

```
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](https://makecode.com)

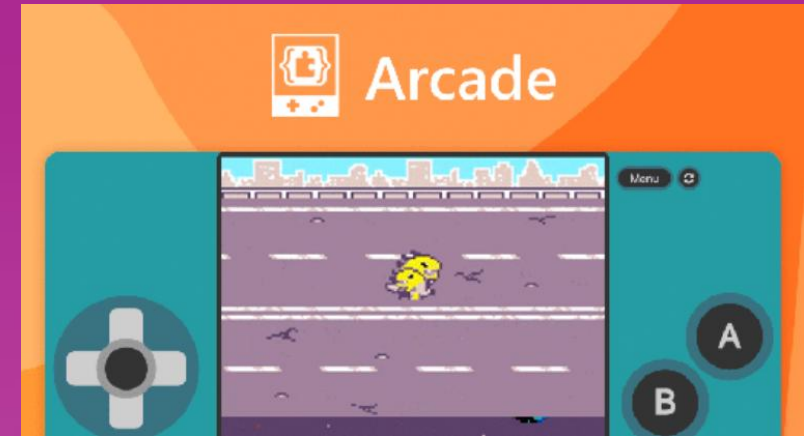
# Three Main Code Editors



Physical Computing with micro:bit

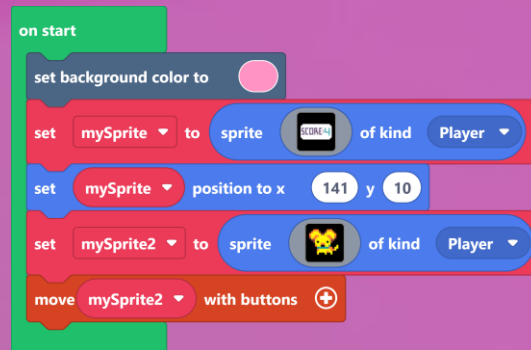


Mods in Minecraft

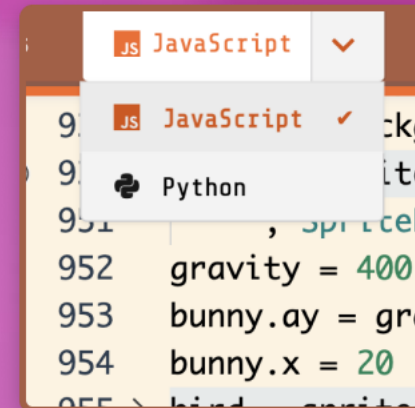


Retro Arcade Games

Blocks



Text



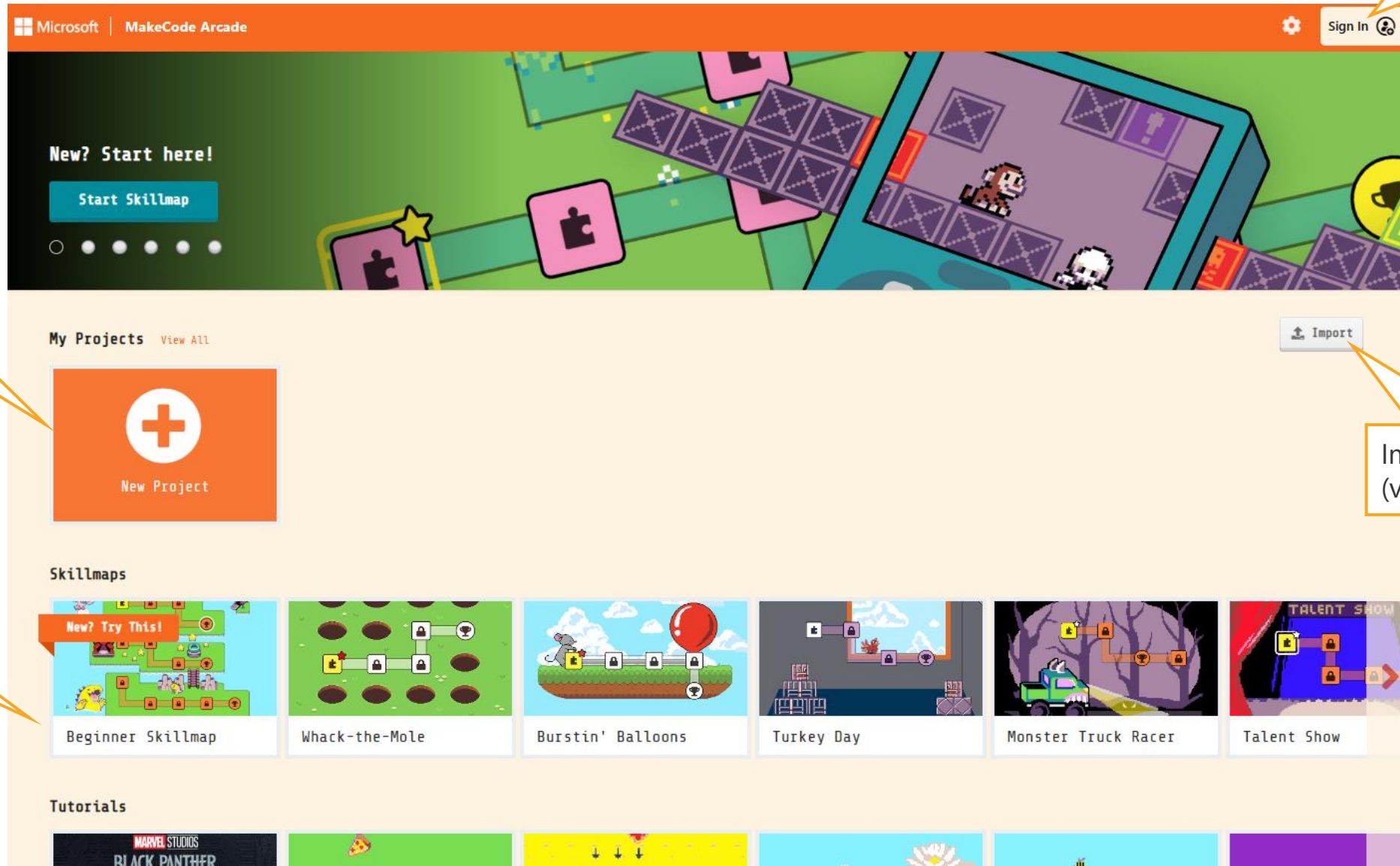
# Let's code a game!

Collect the bones but avoid the cars!



