

Code Rock Paper Scissors on a micro:bit with EduBlocks

Microbit

Beginner

10 Mins

1 Overview

In this project you will learn how to code your micro:bit to play rock, paper, scissors using gestures and the LED matrix. We will start by understanding the basic game logic and how to present rock, paper, and scissors on the LED matrix. You will discover how to use random selection for the micro:bit's choice and show it on the LED matrix, so players can see what the micro:bit picked and work out whether they have won. Through this project you will develop fundamental coding concepts like variables, conditionals and random numbers while creating something fun and interactive. By the end you will have a fully functional game that you can play with friends or challenge the micro:bit yourself.



What you'll learn

- How to import Python libraries
- How to use a while loop
- How to create and use a variable
- How to use the accelerometer
- How to display your own images on the LED matrix
- How to use if, elif and else statements
- How to use comparison operators
- How to randomise choices

2 What you'll need



What you'll need

- The micro:bit EduBlocks editor
- micro:bit
- micro USB cable
- battery pack for the micro:bit (optional)

3 Key words



Key words

Importing Libraries

A library is a ready-made bundle of code that someone else has written for you. Importing a library means bringing it into your project so you can use all its ready-made commands and features, without building them from scratch.

It's like borrowing a recipe book from the library. Instead of working out every recipe yourself, you bring the book in and use the recipes whenever you need them.

Variable

A variable is something that can change. Think of it like a labelled box: you pop something inside, and the label tells you what it is. You can swap what's inside whenever you like.

A variable called score might start at 0, then change to 1, 2 and 3 as you collect points in a game.

While loop

A while loop keeps repeating its instructions for as long as something stays true. The moment that thing stops being true, the loop stops.

"While it's still raining, keep the umbrella up." As soon as the rain stops, you put the umbrella down, and the loop ends.

If statement

An if statement checks whether something is true, and only runs its instructions when it is. It's how code makes decisions.

"If it's cold, put on a coat." If it's warm, you skip the coat, the action only happens when the check is true.

Accelerometer

An accelerometer is a tiny part inside the micro:bit that senses movement, like tilting, shaking or being turned over. It turns that movement into numbers your code can react to.

A bit like the sensor in a games controller that knows when you tilt it to steer, the accelerometer notices when the micro:bit moves.

LED Matrix

The LED matrix is the grid of little red lights on the front of the micro:bit. By switching them on and off in patterns, your code can show numbers, letters and tiny pictures.

Like a mini scoreboard made of dots, the LED matrix lights up to spell out your step count.

Comparison Operators

A comparison operator checks how two things measure up, for example whether they are equal or one is bigger than the other. It gives back a simple true or false answer.

Asking "is 6 == 6?" gives back true, but "is 6 == 2?" gives back false, so your code knows whether the two numbers match.

Random

Random means you can't guess what's coming next, a bit like rolling a dice. The computer picks for you, so you get a different surprise each time.

Asking a computer for a random number between 1 and 6 is just like rolling a dice, you never know if you'll get a 1 or a 6.

4 Set up your micro:bit EduBlocks Editor

1. Open your web browser. We recommend Google Chrome or Microsoft Edge.
2. In the address bar type app.edublocks.org.
3. Under **Create New Project**, select **micro:bit**. Name your project **Rock Paper Scissors**, and make sure **block** is selected under type.

