

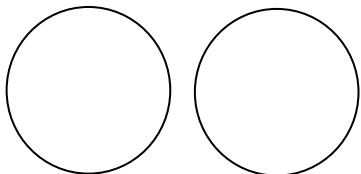
As a team

We want to make sure that our development environment is all setup for the Scrum project simulation

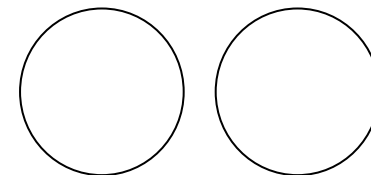
So that we can begin to create something

Acceptance Criteria:

1. We have a PC laptop up and running
2. We have the <http://microbit.org> coding tool up and running in a browser
3. We have a micro:bit attached to the laptop via a USB cable
4. We have a power pack for the micro:bit



Archetype



Dependencies

ExSPS

Reference

Definition of “Done”

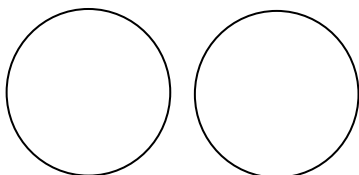
As a Team

We want to create our definition of “Done” for the micro:bit challenge

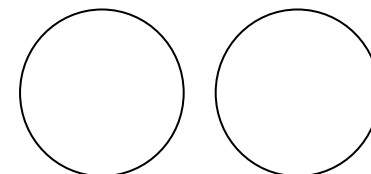
So that we can refer to it during the Sprint and the Sprint Review to understand if items are really “Done”

Acceptance Criteria:

1. We have agreed a definition of “Done” for our Sprints
2. We have written our definition of “Done” on a sheet of flip chart paper and displayed it in our work area



Archetype



Dependencies

ExSPS

Reference

Micro:bit RPSLS: Game Play ^{Size} N/A

Feature:

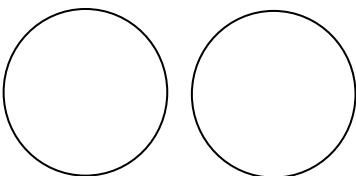
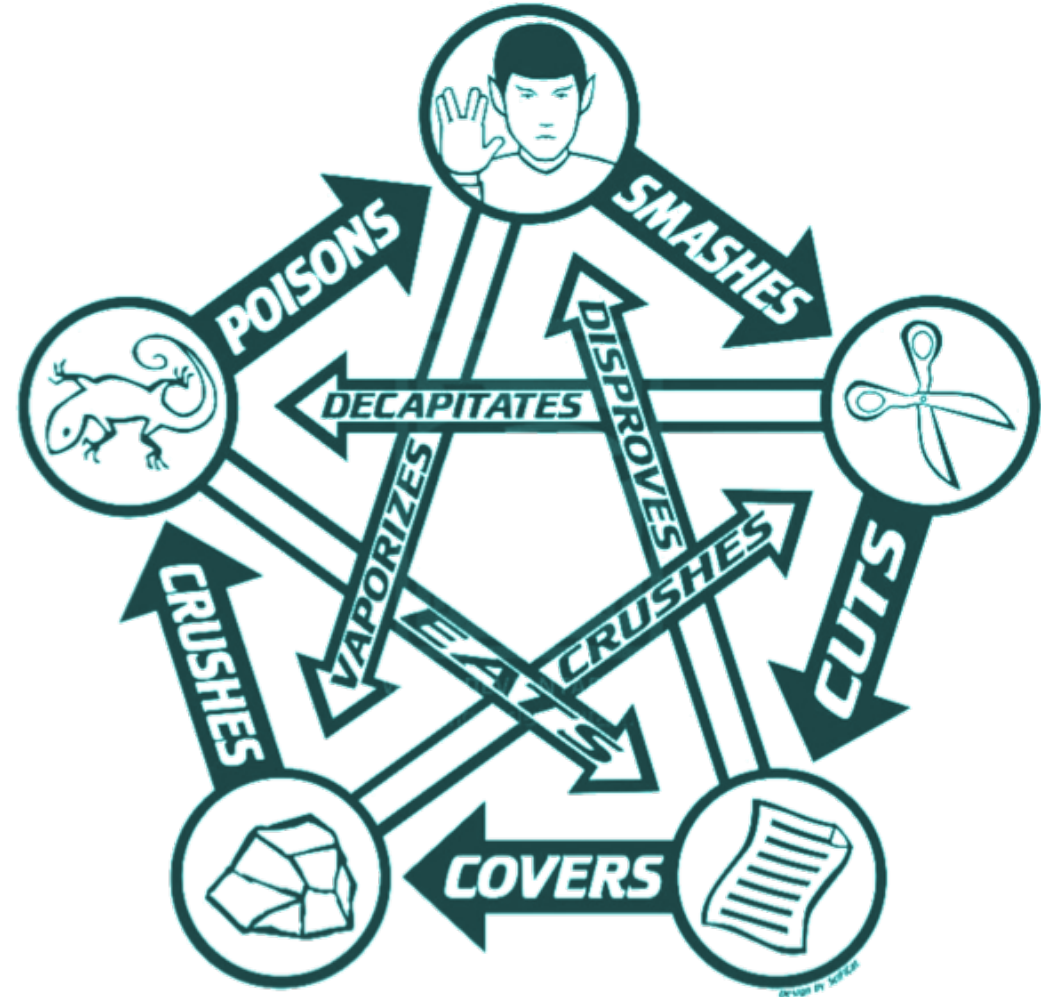
In order to play the Rock, Paper, Scissors, Lizard, Spock game