# Open Browser: [arcade.makecode.com](https://arcade.makecode.com)

Optionally sign in to save to projects cloud



Create a New blank Project

Import Projects (via a File, or URL)

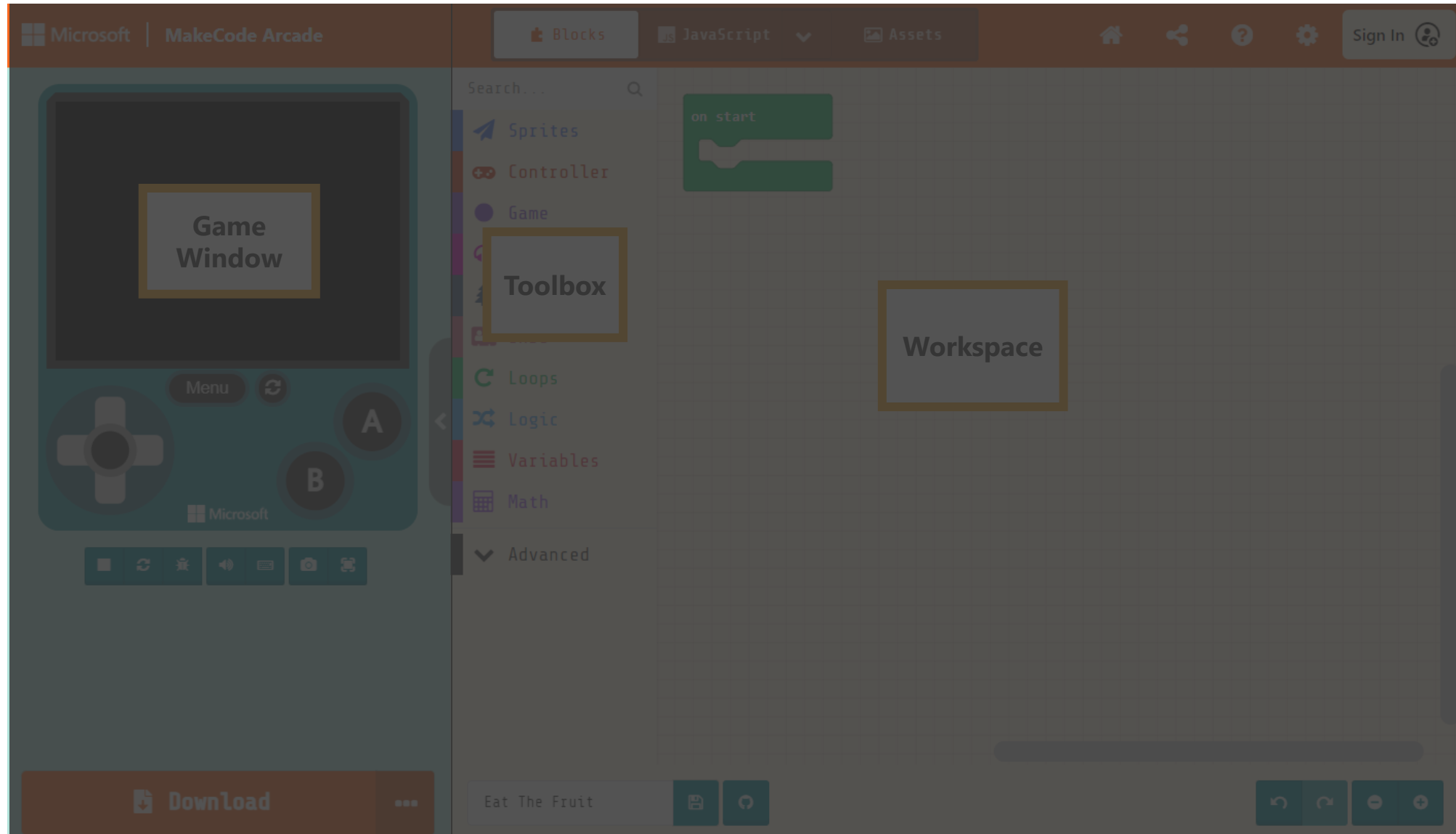
Step-by-step Skillmaps and Tutorials

# 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 icon and a 'Sign In' button. Below the navigation bar is a large banner with the text 'New? Start here!' and a 'Start Skillmap' button. The main content area is divided into sections: 'My Projects' (with a 'View All' link and an 'Import' button), 'Skillmaps', and 'Tutorials'. The 'My Projects' section features a prominent orange square button with a white plus sign and the text 'New Project', which is circled in purple. A modal dialog box is open over this button, titled 'Your project needs a name' with two smiley face emojis and a close button. The dialog contains the text 'Give your project a name.' and a text input field with the value 'Crossy road'. A green 'Create' button with a checkmark is at the bottom right of the dialog. The 'Skillmaps' section includes a 'New? Try This!' banner and several project thumbnails: 'Beginner Skillmap', 'Whack-the-Mole', 'Burstin' Balloons', 'Turkey Day', 'Monster Truck Racer', and 'Talent Show'. The 'Tutorials' section at the bottom shows a 'MARVEL STUDIOS BLACK PANTHER' tutorial and other colorful thumbnails.

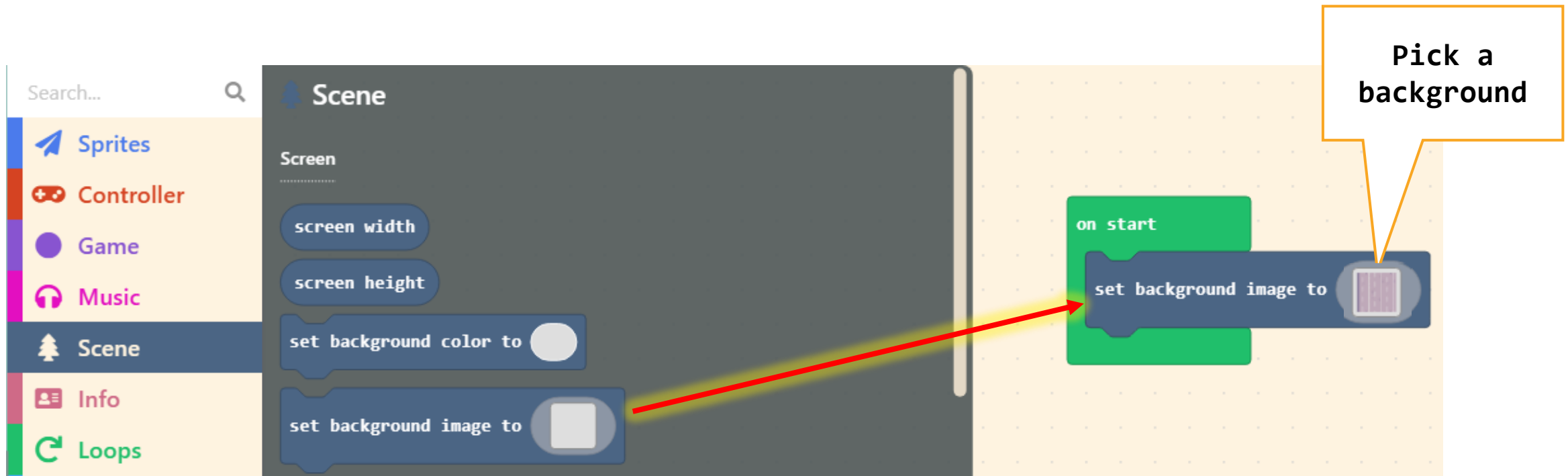