+

×

Create Project

Project Name *

Rock Paper Scissors

Mode *

micro:bit

Type

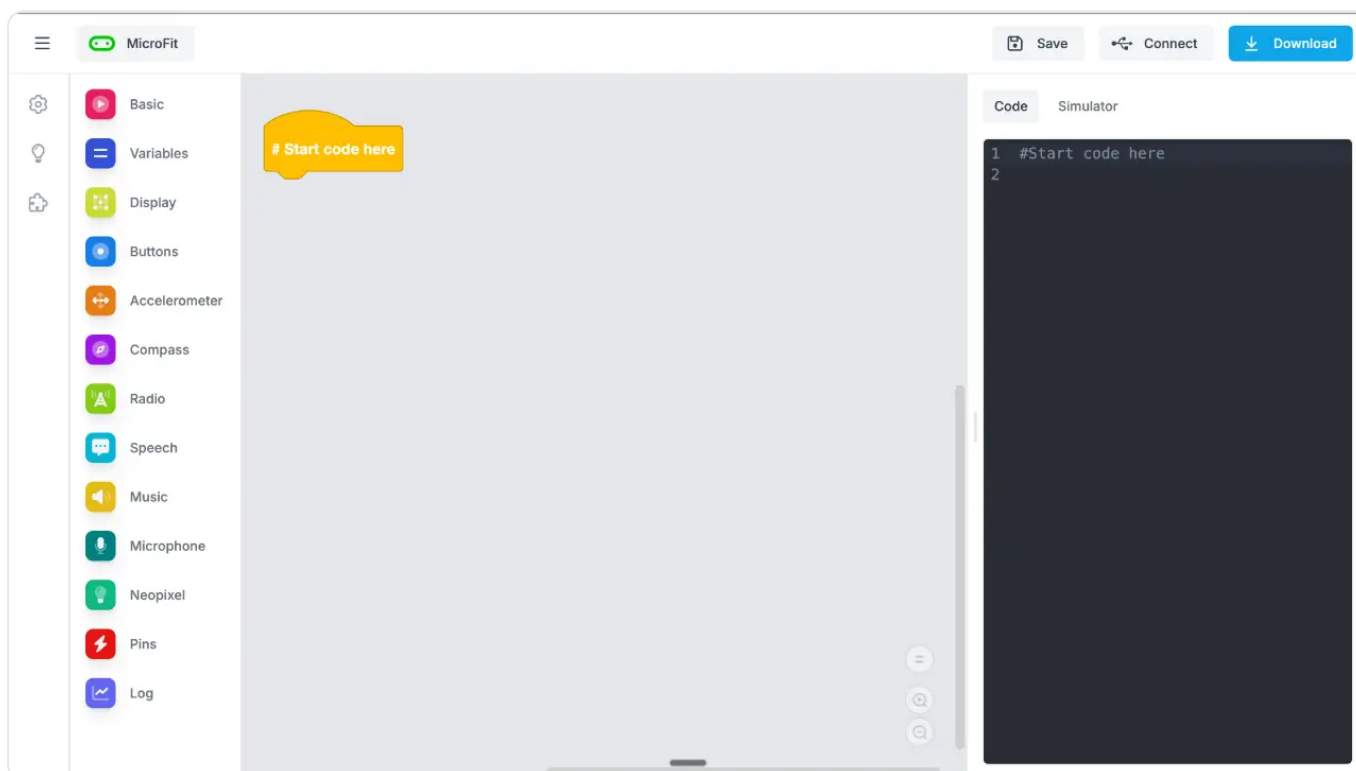
Blocks
Great for beginners

Text only
Ideal for advanced users

Cancel

Create

4. Select **Create** to open the micro:bit editor.



5 Importing MicroPython libraries

1. Select **Basic**, select **Imports**. Select `from microbit import *` and snap it to the `# Start code here` block.
2. Select **Basic**, select **Imports**, select an `import random` block and snap it to the `from microbit import *` block.

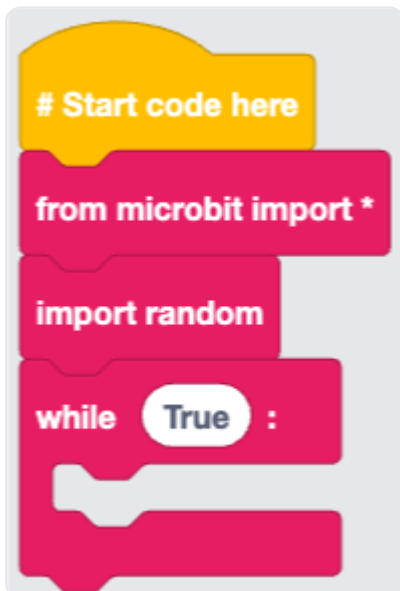
This imports the Python libraries that we need to create our game.



6 Create a While Loop

Select **Basic**, select **Loops**. Select a `while True:` block and snap it to the `import random` block.

This creates a loop that goes around and around forever.



