

```
on start
  set cherry to sprite of kind Food
  set cherry position to x pick random 0 to 160 y pick random 0 to 120
```

```
let cherry = sprites.create(img, SpriteKind.Food)
cherry.setPosition(Math.randomRange(0, 160), Math.randomRange(0, 120))
```

Microsoft MakeCode



< Instructor Name >

< Title >

How to create an Arcade Game



Microsoft MakeCode



Inspiring new generations of technology creators
through immersive, hands-on computing education

Learn more
makecode.com

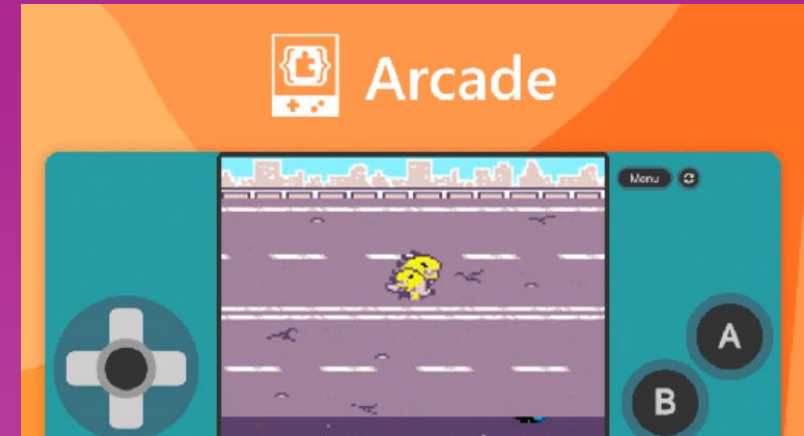
Three Main Code Editors



Physical Computing with micro:bit

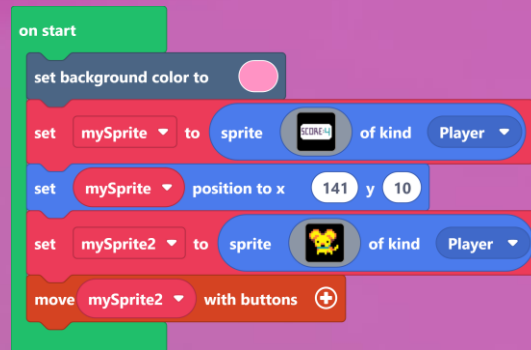


Mods in Minecraft

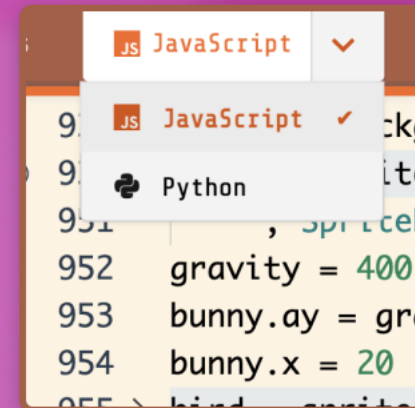


Retro Arcade Games

Blocks



Text



What is an Arcade game?

History of Arcade Games

An arcade game is a coin-operated video game machine installed in a public place like a restaurant, or an amusement park. Arcade games rose to popularity in the 1970's and 1980's.

The first successful Arcade game was called Pong, created by Atari in 1972.



The two paddles return the ball back and forth. The score is kept by the numbers (0 and 1) at the top of the screen.

Examples of Arcade Games?

Pac-Man



Space Invaders

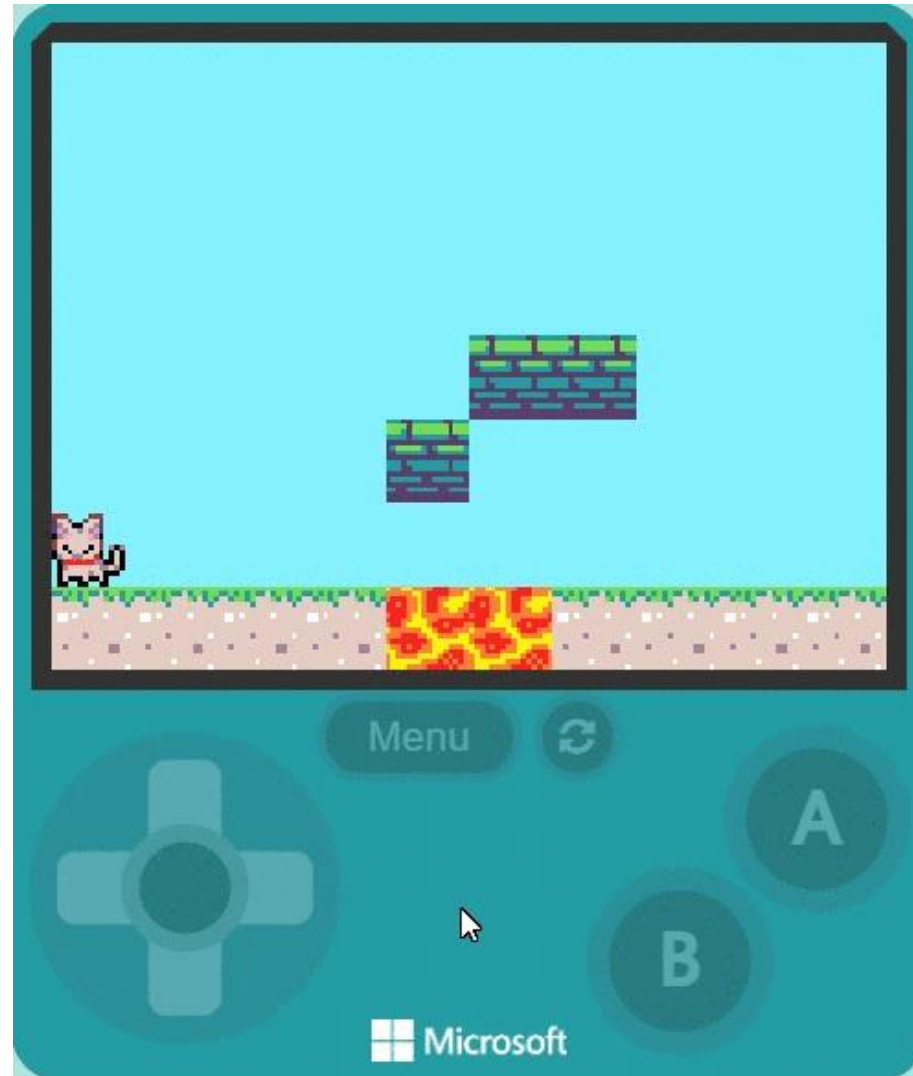


Donkey Kong



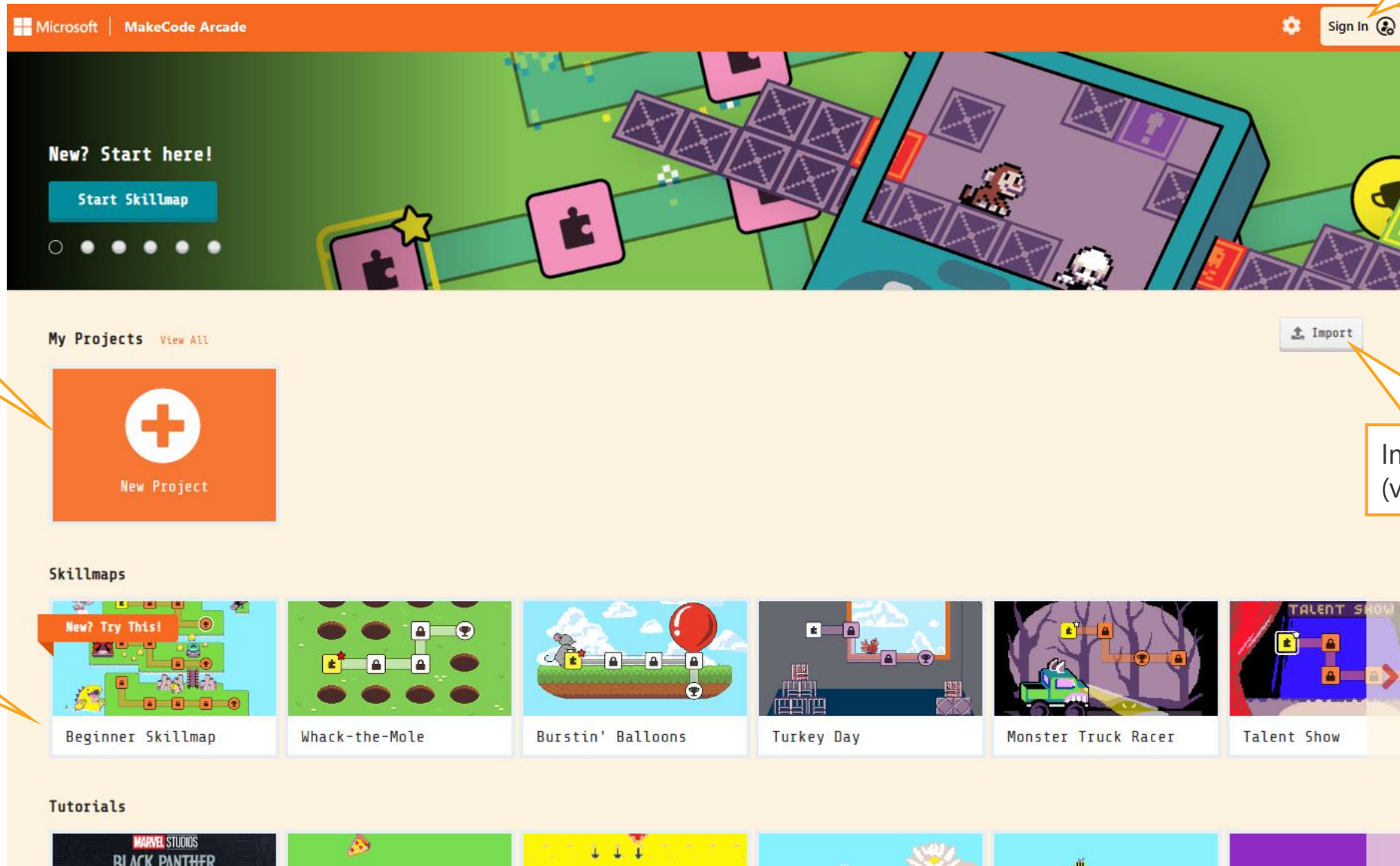
Let's code a game!

Platformer



Open Browser: arcade.makecode.com

Optionally sign in to save to projects cloud



Create a New blank Project

Import Projects (via a File, or URL)