Click on  
New Project →

# Getting familiar with MakeCode Arcade



# Set the background

- From the **Scene** Toolbox drawer, drag a **Set Background image** onto the Workspace
- Drop into the **On Start** block



# Create a Player Sprite

- From the **Sprites** Toolbox drawer, drag a **Set** sprite block onto the Workspace
- Drop into the **On Start** block after the **Set Background** block

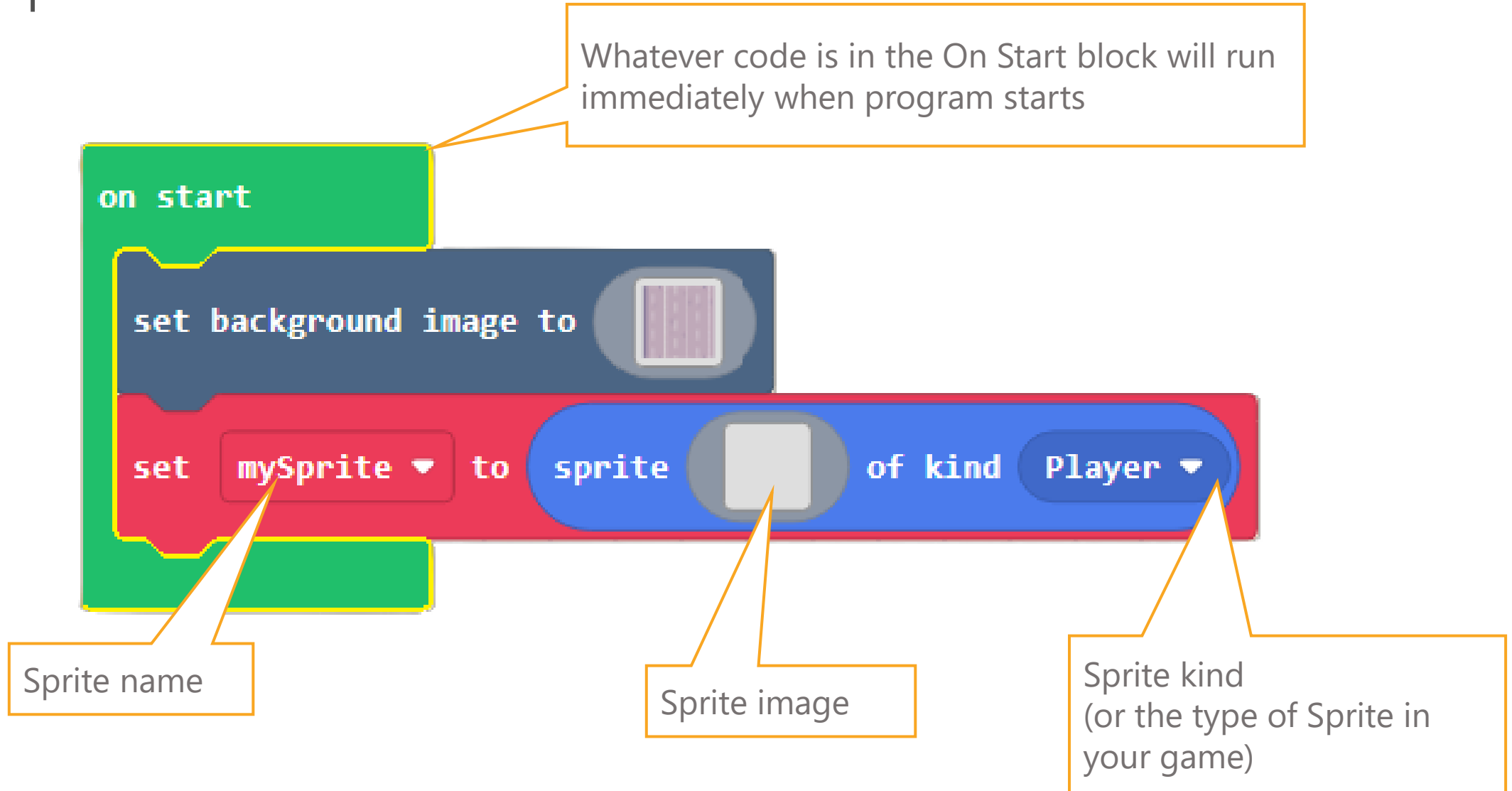


# Sprites

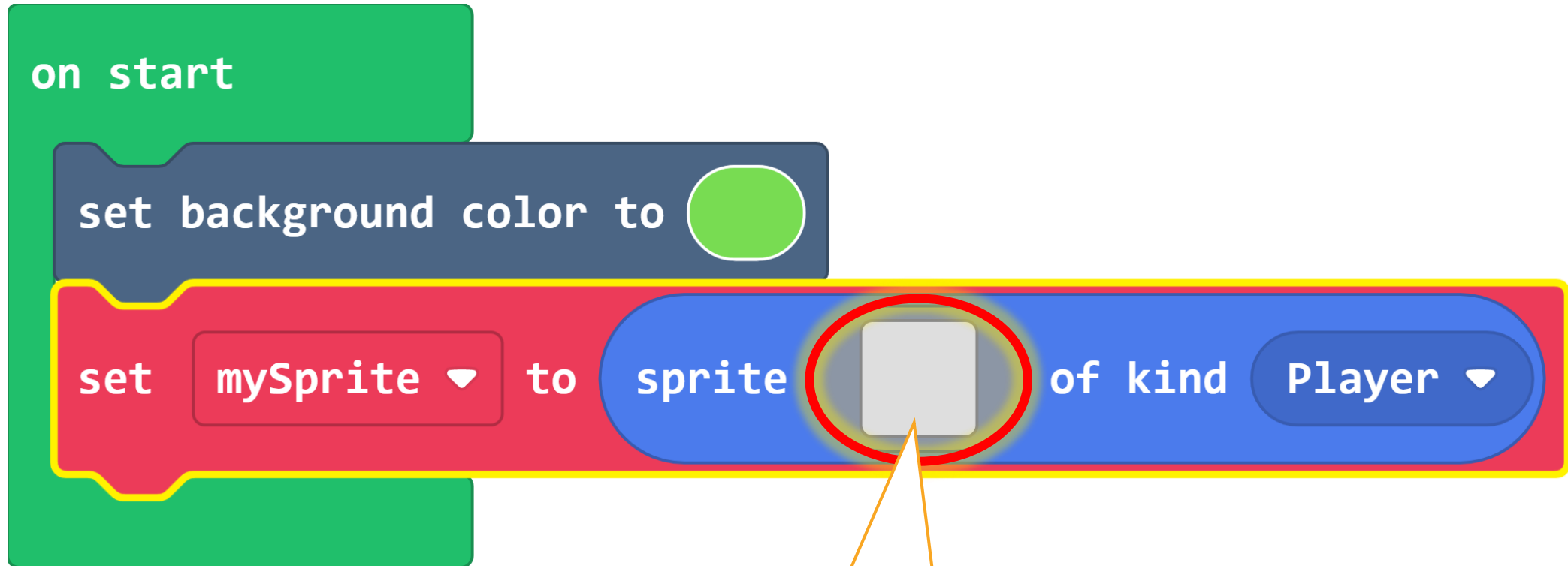


any object in a game that has properties and behaviors

# Set Sprite block



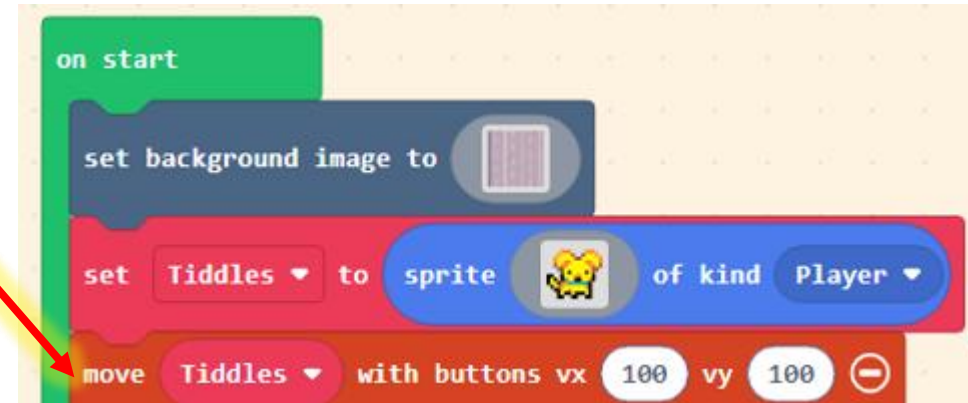
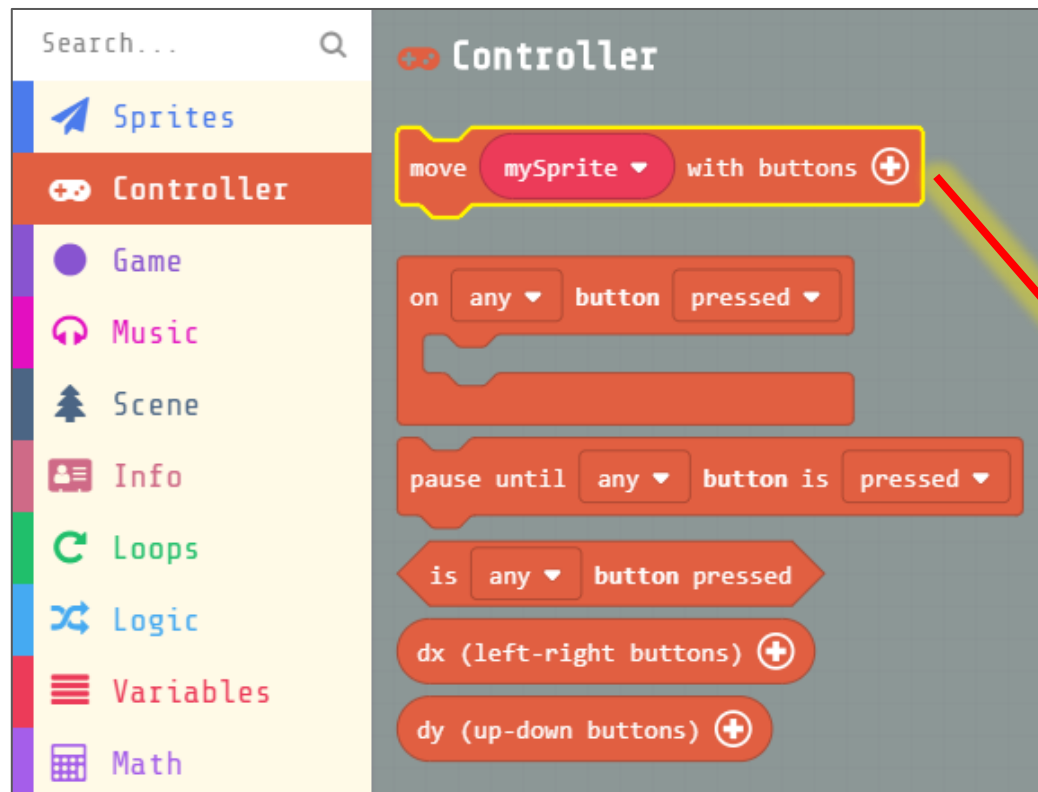
# Set Sprite block



Click to draw Image of your main character

# Control the movement of your Sprite

- From the **Controller** Toolbox drawer, drag a **Move** block onto the Workspace
- Drop into the **On Start** block after the **Set** sprite block



# Try it out in Simulator

- Click on the joystick buttons in the Simulator to move your Sprite around the screen
- Or use the arrow keys on the keyboard (make sure the mouse is hovered over the Game window to activate controls)



# Keep Sprite in Screen

The image shows the Scratch interface with two panels: 'Effects' and 'Scripts'.