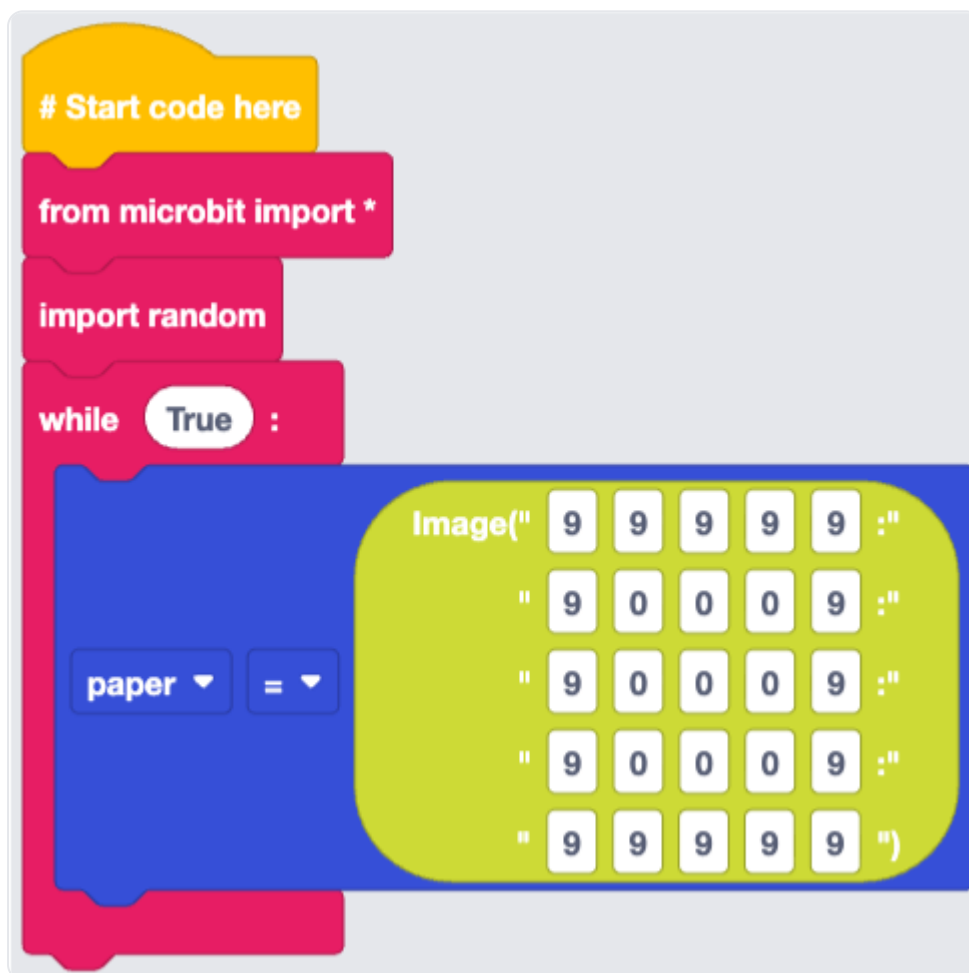
7 Create the Paper Variable

1. Select **Variables**. Select **Create Variable** and name it **paper**.
2. Select a `paper = 0` block and snap it within the `while True` block.

3. Select **Display**. Select an `Image` block and snap it onto the `paper =` block where it says **0**.
4. Create the following pattern within the `Image` block. A **9** means the LED is turned **on** and a **0** means the LED is **off**.



Here is the code so far:

