Rock, paper, scissors

Scratch lesson plan – Code Playground

Lesson overview

In this lesson, students will create a Scratch project based on the classic game rock, paper, scissors. They will learn to use selection and variables to program game logic that lets players choose an option while the computer randomly selects one. This project is part three of our story challenge, it is also a more advanced version of rock, paper, scissors than previously coded.

Time	Key learning outcomes	Resources
45 mins	- Understand and apply selection blocks to create conditional logic - Use variables to store and use random values - Programmatically switch costumes to represent different outcomes - Develop problem-solving skills by debugging and refining game mechanics.	- Laptops or desktop computers - Access to Scratch website - https://scratch.mit.edu.

Content

Activities	Time	Page
Activity introduction	15 mins	3
Activity – Rock, paper, scissors	30 mins	4
Code snippets	-	5
Summary	-	6

Introduction

Look and feel is the third part of our story challenge series. It focuses on using selection and variables to create animations.

> "Through the five parts of the story challenge you will use a variety of different skills that you can use to build up to create a longer project."

This is a more advanced version of an activity from the beginning of the year. It includes more of the skills that have been picked up on other projects.
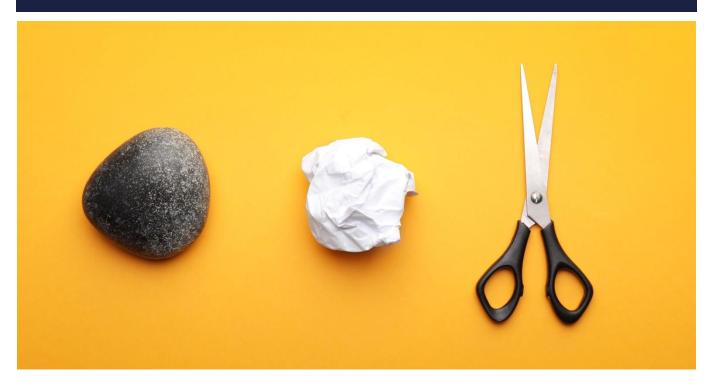
> Class discussion
>
> "What do you think are the rules we need to code to make this game work?"

Scratch practical

Ask the children to log into Scratch and set up their workspace as described in the workbook. Show the rock, paper, scissors video as a guided lesson pausing regularly when the pupils 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 rock, paper, scissors activity.

Activity – Rock, paper, scissors

"Today, we'll program the game Rock, Paper, Scissors in Scratch. You'll learn to use variables and selection blocks to create a fully functional game."

Highlight the coding concepts:

"We'll store the computer's choice as a variable and use selection blocks to decide who wins. This will teach you how to create rules and conditions in a program."

As part of this project pupils will use refactoring to make a piece of code easier to understand and work with.

Explain refactoring:

- Show how the code can be simplified by replacing multiple if / else blocks with more efficient logic.

Scratch practical

Using the video and workbooks 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 does refactoring improve your code? What makes it easier to read and debug?"

Encourage customisation

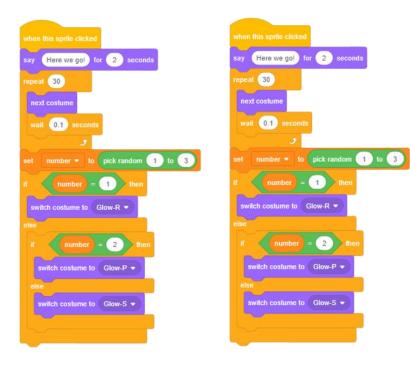Add extra options like lizard or Spock to expand the game.

Use the video sensing extension to allow players to make choices by moving their hands.
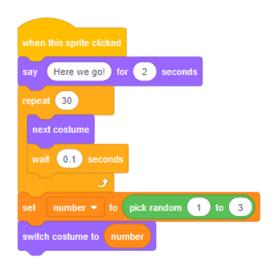
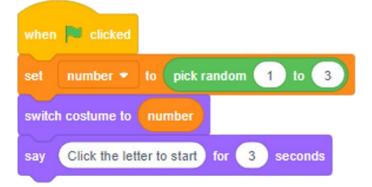Add a scoreboard to track wins, losses, and ties over multiple rounds.

Code snippets

Sprite code

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	Differentiation – Higher ability/extension
• Focus on creating the sprites and programming one basic outcome (e.g. a draw) • Provide starter code to help with random selection and player choice • Pair students for additional support.	• Challenge students to expand the game with more options and outcomes • Encourage them to optimise the code further using custom blocks or variables • Explore adding animations to make the game more interactive.

Plenary

• "What blocks did you use to create the game's logic?"

• "What did you learn about using variables and selection blocks?"

• "How would you improve your game if you had more time?"

Assessment questions

• How did you program the computer to choose an option randomly?

• What blocks did you use to determine the game's outcome?

• How does simplifying your code help when building larger projects?

• What feature would you add to improve the game?