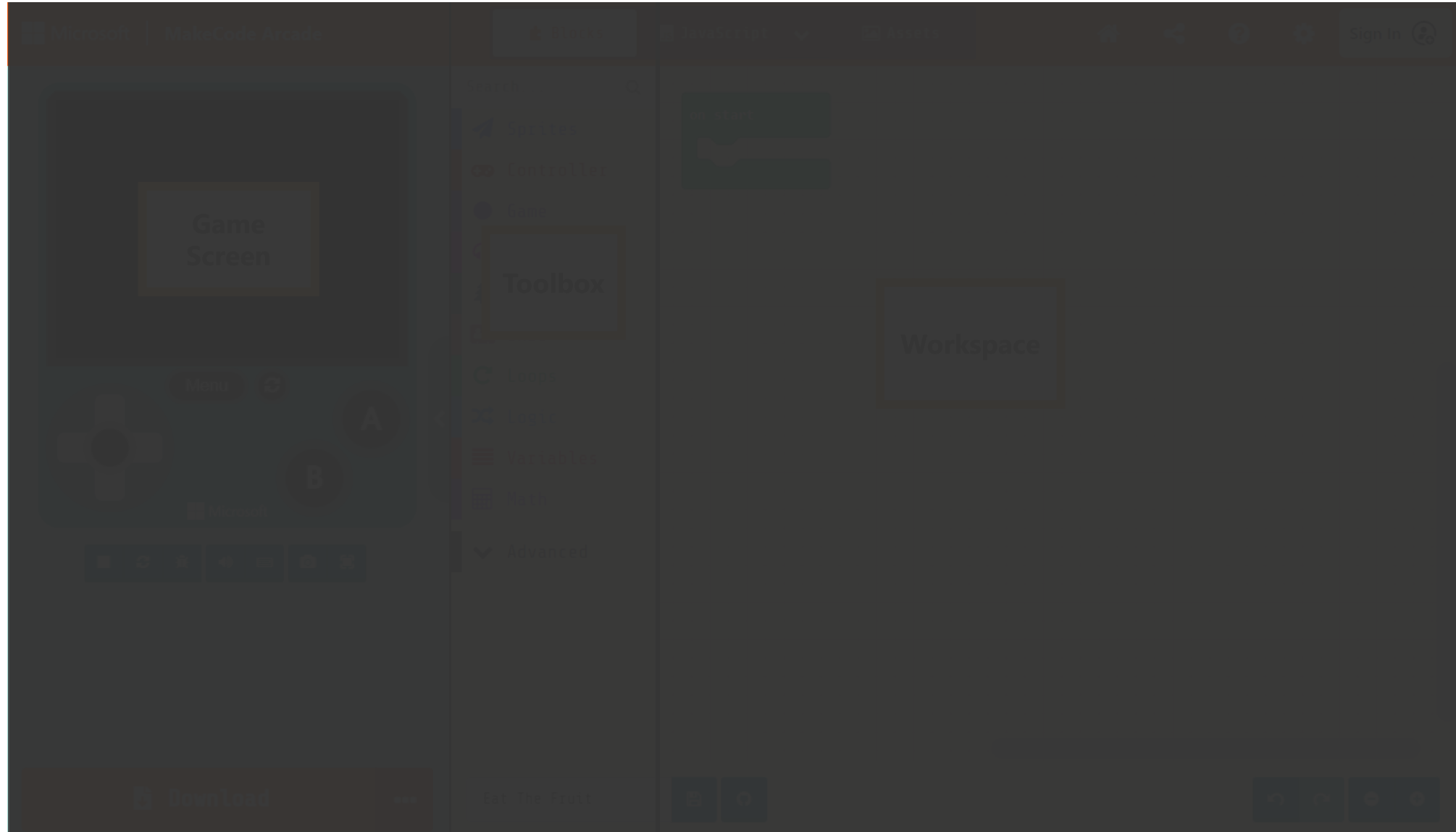
Step-by-step Skillmaps and Tutorials

Create a New Project

The screenshot shows the Microsoft MakeCode Arcade interface. At the top, there's a navigation bar with the Microsoft logo and 'MakeCode Arcade' text, along with a settings gear and a 'Sign In' button. Below this is a hero section with the text 'New? Start here!' and a 'Start Skillmap' button. The main content area is divided into 'My Projects' (with a 'View All' link and an 'Import' button), 'Skillmaps', and 'Tutorials'. In the 'My Projects' section, an orange square button with a white plus sign and the text 'New Project' is circled in purple. A purple handwritten note 'Click on New Project' with an arrow points to this button. A 'Create a Project' dialog box is open, featuring a title bar with two smiley face emojis and a close button. The dialog prompts the user to 'Give your project a name.' with a text input field containing 'Space Game!'. Below the input field is a '> Code options' link and a green 'Create' button with a checkmark. The background shows various project thumbnails, including 'Beginner Skillmap', 'Whack-the-Mole', 'Burstin' Balloons', 'Turkey Day', 'Monster Truck Racer', and 'Talent Show'.

Click on
New Project

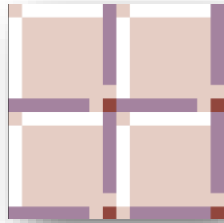
Getting familiar with MakeCode Arcade



Tiles and Tilemaps

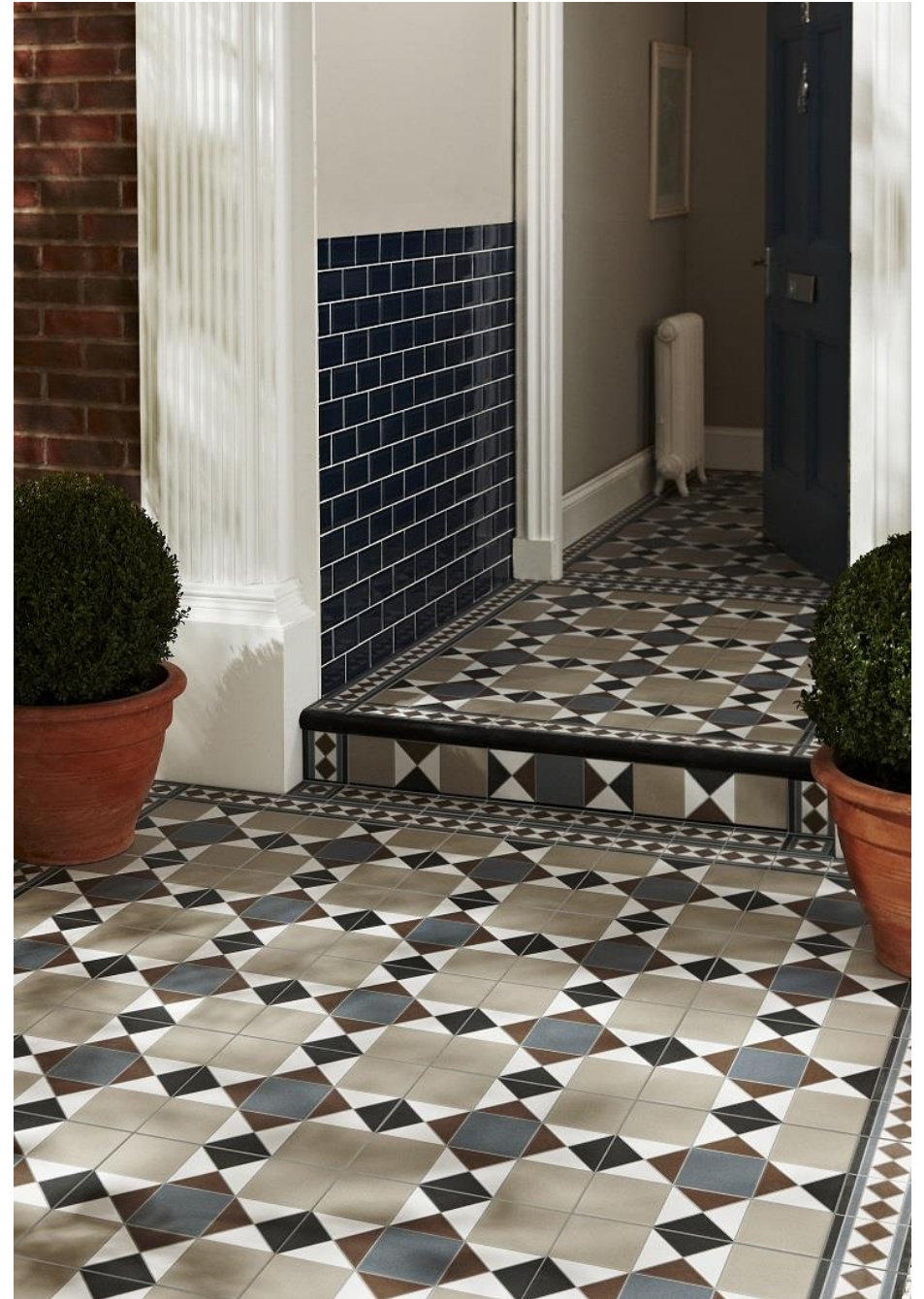
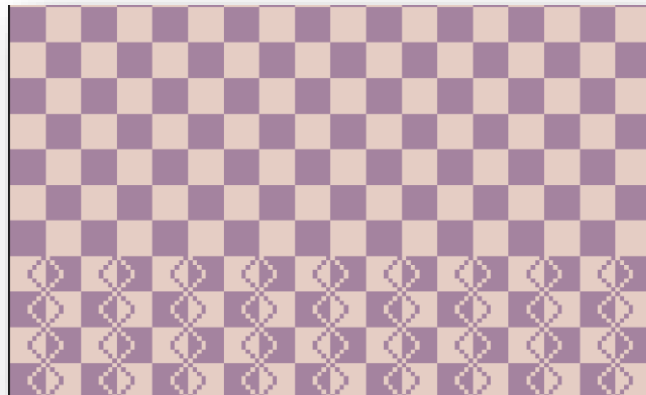
- What is a Tile?

A square pattern



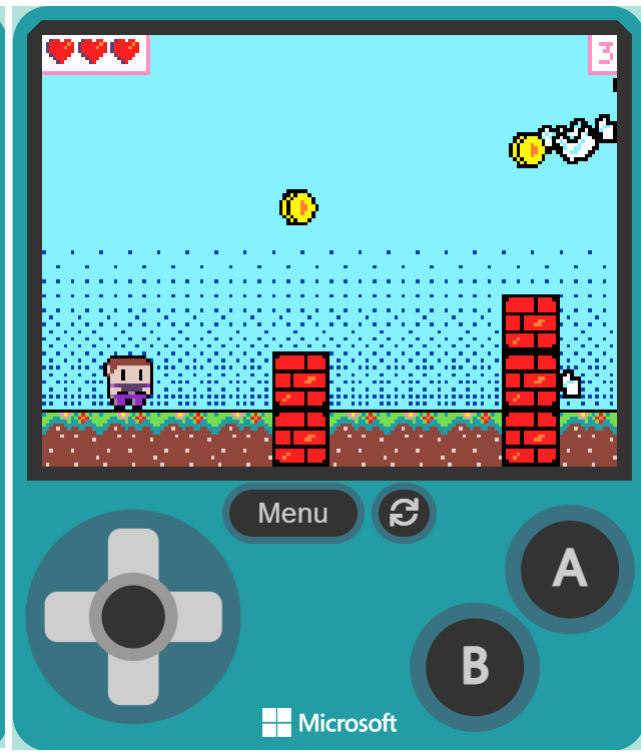
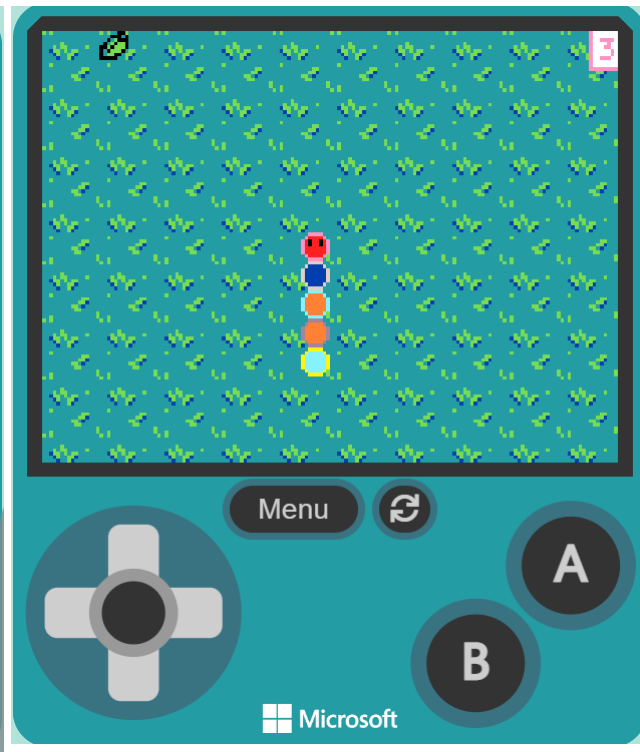
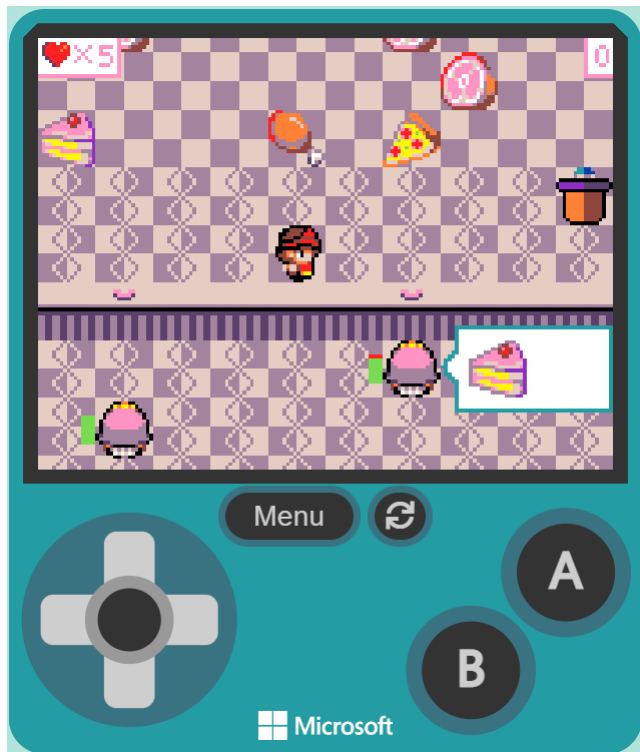
- What is a Tilemap?

A collection of tiles



Tilemap Games

Tilemaps are good for top-down perspective (maze, adventure) or side-view perspective games (platformer, side-scroller)



Create the Tile Map

The screenshot shows the Scratch interface with the 'Scene' category selected in the left sidebar. The 'Tilemaps' section is active, displaying several blocks:

- Effects:**
 - start screen **confetti** effect (+)
 - end screen **confetti** effect
- Tilemaps:**
 - set tilemap to tilemap** (highlighted with a yellow border and a red arrow pointing to the zoomed-in view)
 - tilemap (grey box)
 - on **sprite** of kind **Player** overlaps **tilemap** at **location**

The zoomed-in view shows an **on start** block containing a **set tilemap to tilemap** block. The grey box next to the 'tilemap' text is highlighted with a yellow border, and a red arrow points from this box to the zoomed-in view.