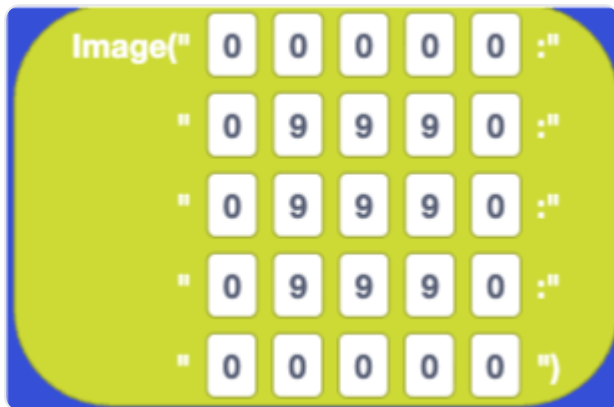


8 Create the Rock Variable

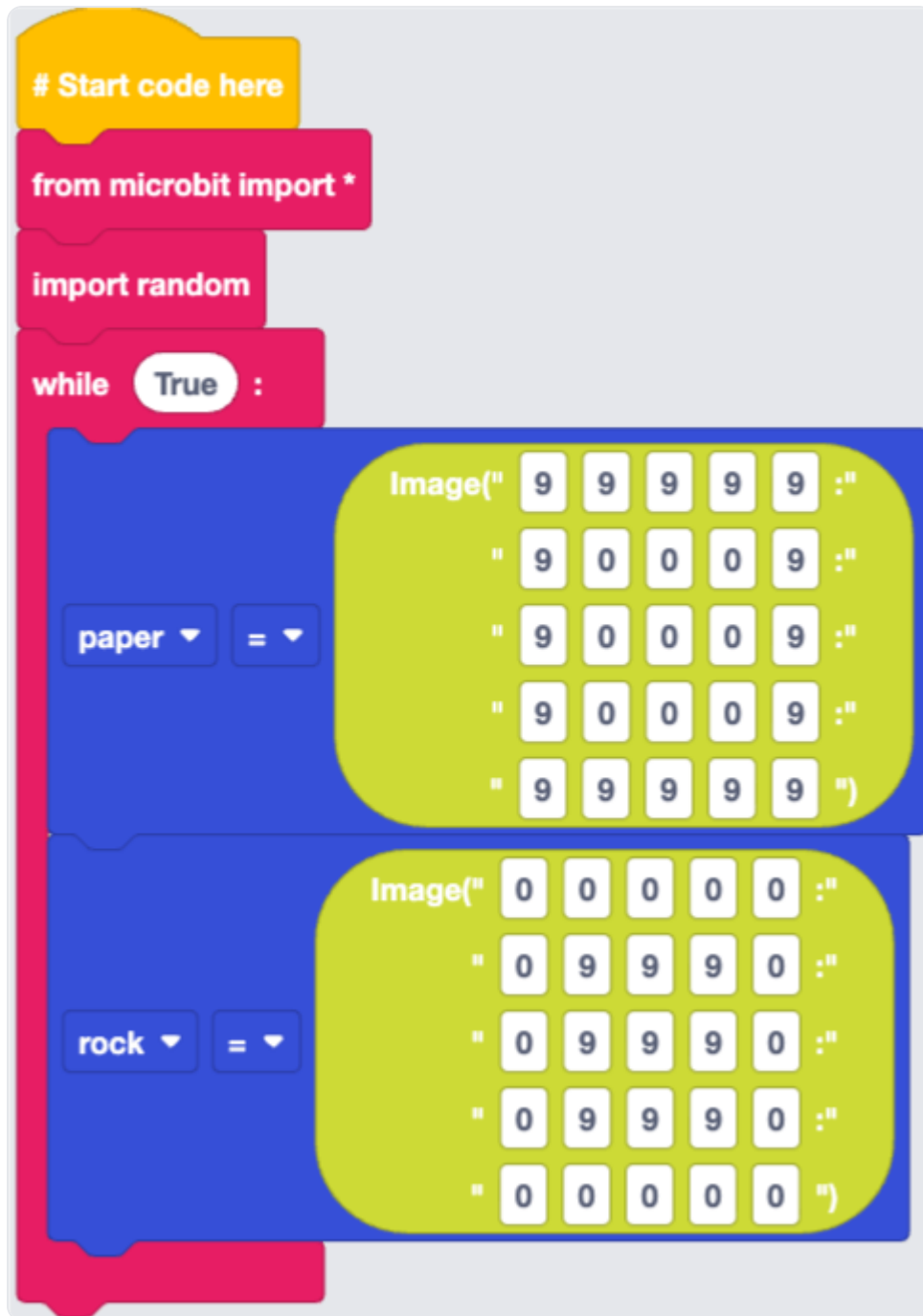
1. Select **Variables**. Select **Create variable** and name it **rock**.
2. Select a `rock = 0` block and snap it below the `paper = Image ("99999:", "90009:", "90009:", "90009:", "99999")` block.

3. Select **Display**. Select an `Image` block and snap it onto the `rock =` block where it says **0**.

4. Create the following pattern within the `Image` block.

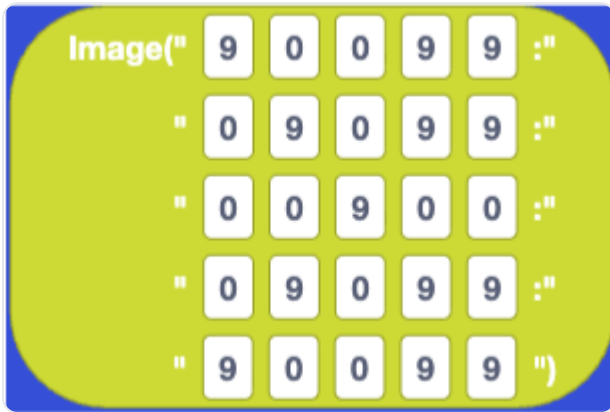


Completed code so far:



9 Create the Scissors Variable

1. Select **Variables**. Select **Create Variable** and name it **scissors**.
2. Select a `scissors = 0` block and snap it below the `rock = Image ("00000:", "09990:", "09990:", "09990:", "00000")` block.
3. Select **Display**. Select an `Image` block and snap it onto the `scissors =` block where it says **0**.
4. Create the following pattern within the `Image` block.



Completed code so far with the scissors variable added.