As a player

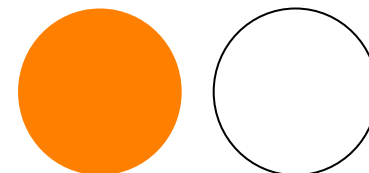
I want to be able to play the Rock, Paper, Scissors, Lizard, Spock game

So that I can experience building something within an Agile context

Theme



Archetype



Dependencies

ExSPS

Reference

Micro:bit RPSLS: Paper Icon

Feature: In order to play the Rock Paper Scissors game

As a player

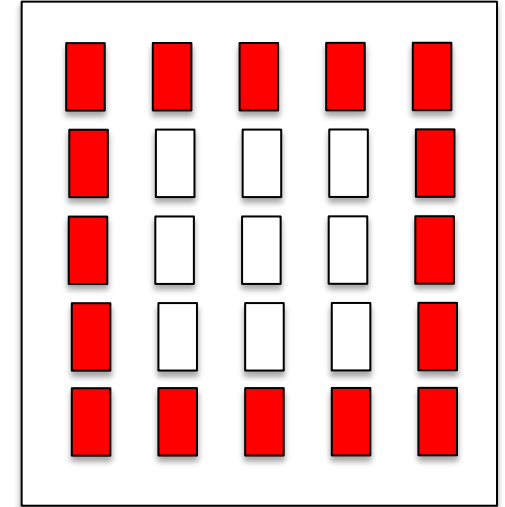
I want to see the paper icon output on the micro:bit

Scenario: A player selects button A and the paper icon is displayed on the micro:bit

Given that our program is running on the micro:bit

When a user selects <button> on the micro:bit

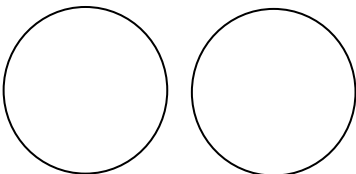
Then <output> is displayed



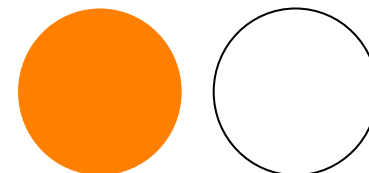
Theme

Examples:

```
| button | output |
| A | {paper icon}
```



Archetype



Dependencies

ExSPS

Reference

Feature: In order to play the Rock Paper Scissors game

As a player

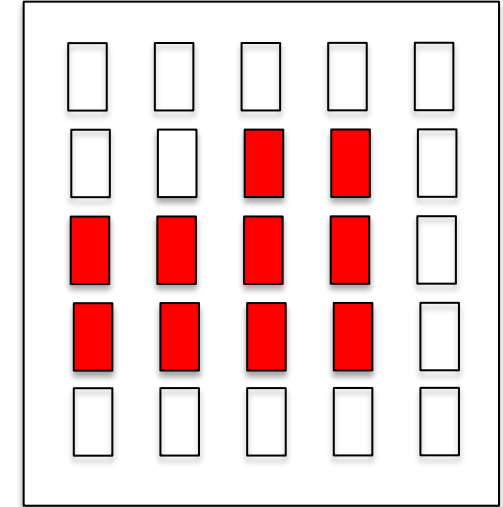
I want to see the rock icon output on the micro:bit

Scenario: A player selects button A and the rock icon is displayed on the micro:bit

