



# Trash Panda 2: Garbage Day

Created by John Park



<https://learn.adafruit.com/trash-panda-2-dumpster-dive>

Last updated on 2024-06-03 02:49:05 PM EDT

# Table of Contents

Overview	3
<hr/>	
• Parts	
Code Trash Panda 2: Garbage Day in MakeCode Arcade	5
<hr/>	
• MakeCode Arcade	
• Garbage Day	
• Load the Code	
• Play It	
• Debug Mode	
• Tile Mapped Hiding Spots	
• Character Setup	
• Night Mode	
• Gameplay	
• Gravity	
• Overlap	
Update the PyBadge/PyGamer Bootloader	15
<hr/>	
• PyBadge/PyBadge LC Bootloader	
• PyGamer Bootloader	
• Hardware Checks	
Load a MakeCode Game on PyGamer/PyBadge	17
<hr/>	
• Board Definition	
• Change Board screen	
• Bootloader Mode	
• Drag and Drop	
• Play!	
Troubleshooting MakeCode Arcade	21
<hr/>	

---

# Overview

In **Trash Panda 2: Garbage Day**, you play as the suburban dweller just trying to get some sleep when the raccoons and cats decide its time to make noise and throw garbage our of the trash bins! You must try to stop them by shining your flashlight on them. But you can only play at night, so be sure that your PyGamer or PyBadge's light sensor indicates it's dark out!

With this game, you'll learn about tile arrays, gravity, using the light sensor, and more! Plus, you can have fun challenging your friends to see who can get the highest score!

## Parts

You can play Trash Panda 2 in the browser by turning on "debug" mode, but for the full experience and to use the light sensor, you can play it on a PyGamer or PyBadge.



### [Adafruit PyGamer Starter Kit](https://www.adafruit.com/product/4277)

Please note: you may get a royal blue or purple case with your starter kit (they're both lovely colors)What fits in your pocket, is fully Open...

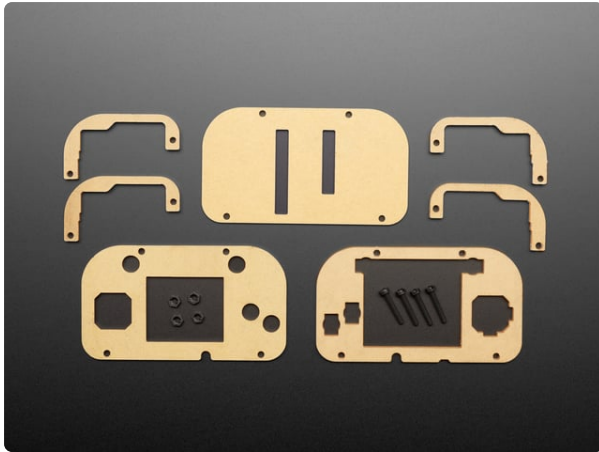
<https://www.adafruit.com/product/4277>



### [Adafruit PyGamer for MakeCode Arcade, CircuitPython or Arduino](https://www.adafruit.com/product/4242)

What fits in your pocket, is fully Open Source, and can run CircuitPython, MakeCode Arcade or Arduino games you write yourself? That's right, it's the Adafruit...

<https://www.adafruit.com/product/4242>



### Adafruit PyGamer Acrylic Enclosure Kit

You've got your PyGamer, and you're ready to start jammin' on your favorite arcade games. You gaze adoringly at the charming silkscreen designed by Adafruit...

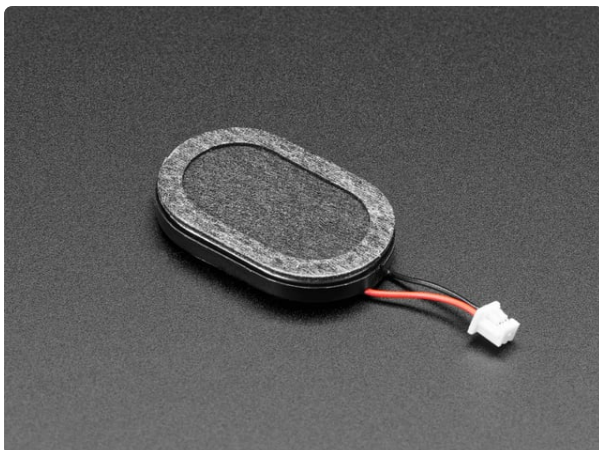
<https://www.adafruit.com/product/4238>



### Lithium Ion Polymer Battery with Short Cable - 3.7V 350mAh

Lithium-ion polymer (also known as 'lipo' or 'lipoly') batteries are thin, light, and powerful. The output ranges from 4.2V when completely charged to 3.7V. This...