Start code here

from microbit import *

import random

while True :

paper ▾

= ▾

```
Image(" 9 9 9 9 9 :"  
      " 9 0 0 0 9 :"  
      " 9 0 0 0 9 :"  
      " 9 0 0 0 9 :"  
      " 9 9 9 9 9 ")
```

rock ▾

= ▾

```
Image(" 0 0 0 0 0 :"  
      " 0 9 9 9 0 :"  
      " 0 9 9 9 0 :"  
      " 0 9 9 9 0 :"  
      " 0 0 0 0 0 ")
```

scissors ▾

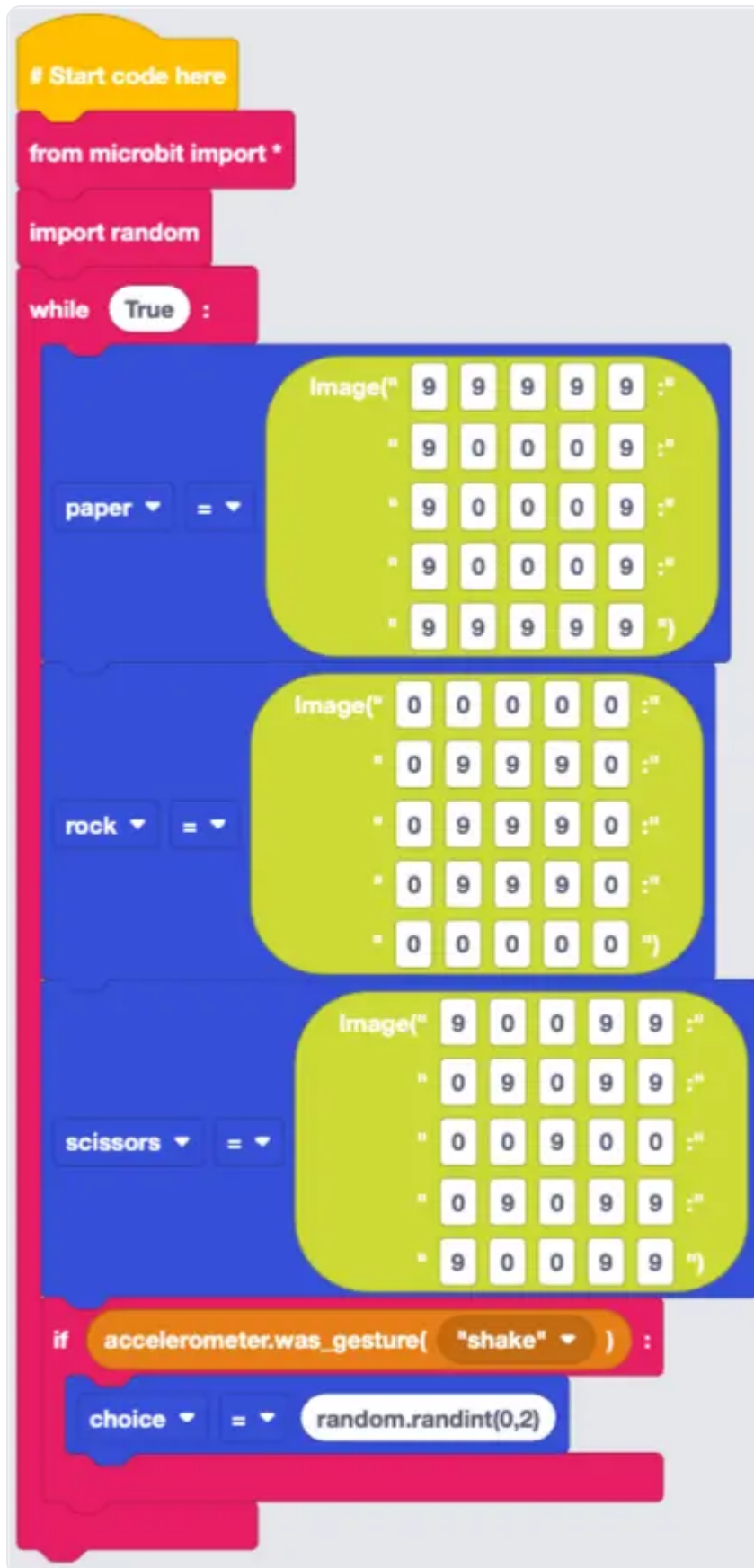
= ▾

```
Image(" 9 0 0 9 9 :"  
      " 0 9 0 9 9 :"  
      " 0 0 9 0 0 :"  
      " 0 9 0 9 9 :"  
      " 9 0 0 9 9 ")
```

10 Check for a Shake

1. Select **Basic**, select **Logic**, select an `if True:` block and snap it to the `scissors = Image ("90099:", "09099:", "00900:", "09099:", "90099")` block.
2. Select **Accelerometer**. Select an `accelerometer.was_gesture ("shake")` block and snap it onto the `if` block where it says **True**.
3. Select **Variables**, select **Create Variable** and name it **choice**. Select a `choice = 0` block and snap it inside the `if accelerometer.was_gesture ("shake"):` block.
4. Within the **0** of the `choice` block type **random.randint(0, 2)**

This will check to see if the micro:bit has been shaken and if it had it will set choice to a random number between 0 and 2.

