Rock, Paper, Scissors, Well, Bull! (RPS+X)

Year level band: 7-8

Description: Using the concept of Rock, Paper, Scissors, Well, Bull! students will design an interactive board game with Micro:bits for a target audience.

Resources: Micro:bit Go x 5, computers
Micropython or https://codewith.mu/
Trello.com
Paper, Pen

Prior Student Learning: An understanding of Micro python or visual programming for the Micro: bit would be beneficial, or allow time for students to gain this knowledge throughout the unit of work. An understanding of variables as prior knowledge is required.

Australian Curriculum alignment summary

This lesson provides opportunity for students to learn to code a Microbit to play Rock, Paper, Scissors plus add on additional elements of Well and Bull. Extending this lesson to develop code to include additional features of dice and head/tails and design a board game to incorporate all of these elements.

| Year	Content Descriptors
Year 7-8	● Design the user experience of a digital system, generating, evaluating and communicating alternative designs (ACTDIP028)
	● Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors (ACTDIP029)
	● Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language (ACTDIP030)
	● Plan and manage projects that create and communicate ideas and information collaboratively online, taking safety and social contexts into account (ACTDIP032)

Element	Summary of tasks
Learning hook	In an increasingly digital world, board games are growing in popularity for players of all ages. You have been commissioned to produce a new and engaging game based on the traditional Rock, Paper, Scissors but with additional complexity to include Well, Bull/Spock, Lizard or entirely new elements. Your board game will also require at least one additional function such as dice, head/tails. You will choose your Target audience, design and produce the board game together with a sales pitch to promote your product.
Achievement Standards Learning Map (Sequence)	By the end of Year 8, students plan and manage digital projects to create interactive information. They define and decompose problems in terms of functional requirements and constraints. Students design user experiences and algorithms incorporating branching and iterations, and test, modify and implement digital solutions. They evaluate information systems and their solutions in terms of meeting needs, innovation and sustainability
	In this task students will work collaboratively as a team to identify elements of game design and apply that knowledge to the creation of an engaging board game using Rock, Paper, Scissors as the starting point. Online collaboration and project management requirements will allow students to work together on a shared platform such as google docs, Office 365 or Trello. They will develop key programming skills using Micropython and Micro:bits.
Learning input	Explore Micro: bit features, storage and safety instructions. Model and share how to create the pseudocode with the class. Plan your program using pseudocode and/or flowchart Common pseudocode notations INPUT - user inputting something OUTPUT - displayed on screen, sound etc WHILE - loop that has a condition at beginning FOR - counting loop REPEAT- UNTIL - loop that has a condition at the end IF-THEN-ELSE - decision to be made https://www.bbc.com/education/guides/z3bq7ty/revision/2 Basic Rock, Paper, Scissors Create variable for RPS = elements On shake Pick random number between 0&2 (3 options of 0,1 & 2) If 0 show image representing paper or show string 'P' Else if 1 show image representing rock or 'R' Else show image representing scissors or 'S' Add scoring Press button a to increase score by 1

	And show string 'WIN' Press button b to show score *Include additional elements to increase odds of probability *Add sound capability for accessibility

Learning construction	Provide student with sample code to create custom images on the device and ask them to add comments to demonstrate understanding.
	
	The following code for the project is below.
	
	``python
	from microbit import*
	While True:
	    if button_a.is_pressed() and button_b.is_pressed():
	        display.show(Image.CONFUSED)
	    elif button_a.is_pressed():
	        display.show(Image.HAPPY)
	    elif button_b.is_pressed():
	        display.show(IMAGE.SILLY)
	    else:
	        display.show(IMAGE.DUCK)
	``
	
	DIY Images – Each LED pixel (5X5) on the device can be controlled using numerals between 0and 9. – Create own Image.
	
	Animation: Fade the Monster out by creating six images in the same way as your previous DIY Images task.
	