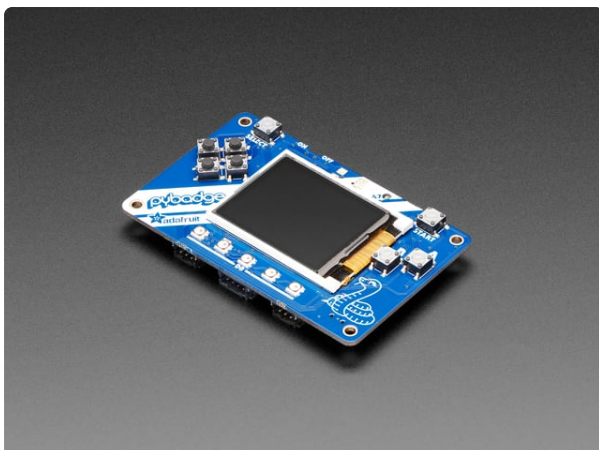
<https://www.adafruit.com/product/4237>



### Mini Oval Speaker with Short Wires - 8 Ohm 1 Watt

Hear the good news! This wee speaker is a great addition to any audio project where you need 8 ohm impedance and 1W or less of power. We particularly like...

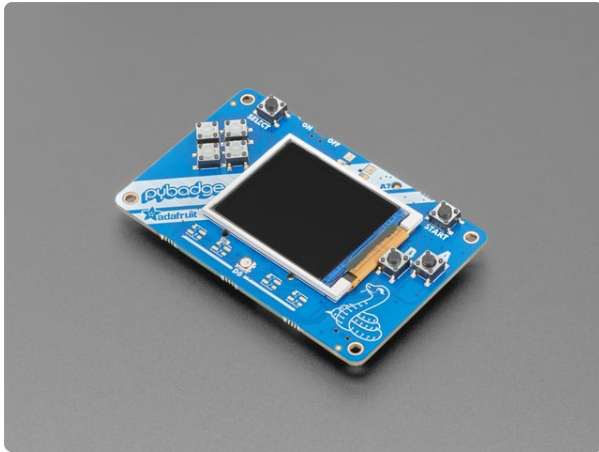
<https://www.adafruit.com/product/4227>



### Adafruit PyBadge for MakeCode Arcade, CircuitPython, or Arduino

What's the size of a credit card and can run CircuitPython, MakeCode Arcade or Arduino? That's right, its the Adafruit PyBadge! We wanted to see how much we...

<https://www.adafruit.com/product/4200>



Adafruit PyBadge LC - MakeCode Arcade, CircuitPython, or Arduino

What's the size of a credit card and can run CircuitPython, MakeCode Arcade or Arduino even when you're on a budget? That's right, it's the Adafruit...

<https://www.adafruit.com/product/3939>

---

# Code Trash Panda 2: Garbage Day in MakeCode Arcade

## MakeCode Arcade

MakeCode Arcade is a free Microsoft block programming environment designed specifically to make games. Learning to use MakeCode is easy & fun.

If you're not already familiar with the basics of MakeCode Arcade, [check out this guide \(https://adafru.it/Elc\)](https://adafru.it/Elc) on creating a character sprite and moving it with controls.

To start, open a new Chrome browser window (Chrome works best) and go to [MakeCode Arcade \(https://adafru.it/DCY\)](https://adafru.it/DCY).

These MakeCode Arcade guides are designed to take you through the fundamentals before tackling more complex games:

- [Pixel Art \(https://adafru.it/EOI\)](https://adafru.it/EOI)
- [Animation \(https://adafru.it/EOk\)](https://adafru.it/EOk)
- [Level Design \(https://adafru.it/EOj\)](https://adafru.it/EOj)
- [Sparky Invaders \(https://adafru.it/EYf\)](https://adafru.it/EYf)
- [Next Level Game Techniques \(https://adafru.it/EYg\)](https://adafru.it/EYg)

For intermediate-level techniques, check out:

- [Re-MakeCode the Classics: Arkanoid \(https://adafru.it/E-o\)](https://adafru.it/E-o)
- [Re-MakeCode the Classics: Py Hunter \(https://adafru.it/F2T\)](https://adafru.it/F2T)

And, for the full immersion in Trash Panda game lore, check out [Trash Panda: The Game](https://adafru.it/F7W) (<https://adafru.it/F7W>)!