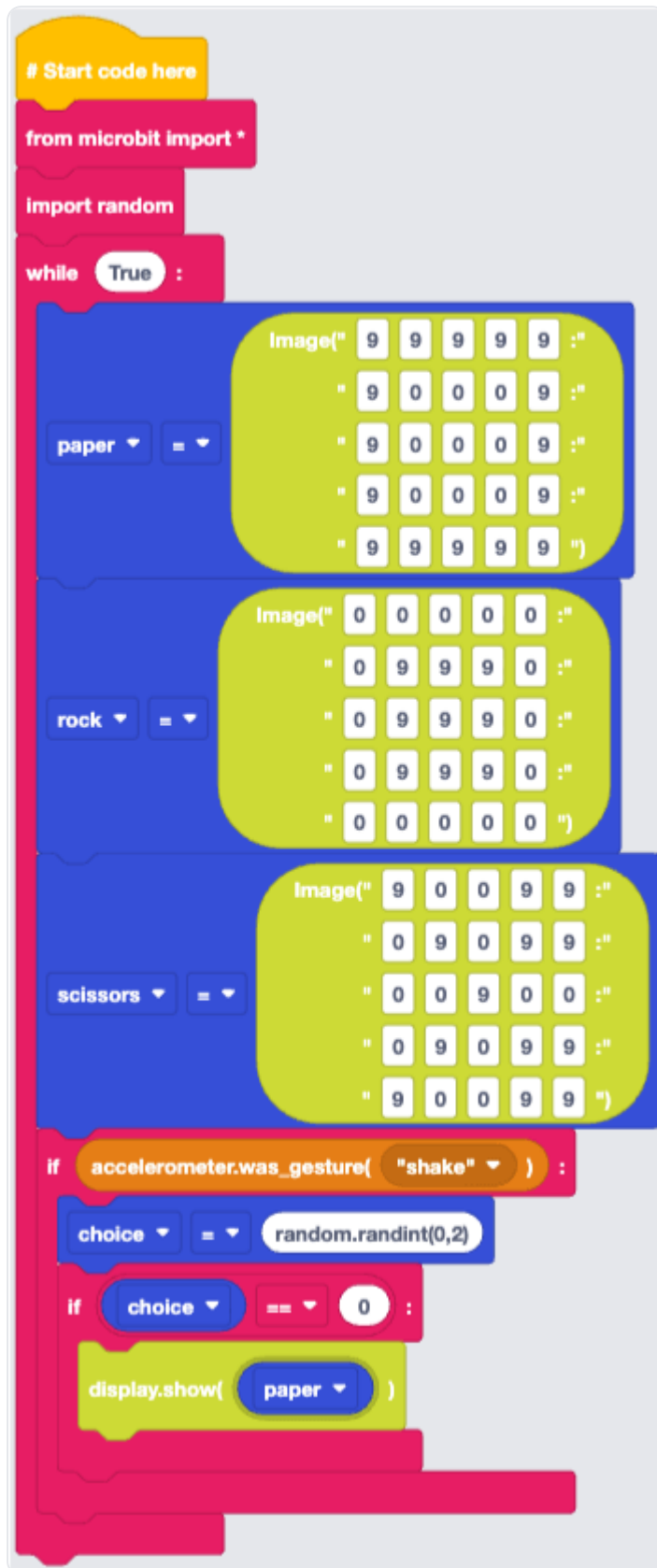


11 Create the If Condition

1. Select **Basic**, select **Logic**, select an `if True:` block and snap it to the `choice = random.randint(0, 2)` block.

2. Select **Basic**, select **Logic**, select a `0 == 0` block and snap it on to the `if` block where it says **True**.
3. Select **Variables**. Select a `choice` block and snap it onto the `if` block before the `==` sign.
4. Select **Display**. Select a `display.show(10)` block and snap it inside the `if choice == 0:` block.
5. Select **Variables**. Select a `paper` block and snap it onto the `display.show` block where it says **10**.

This will check to see if choice is equal to 0 and if it is paper will be displayed on the LED matrix.