Given that our program is running on the micro:bit

When a user selects <button> on the micro:bit

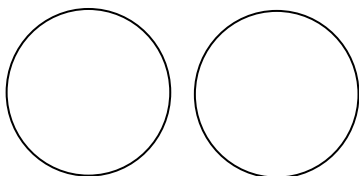
Then <output> is displayed



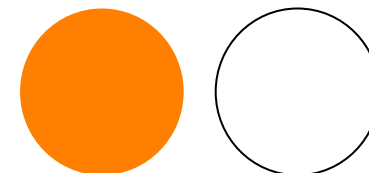
Theme

Examples:

```
| button | output |
| A | {rock icon}
```



Archetype



Dependencies

ExSPS

Reference

Feature: In order to play the Rock Paper Scissors game

As a player

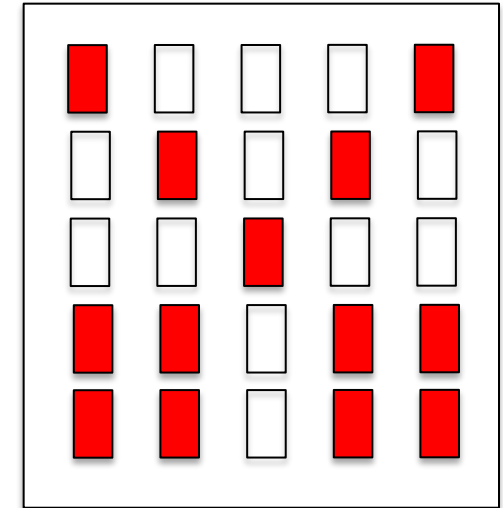
I want to see the scissors icon output on the micro:bit

Scenario: A player selects button A and the scissors icon is displayed on the micro:bit

Given that our program is running on the micro:bit

When a user selects <button> on the micro:bit

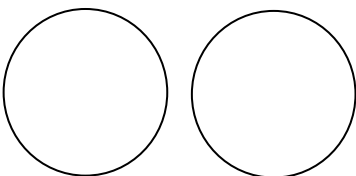
Then <output> is displayed



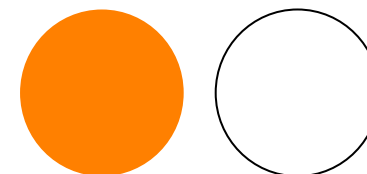
Theme

Examples:

```
| button | output |
| A | {scissors icon}
```



Archetype



Dependencies

ExSPS

Reference

Feature: In order to play the Rock Paper Scissors Lizard Spock game

As a player

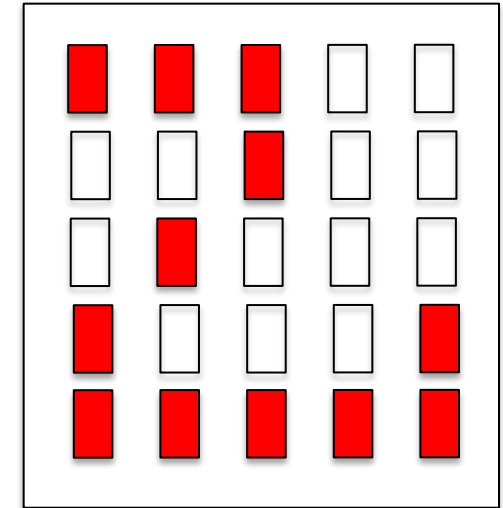
I want to see the lizard icon output on the micro:bit

Scenario: A player selects button A and the lizard icon is displayed on the micro:bit