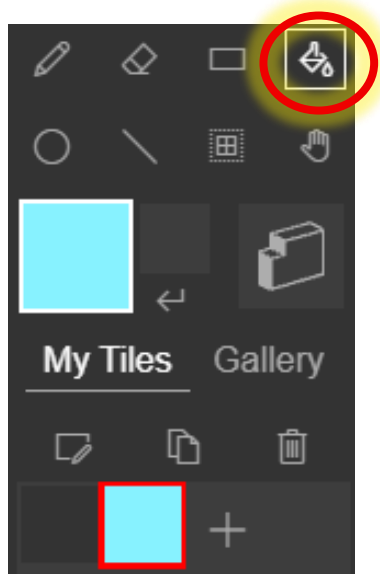
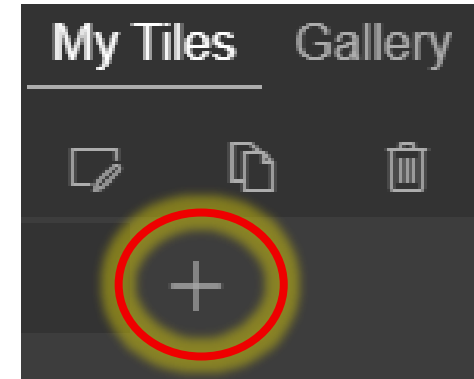
Click on grey box to open Tilemap Editor

Set the Canvas

- In the lower left corner, set the Tilemap size to **50** x **8** pixels



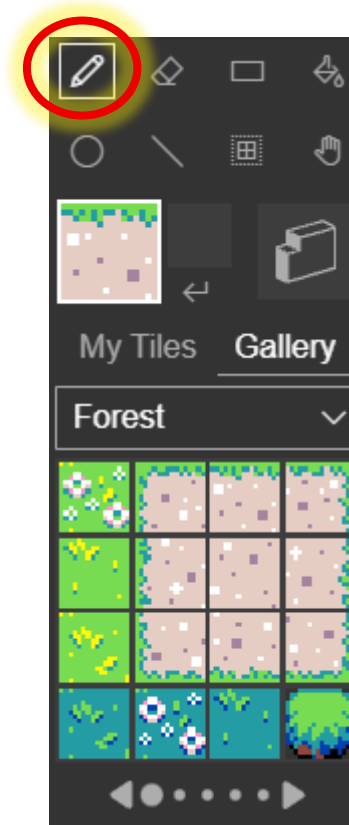
- Click on **My Tiles**, and Click on **+** to draw your tile to serve as the background (sky?)



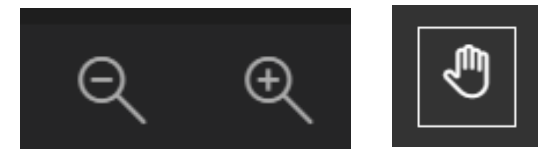
- Use the **Fill** tool to paint your whole Tilemap

Draw the ground

- From the **Gallery**, pick a tile to use as the ground (or draw your own in My Tiles)
- Paint the tile along the bottom of your Tilemap

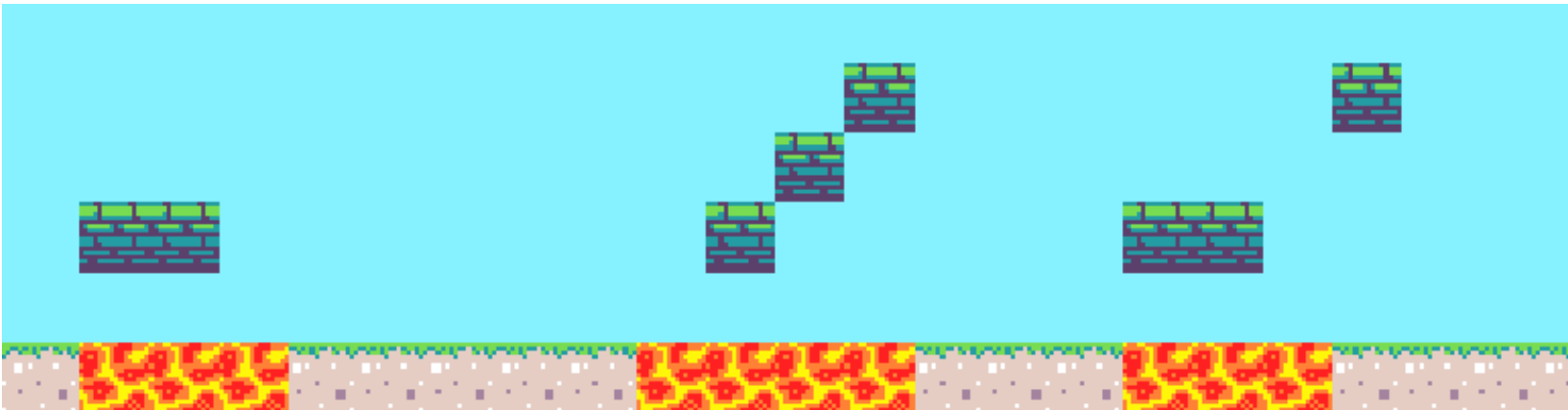


Note – you can use the **Zoom** buttons in the bottom right corner to zoom out to see your whole tilemap, and the **Pan** tool to move your tilemap around



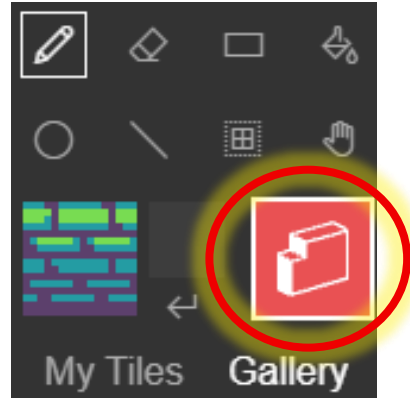
Draw pits and platforms

- From the Gallery, pick a tile to use as the pits (or draw your own in My Tiles)
- Paint the pits in places along the ground
- From the Gallery, pick a tile to use as the platforms (or draw your own in My Tiles)
- Paint the platforms above pits where your player can jump up to

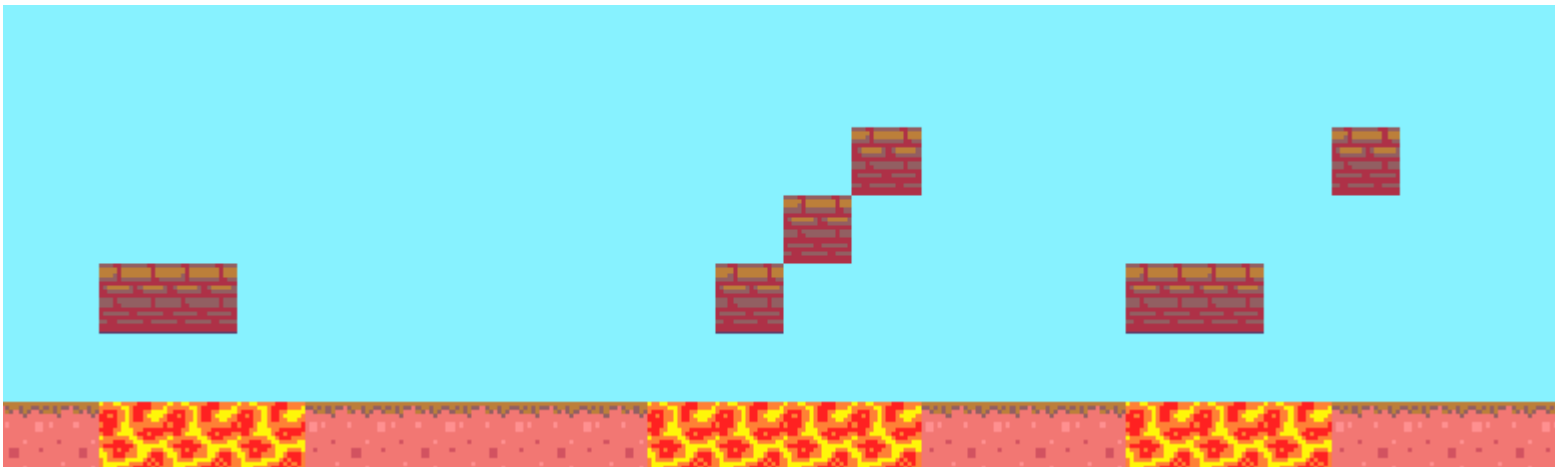


Add walls (tiles player can't go through)

- Select the **Wall** Tool

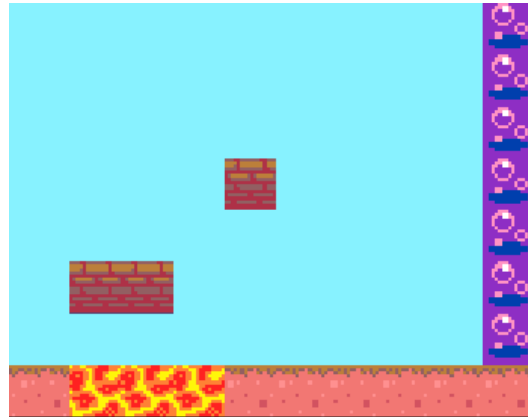


- Paint walls on the ground and the platform tiles



Add Finish Line

- From the Gallery, pick a tile to use as the ending point (or draw your own in My Tiles)
- Paint tile along the right-most edge of your Tilemap

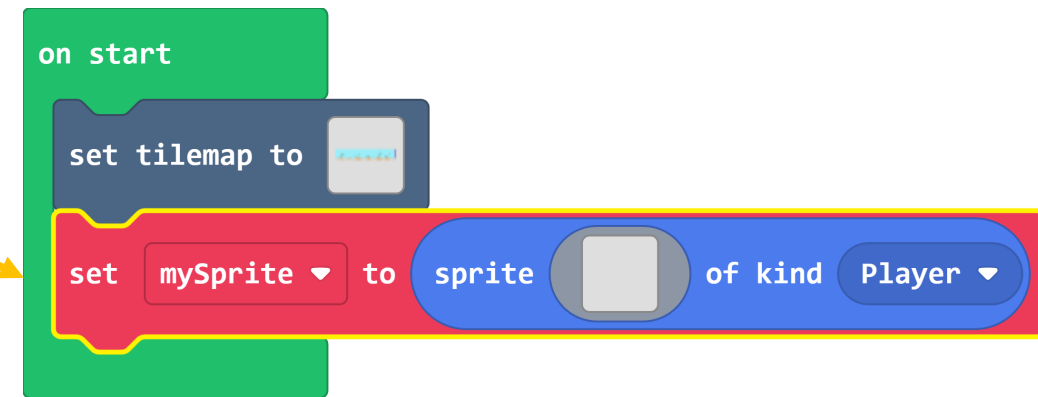
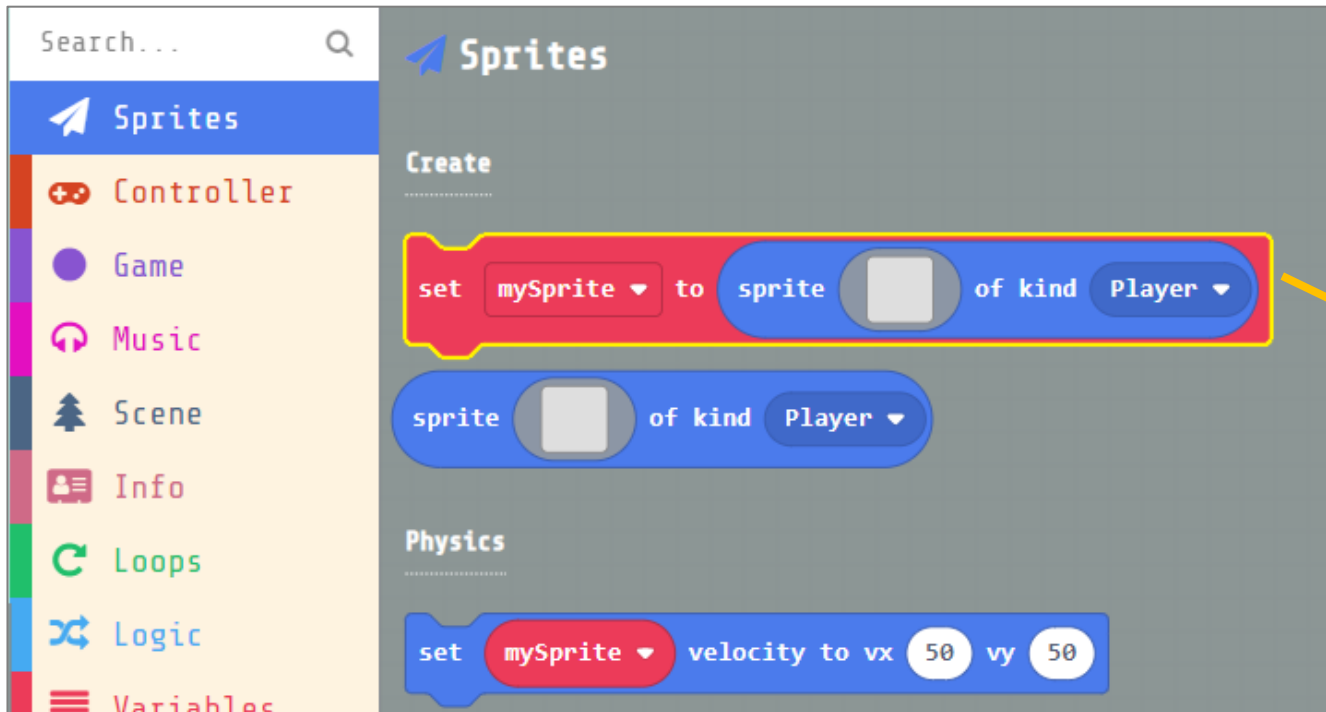