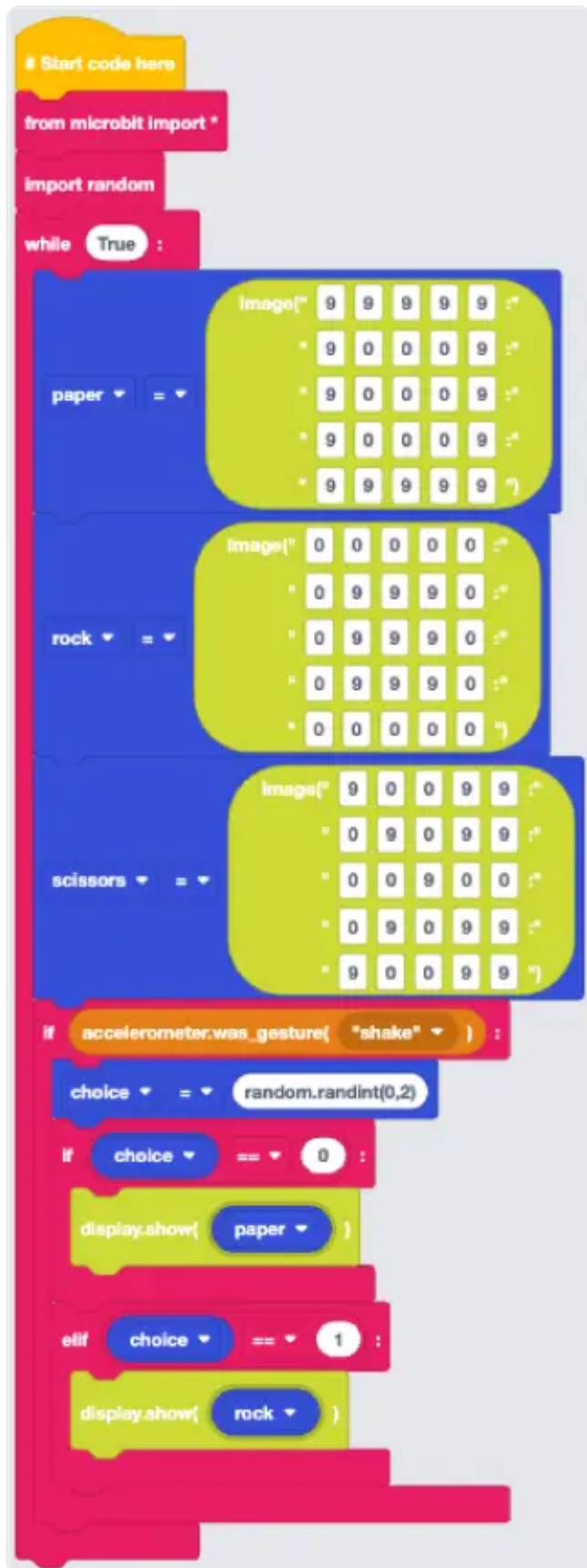


12 Create the elif Condition

1. Select **Basic**, select **Logic**. Select an `elif True:` block and snap it to the `if choice == 0:` block.

2. Select **Basic**, select **Logic**. Select a `0 == 0` block and snap it onto the `elif` block where it says **True**. Change the **0** to **1**.
3. Select **Display**. Select a `display.show(10)` block and snap it into the `elif choice == 1:` block.
4. Select **Variables**. Select a `rock` block and snap it onto the `display.show` block where it says **10**.

This will check to see if choice is equal to 1 and if it is rock will be displayed on the LED matrix.



13 Create the Else Condition

1. Select **Basic**, select **Logic**. Select an `else:` block and snap it to the `elif choice == 1:` block.

2. Select **Display**. Select a `display.show(10)` block and snap it inside the `else:` block.
3. Select **Variables**. Select a `scissors` block and snap it onto the `display.show` block where it says **10**.

This will display scissors on the LED matrix if choice does not equal 0 or 1.