Only use the Google Chrome browser with MakeCode!

## Garbage Day

To start with, let's load the completed version of **Trash Panda 2: Garbage Day** and try it out.



Start by launching [MakeCode Arcade](https://adafru.it/DCY) (<https://adafru.it/DCY>) using the Google Chrome web browser. Then, download the **arcade-Trash-Panda-2-Garbage-Day.png** file above by right-clicking on the image and saving it to your computer.

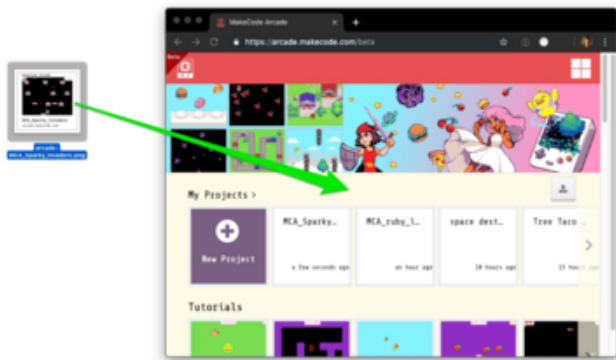




## Load the Code

This is a special .png file that contains not only an image, but the entire game is embedded in it as well!

Simply drag it from the location to which you saved the image on your computer (such as the desktop as shown here) onto the Chrome browser window that is already running MakeCode Arcade (MCA). Note that the image in this graphic is of a different game, but you'll be dragging the Trash Panda 2: Garbage Day .png file.



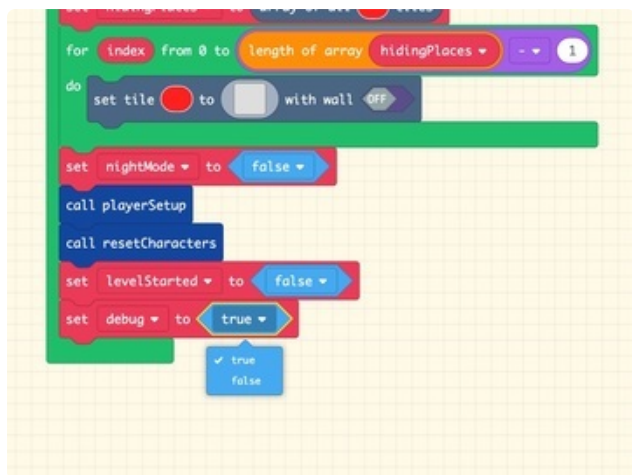
This will open the code into the MCA editor.

If you're ever unsure where a MakeCode block comes from, you can often find it by matching the block's color to a category on the left side of the editor. You can also use the handy search function!

## Play It

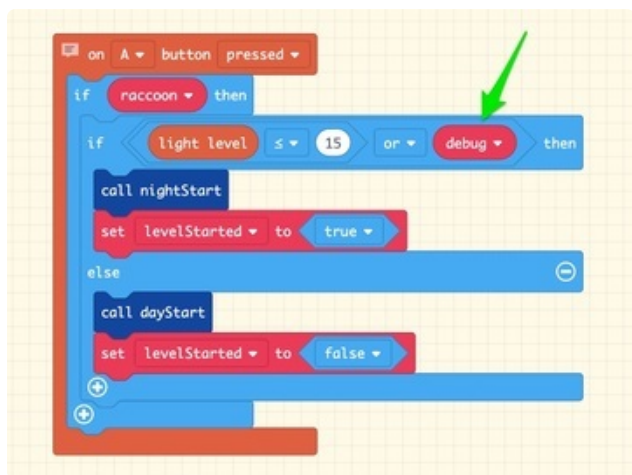
Time to try out the game! The instructions will let you know that it needs to be played in the dark if you're playing on your PyGamer or PyBade, however, you can switch to debug mode in order to play in the browser for now.

To do so, go to the **on start** block and flip the **debug** variable value to **true**.



## Debug Mode

Setting up a debug mode can be very helpful while you're developing a game. It allows you to do things such as skip over a set of intro screens, or give yourself unlimited lives.



In this case, it's part of a conditional check done to see if either the **light level** is low enough or the **debug** is true.

Once you're in **debug** mode, try the game in the browser simulator.