Given that our program is running on the micro:bit

When a user selects <button> on the micro:bit

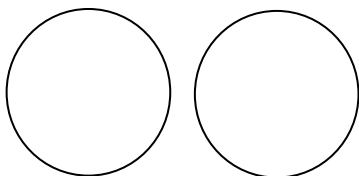
Then <output> is displayed



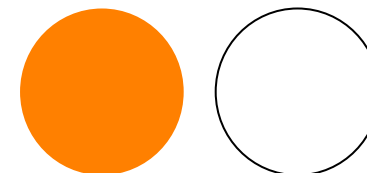
Theme

Examples:

```
| button | output |
| A | {lizard icon}
```



Archetype



Dependencies

ExSPS

Reference

Micro:bit RPSLS: Spock Icon

Feature: In order to play the Rock Paper Scissors Lizard Spock game

As a player

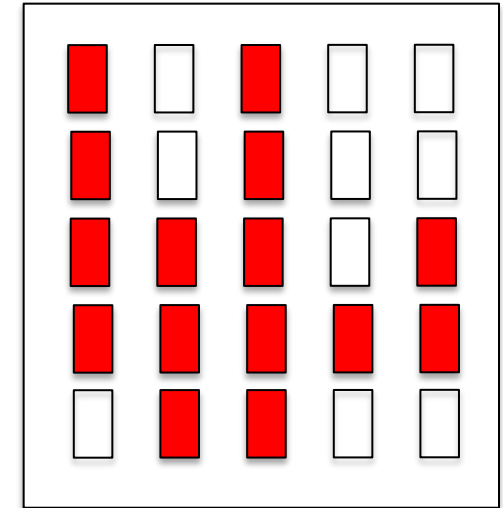
I want to see the Spock icon output on the micro:bit

Scenario: A player selects button A and the Spock icon is displayed on the micro:bit

Given that our program is running on the micro:bit

When a user selects <button> on the micro:bit

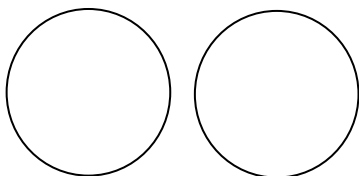
Then <output> is displayed



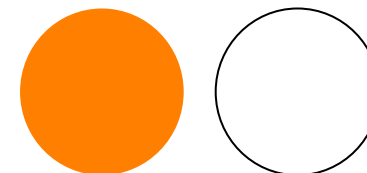
Theme

Examples:

```
| button | output |
| A | {Spock icon}
```



Archetype



Dependencies

ExSPS

Reference

Feature: In order to play the Rock Paper Scissors Lizard Spock game

As a player

I want to browse through the icons using a button on the micro:bit

Scenario: A player selects button A and the next icon is displayed on the micro:bit

Given that our program is running on the micro:bit

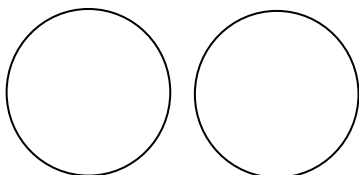
And <current icon> is displayed

When a user selects <button> on the micro:bit

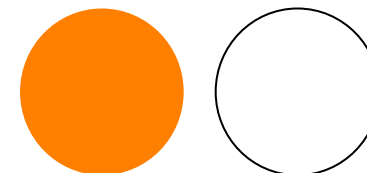
Then <next icon> is displayed

Examples:

```
| current icon | button | next icon |
| {rock icon} | A | {paper icon}
| {paper icon} | A | {scissors icon}
| {scissors icon} | A | {lizard icon}
| {lizard icon} | A | {Spock icon}
| {Spock icon} | A | {paper icon}
```



Archetype



Dependencies

ExSPS

Reference

Feature: **In order to** play the Rock Paper Scissors Lizard Spock game

As a player

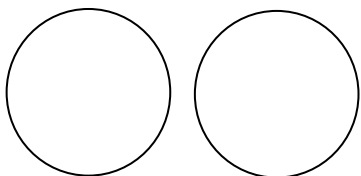
I want to select an icon at random by simply shaking the micro:bit

Scenario: A player shakes the micro:bit and the next randomly selected icon is displayed on the micro:bit

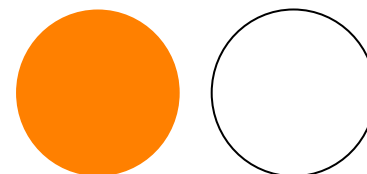
Given that our program is running on the micro:bit

When a user shakes the micro:bit

Then a randomly selected icon is displayed



Archetype



Dependencies

ExSPS

Reference

Feature: In order to play the Rock Paper Scissors Lizard Spock game

As a player

I want to send my selected icon to the other player's micro:bit

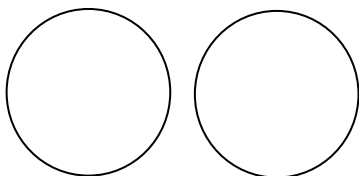
So that the winner in the game can be determined and their score recorded

Scenario: A player sends their selected icon the other player's micro:bit

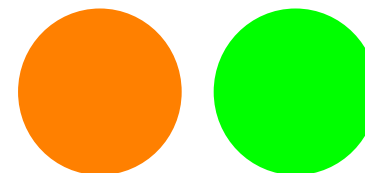
Given that our program is running on the micro:bit

When a user selects button 'B'