- Click **Done** to close Tilemap Editor



Create your Player Sprite

- From the **Sprites** Toolbox drawer, drag a **Set mySprite** block into the **On Start** block

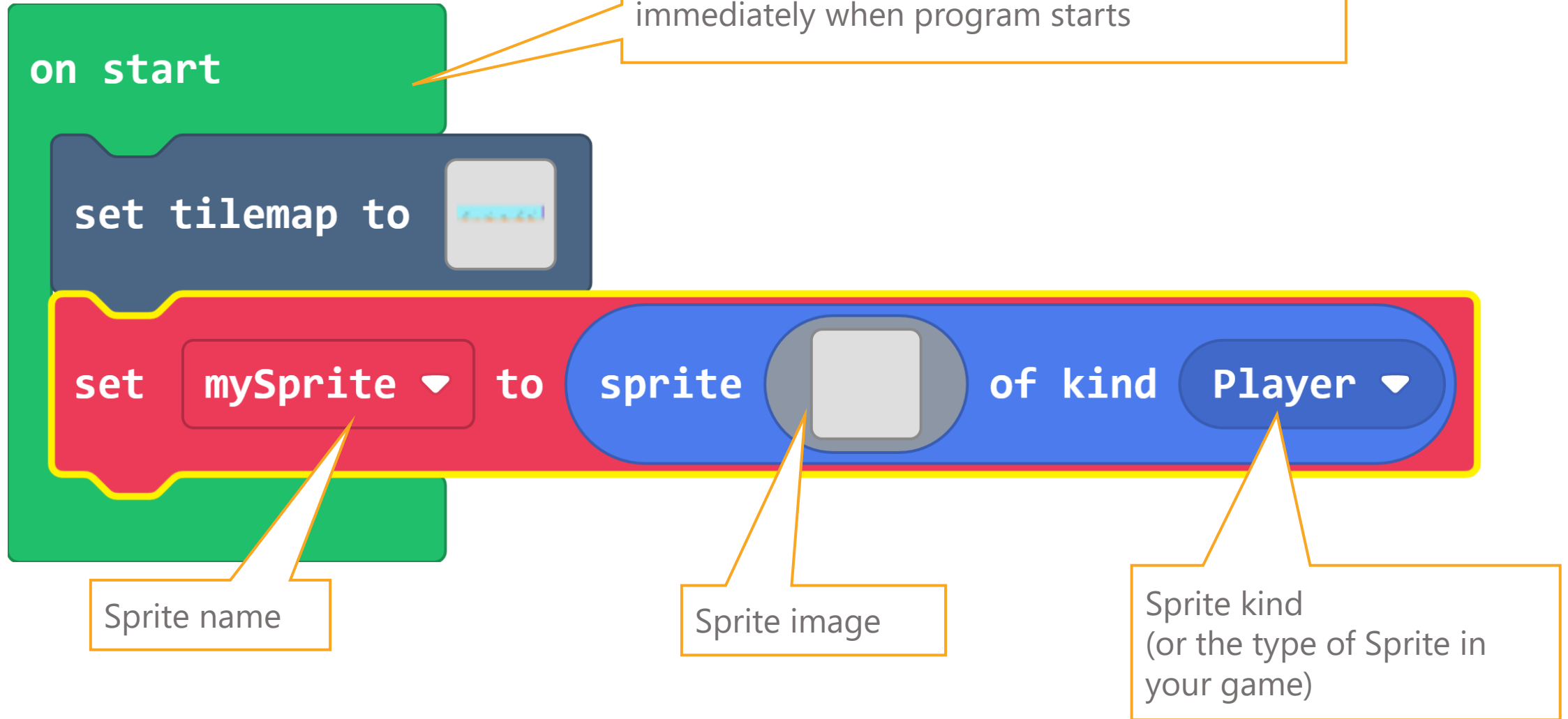


Sprites



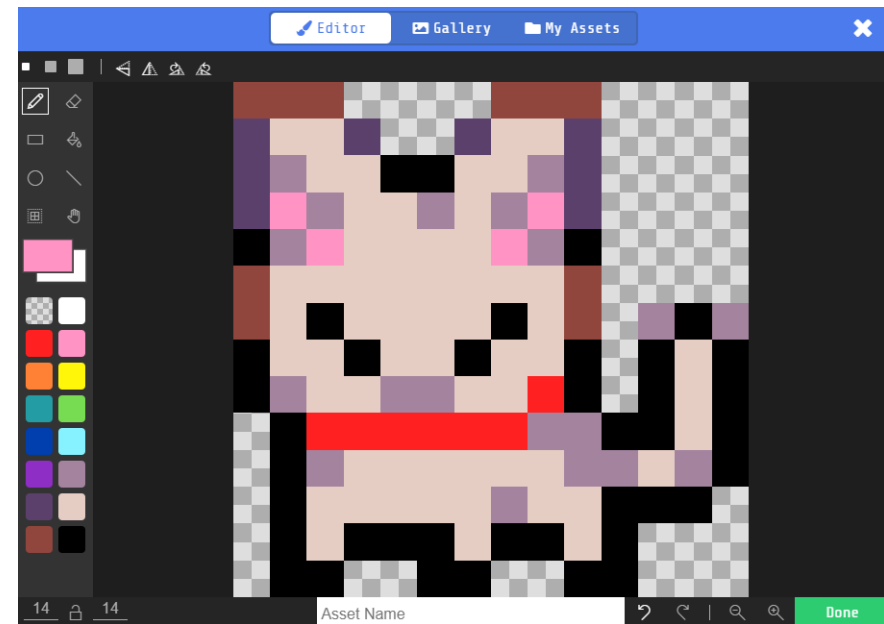
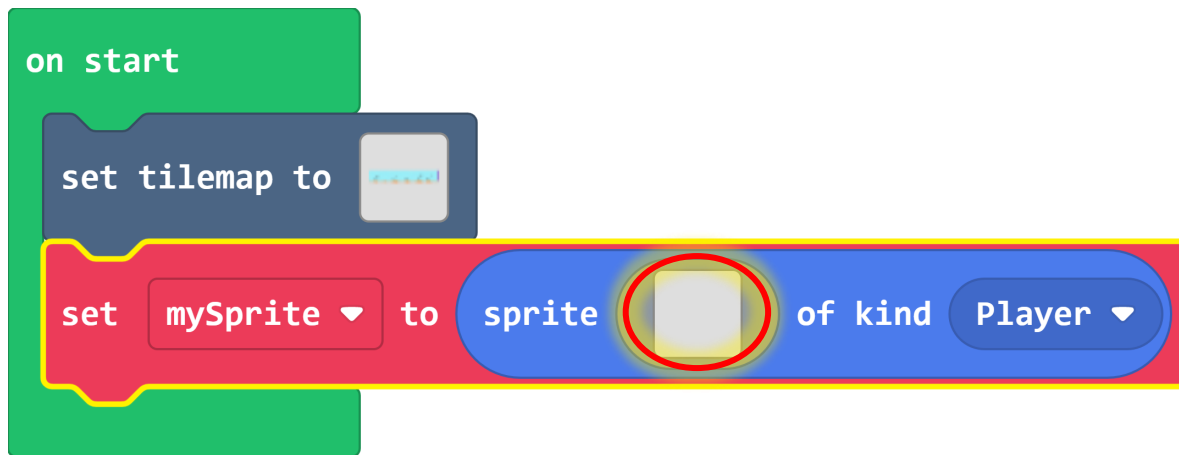
any object in a game that has properties and behaviors

Set Sprite block



Draw your character

- In the **Set mySprite** block, click on the grey square to open the Image Editor
- Draw your Player Character
- Or pick an existing image from the Gallery



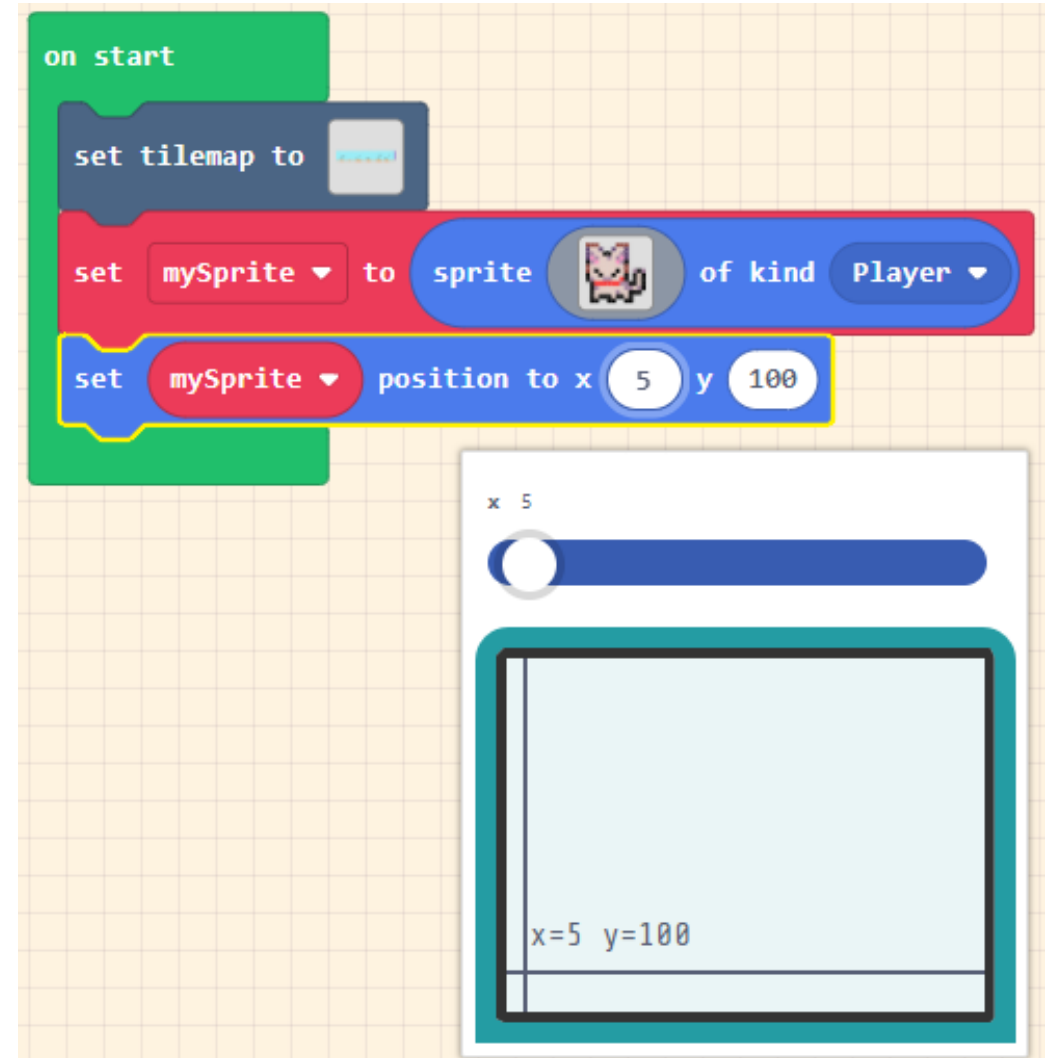
Set the starting position

- From the **Sprites** Toolbox drawer, drag a **Set mySprite Position** block into the **On Start** block

The image shows a software interface with a toolbox on the left and a script editor on the right. The toolbox is titled 'Sprites' and contains several categories: Sprites, Controller, Game, Music, Scene, Info, Loops, Logic, Variables, Math, and Advanced. The 'Sprites' category is selected, showing a 'Create' section with a 'set mySprite2 to sprite of kind Player' block and a 'Physics' section with 'set mySprite velocity to vx 50 vy 50' and 'set mySprite position to x 0 y 0' blocks. A red arrow points from the 'set mySprite position to x 0 y 0' block in the toolbox to the 'on start' script block on the right. The 'on start' script block contains three blocks: 'set tilemap to', 'set mySprite to sprite of kind Player', and 'set mySprite position to x 0 y 0'.

