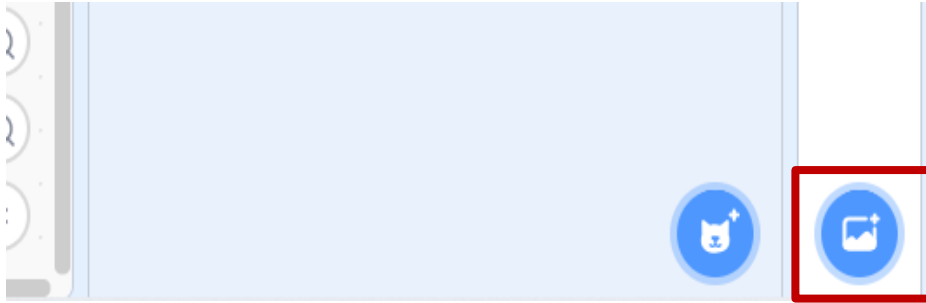


To complete this script we need to tell it all when to start. Find the block shown here and drag it to attach on top of the forever loop block

Now to test if the program works...

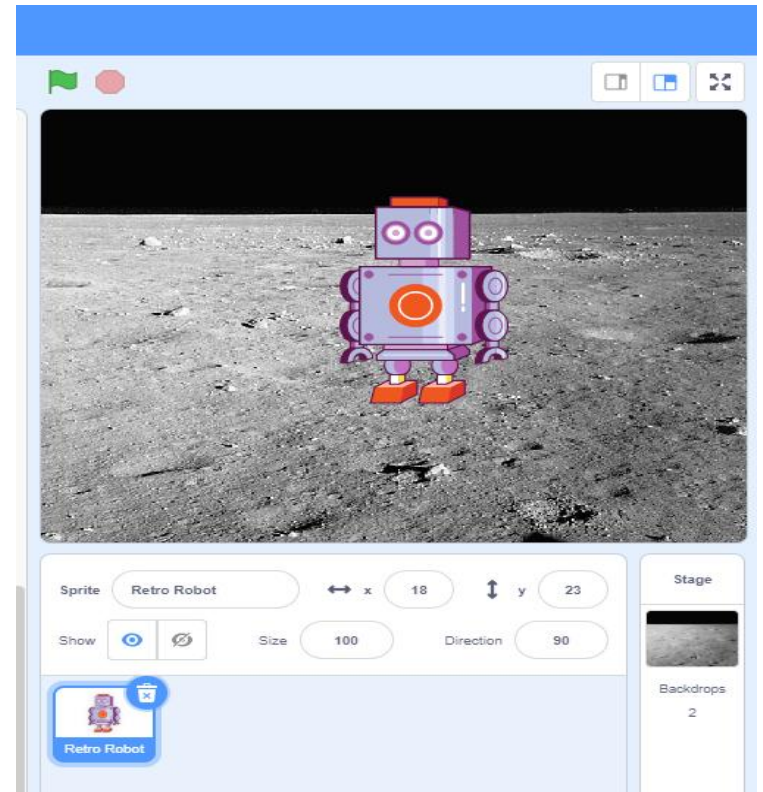


Adding a background



Click on the add background icon at the bottom of the **stage contents** window. See above image.

Select the moon surface image and your stage should now look like the one shown on the right.



Waiting to move on? Start thinking about what else the game needs

Collecting Items

First, we need an item to collect. This means we need to add another sprite. Try to remember how you added the robot onto the **sprite list** and do the same, this time selecting 'Rocks' instead of 'Retro Robot'.

You should now have both the robot and a rock pile on your **stage**.

Waiting to move on? What do you think will happen if we run the program now?

Now to test if the program works...

- The rock should remain still.
- The robot will move through the rock as if it wasn't there
- Nothing happens when the robot and rock touch

The robot needs to pick up the rocks. How do we do this?