	``python
	monster1=Image("09090:""99999:""90009:""09990:""99099:")
	monster2=Image("05050:""55555:""50005:""05550:""55055:")
	monster3=Image("02020:""22222:""20002:""02220:""22022:")
	
	all_monsters=[monster1, monster2, monster3]
	``
	
	*This will only fade the monster out to a brightness of 2. *Can you add in the other lines of code to make it fade slowly from 9 to 0?*
	
	Students decide on images for their RPS+?, create and test with peers to determine successful recognition. Make changes based on feedback. Use the RPS guide below to plan out images.

Learning demo	Provide students with basic RPS code.
	
	from microbit import*
	import random
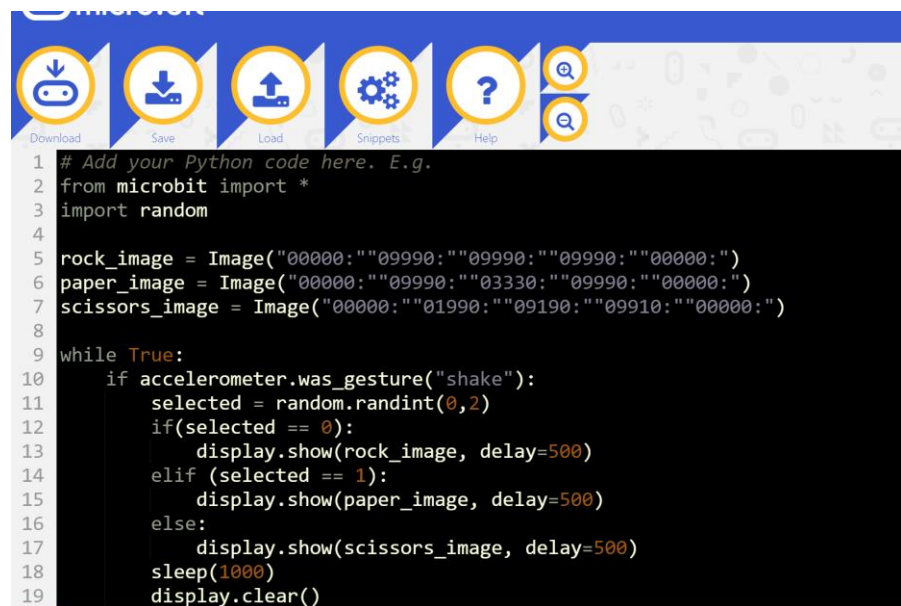	
	#define your custom rock, paper, scissors etc... images
	
	#create a list containing your images
	
	While True:
	   if accelerometer.was_gesture("shake")
	     #display a random image from list
	
	** This is very basic code and is a sample code of have students comment code.
	
	Alternative code with user defined images for RPS elements


```python
1  # Add your Python code here. E.g.
2  from microbit import *
3  import random
4
5  rock_image = Image("00000:""09990:""09990:""09990:""00000:")
6  paper_image = Image("00000:""09990:""03330:""09990:""00000:")
7  scissors_image = Image("00000:""01990:""09190:""09910:""00000:")
8
9  while True:
10     if accelerometer.was_gesture("shake"):
11         selected = random.randint(0,2)
12         if(selected == 0):
13             display.show(rock_image, delay=500)
14         elif (selected == 1):
15             display.show(paper_image, delay=500)
16         else:
17             display.show(scissors_image, delay=500)
18         sleep(1000)
19         display.clear()
```

CODE EXAMPLE:

Code required:
#Rock, Paper, Scissor
#shake the microbitand it will pick a shape at random
from microbit import *
import random
#Images of R, P and S
rock = Image("00000:"
     "09990:"
     "90009:"
     "92229:"
     "29992")
paper = Image("99950:"
     "90992:"
     "90092:"
     "90092:"

CSER Group, The University of Adelaide

```
          "99992")
 scissors = Image("90009:"
         "09090:"
         "00900:"
         "55055:"
         "55055")

rock_paper_scissors=[rock,paper,scissors]

display.show(rock_paper_scissors, delay=500)
display.clear()
```

Test this code and record results.

