



Rock, paper, scissors

Micro:bit introduction lesson plan – Code Playground



Lesson overview

In this lesson, students will build a 'Rock, paper, scissors' game using the Micro:bit emulator. They'll explore the concepts of randomness, inputs (pressing button) and outputs (LED display). Students will explore how random numbers can create game outcomes, and see how block code converts into JavaScript, bridging the gap between beginner-friendly and text-based coding.

Time	Key learning outcomes	Resources
60 mins	<ul style="list-style-type: none">Understand how to use the Micro:bit emulatorCreate a 'Rock, paper, scissors' game in MakeCode using blocksExplore selection and how randomness can create fair outcomes in gamesRecognise how block-based code can translate into JavaScript text-based coding.	<ul style="list-style-type: none">Laptops or desktop computersAccess to Micro:bit website https://microbit.org/

Content

Activities	Time	Page
Introduction	10 mins	3
Activity – Rock, paper, scissors	50 mins	4
Code snippets	-	5
Summary	-	6

Introduction

Begin with a discussion of the classic game 'Rock, paper, scissors', by playing a round together. This could be as a whole class versus the teacher or in pairs with a partner. Explain that students will explore the concept of selection in coding, understanding how a computer makes decisions between different options, with outcomes determined by random choices made by the computer.

Link this to making choices – “When you choose what to wear in the morning, how do you decide between a jumper and a T-shirt?”. Highlight that computers also need ways to make choices, and that’s what selection in code means.

Class discussion on the rules of the game:

“What happens if two people choose different options? How do we know who wins?”

Explain that the emulator will detect a button being pressed then will randomly pick rock, paper or scissors and display the choice with an icon.

How to tie these concepts to a real-world example: “Rolling a dice or spinning a wheel – the outcome depends on a random choice. If the music shuffle is on, the music app randomly selects the next song to play, just like the Micro:bit will do in our game.”

Micro:bit practical

Ask the children to log into Micro:bit and set up their workspace as described in the workbook. Show the Micro:bit introduction – rock, paper, scissors video as a guided lesson pausing regularly when the students need to catch up.

- Children should be able to follow along with the workbook or the guided lesson video
- By the end of the lesson children should be able to create a working program showcasing the Micro:bit introduction - rock, paper, scissors project.

Activity – Rock, paper, scissors

Reinforce that the project will use selection which allows the computer to make a choice:

- If the random number chosen is 1, it displays rock
- If the random number chosen is 2, it displays paper
- If the random number chosen is 3, it displays scissors.

Class question: “If the Micro:bit didn't use selection and only had one outcome; would it still be a fair game? Why or why not?”

Micro:bit practical

Using the video and workbook support the children to follow the instructions and complete the coding project. Have them think of other ways to enhance the project if they have extra time.

Activity wrap up

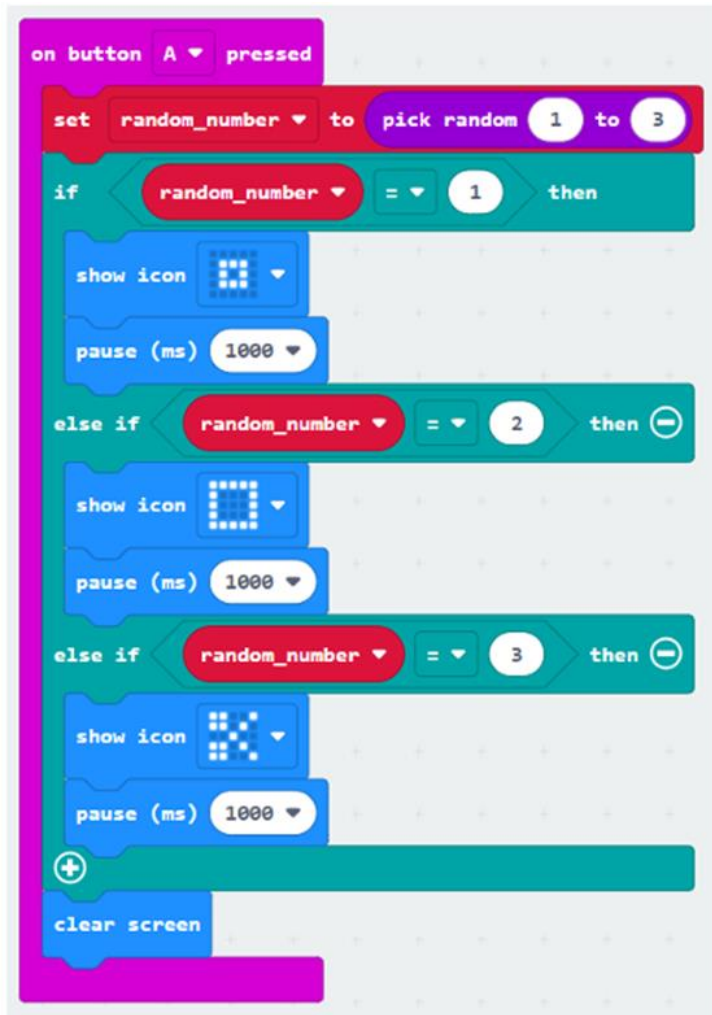
Prepare to share your project with the class:

- “How did using variables help the Micro:bit decide between rock, paper, scissors?”
- “If you wanted to add a fourth option, like ghost or heart, what changes would you need to make to your code?”

Encourage customisation

- Can pupils add sound effects to their project?
- Explore custom LED patterns for rock, paper and scissors instead of icons
- Could string messages be added like “You chose rock”.

Code snippets



Summary

The following information is an example of what a child at an expected level would be able to demonstrate when completing these activities with additional examples to demonstrate how this would vary for a child with emerging or exceeding achievements.

Assessment guidance

Differentiation – Lower ability/ASN

- Focus on programming just one output
- Use visual prompts so learners can match the random number to the right icon more easily
- Pair students for additional support.

Differentiation – Higher ability/extension

- Challenge students to add more variables for a scoring system
- Add LED animations instead of static icons
- Explore the JavaScript in MakeCode and challenge students to edit a small part.

Plenary

- Where else might randomness be useful in computer programs? (password generation, music shuffle)
- Can you name any other games or applications where selection is important?
- How does the Micro:bit know whether to show rock, paper, or scissors?

Assessment questions

1. Which block makes the game fair?
2. How could you personalise the game?
3. Which block makes the output visible?
4. Why might it be useful to see the JavaScript version of your code?