First make sure you are selected on the Rocks in the **Sprite List**, and then start the new script in the **Script Area** with this:

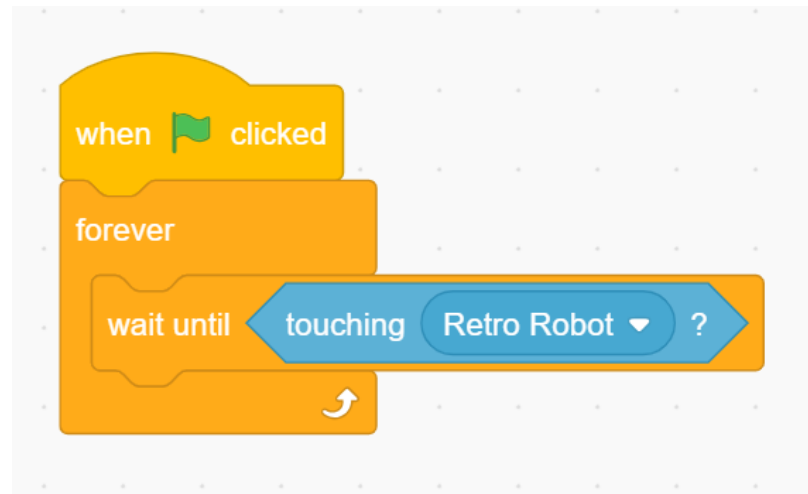


Waiting to move on? Have a look at the block list and see if you can find blocks that will fit in the 'wait until' hexagon

We need to tell the rocks to 'wait until' they are 'touching' the robot.
Can you find and insert the correct instruction block from the **Block List**?

*Waiting to move on? How could we tell if this new section
of script works?*

We need to tell the rocks to 'wait until' they are 'touching' the robot.
Can you find and insert the correct instruction block from the **Block List**?



Waiting to move on? How could we tell if this new section of script works?

To test if this new instruction works we need to add another instruction inside the forever loop (below the wait until command) that will play a pop sound until done. Can you find and add the block into the script?

Waiting to move on? You can alter the sound of the pop in the 'sounds' tab above the block list.

Now to test if the program works...

- The rock should remain still.
- The robot will move through the rock as if it wasn't there
- When the robot and the rock touch a popping noise will happen until they are moved apart

Lots of popping but no collecting....

The rocks can now detect the robot touching it and responds with a sound. We need it to be 'picked up' by the robot.

We can do this by making the rocks move each time the robot touches them.

Add a block to tell the rocks to 'go to random position' in the forever loop below the play sound instruction.

Waiting to move on? You can alter the appearance of the rock in the 'costumes' tab above the block list

Now to test if the program works...

- The rock moves whenever touched by the robot
- There is a popping sound when the robot and rock touch

That pile of rocks is mean and seems to keep dodging the robot rather than being collected.

How can we make it seem more like the robot is collecting them?

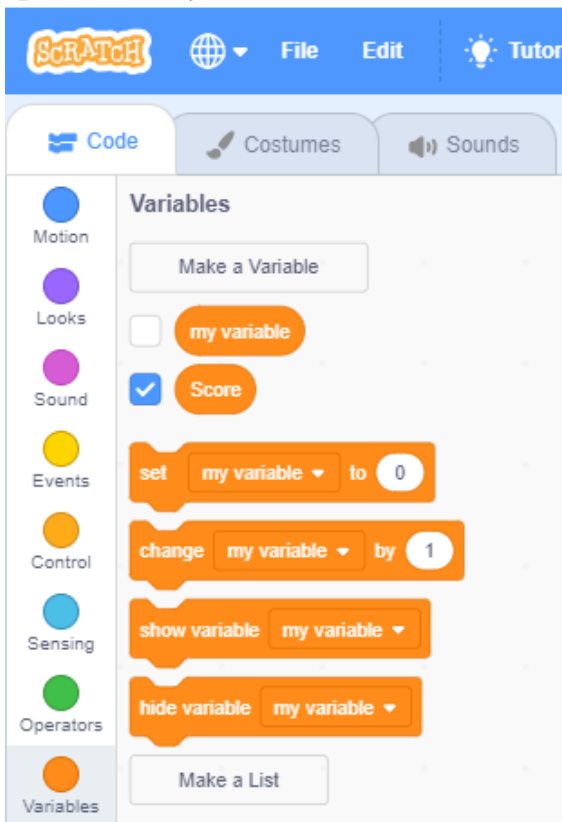
Scoring System

By adding a score to the screen that goes up every time the player collects a rock will make this seem more like a game.