Feedback Questions:
- Do the results demonstrate correct statistics?
- How do you know that it is a new image displaying if the same element shows more than once in a row?
- How can you add conditional statement to improve the code?
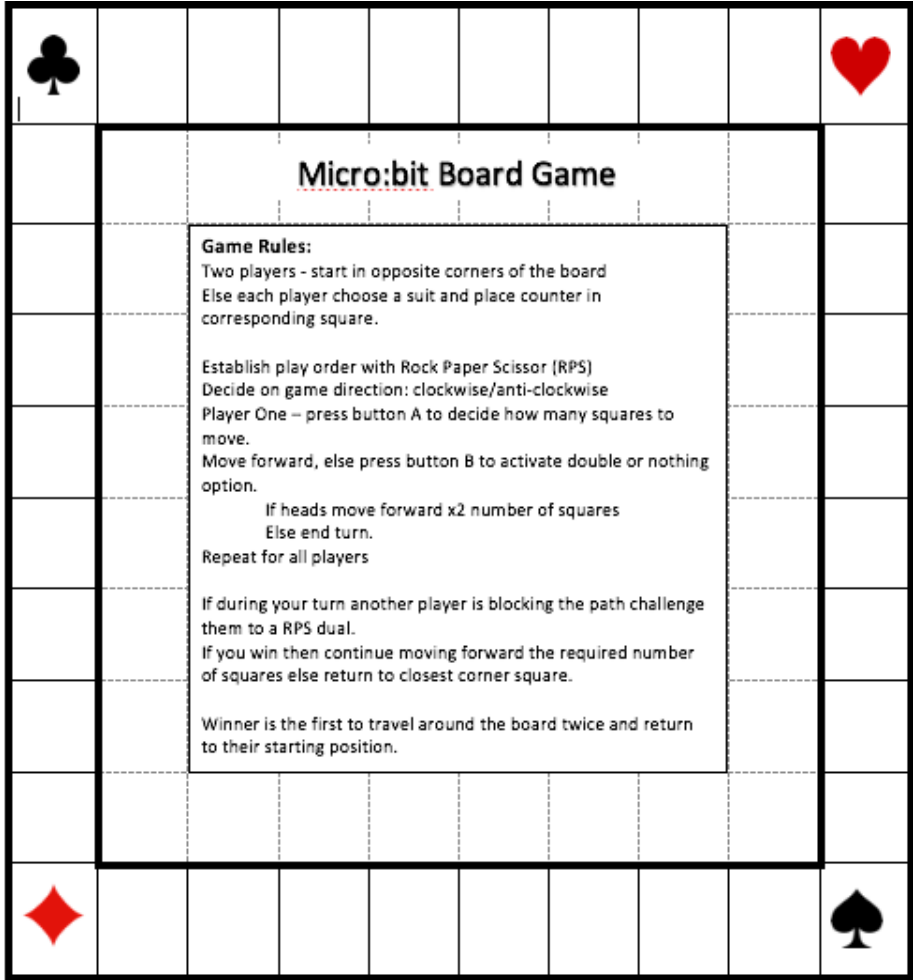
Students to design a board game for a target audience of their choice and to complement RPS+X. What other elements are required for a board game? Eg dice, counters etc. Have them examine a range of other games to identify common features, rules and gameplay.

RPS code can be adapted to include dice and head/tails - note that a different user input will need to be determined.

Extension: Students can design a casing for the Micro:bit to ensure durability and include additional design elements.

Example Gameboard

	Micro:bit Board Game
	
	Game Rules:
	Two players - start in opposite corners of the board
	Else each player choose a suit and place counter in corresponding square.
	
	Establish play order with Rock Paper Scissor (RPS)
	Decide on game direction: clockwise/anti-clockwise
	Player One – press button A to decide how many squares to move.
	Move forward, else press button B to activate double or nothing option.
	    If heads move forward x2 number of squares
	    Else end turn.
	Repeat for all players
	
	If during your turn another player is blocking the path challenge them to a RPS dual.
	If you win then continue moving forward the required number of squares else return to closest corner square.
	
	Winner is the first to travel around the board twice and return to their starting position.

Learning reflection	Students present their game pitches to the class, or to a small group. Invite them to explain how they designed and implemented the game, but also the following (either as part of their presentation or a reflection piece of writing): ● Is the code robust and reflective of random probability? ● Have you met the needs of your target audience for the Board Game? Is it engaging? How can you demonstrate or support this? ● Was the response to your sales pitch positive? How did you measure this?