Set the starting position

- In the **Set mySprite Position** block, click on the **x** value to open the coordinate picker
- Move your cursor to the bottom left of the screen and click to select the starting position for your sprite
- Check the location in the Game Window – your sprite should be starting on the left of your tilemap above the ground



Control the movement of your Sprite

- From the **Controller** Toolbox drawer, drag a **Move mySprite** block into the **On Start** block

The image shows the Scratch interface. On the left is the 'Controller' toolbox drawer, which is open to the 'Controller' category. The 'move mySprite with buttons' block is highlighted with a yellow border and a red arrow pointing to its right side. On the right, the 'on start' script area contains several blocks: 'set tilemap to', 'set mySprite to sprite of kind Player', 'set mySprite position to x 5 y 100', and the 'move mySprite with buttons' block, which is also highlighted with a yellow border. The 'move mySprite with buttons' block in the script is connected to the 'set mySprite position to x 5 y 100' block.

Move only left and right

- In the **Move mySprite** block, click the plus (+) icon to expand
- Set the Velocity **vy** value to **0**

The image shows a Scratch script starting with an 'on start' block. It contains four blocks: 'set tilemap to' (with a tilemap icon), 'set mySprite to sprite of kind Player' (with a cat sprite icon), 'set mySprite position to x 5 y 100', and 'move mySprite with buttons vx 100 vy 0'. The 'move' block is expanded, showing a minus sign icon next to the 'vy' field. Two callout boxes point to the 'vx' and 'vy' fields: 'Speed along the horizontal X axis' points to '100', and 'Speed along the vertical Y axis' points to '0'.

```
on start
  set tilemap to [tilemap icon]
  set mySprite to sprite of kind Player
  set mySprite position to x 5 y 100
  move mySprite with buttons vx 100 vy 0
```

Speed along the horizontal X axis

Speed along the vertical Y axis

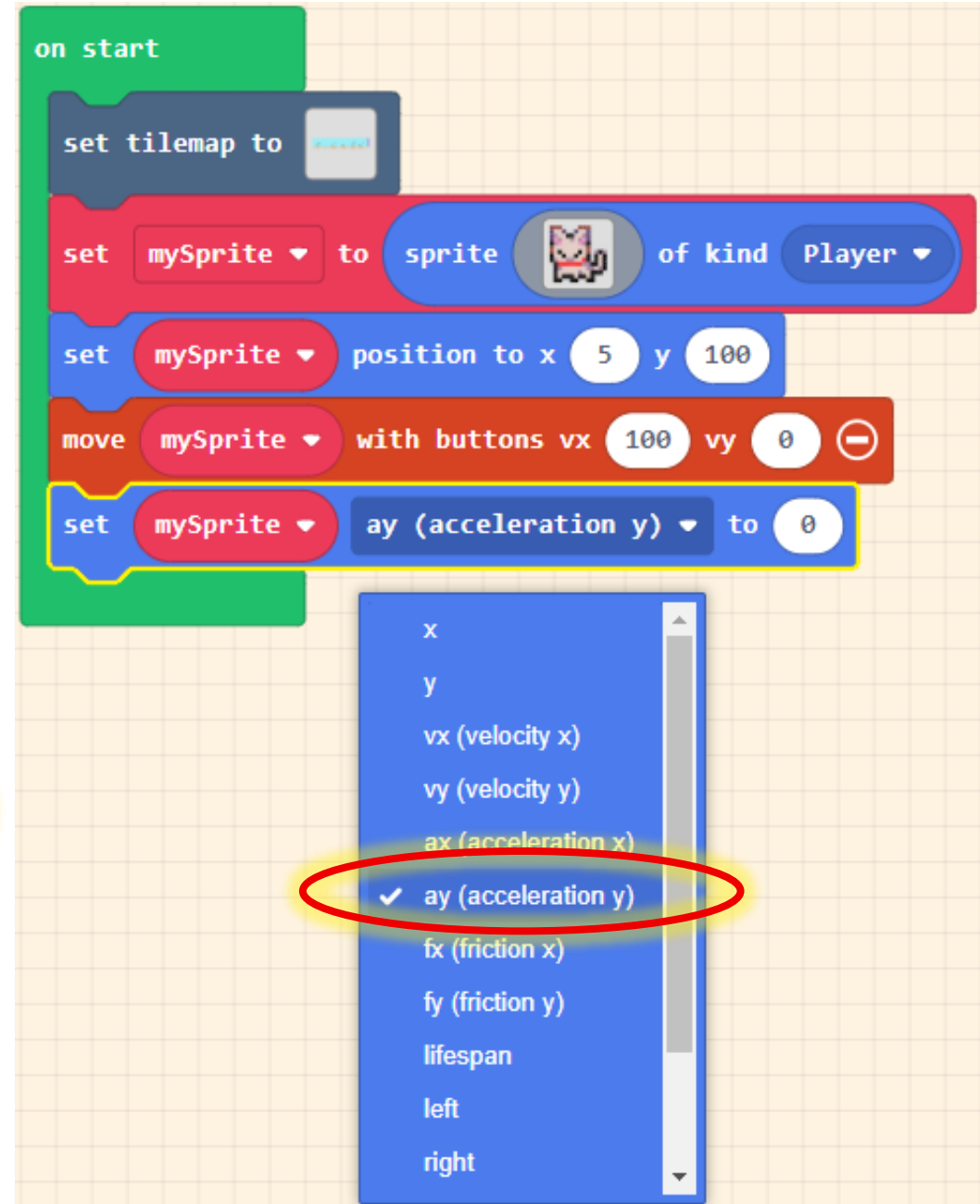
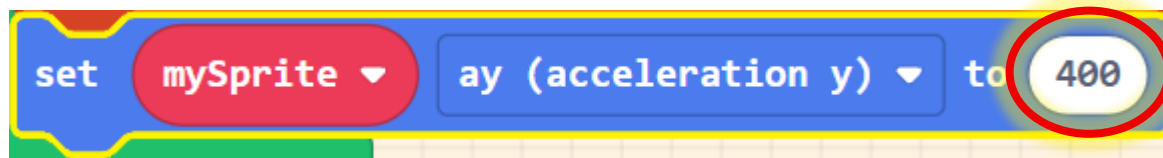
Add Gravity

- From the **Sprites** Toolbox drawer, drag a **Set mySprite** property block into the **On Start** block

The image shows two parts of a game development interface. On the left is the 'Sprites' toolbox drawer, which has a search bar and a list of categories: Sprites, Controller, Game, Music, Scene, Info, Loops, Logic, Variables, Math, and Advanced. Under the 'Sprites' category, there are several blocks. In the 'Create' section, there is a 'set mySprite2 to sprite of kind Player' block. In the 'Physics' section, there are several 'set mySprite' blocks: 'set mySprite velocity to vx 50 vy 50', 'set mySprite position to x 0 y 0', 'set mySprite x to 0', and 'change mySprite x by 0'. A red arrow points from the 'set mySprite x to 0' block in the toolbox to the right. On the right is an 'on start' script block containing several other blocks: 'set tilemap to', 'set mySprite to sprite of kind Player' (with a cat sprite icon), 'set mySprite position to x 5 y 100', 'move mySprite with buttons vx 100 vy 0', and 'set mySprite x to 0' (highlighted with a yellow border).

Add Gravity

- In the **Set mySprite** property block, click on the **x** drop-down menu and select **ay** acceleration along the y axis (in the vertical direction)
- Set the value to be **400**



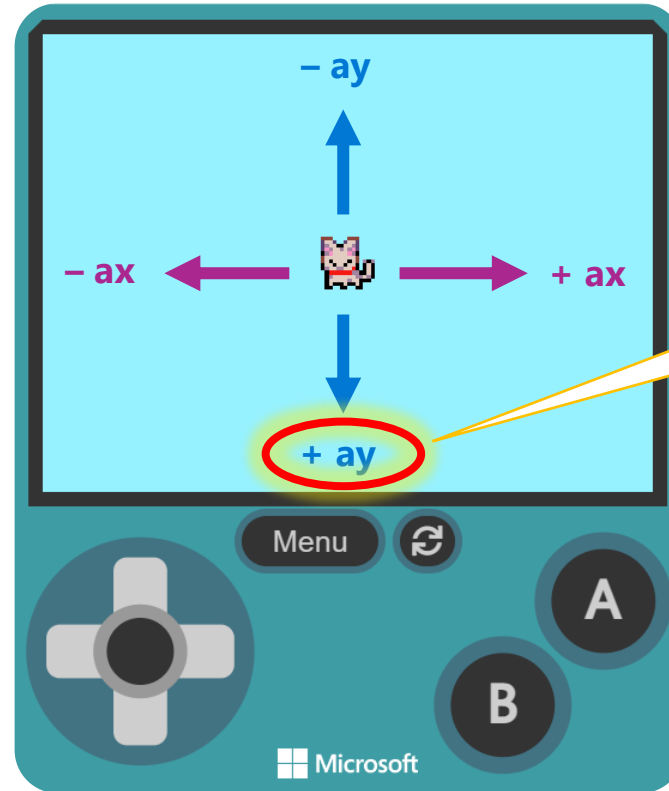
Acceleration = change in speed

ax = horizontal acceleration

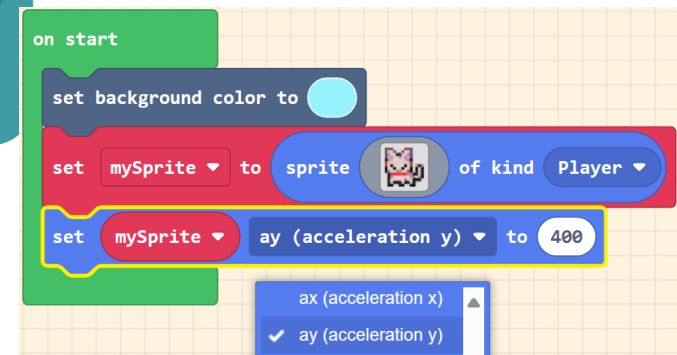
- positive value = left to right
- negative value = right to left

ay = vertical acceleration

- positive value = top to bottom
- negative value = bottom to top



Positive **ay** value simulates gravity pulling sprite down



The value for acceleration determines how quickly the speed of the sprite will change

Camera to follow Sprite

The image shows the Scratch interface with the 'Camera' extension menu open. The menu items are: Info, Loops, Logic, Variables, Math, and Extensions. The 'camera follow sprite mySprite' block is highlighted with a yellow border. A red arrow points from this block to the 'on start' script area on the right.

The image shows the 'on start' script area with the following blocks:

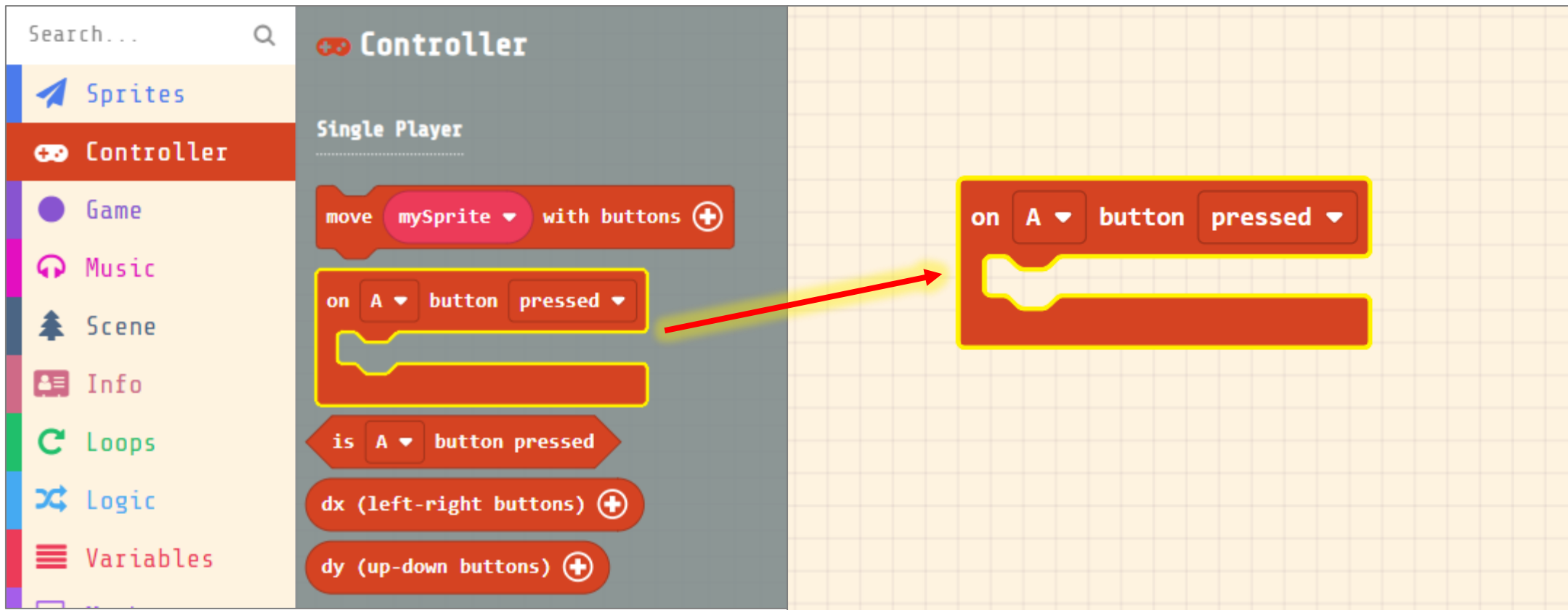
- set tilemap to [tilemap]
- set mySprite to sprite [Player] of kind Player
- set mySprite position to x 5 y 100
- move mySprite with buttons vx 100 vy 0
- set mySprite ay (acceleration y) to 400
- camera follow sprite mySprite