**Effects Panel:**

- mySprite start spray effect
- clear effects on mySprite
- destroy mySprite
- mySprite say ":"
- set mySprite stay in screen ON (highlighted with a yellow box)
- set mySprite bounce on wall ON
- set mySprite auto destroy OFF

**Scripts Panel:**

- on start
  - set background image to [background image]
  - set hero to sprite [hero sprite] of kind Player
  - move hero with buttons
  - set hero stay in screen ON (highlighted with a red box and a red arrow pointing from the 'set mySprite stay in screen ON' block in the Effects panel)

# Decide where tiddles starts

The image shows a game engine's code editor with a sidebar on the left and a main workspace on the right. The sidebar contains a search bar and a list of categories: Sprites, Controller, Game, Music, Scene, Info, Loops, Logic, Variables, and Math. The main workspace is divided into two panels. The left panel, titled 'Sprites', has a 'Create' section with a red-bordered block containing 'set mySprite to sprite of kind Player' and a blue-bordered block containing 'sprite of kind Player'. Below this is a 'Physics' section with a blue-bordered block 'set mySprite velocity to vx 50 vy 50' and a yellow-bordered block 'set mySprite position to x 0 y 0'. The right panel, titled 'on start', contains a green 'on start' block followed by several other blocks: 'set background image to', a red-bordered 'set Tiddles to sprite of kind Player' block, a red 'move Tiddles with buttons vx 100 vy 100' block, a blue 'set Tiddles position to x 5 y 55' block, and a blue 'set Tiddles stay in screen ON' block. A red arrow points from the '0' in the 'set mySprite position to x 0 y 0' block to the '5' in the 'set Tiddles position to x 5 y 55' block.

# Add countdown and lives

The image shows the Scratch code editor interface. On the left is a sidebar with a search bar and a list of categories: Sprites, Controller, Game, Music, Scene, Info, Loops, Logic, Variables, Math, and Animation. The 'Info' category is selected, showing a list of blocks including 'set life to 3', 'change life by -1', 'on life zero', and a 'Countdown' section with 'countdown', 'start countdown 10 (s)', 'change countdown by 0 (s)', and 'stop countdown'.

On the right, the script area contains an 'on start' block with the following code:

- set background image to [background image]
- set Tiddles to sprite [Tiddles] of kind Player
- move Tiddles with buttons vx 100 vy 100
- set Tiddles position to x 5 y 55
- set Tiddles stay in screen ON
- start countdown 15 (s)
- set life to 3

Two red arrows indicate the flow of code: one from the 'set life to 3' block in the palette to the 'set life to 3' block in the script area, and another from the 'start countdown 10 (s)' block in the palette to the 'start countdown 15 (s)' block in the script area.

# Create a flying car Sprite every 1 second

- From the **Game** Toolbox drawer, drag a **On Game Update Every** block onto the Workspace (put this anywhere)

The image shows the Scratch Game toolbox on the left and the workspace on the right. The toolbox has a search bar and categories: Sprites, Controller, Game, Music, Scene, Info, Loops, Logic, Variables, and Math. The Game category is selected, showing blocks like 'on game update', 'on game update every 500 ms', 'time since start (ms)', and 'reset game'. The 'on game update every 500 ms' block is highlighted with a yellow border and a red arrow pointing to it in the workspace. A callout box points to the block with the text: 'Code in this block will execute on a given time interval (500ms = 1/2 second)'. The workspace also shows a 'start' block with sub-blocks: 'set background to...', 'set music to...', 'move...', 'set...', 'set scene to...', and 'start'.

# Create a enemy flying car Sprite

- From the **Sprites** Toolbox drawer, drag a **Set Projectile from side** block into the **On Game Update Every** block



# Velocity = speed and direction

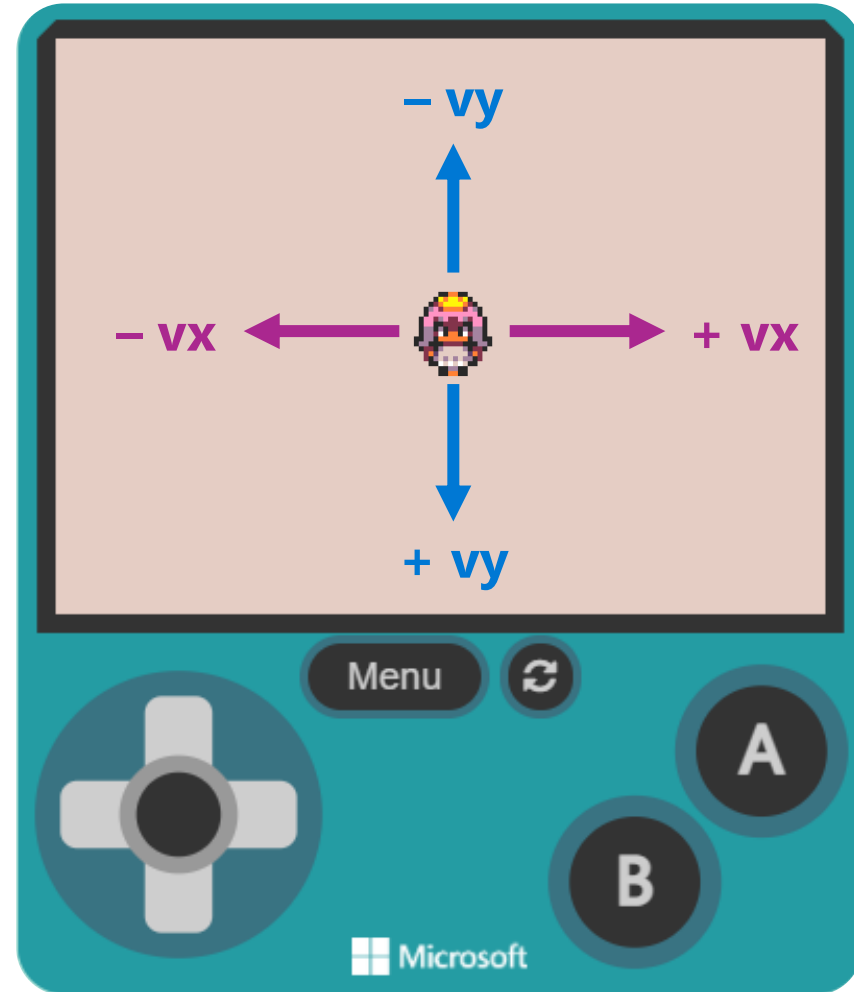
**vx** = horizontal movement

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

**vy** = vertical movement

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

from side with **vx** 50 **vy** 0



# Set the car starting position

- From the **Sprites** Toolbox drawer, drag a **Set Position** block into the **On Game Update Every** block
- Change mySprite variable to **projectile**

The image shows a game engine 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 Extensions. The 'Sprites' category is selected, and a 'Set Position' block is highlighted. The script editor shows an 'on game update every' block with a duration of 500 ms. Inside this block, there are two 'Set Position' blocks. The first block is set to 'projectile' and has a velocity of vx 50 and vy 0. The second block is also set to 'projectile' and has a position of x 0 and y 0. A dropdown menu is open for the second 'Set Position' block, showing a list of variables: mySprite, projectile (checked), New variable..., Rename variable..., and Delete the "projectile" variable. A red arrow points from the 'Set Position' block in the toolbox to the 'Set Position' block in the script editor.