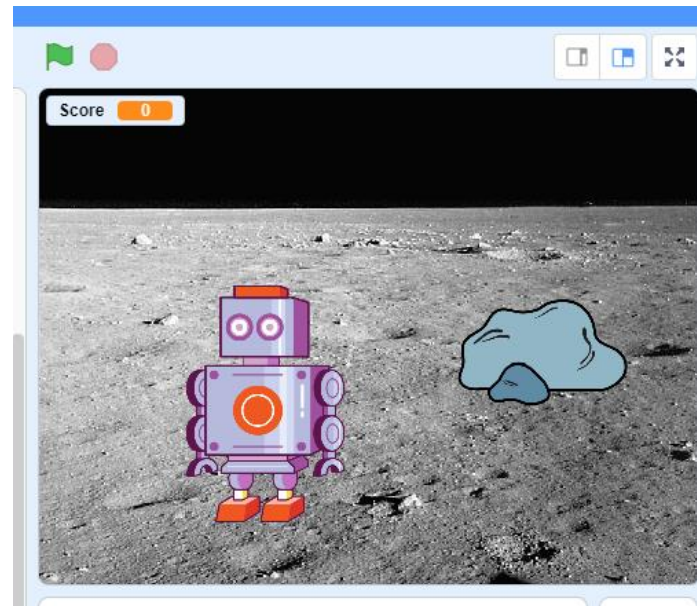
To do this we will have to create our own variable –a place for storing values that can be used in the scripts

Find ‘ Make a Variable’ on the block list, click on it and name it
Score

Scratch Desktop



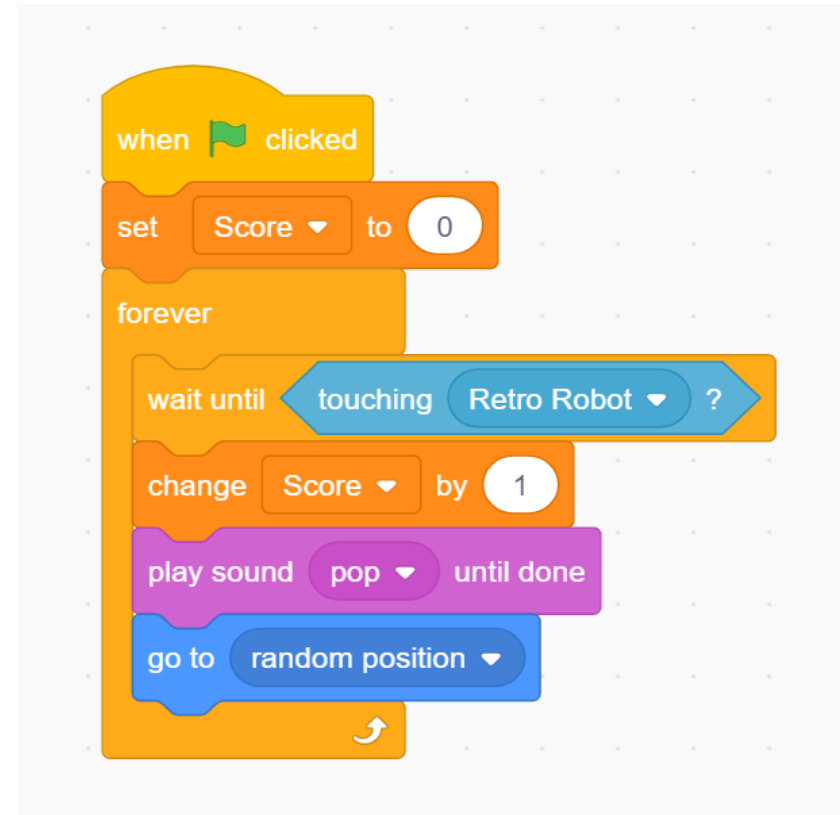
Your new variable should now be visible in the Block List. The tick in the box next to it means that this is visible on the stage.