Test your game



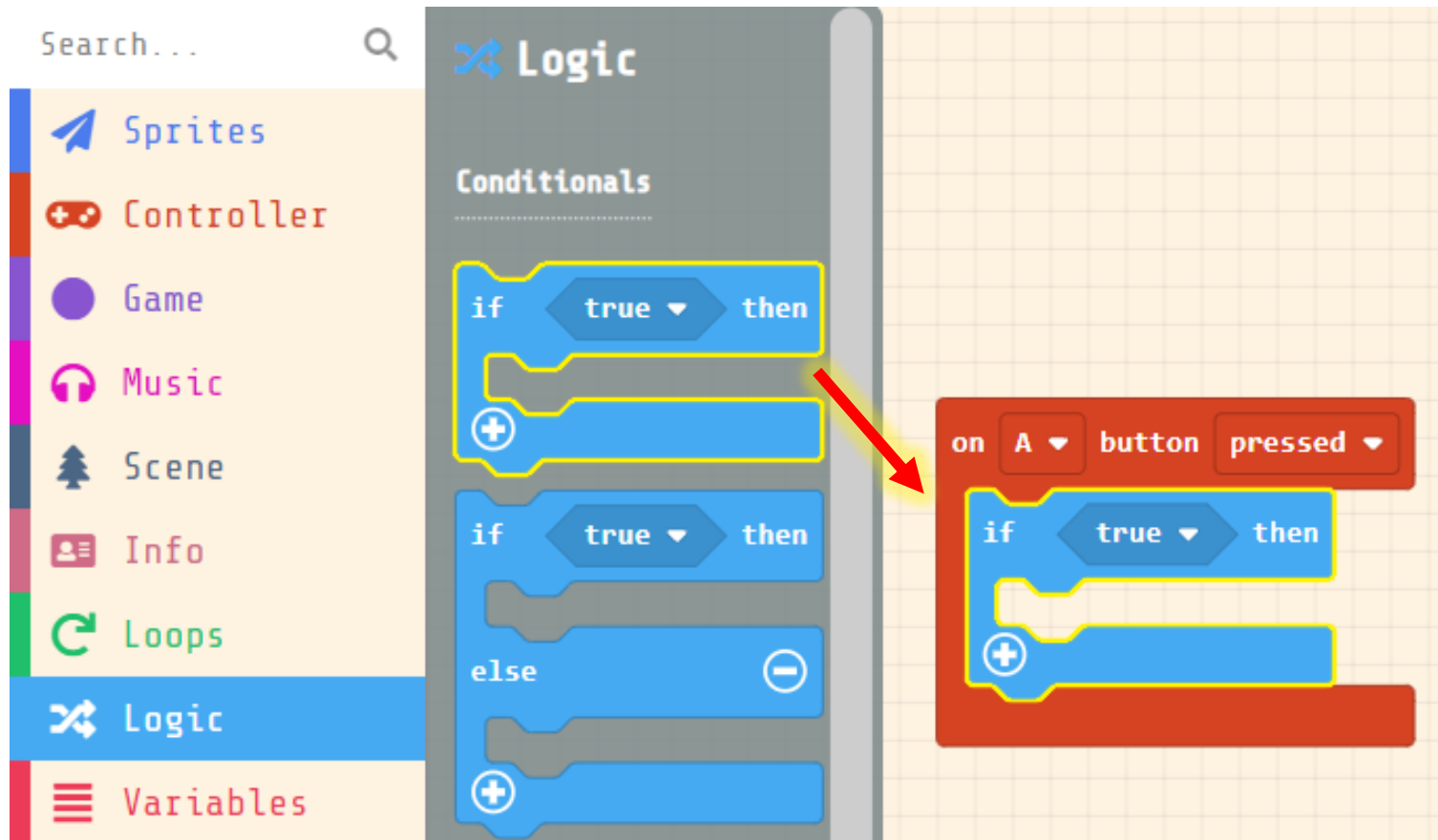
Add a Jump when you press A button

- From the **Controller** Toolbox drawer, drag an **On Button A Pressed** block onto the Workspace



Check to see if you're on the ground

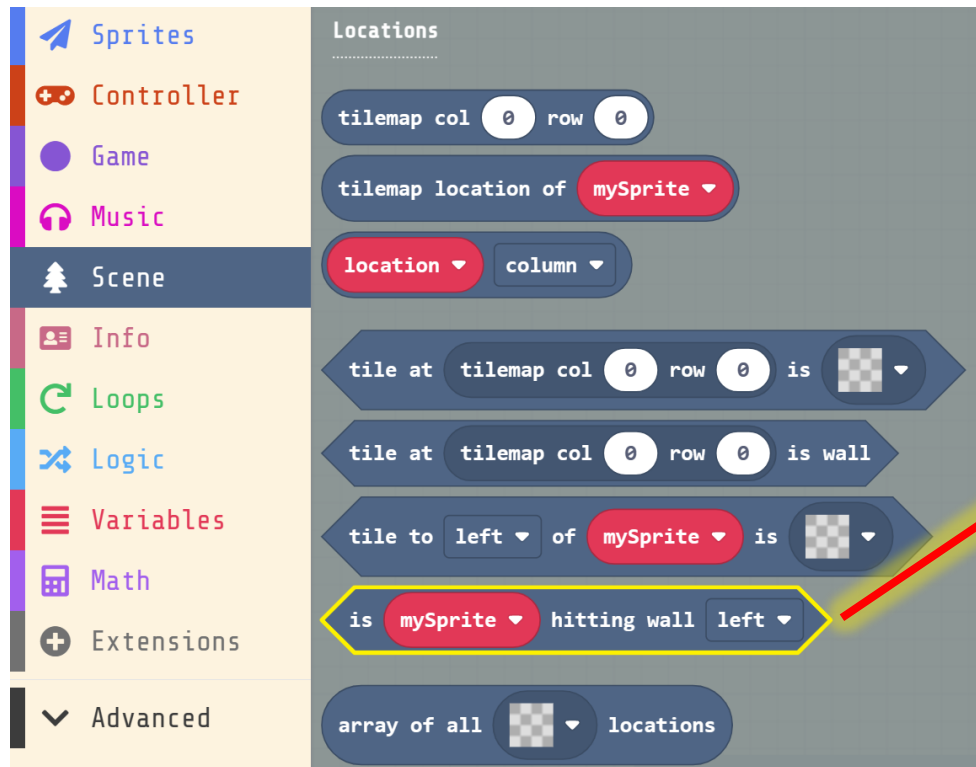
- From the **Logic** Toolbox drawer, drag an **If Then** block into the **On A Button Pressed** block



Note that it's too easy to jump multiple times in mid-air, so let's only allow our player to jump when they are touching the ground or a platform

Check to see if you're on the ground

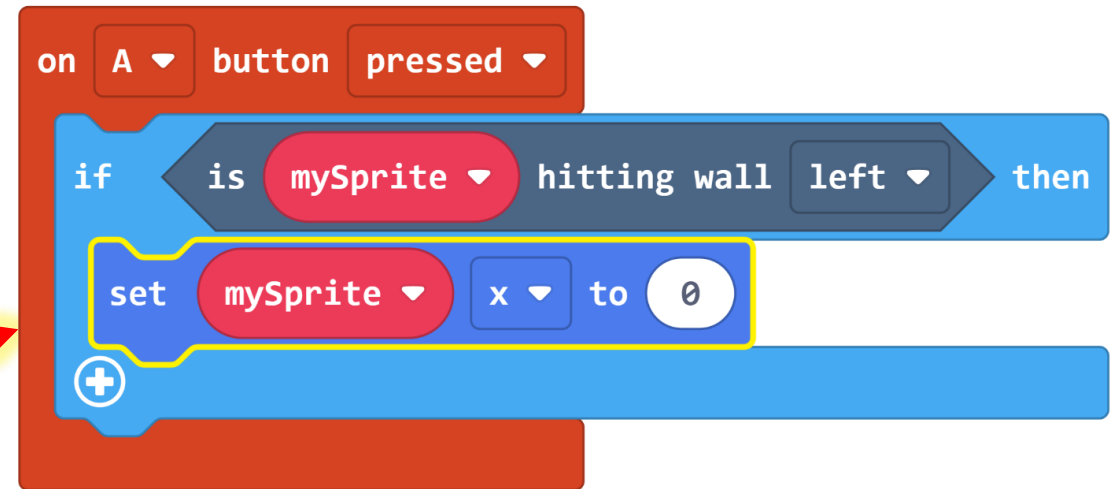
- From the **Scene** Toolbox drawer, drag a **Is mySprite Hitting Wall** block into the **If Then** block, replacing **True**



- In the **Is mySprite Hitting Wall** block, click on left drop-down menu and select **bottom**

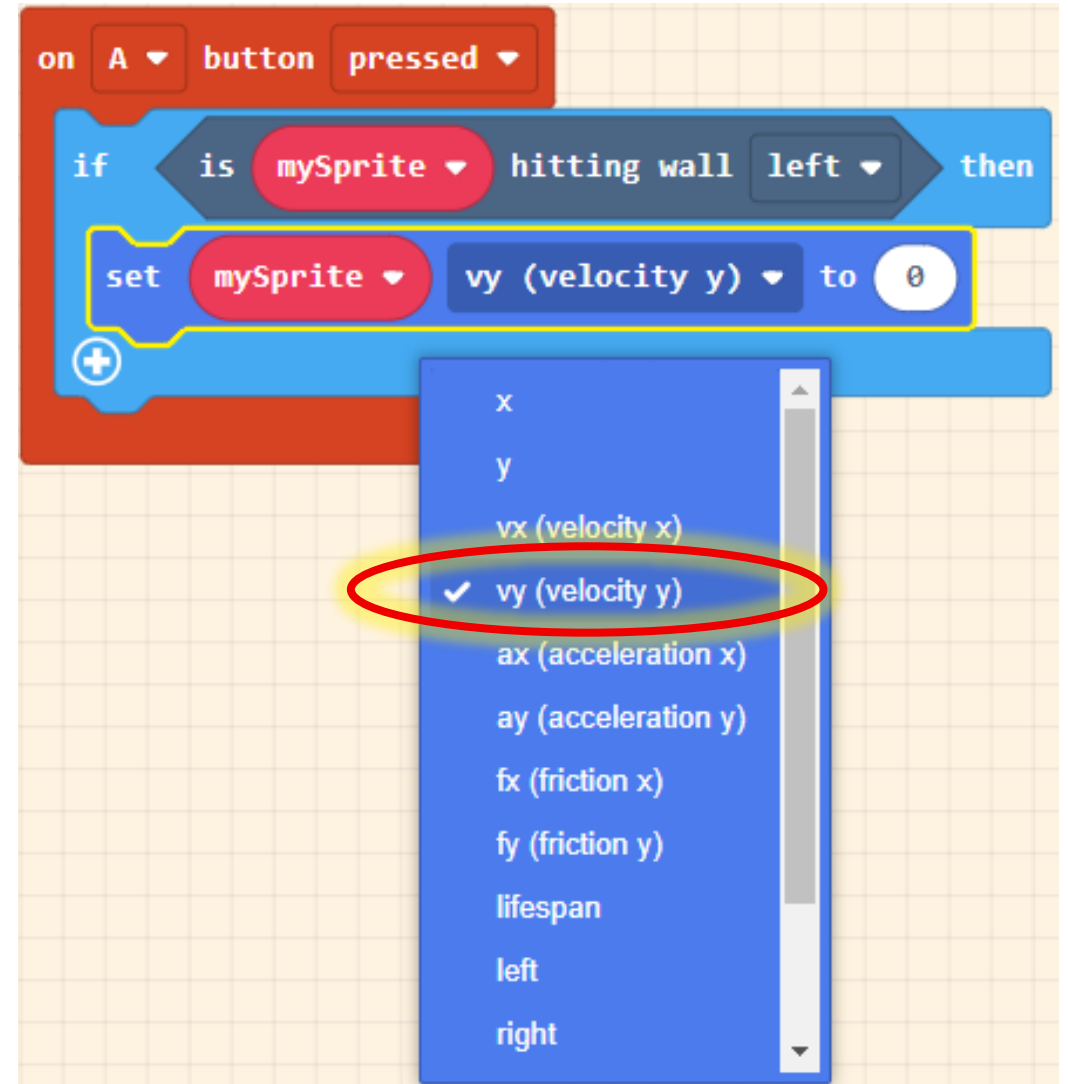
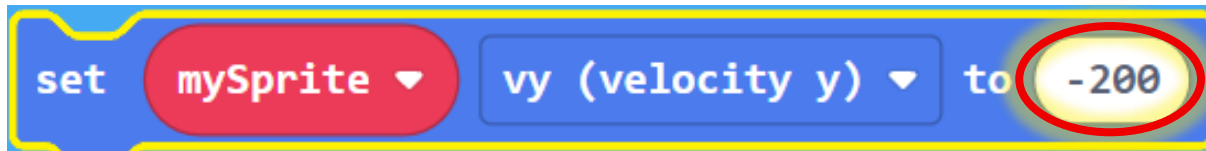
Set jumping velocity!