# Coordinates

The Arcade game screen dimensions are:

**160** pixels wide x **120** pixels high

x is horizontal

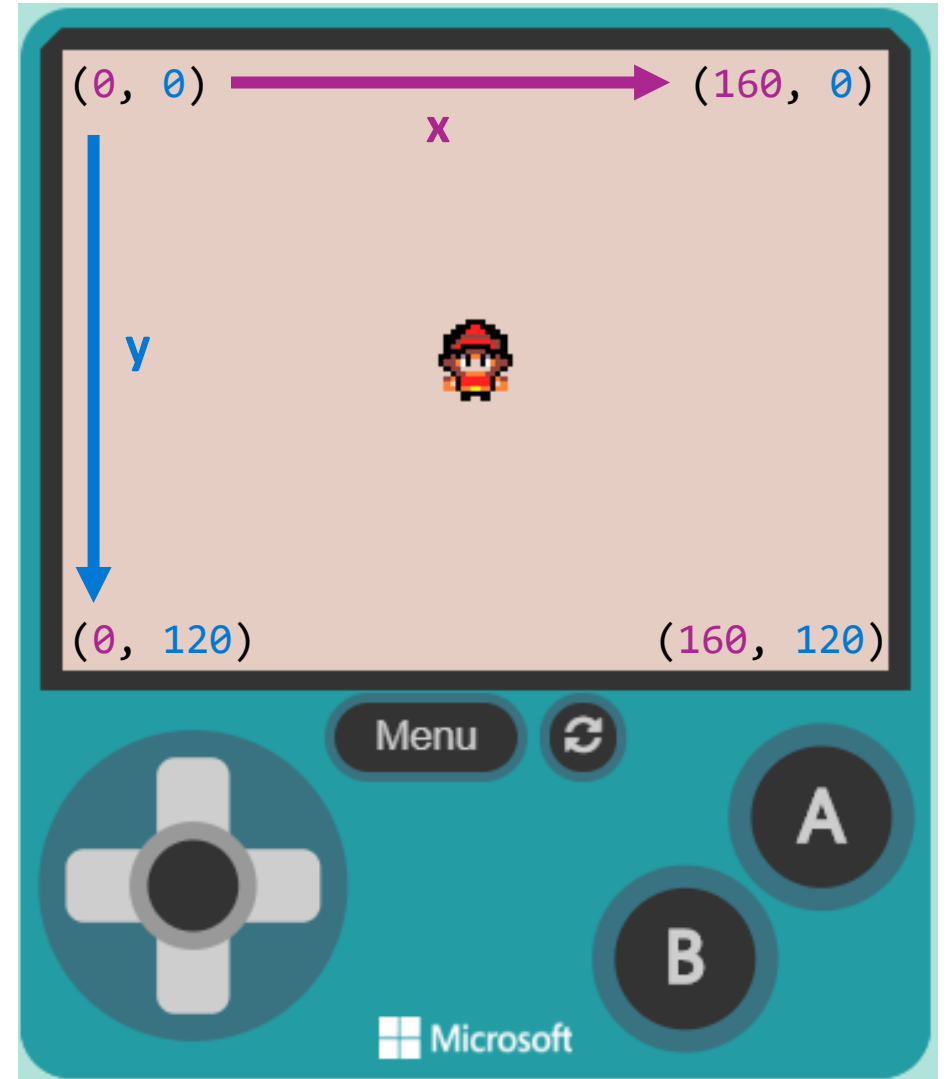
y is vertical

position to x

0

y

0



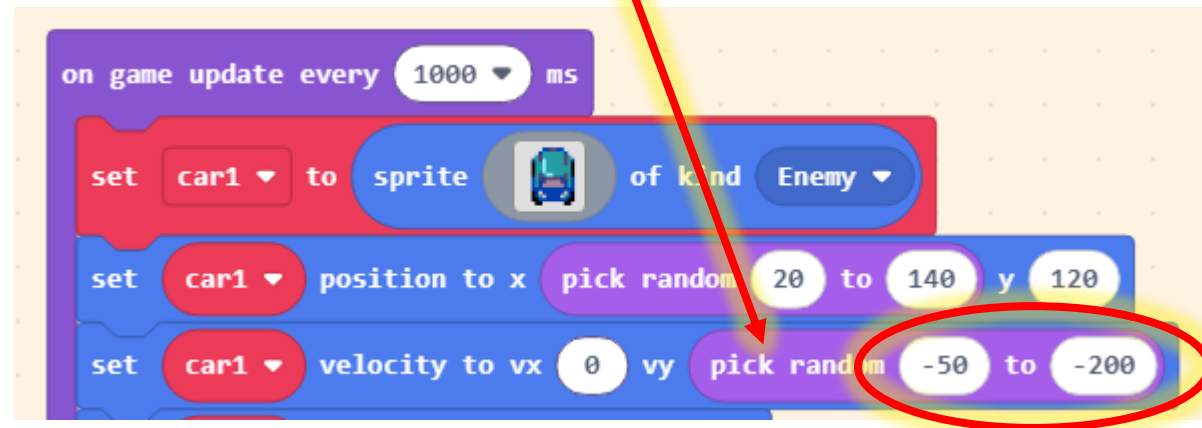
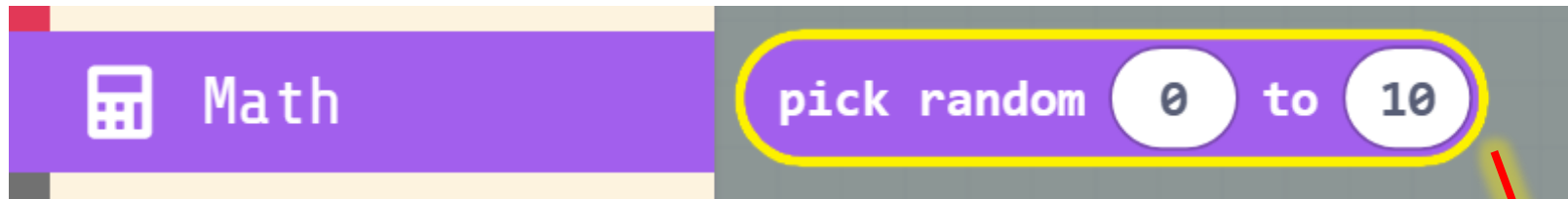
# Start from the bottom at random places

- From the **Math** Toolbox drawer, drag a **Pick Random** block, and drop into the **x** field of the **Set Position** block
- In Pick Random block, change values from 20 to 140 value to **120**



# Make the speeds random

- From the **Math** Toolbox drawer, drag a **Pick Random** block, and drop into the **y** field of the **Set Position** block



# Add a bone to collect

Search...

**Sprites**

- Controller
- Game
- Music
- Scene
- Info
- Loops
- Logic
- Variables

**Sprites**

Create

on start

set mySprite to sprite of kind Player

sprite of kind Player

Physics

set mySprite velocity to vx 50 vy 50

set mySprite position to x 0 y 0

on start

set background image to

set Tiddles to sprite of kind Player

move Tiddles with buttons vx 100 vy 100

set Tiddles position to x 5 y 55