Waiting to move on? Think about how you are going to score this game. Is a rock worth 1 point or more?

We need to tell the program that the score needs to be zero when we start the game and to add points every time we touch the rock.

To do this, we add the information into the Rocks script as shown here.



Waiting to move on? Why is the 'set score to 0' instruction not in the forever loop?

Now to test if the program works...

- The rock should pop and move whenever the robot touches it
- The score should start at zero
- The score goes up as rocks are collected
- On restart the score should be zero again

How is this a challenge for the player?

We need to introduce something else to the game to make it
more challenging

What are we missing?

How is this a challenge for the player?

We need to introduce something else to the game to make it
more challenging

What are we missing?

A competitor for the rocks

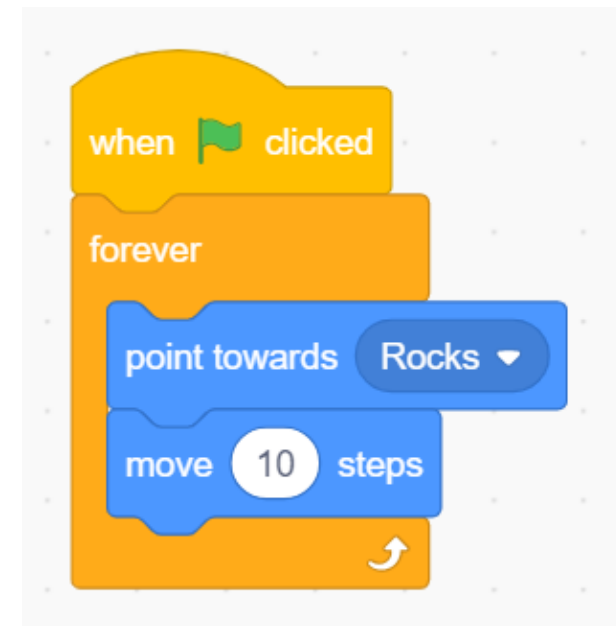
We need to introduce a rival computer player to try and steal the rocks first. It might be a competing robot, an alien, or something else entirely.

It is time to add a new sprite to the program, this time you can choose which one you wish to use as your game's 'enemy'

Waiting to move on? What would you like the computer player to do? What is the back-story?

Now you having chosen the look of your computer player character we need to instruct the program on how to move in the game. Do you want it to aim at the player or steal the rocks?

Let's first show you how to keep it moving towards the rocks. You'll need this basic movement script for the enemy sprite.



Waiting to move on? How would changing the number of steps alter the way the enemy sprite moves?

Now to test if the program works...

- The enemy sprite should move to the rock and then hover over it in a 'glitchy' way
- The robot can still pick up the rocks
- Nothing happens when the robot and enemy touch

What Are the Two Main Problems with the Computer Player?

What Are the Two Main Problems with the Computer Player?