Then the currently displayed icon is transmitted to the other player's micro:bit



Archetype



Dependencies

ExSPS

Reference

Feature: **In order to** play the Rock Paper Scissors Lizard Spock game

As a player

I want the scoring micro:bit to receive the other player's selected icon

So that the winner in the game can be determined and their score recorded

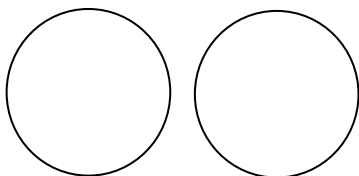
Scenario: A player sends their selected icon to the other player's micro:bit which is received and displayed

Given that our program is running on the micro:bit

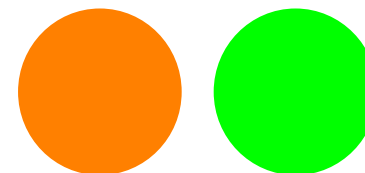
And the micro:bit is ready to receive icons

When a player sends an icon

Then the sent icon is received and displayed on the micro:bit



Archetype



Dependencies

ExSPS

Reference

Micro:bit RPSLS: Win Icon

Feature: **In order to** play the Rock Paper Scissors
Lizard Spock game

As a player

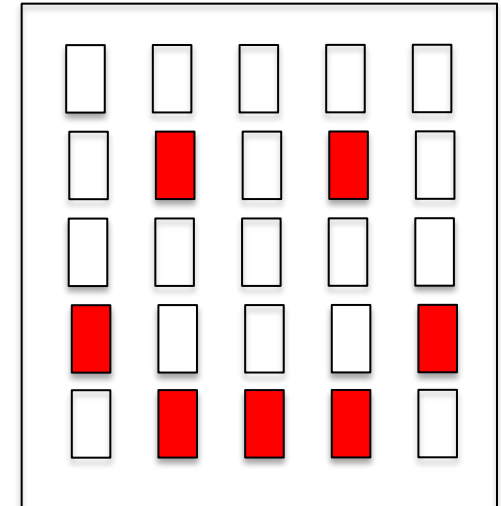
I want to see the win icon output on the
micro:bit

Scenario: A player wins the game and the win icon is
displayed on the micro:bit

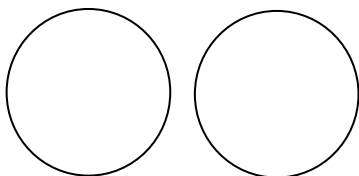
Given that our program is running on the micro:bit

When the player wins the game

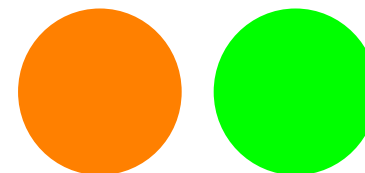
Then the win icon is displayed on their micro:bit



Theme



Archetype



Dependencies

ExSPS

Reference

Micro:bit RPSLS: Lose Icon

Feature: **In order to** play the Rock Paper Scissors
Lizard Spock game

As a player

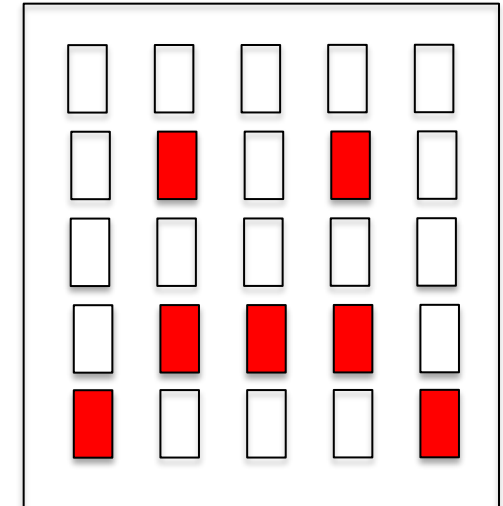
I want to see the lose icon output on the
micro:bit

Scenario: A player loses the game and the lose icon is
displayed on the micro:bit

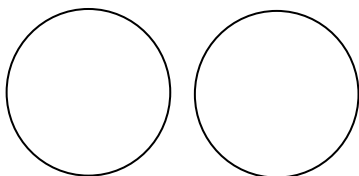
Given that our program is running on the micro:bit

When the player loses the game

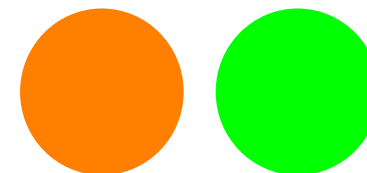
Then the lose icon is displayed on their micro:bit



