Extension: Co-op Mode - Player 1 Side

Goal: We're going to play co-op mode! Create a Player 1 that displays the moves to be played, and receives the actions

Task 1.1: Configure the Radio

We'll need to start a new file for our game master!

1. Create a new file, and save it as **`coopplayer1.py`**.
2. At the top of your file, `import` the **`micro:bit, random`** and **`radio`** modules.
3. Turn the radio on.
4. Configure the radio to use the group channel that the room coordinator gave you.

Task 1.2: Set up the game

Let's set up the variables we need!

1. Create a variable called **`score`**, and set it to **`0`**.
2. Create a list called **`actions`**, and have it include `"button a"` and `"button b"`.
3. Create a variable called **`action`**, and randomly select from **`actions`**.

Task 1.3: Keep Playing!

Let's keep checking moves!

1. Create a `while` loop that keeps running while `running_time()` is less than `10000`.

2. Inside the `while` loop, create an `if` statement that checks to see if the `action` is equal to `"button a"`. If it is, `display` a left arrow.

3. Create a second `if` statement that checks to see if the `action` is equal to `"button b"`. If it is, `display` a right arrow.

4. At the end of your `while` loop, `sleep` for `200` seconds and then randomly select a new `action`.

Task 1.4: Right move?

Let's see if the other player gets the correct move!

1. Inside your `while` loop, before the `sleep`, check to see `if` the radio message received matched the action. If it did, `display` a happy face and increase the `score` by `1`.

2. If it didn't, just `continue`.

Task 1.5: Game Over!

When the time is up, let everyone know the score!

1. Send a radio message that says `"finished"`.

2. `Scroll` the `score` across the LED screen.

Task 1.6: Stop! Testing time!

Play your game!

1. Try out your game! Don't forget you'll need to say what the action the other player will need to do!

☑ CHECKPOINT ☑

If you can tick all of these off you have finished this Extension:

☐ You have configured your radio using the group number the room coordinator gave you

☐ Moves are randomly selected while the game is running

☐ The correct arrow is displayed for the chosen action.

☐ You increase the score by 1 when the correct action is received

☐ At the end of the game, the score is displayed

Extension: Co-Op Mode - Player 2 Side

Goal: We're going to play co-op mode! Create a Player 2 that sends the action that needed to be made!

Task 2.1: Configure the Radio

We'll need to start a new file for our game master!

1. Create a new file, and save it as **`coopplayer2.py`**.
2. At the top of your file, `import` the **`micro:bit`** and **`radio`** modules.
3. Turn the radio on.
4. Configure the radio to use the group channel that the room coordinator gave you.

Task 2.2: Play the game!

Let's play!

1. Create a `while` loop that keeps running until a radio message is received saying `"finished"`.
2. Inside the `while` loop, **`display`** a question mark.
3. Create an `if` statement that checks to see if **`button a`** was pressed. If it was, use the radio to send the message `"button a"`.
4. Add an `elif` that checks to see if **`button b`** was pressed. If it was, use the radio to send the message `"button b"`.
5. Otherwise, just `continue`.
6. At the end of your `while` loop, **`display`** an exclamation mark image, and then sleep for **`200`** milliseconds.

Task 2.3: Game over!

Play your game!

1. Try out your game! Don't forget you'll need to press what button the other player told you to!

☑ CHECKPOINT ☑

If you can tick all of these off you have finished this Extension:

☐ You have configured your radio using the group number the room coordinator gave you

☐ Your while loop runs while the message "finished" hasn't been received

☐ You send a message saying which button has been pressed