1. Our new sprite does not stay the right way up
2. It glitches when it gets to the rocks

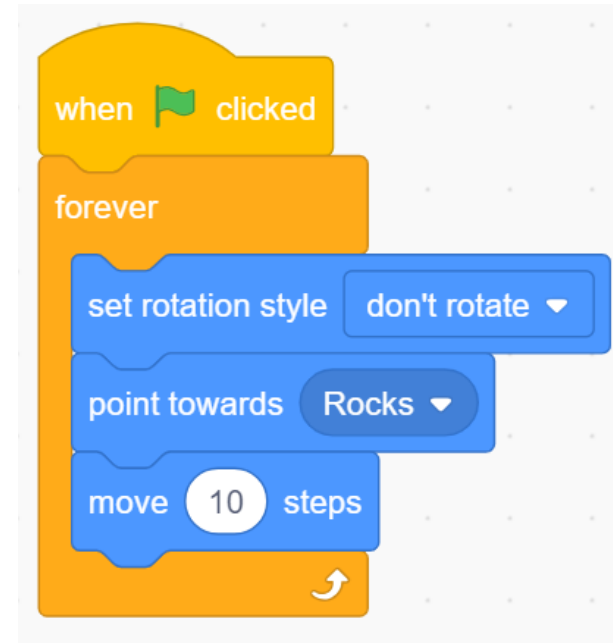
Rotation Issue

All we need to do to stop the random spinning of our computer player is add an instruction into the sprite's forever loop that says 'set rotation style don't rotate'

Waiting to move on? Try changing the number of movement steps to see what effect

Rotation Issue

All we need to do to stop the random spinning of our computer player is add an instruction into the sprite's forever loop that says 'set rotation style don't rotate'



Waiting to move on? Try changing the number of movement steps to see what effect

Stealing the Rocks

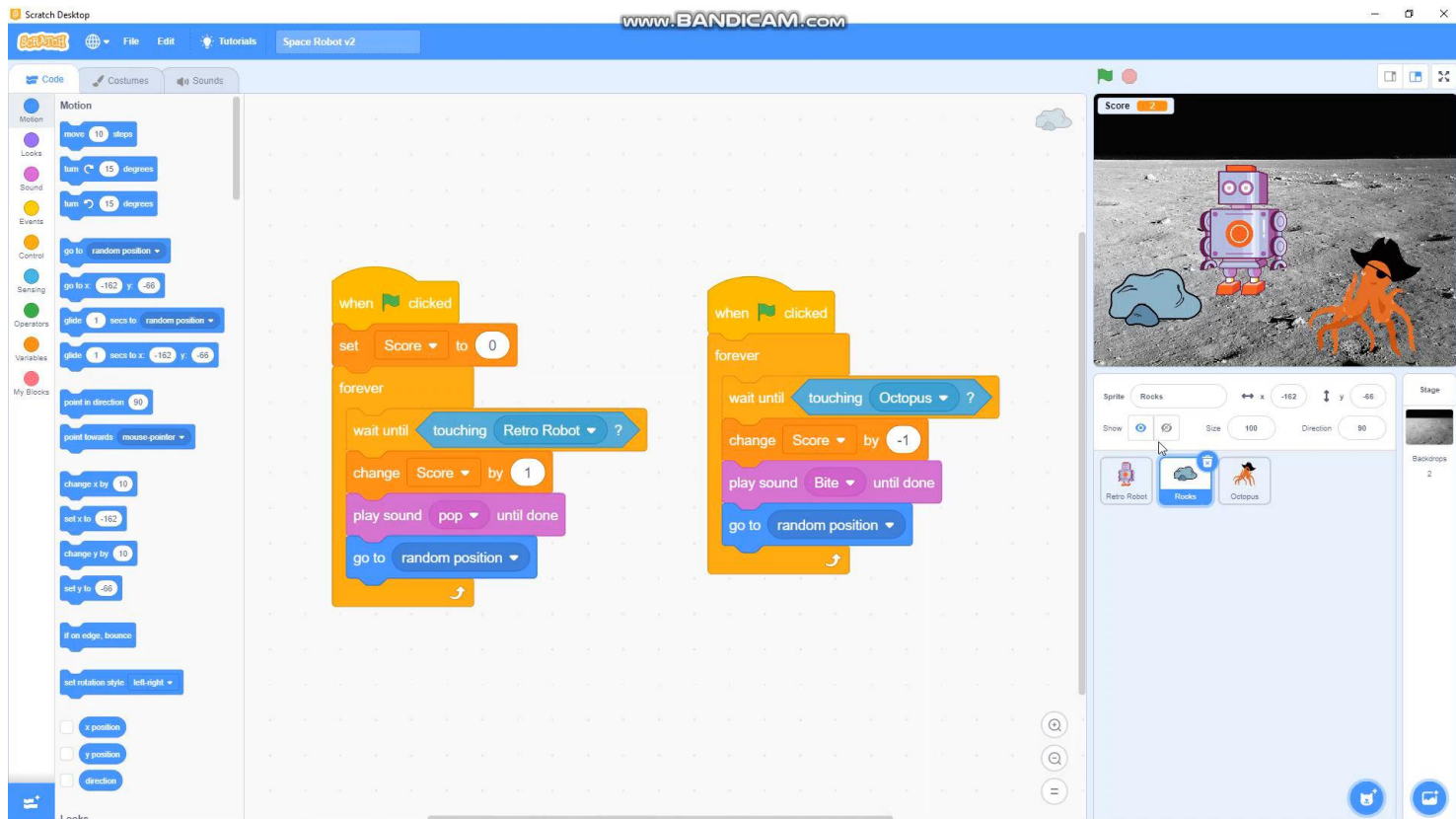
To stop the glitching behaviour, we need to have the computer player interact with the rocks.