set Tiddles stay in screen ON

set bone to sprite of kind Food

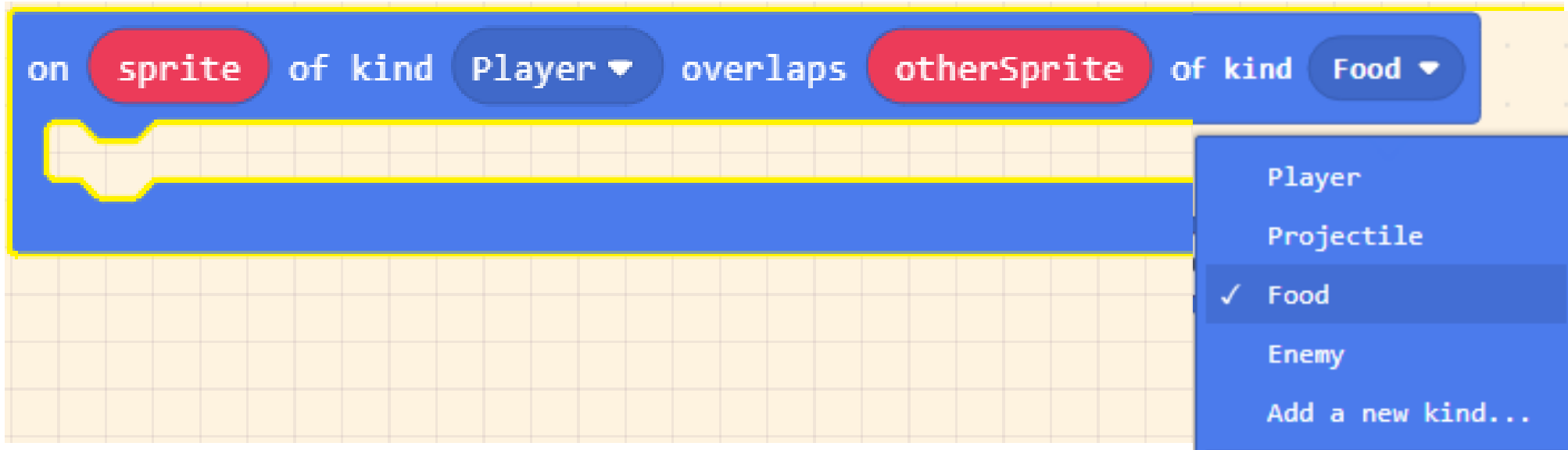
set bone position to x 150 y pick random 5 to 100

start countdown 15 (s)

set life to 3

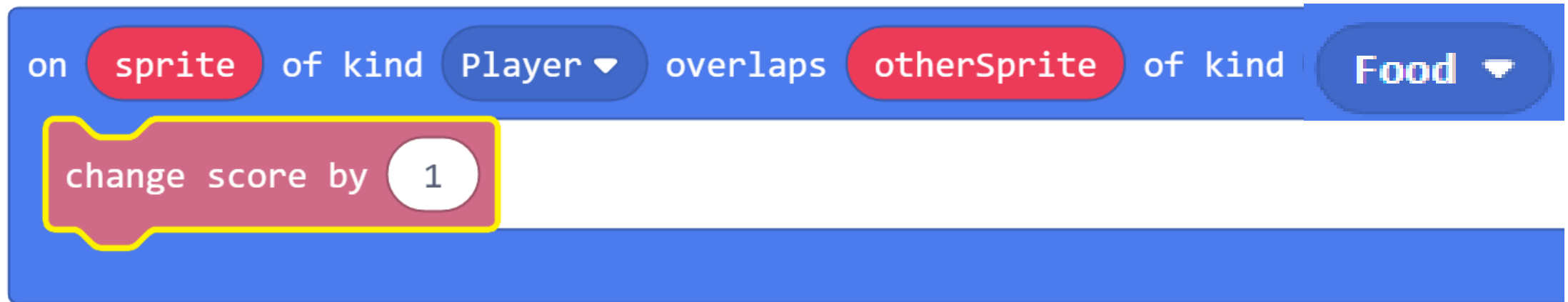
# Add a point when you eat bone

- From the **Sprites** Toolbox drawer, under the **Overlaps** category, drag an **On Sprite Overlaps** block onto the Workspace
- Change the second sprite kind to **food**



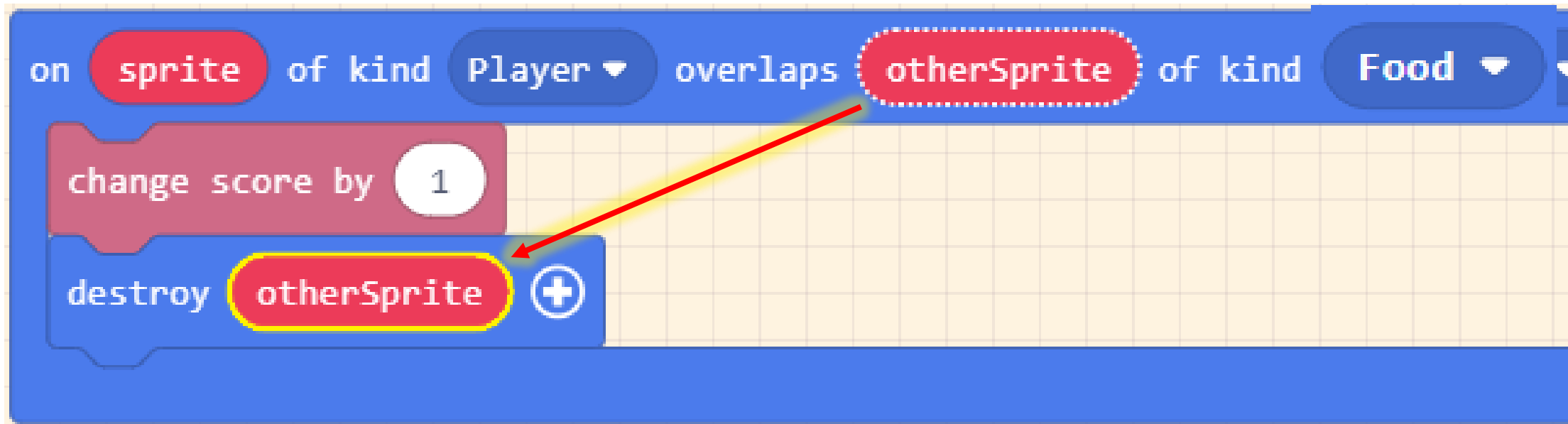
# Add a point when you eat fruit

From **Info** Toolbox drawer, add **Change Score** block



# Destroy the Fruit

- From the **Sprites** Toolbox drawer, Add **Destroy mySprite** block
- Drag the **otherSprite** local variable block into the **Destroy** block



# Add effects and music

- In the **Destroy** block, click the plus (+) icon
- Select an effect to show when our fruit is destroyed
- From the **Music** Toolbox drawer, drag a **Play Sound** block into the **On Sprite Overlaps** block

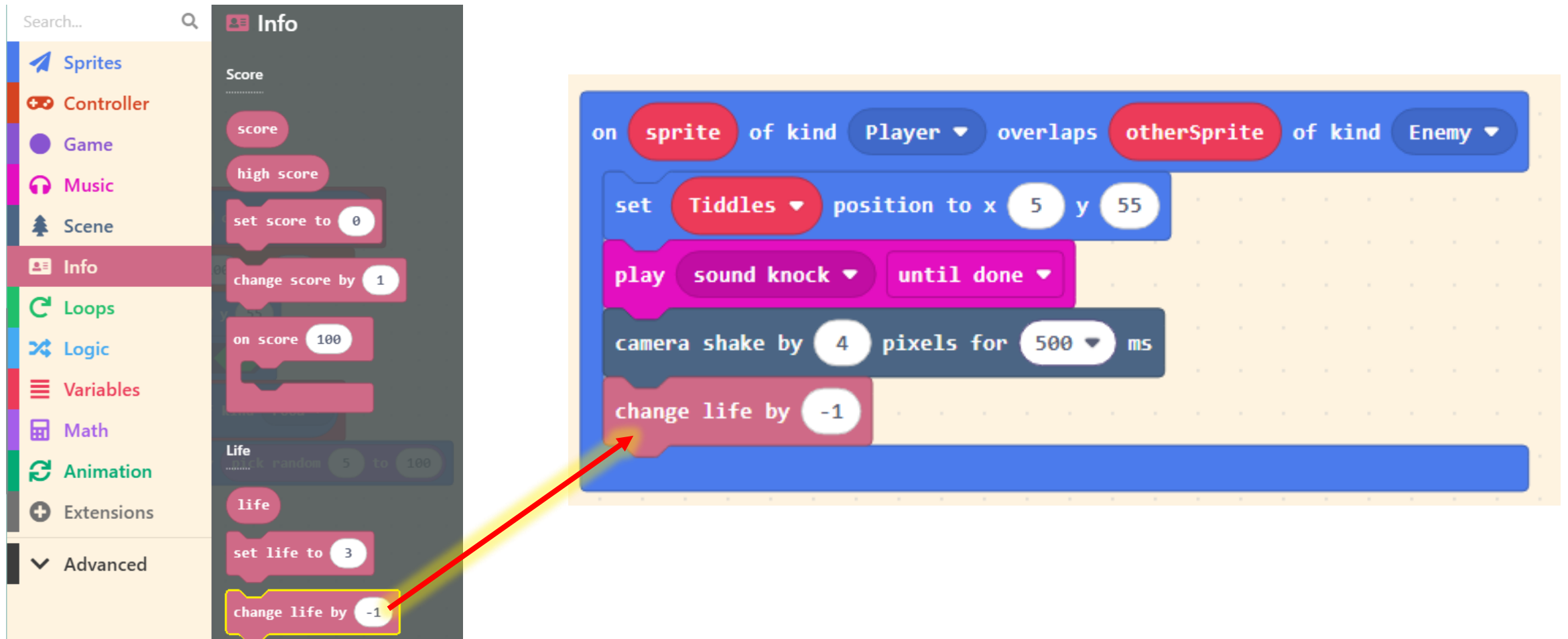


Add a pause before the next bone comes

Send Tiddles back across the road