Theme



Archetype



Dependencies

ExSPS

Reference

Feature: **In order to** play the Rock Paper Scissors
Lizard Spock game

As a player

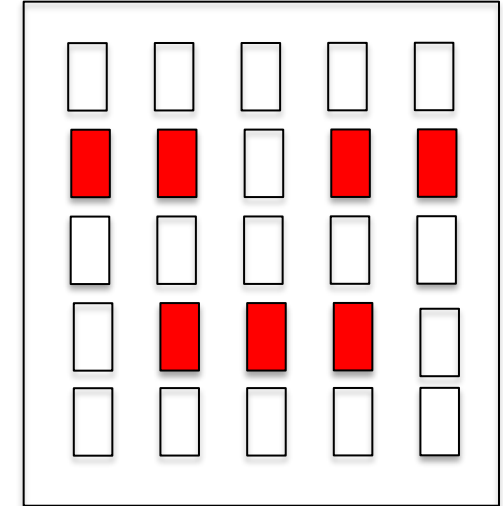
I want to see the lose icon output on the
micro:bit

Scenario: A player draws the game and the draw icon
is displayed on the micro:bit

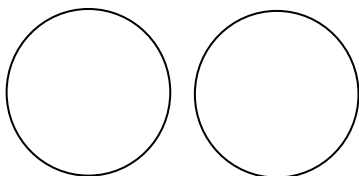
Given that our program is running on the micro:bit

When the player draws the game

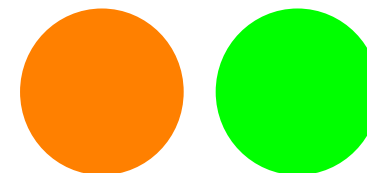
Then the draw icon is displayed on their micro:bit



Theme



Archetype



Dependencies

ExSPS

Reference

Feature: **In order to** play the Rock Paper Scissors Lizard Spock game
As a player
I want my micro:bit to evaluate if I have won or lost the game

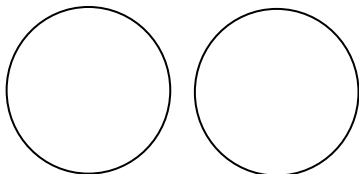
Scenario: After both players have sent their icons to each others' micro:bit, it will determine if the micro:bit is the winner or loser and display winning or losing icons

Given that our program is running on the micro:bit
And both players have sent their icons to each others' micro:bits
When player 1 has <p1 icon> displayed on their micro:bit
And player 2 has <p2 icon> displayed on their micro:bit
Then the <p1 result> is displayed on the player 1's micro:bit
And the <p2 result> is displayed on player 2's micro:bit

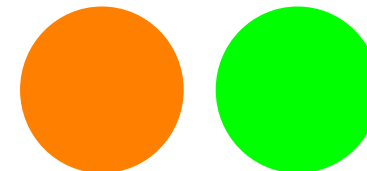
Examples:

p1 icon	p2 icon	p1 result	p2 result
rock	paper	lose	win
rock	lizard	win	lose
rock	spock	lose	win
rock	scissors	win	lose
rock	rock	draw	draw
paper	lizard	lose	win
paper	spock	win	lose
paper	scissors	lose	win
paper	paper	draw	draw
scissors	lizard	win	lose
scissors	spock	lose	win
scissors	scissors	draw	draw
spock	lizard	lose	win
spock	spock	draw	draw
lizard	lizard	draw	draw

Theme



Archetype



Dependencies

ExSPS

Reference

Feature: **In order to** play the Rock Paper Scissors Lizard Spock game

As a player

I want the micro:bit to be reset

So that we can begin a new game

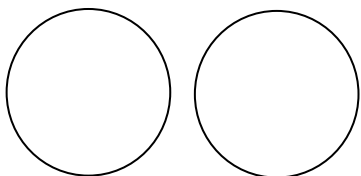
Scenario: A player resets the micro:bit so that a new game can be started

Given that our program is running on the micro:bit

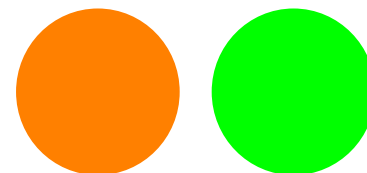
And the winning or losing icon is displayed on the micro:bit

When a user selects button 'A'

Then the micro:bit is reset and ready to begin a new game



Archetype



Dependencies

ExSPS

Reference