Now is a good time to tell you that the script area for a sprite can have more than one script built in it.

By copying the script that is already in the rocks sprite Scripts Area and then changing what is needed for the computer player you should be able to make the computer player steal the rocks.

Waiting to move on? You can alter which sound effect is used for the computer player, and how it may affect the scoring

Now to test if the program works...



Game Over

At the moment the game continues until the player ends the program. Instead we need to set a point at which the game stops itself.

Let us do this by telling the game to stop if the player crashes into the computer player. Go into the retro robot sprite and add another script to the scripts area, start with:

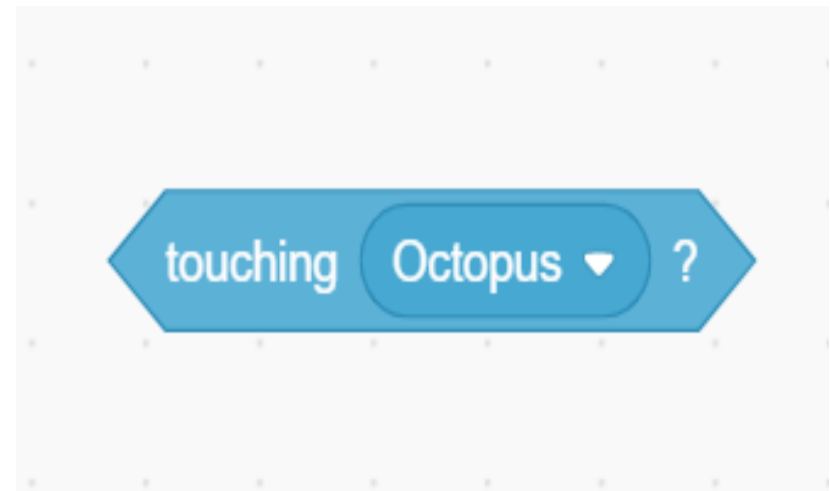


Which block do we need in the wait until command to finish this instruction?

Game Over

At the moment the game continues until the player ends the program. Instead we need to set a point at which the game stops itself.

Let us do this by telling the game to stop if the player crashes into the computer player. Go into the retro robot sprite and add another script to the scripts area, start with:



One More Thing

We need to make the players and rocks smaller to give more space for gameplay. To do this we need to add one block to each sprite's scripts telling it to change the size.

For example I have added the block into the Retro Robot script to shrink the player to half it's size. Can you do the same for the rocks and computer player as well?



Further Ideas

You have now managed to create a space game to challenge your friends to. However, this is just the beginning. Perhaps you want to add more sound effects? Use different sprites and backgrounds? Fine-tune the game to make it harder or easier? Add a computer player score counter? Have a start screen and game over screen?

It's your game to do with as you wish.

Enjoy!