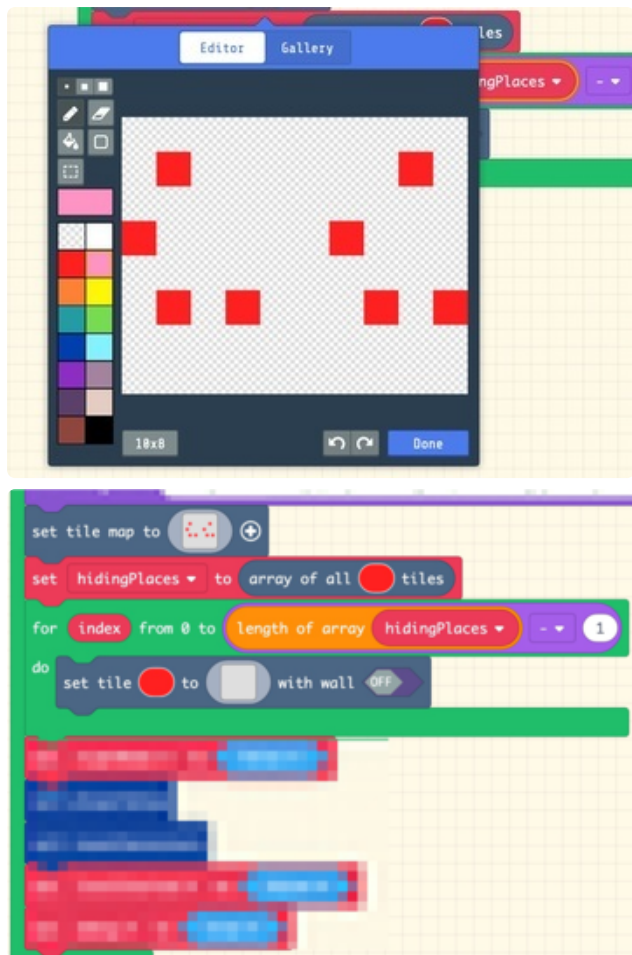
You'll use the direction keys/arrows to move the flashlight around to catch the critters from making a mess!

The gameplay is based on quickly shining the light on either the raccoon or the cat. They have glowing eyes with subtle difference that the player will learn to spot, and they also throw different trash out -- raccoons throw cans, cats throw fishbones.

Since the raccoons are worth more points, players will need to determine if its worth seeking them out, or going for the quicker points when the cat is nearer to the spotlight.

Next, we'll take a look at how some of the game's unique features work. Since many of the blocks are typical to most MakeCode Arcade games (see the list of links above), we won't go over every single block, but instead take a closer look at the concepts and techniques that are unique to this game.





## Tile Mapped Hiding Spots

We need to set up a set of positions for the characters to hide. To do this, we'll create an array of red tile maps to use for spawning the raccoon and cat sprites.

With the tile map created, we created a variable called **hidingPlaces** and set it to the **array of all red tiles**. This makes it convenient to iterate those tiles with a **for** loop.

We want to set the tiles to have no sprite, so they just act like coordinate spots but remain invisible, so in our **for loop** we run through the array and for each red tile use the **set tile red to sprite** block without setting any pixels into the sprite.

## Character Setup

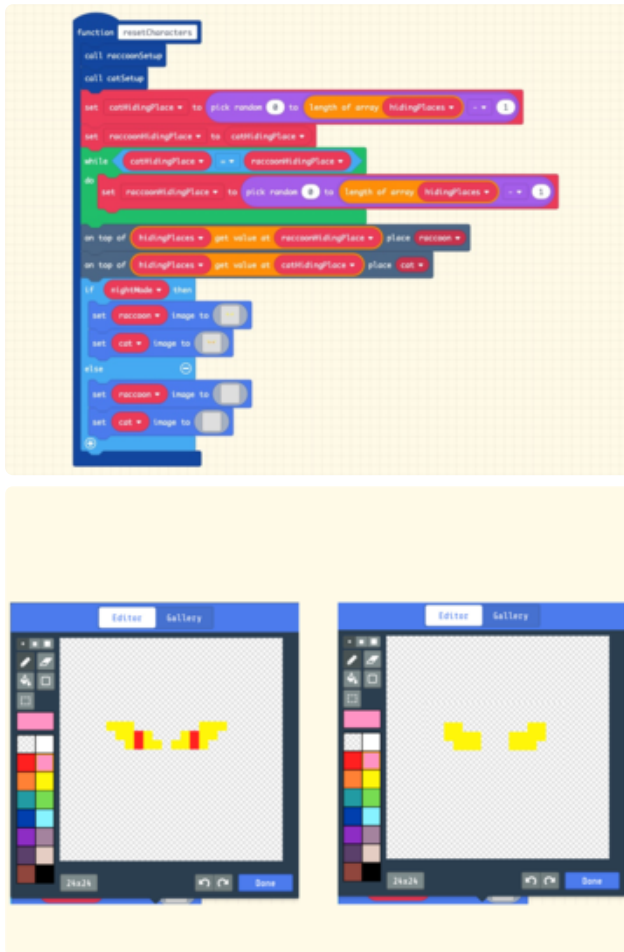
When the level begins and each time one of the characters is spotted with the flashlight, we'll call the **resetCharcters** function to set them up.