# What happens if you get hit by a car?

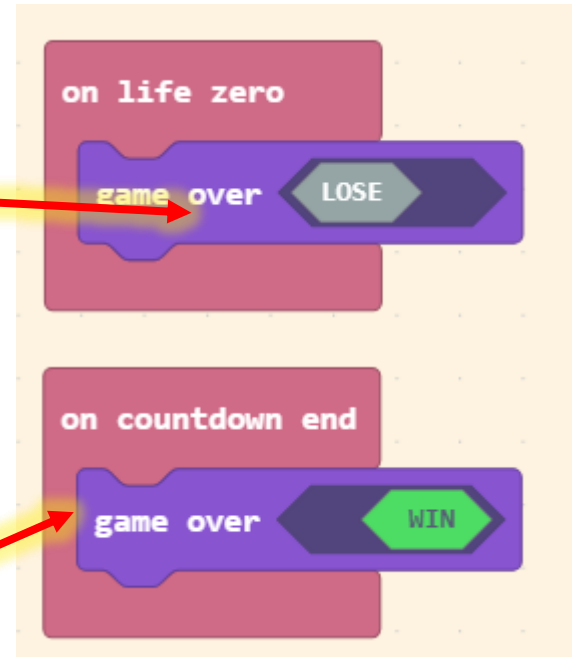
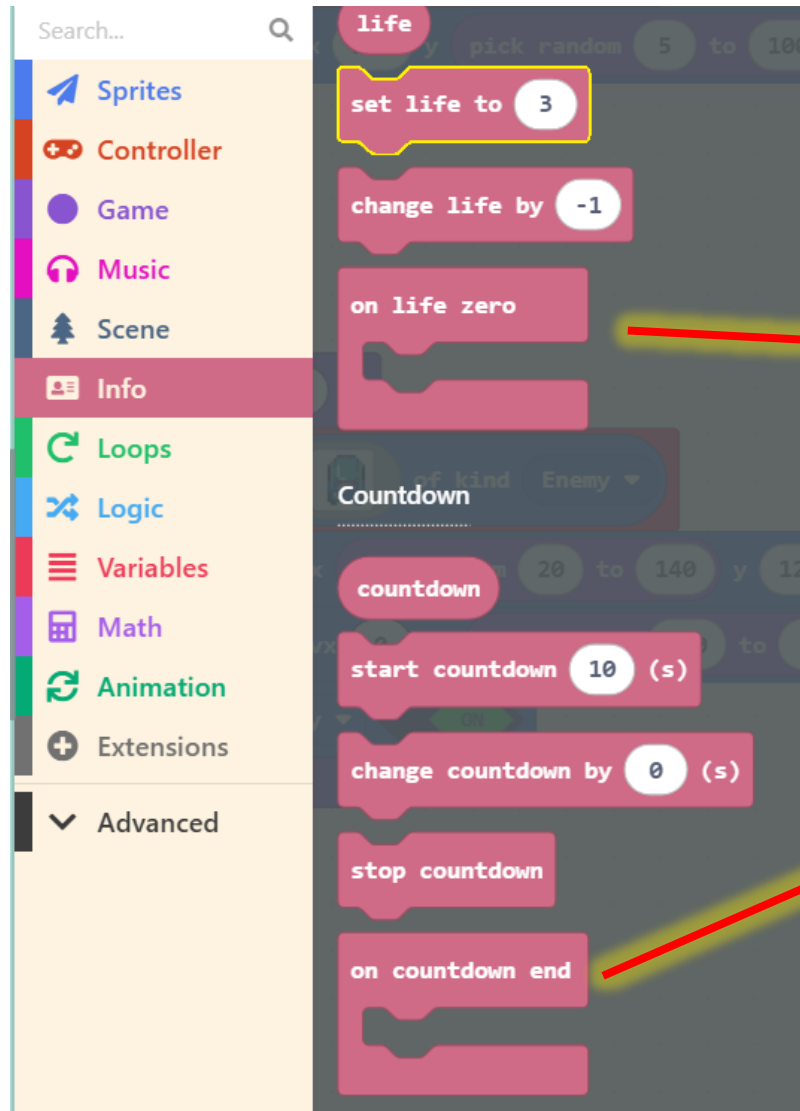
- Use the same tools but this time- when player overlaps enemy (car)



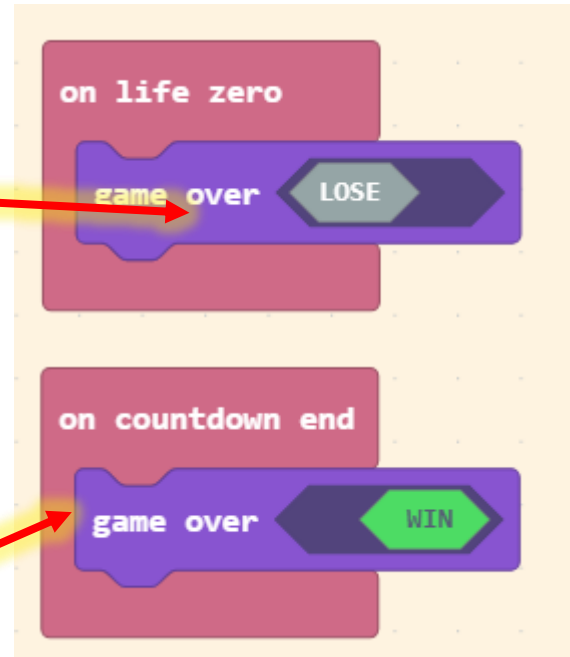
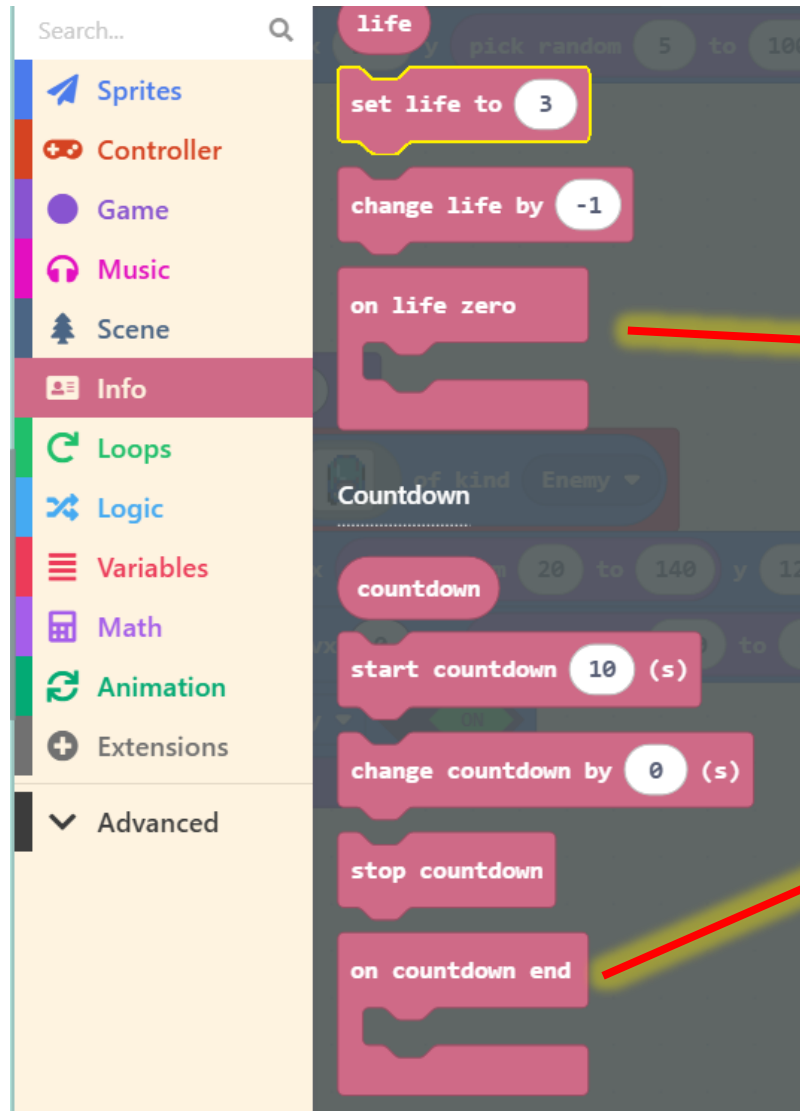
The image shows a Scratch code editor interface. On the left is a sidebar with categories: Sprites, Controller, Game, Music, Scene, Info, Loops, Logic, Variables, Math, Animation, Extensions, and Advanced. The 'Info' category is selected, showing a 'Life' variable with a 'change life by -1' block highlighted in yellow. A red arrow points from this block to a script in the main workspace. The script is triggered by the event 'on sprite of kind Player overlaps otherSprite of kind Enemy'. The script contains the following blocks: 'set Tiddles position to x 5 y 55', 'play sound knock until done', 'camera shake by 4 pixels for 500 ms', and 'change life by -1'.

```
on sprite of kind Player overlaps otherSprite of kind Enemy
  set Tiddles position to x 5 y 55
  play sound knock until done
  camera shake by 4 pixels for 500 ms
  change life by -1
```

# Game over?



# Game over?



# Try modding your game

- Add a beginning Splash screen
- Change the speed of the food or player
- Add other types of Fruit for different points
- Add more sounds and effects
- Change how often each type of food appears
- Change the direction of the food sprites
- Add Animations
- Move the bone around each time it is collected
- Animate the character

```
on start splash "Eat as many cherries as you can!"
```

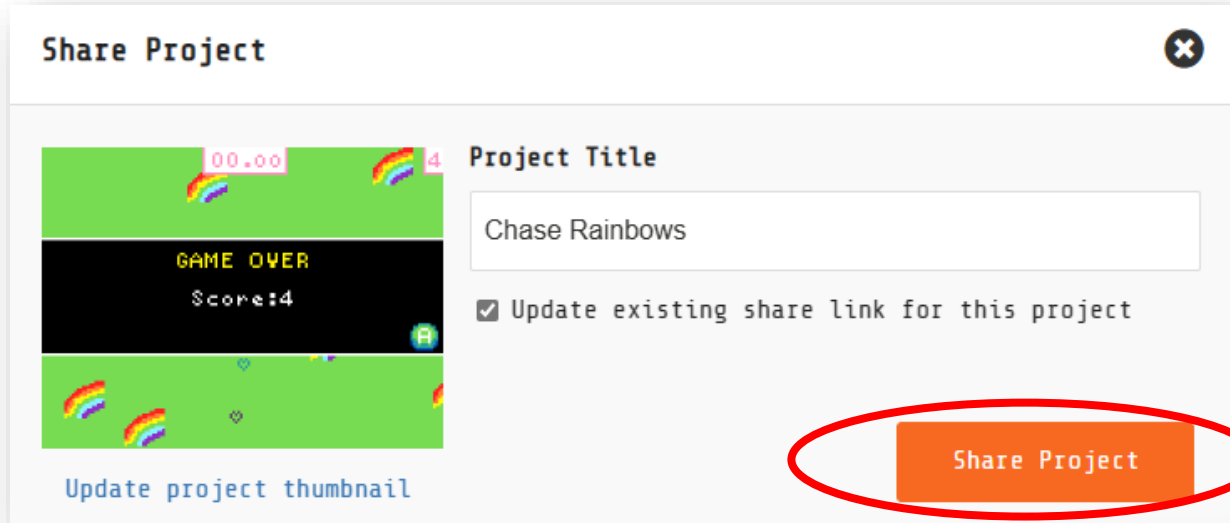
```
on destroyed sprite of kind Food  
set bone to sprite of kind Food  
set bone position to x 150 y pick random 5 to 100
```

```
on right button pressed  
animate Tiddles  
frames 5  
interval (ms) 500  
loop
```

```
on left button pressed  
animate Tiddles  
frames 5  
interval (ms) 500  
loop
```