- From the **Sprites** Toolbox drawer, drag a **Set mySprite** property block into the **If Then** block



Set Velocity

- In the **Set mySprite** property block, click on the **x** drop-down menu and select **vy** – velocity along the y axis (in the vertical direction)
- Set the value to be **-200**



Velocity = speed and direction

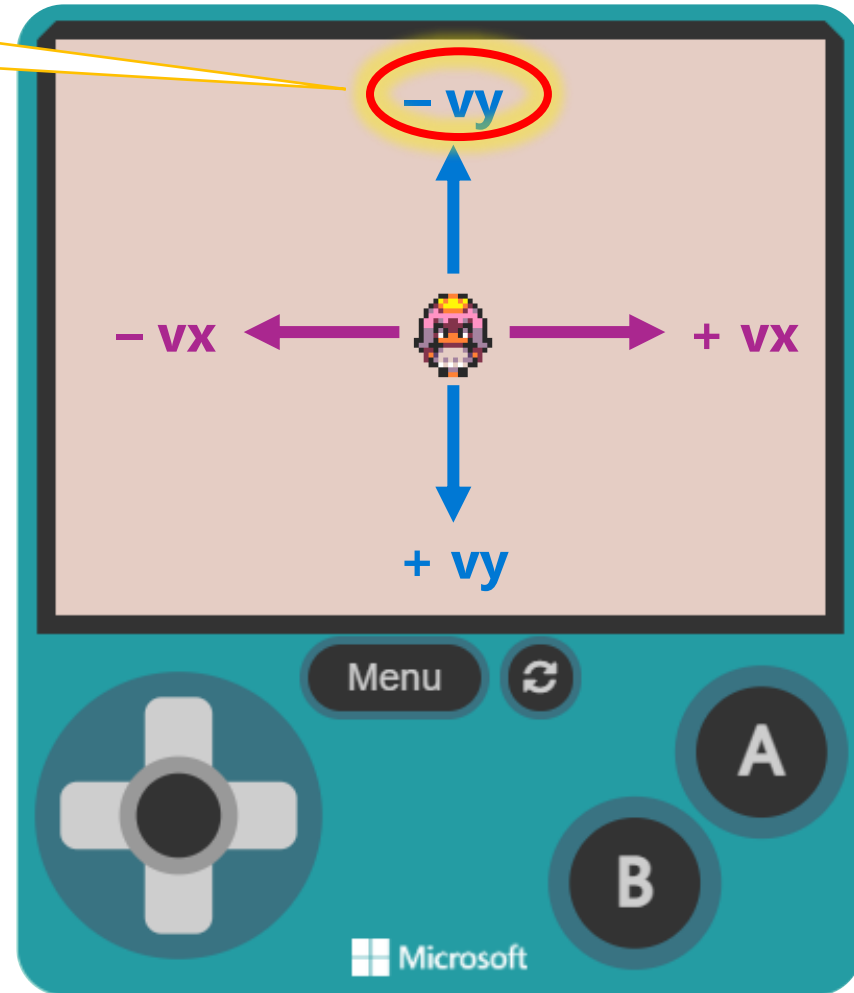
Our jumping velocity is a negative v_y value, so our sprite will jump towards the top of the screen

v_x = horizontal movement

- positive value = left to right
- negative value = right to left

v_y = vertical movement

- positive value = top to bottom
- negative value = bottom to top



Test your game

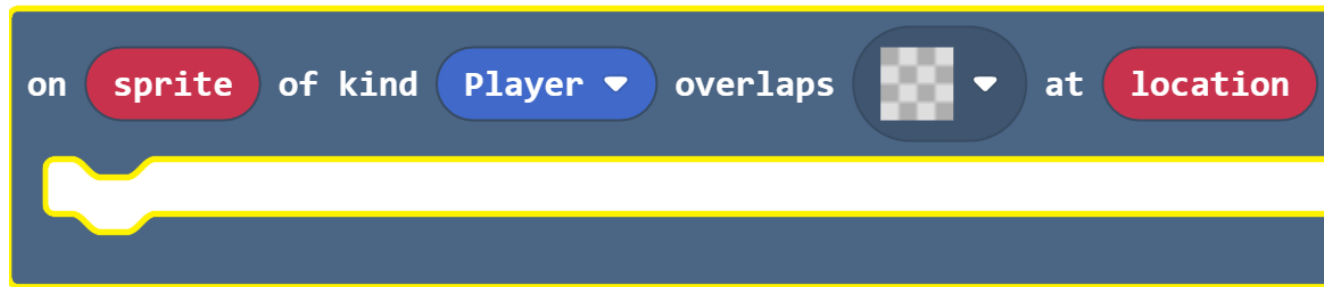


```
on start
  set tilemap to [tilemap]
  set mySprite to sprite [cat] of kind Player
  set mySprite position to x 5 y 100
  move mySprite with buttons vx 100 vy 0
  set mySprite ay (acceleration y) to 400
  camera follow sprite mySprite
```

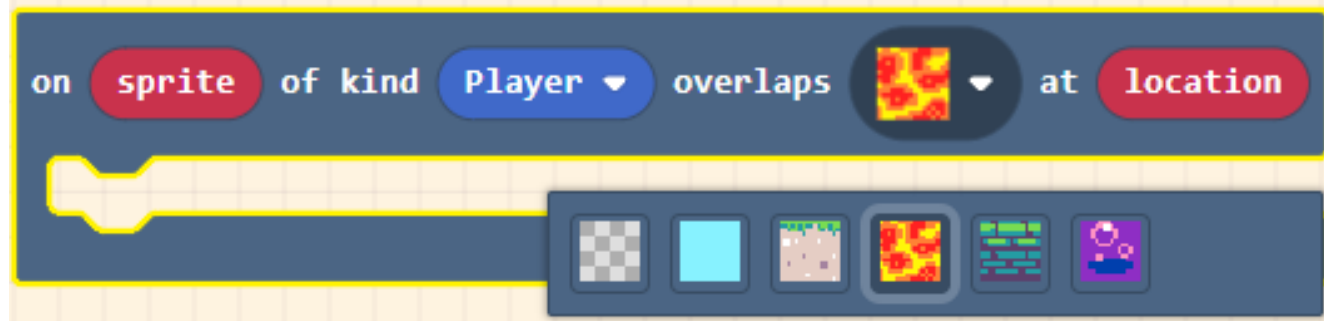
```
on A button pressed
  if is mySprite hitting wall bottom then
    set mySprite vy (velocity y) to -200
```

Lose when you fall into a pit

- From the **Scene** Toolbox drawer, drag a **On Sprite Overlaps Tile** block onto the Workspace

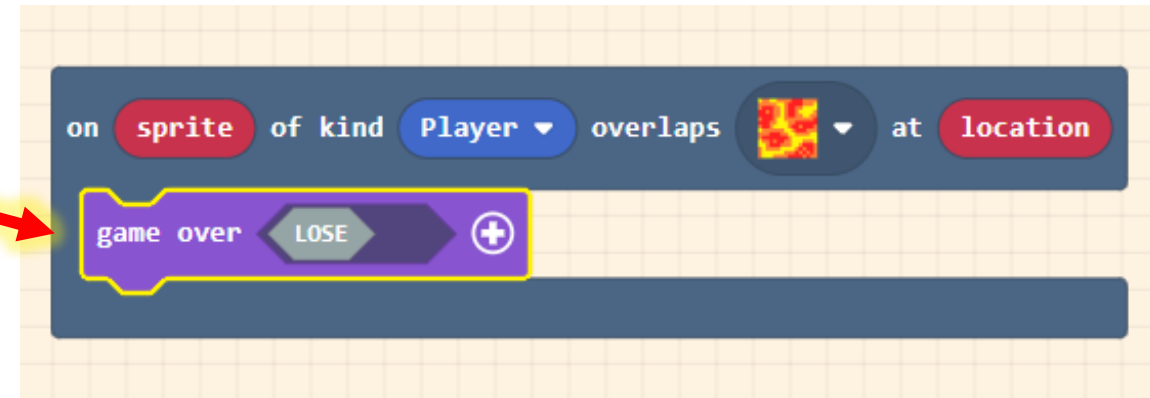


- Click on the checkered Tile drop-down to select your Pit Tile



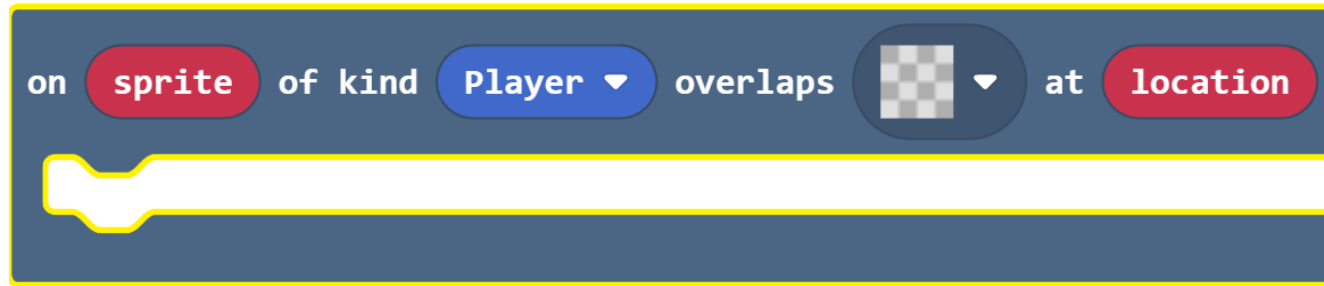
Lose when you fall into a pit

- From the **Game** Toolbox drawer, drag a **Game Over** block onto the **On Sprite Overlaps Tile** block
- Set the value to **LOSE**

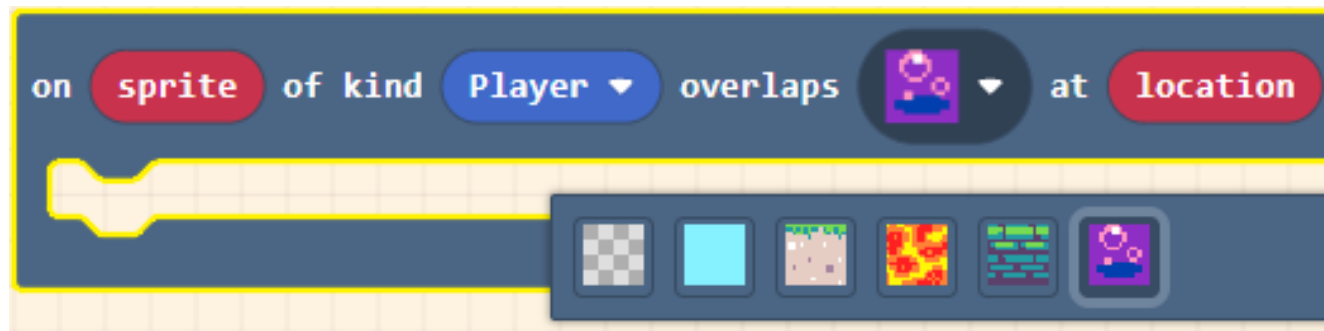


Win when you reach the end

- From the **Scene** Toolbox drawer, drag another **On Sprite Overlaps Tile** block onto the Workspace

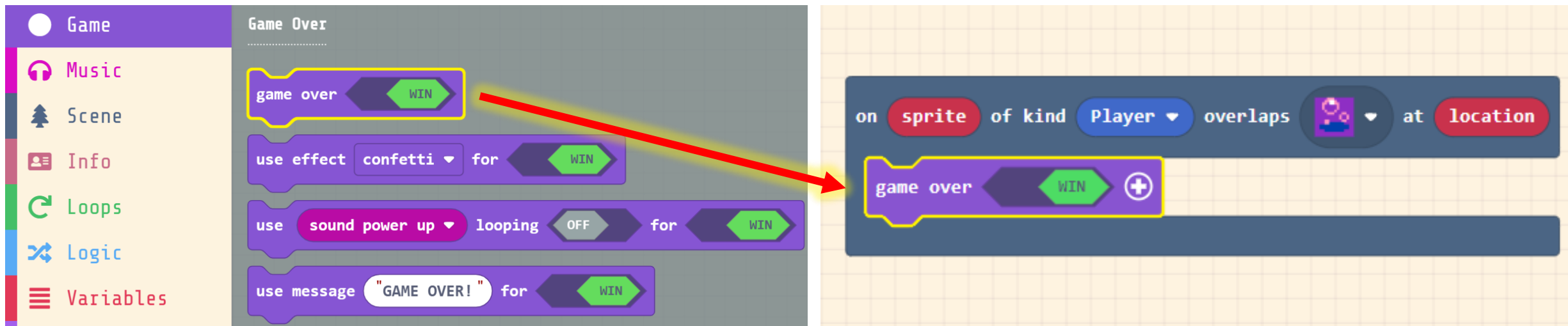


- Click on the checkered Tile drop-down to select your Finish Line Tile



Win when you reach the end

- From the **Game** Toolbox drawer, drag a **Game Over** block onto the **On Sprite Overlaps Tile** block



The image shows two parts of the Scratch interface. On the left is the 'Game' toolbox drawer, which contains several blocks: 'game over', 'use effect confetti', 'use sound power up', and 'use message "GAME OVER!"'. A red arrow points from the 'game over' block in the toolbox to a script block on the right. The script block is an 'on sprite of kind Player overlaps [tile icon] at location' block. The 'game over' block is being dragged into the script block's slot, as indicated by the red arrow and the plus sign on the block.

