Extension: Basic Radio – Player Side

Right now, we can only compete against ourselves, which is getting a bit boring! Let's see if we can link the micro:bits. We'll have one micro:bit which is the game master, and then all the other micro:bits will be players!

Goal: Configure the Micro:bit's radio to start the game, and send a response when you get to 10 points!

Task 1.1: Configure the Radio

We need to configure the radio to start off with

1. At the top of your program, `import radio`.

2. After the target image is displayed, turn the radio on with `radio.on()`

3. Then configure the radio's group with `radio.config(group=100)`. Your room coordinator will tell you what number to use.

Task 1.2: Ready, Set, Go!

Make the micro:bit wait until it's been told to start!

1. Before your main game `while` loop, add a new `while` loop that keeps running until the radio receives a message of `"start"`.

2. Inside the `while` loop, add a `pass` statement.

Hint - Radio Messages

You can read the message that the radio has received with the following code:

```
incoming = radio.receive()
```

Task 1.3: Game over!

Send a message to the game master when you’ve reached 10 points!

1. Update your main game while loop so it only runs if the score is less than 10.
2. At the end of your code, and outside of the main game while loop, send the player’s name via the radio!

Hint - Radio Send

You can send a message using the radio with the following code:

```
radio.send("I won!")
```

☑ 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.

□ The game doesn’t start until the game master says start!

□ When you have reached 10 points, the player’s name is sent to the game master.

Extension: Basic Radio - Game Master Side

Let's try programming our own Game Master code, so we can play Bop It! whenever we want!

Goal: Create the Game Master code

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 `gamemaster.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: Set up the game

Let's set up the variables we need!

1. Create a variable called `winner`, and set it to `None`.
2. Constantly scroll a message that says `"PRESS A to Start"`.
3. Make sure your message has a wait of `False`.

Hint - Scrolling messages

To make a message scroll constantly, and have a wait of false, you can use the following code:

```
display.scroll("Welcome to GPN", wait=False, loop=True)
```

Task 2.3: Start the game!

Send a message to the players when you're ready to start the game!

1. At the end of your code, create a while loop that keeps running while winner is equal to None.
2. Inside the `while` loop, add an `if` statement that checks to see if **`button_a`** was pressed.
3. If **`button_a`** was pressed, send a message using the radio with the message `"start"`.
4. Outside of the `if` statement, but still inside the `while` loop, set the value of **`winner`** to be the message the radio receives.

Task 2.4: Configure the Radio

Display the winner!

5. At the end of your code, **`scroll`** who the **`winner`** was continuously!

☑ 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 radio sends a message of "start" when button_a is pressed.

□ When there is a winner, their name is displayed!

★ BONUS 1.5: Images!

Our game master doesn't really do anything when it's waiting for a winner. Let's make it display some images!

1. In your code, just before when the `winner` variable is created, create a new list called `images`. Add as many images as you want in this, such as `Image.CHESSBOARD` and `Image.CHESSBOARD.invert()`.

2. Inside your `if` statement before the start radio message is sent, start displaying the images on repeat by using the following code:
   ```
   display.show(images, wait=False, loop=True)
   ```