Assessment:

Please find some suggested assessment approaches below.

Formative:

- Students provide early feedback on initial game designs, and use feedback to improve designs and inform the final game design and implementation.
- Collect student design artefacts and provide feedback on initial ideas.
- Observe students as they work on implementing the game and in designing their own - are they demonstrating particular skills and can they articulate knowledge from the content descriptors/achievement standards?

Summative:

- Students present their final game designs and the finished projects as a presentation to the class or as a recorded video.
- Self-assessment: students reflect on their own learning progress during the projects. Invite them to write a reflection or to orally explain what they have learned and what they would like to continue to work on in the future.
- Students play one another's games and peer-assess using a rubric or checklist (can be co-designed with students around good game design characteristics or defined by the teacher).

	Quantity of knowledge			Quality of understanding	
Criteria	**Pre-structural**	**Uni-structural**	**Multi-structural**	**Relational**	**Extended abstract**
User interface	No evidence supplied to support planning of board game	Little evidence supplied to support planning and/or evaluation of board game to address identified audience needs.	Evidence supplied to support planning and evaluation of target audience needs eg: storyboard, mockups	Evidence supplied to demonstrate an iterative approach to planning and evaluation of board game as a result of testing prototypes. Target audience needs identified and addressed.	Evidence supplied to demonstrate an iterative approach to planning and evaluation of board game as a result of testing prototypes. Additional inclusions for accessibility functionality
Algorithm	Code is basic and/or does not work. No comments	Code is basic or copied from teacher supplied	Code has been adapted to include students own images	Code is complex, evident of being independently developed program with all	Code demonstrates independent learning and includes complex parallel

CSER Group, The University of Adelaide

	included No flowchart or pseudocode	with minimal changes. Some comments included Basic flowchart or pseudocode supplied.	for game elements. Comments included for all code. Flowchart or pseudocode provided to support student understanding of algorithm	the code functioning correctly. All code is commented effectively. Flowchart or pseudocode reflects the completed code and alterations or changes have been clearly noted.	programs that are interrelated and extend beyond the brief. Comments included for all code to reflect understanding Flowchart and/or pseudocode includes comments/chang reflective of iterative approach or evaluation processes
Project Management	No evidence of collaboration with team members	Some evidence of collaboration with team members	Evidence provided to demonstrate allocation of team roles and contribution to project	Evidence provided to support collaboration with team members, organisation of team roles and responsibilities including sequencing of tasks and system to identify when completed.	Documentation supplied to support collaboration and leadership in team. Clear outline of roles and responsibilities, timeline and recording when tasks completed.

Teacher/Student Instructions:

MicroPython guide to BBC micro:bit https://microbit.org/guide/python/ This link provides a basic overview of coding the device using python and is useful if the students are not familiar with MicroPython.

This lesson can also be adapted to use visual coding if students are not familiar with programming or for differentiation where required.

CSER Professional Learning:

This lesson plan corresponds to professional learning in the following CSER Digital Technologies MOOCs:

If this is a brand new subject for you, for background on the fundamental concepts involved, you may consider our F-6 Digital Technologies: Foundations MOOC.

● Unit 7: Algorithms and Programming & Unit 8: Visual Programming

Otherwise, we have a Years 7 & 8 Digital Technologies: Next Steps Making Apps MOOC, in which the concepts and key learnings can be applied to the game context.

- Unit 1 - Next Steps 7&8 (Introduction to Years 7&8)
- Unit 2 - Introduction to Making Apps (For introduction to key curriculum concepts)
- Unit 5 - Designing Solutions with Apps (For advice to support students in their design of solutions)
- Unit 6 - Developing & Implementing Apps (For support around taking designs and implementing them in general-purpose programming)

See: http://csermoocs.adelaide.edu.au/moocs

Further Resources:

- Digital Technologies Hub: www.digitaltechnologieshub.edu.au
- CSER: https://csermoocs.adelaide.edu.au

Author: Toni Falusi