14 Downloading the Code

1. Connect the micro:bit to the computer with the micro USB cable.

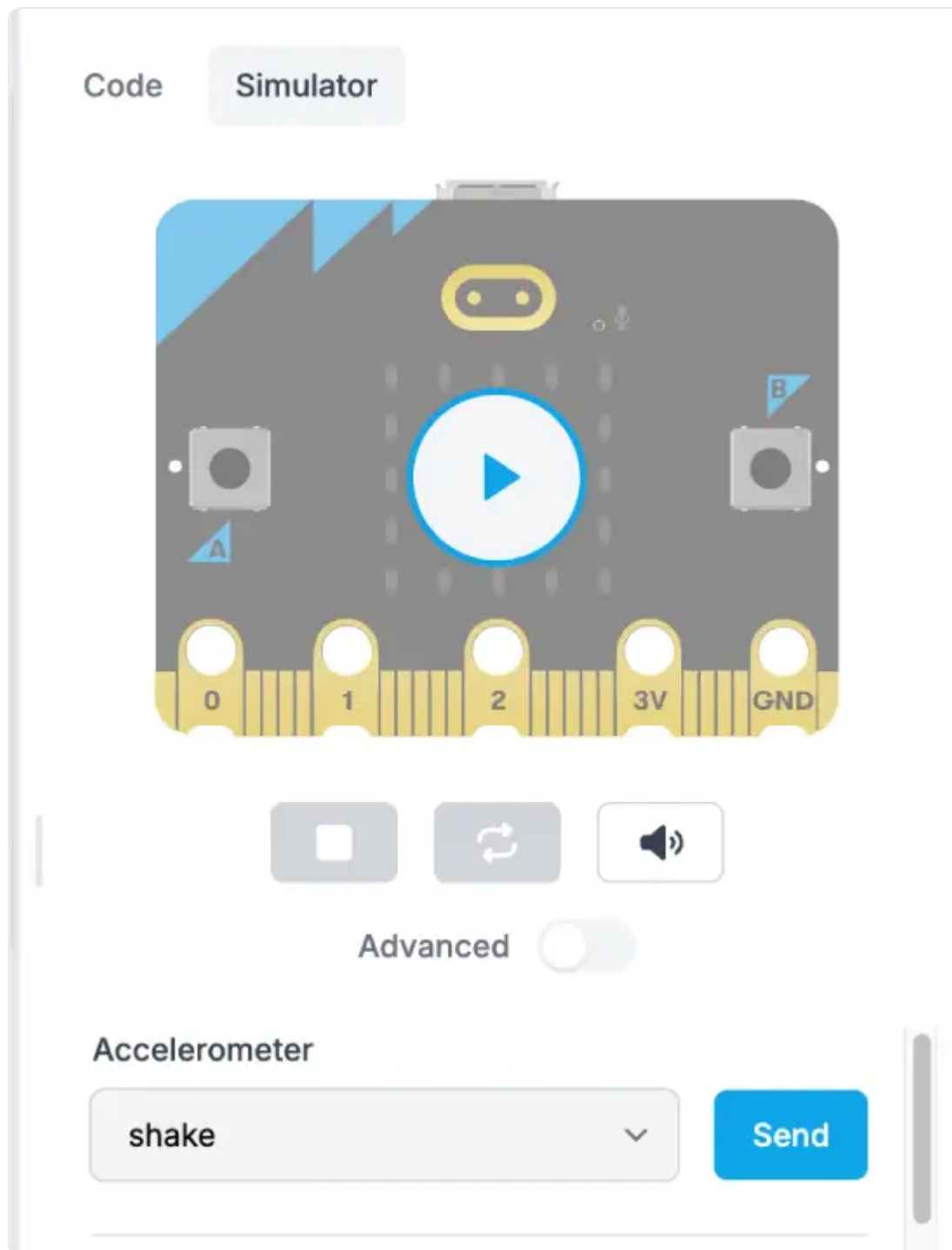
2. Select **Connect** and follow the on-screen prompts to pair the micro:bit with your web browser.
3. Select **Download** to send the code to your micro:bit.

15 Run it and Watch

In the web browser

EduBlocks has a built-in micro:bit simulator, handy if you haven't got a micro:bit to hand.

1. On the right-hand side of the screen, select **Simulator**.
2. From the accelerometer menu below the simulator, select **Shake**.
3. Select **Send**. This simulates a shake of the micro:bit and you will see a rock, paper, or scissors icon display on the LED matrix.



On a real micro:bit

Once the code has downloaded, give the micro:bit a shake and see an icon display on the LED matrix. Why not find a friend and see who can beat the micro:bit?

16 Try it yourself



Try it yourself

Challenge: Can you add a sound so your micro:bit gives a little beep each time it makes a choice? Try snapping an `audio.play(Sound.HAPPY)` block from **Audio** inside your `if accelerometer.was_gesture("shake"):` check, just after you set the choice. For an extra challenge, pick a different sound for rock, paper and scissors!

Good to know: the sound plays through the built-in speaker, which only the micro:bit V2 has. If you have a V1, you can still try this challenge by clipping a buzzer or headphones to the pins. Everything else in this project works happily on both V1 and V2.

17 Stuck? Quick fixes

Problem	Try this
Nothing happens when I shake it	Give it a firm shake, not too gentle. Also check your blocks are snapped inside the <code>if accelerometer.was_gesture("shake"):</code> block.
The same icon keeps showing	That's normal! The choice is random, so shake a few more times and the others will appear.
My icons look wrong or blank	Open each Image block and check you've selected the right squares for rock, paper and scissors.
It only ever shows two icons	Check your <code>elif</code> is set to <code>choice == 1</code> , and that <code>random.randint</code> goes from 0 to 2.
It won't download to my micro:bit	Make sure the micro:bit is plugged in with the USB cable, then try Connect again.