Complete Code

```
on start
  set tilemap to [tilemap]
  set mySprite to sprite [player] of kind Player
  move mySprite with buttons vx 100 vy 0
  set mySprite ay (acceleration y) to 400
  camera follow sprite mySprite
```

```
on A button pressed
  if is mySprite hitting wall bottom then
    set mySprite vy (velocity y) to -200
```

```
on sprite of kind Player overlaps [obstacle] at location
  game over [LOSE]
```

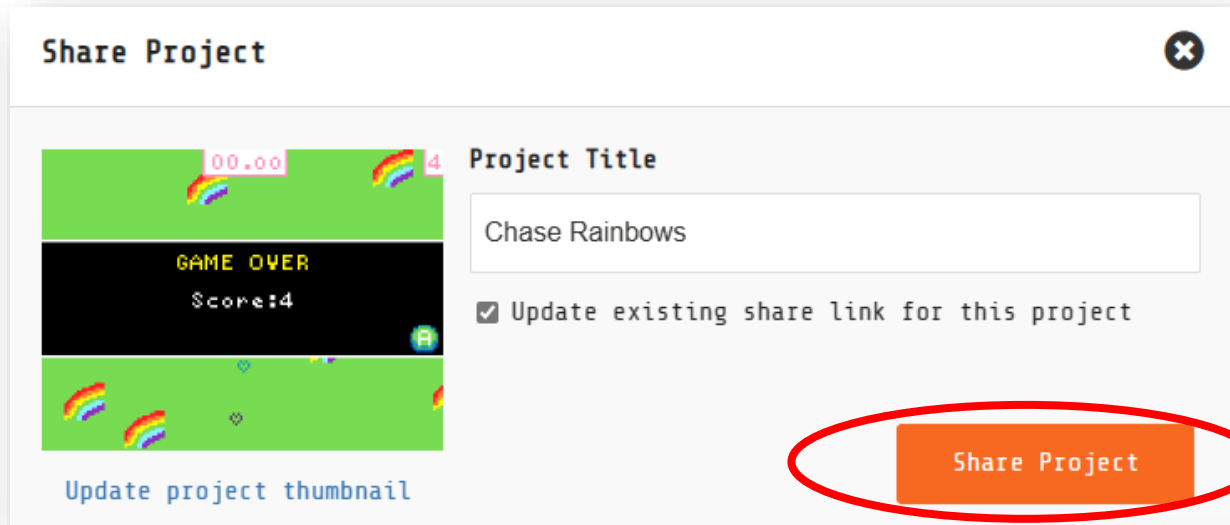
```
on sprite of kind Player overlaps [goal] at location
  game over [WIN]
```

Platformer Mod Ideas

- Add some tiles as objects to collect and gain points
- Add enemies to avoid or battle!
- Add Life & Score
- Add Music and Sound Effects
- Add Animations
- Add multiple Levels

Share your game!

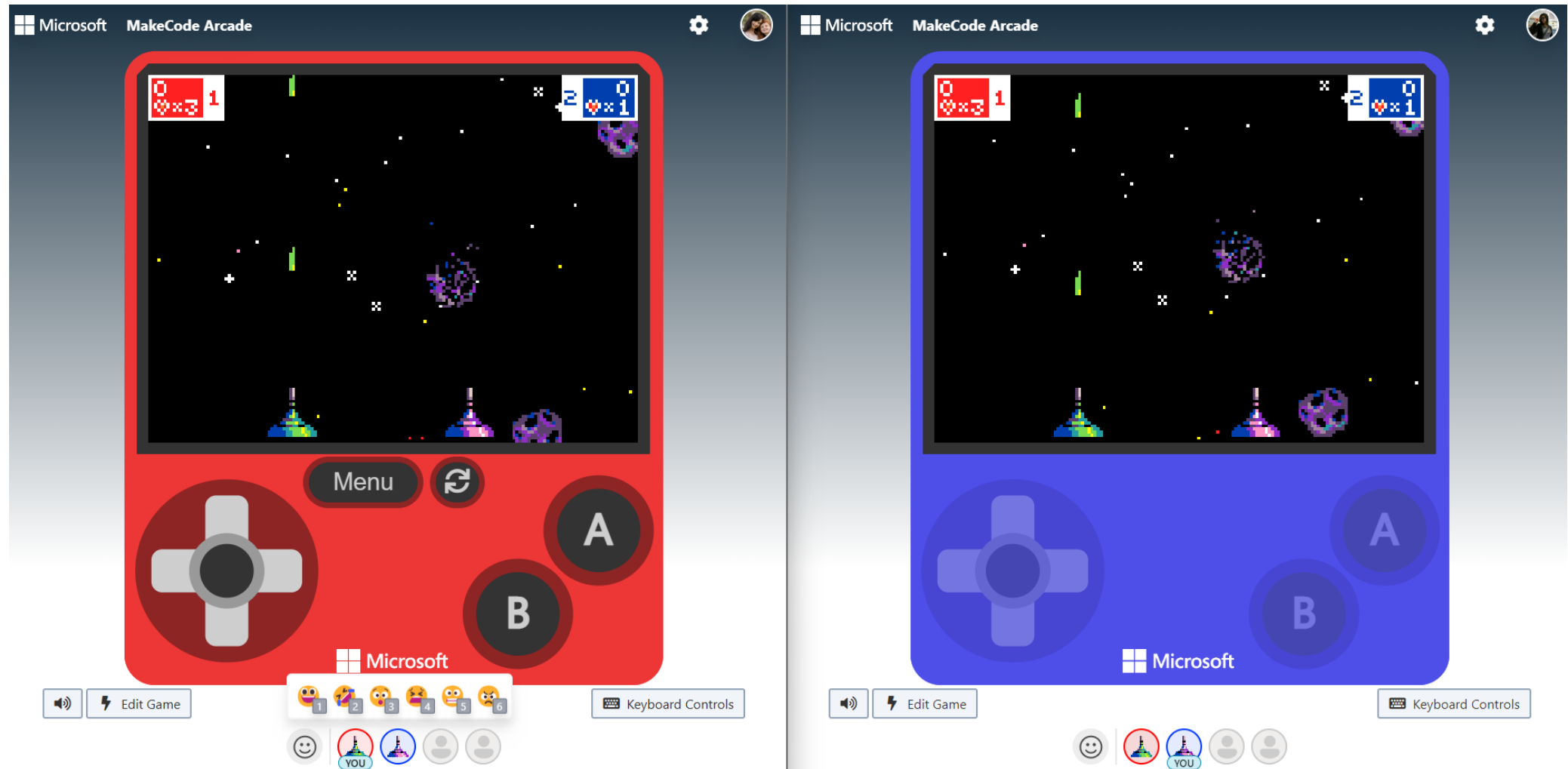
Click the Share button in the top left of the screen



Copy and Paste link
into Chat window



Playing games together



arcade.makecode.com/--multiplayer

Arcade Hardware

Boards

These boards run MakeCode Arcade games. Choose a board to find out more about it and where you can get one!



BrainPad Arcade

Learn how BrainPad Arcade lets you run games on a small handheld console.



Meowbit

A retro game console for STEM education from Kittenbot team.



Adafruit PyBadge

It's a badge, it's an arcade, it's a PyBadge.



Adafruit PyGamer

The upgraded PyBadge.



Kitronik ARCADE

ARCADE is a programmable gamepad for use with MakeCode Arcade.



Ovobot Xtron

A programmable microcomputer that can be used for making MakeCode Arcade games.



Adafruit EdgeBadge

It's the PyBadge with a zest of Machine learning.



Adafruit M4

Learn how to run your games on micro-controllers from Adafruit.



Adafruit Joy Bonnet

Learn how to run your games on Raspberry Pi Zero and Adafruit Joy Bonnet.

arcade.makecode.com/hardware



Arcade Cabinets & Controllers



Cardboard Panel

Turn a cardboard box into a tabletop arcade.



Arcade table

Turn an IKEA FLISAT table into an arcade.

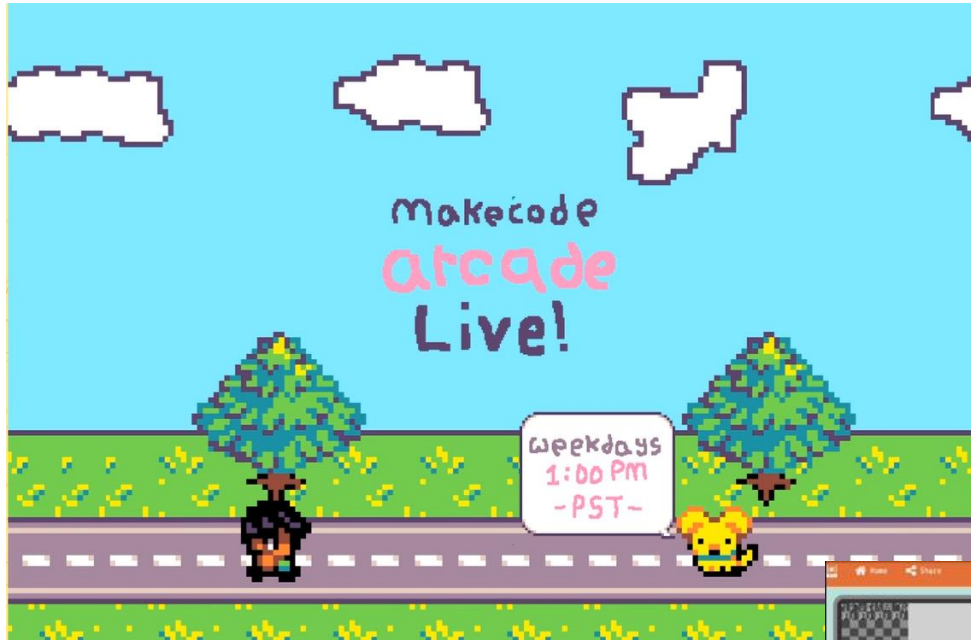


Arcade Kiosk Mode



arcade.makecode.com/hardware/kiosk

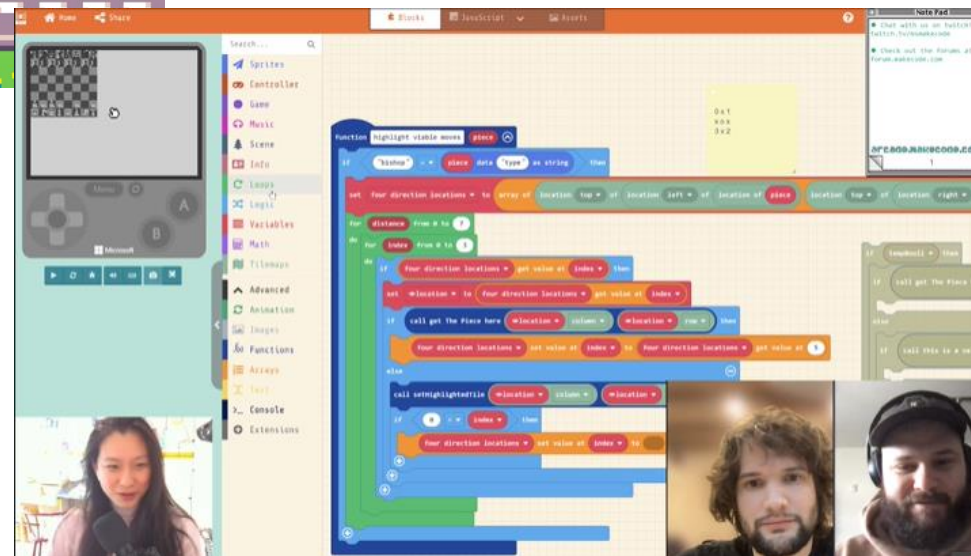
MakeCode Arcade Live Stream & Forum



1pm Pacific / 3pm Eastern MWF

twitch.tv/msmakecode

forum.makecode.com



MakeCode
Engineers

Arts & Crafts



<https://arcade.makecode.com/arts-and-crafts>



<https://arcade-stencils.glitch.me>



Thank You!

```
on start
  set cherry to sprite of kind Food
  set cherry position to x pick random 0 to 160 y pick random 0 to 120
```

```
let cherry = sprites.create(img'', SpriteKind.Food)
cherry.setPosition(Math.randomRange(0, 160), Math.randomRange(0, 120))
```