This function creates the sprite for the **raccoon** and **cat** and then creates a variable called **catHidingPlace**, which is set to a random pick of the **hidingPlaces** tile array.

We then us a **while** loop to pick a random tile for the **raccoonHidingPlace** variable. The **while** loop will run until the two hiding places are different numbers, so that the characters don't appear on the same spot.

Once the variables are set to unique numbers (from 0-7) we us **on top of hidingPlaces** **get value at raccoon/ catHidingPlace** to place the sprites.

Then, we check if it's **nightMode** or **dayMode** and set the images to either glowing eye sprites, or nothing at all so they don't appear during the day.



## Night Mode

We'll use the light sensor on the PyGamer or PyBadge to determine if it's night (or at least dark!) when the player plays the game.

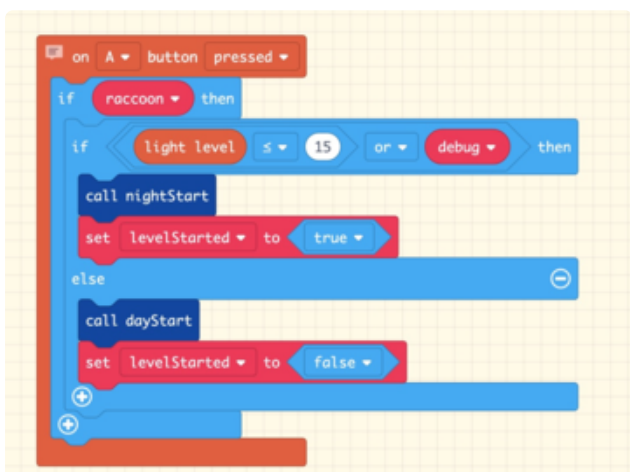
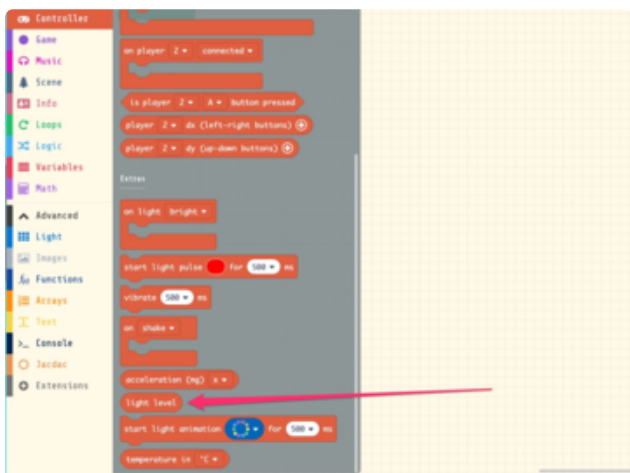
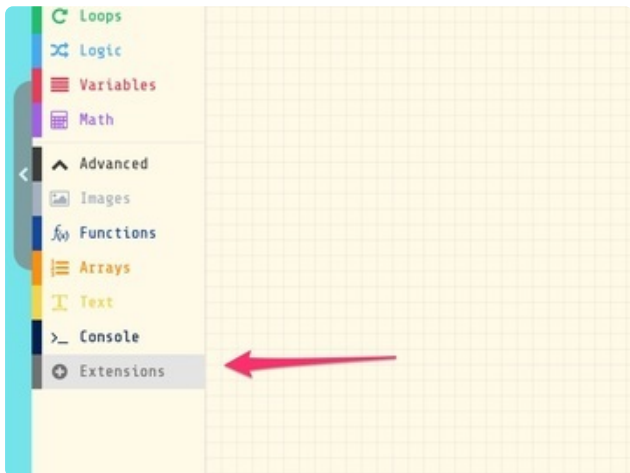
To do this, we need to add the **controller** extension to MakeCode Arcade. This adds a number of additional controller blocks that can be used with handheld devices, including light sensing, vibration/accelerometer sensing, temperature sensing, and more.