# Share your game!

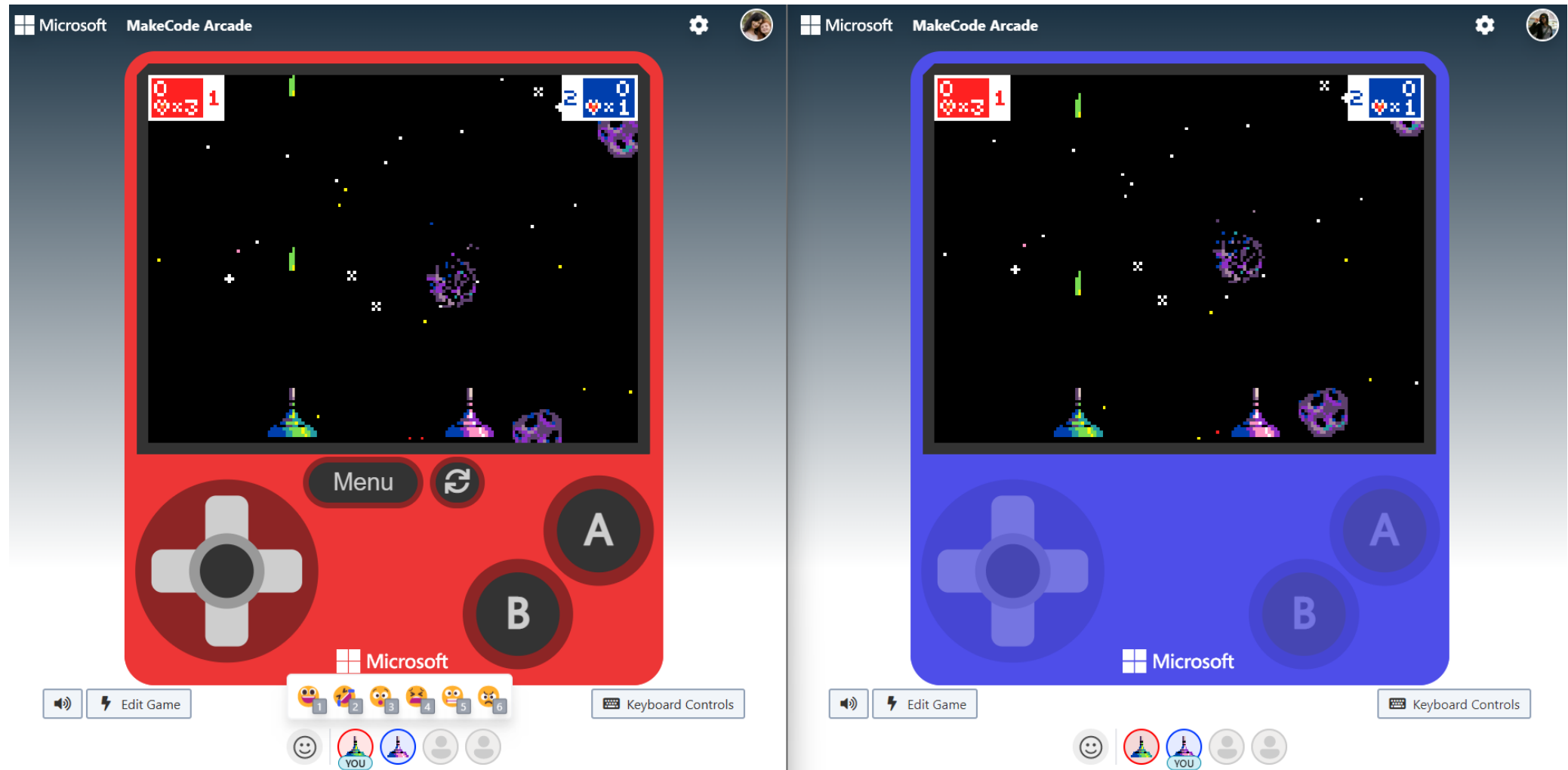
Click the Share button in the top left of the screen



Copy and Paste link  
into Chat window



# Playing games together



[arcade.makecode.com/--multiplayer](https://arcade.makecode.com/--multiplayer)

# Eat the Fruit – Multiplayer!



<https://arcade.makecode.com/S73249-29024-14612-80286>

# 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](https://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.

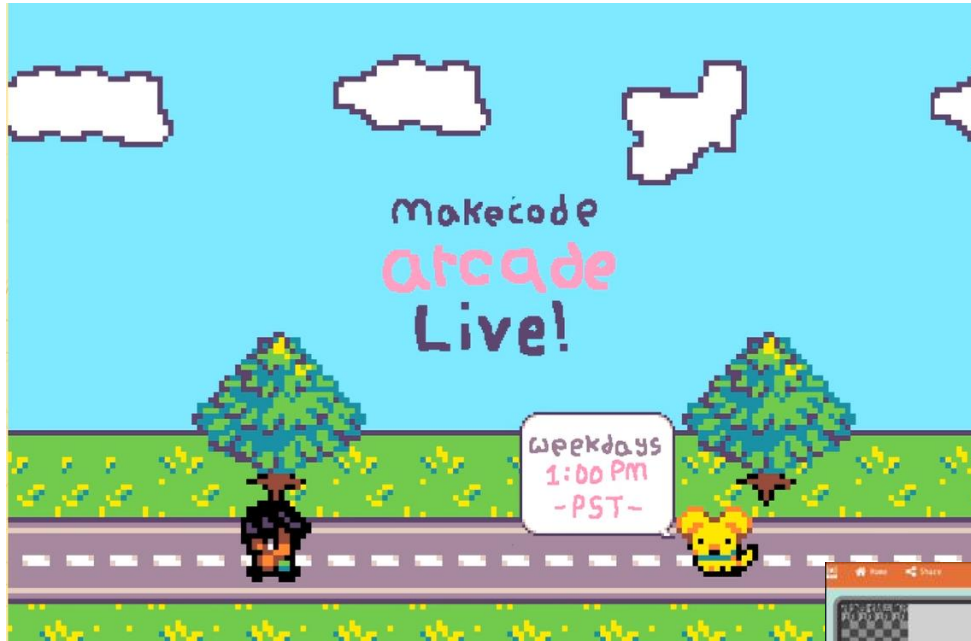


# Arcade Kiosk Mode



[arcade.makecode.com/hardware/kiosk](https://arcade.makecode.com/hardware/kiosk)

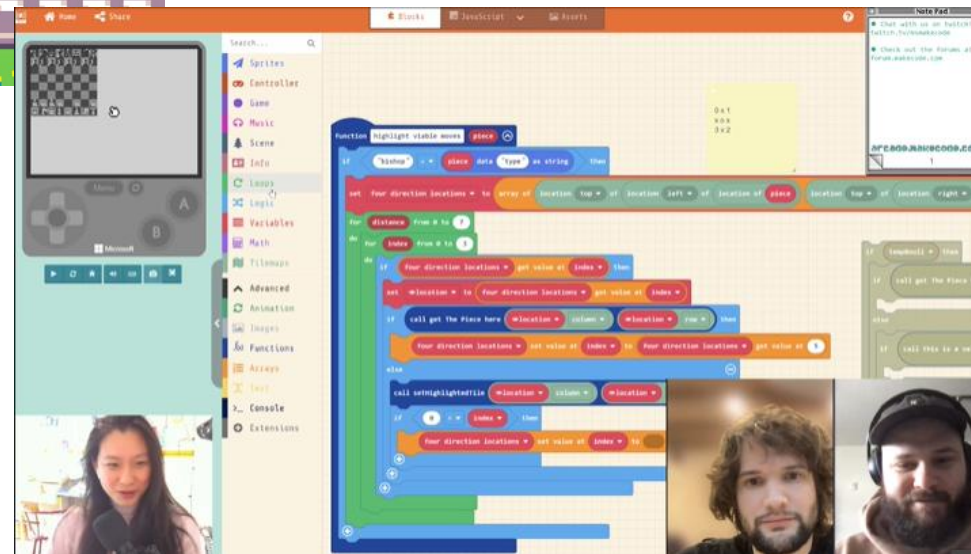
# MakeCode Arcade Live Stream & Forum



1pm Pacific / 3pm Eastern MWF

[twitch.tv/msmakecode](https://twitch.tv/msmakecode)

[forum.makecode.com](https://forum.makecode.com)



MakeCode  
Engineers

# 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))
```