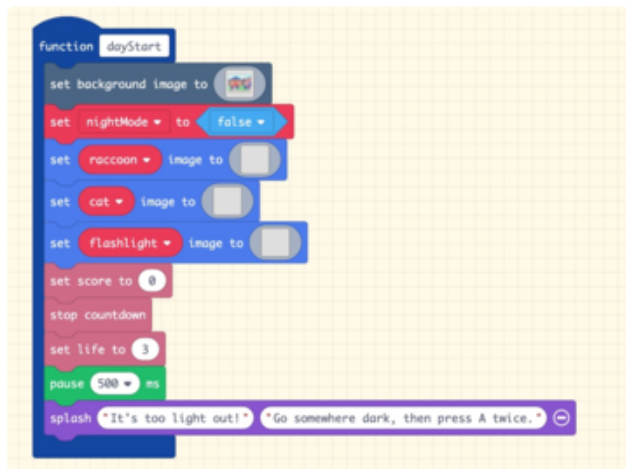
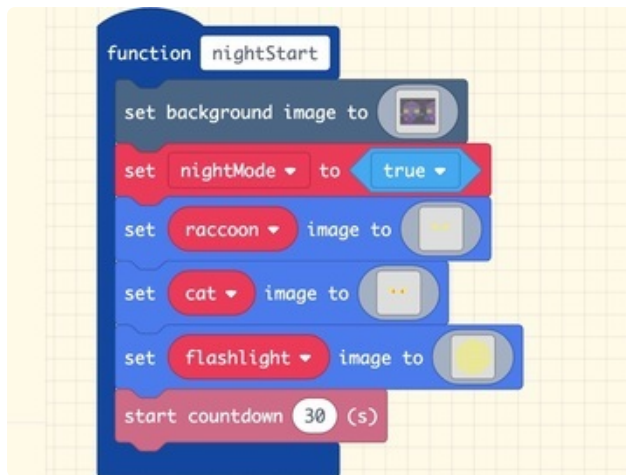
We'll use the **light level** block, which returns a value from 0-255 depending on the dark-to-light levels seen by the sensor on the PyGamer or PyBadge.

Once the player has gotten through the introduction of the game, they'll press the A button to begin. Here we see that this then checks that the **raccoon** sprite has been created (this is a way of avoiding a chicken-or-the-egg paradox if the button is pressed before the game is ready).

Next, we check the **light level** to see that it is dark. You can fine tune this value, but I found that **15** is a good setting for allowing the player to hold the PyGamer under a desk or table and still play in a partially lit room.

If the **light level** comes back as dark enough, we call the **nightStart** function and flip the **levelStarted** variable to **true** -- otherwise, we go to **dayStart** and show the daytime scene, asking the player to go somewhere dark and try again!





## Gameplay

Once the level is set up, let's look at the actual gameplay loops. There are just a few things that happen:

- 30 second countdown timer begins
- a raccoon eyes and cat eyes sprite each are placed on a random red (hidden) tile
- the raccoon starts throwing a can in a random direction every two seconds

- the cat starts throwing fish bones in random directions every two and a half seconds
- when the player overlaps the flashlight spotlight with a raccoon or cat, the image of the critter is revealed, points are scored, the characters are reset to new positions, and the player flashlight is reset to a starting position

## Gravity

Let's take a look at how the critters are throwing the trash.



In this example, if the level has started, we check the **cat** sprite to see which quadrant of the screen it's on via the **x** and **y** values.

We then set a variable **fishVX** to a random number in a sensible range depending on the **cat's** position, so that it won't throw the trash off screen. This variable will be used to set the **velocity x** of the fishbone.

The same is done for the **fishVY** variable, except for the vertical throw. If the cat is lower on screen, the throw will be stronger.

Then, the **fishbone** sprite projectile is created with an appropriate pixel art representation and the chosen velocities on **x** and **y**.

So that we can have the appearance of gravity, we'll set the **fishbone ay (acceleration y) to 250**. This way, after the initial velocity impulse shoots the fishbone into the air,

the downward acceleration will take over, making it arc and fall to the bottom of the screen.







## Overlap

When the player shines the light on one of the critters, here's what we do:

Play a complaining sort of sound glide using the

`music.playSound('340,400^220')` notation in JavaScript as we did in the original Trash Panda game (<https://adafru.it/F7X>)

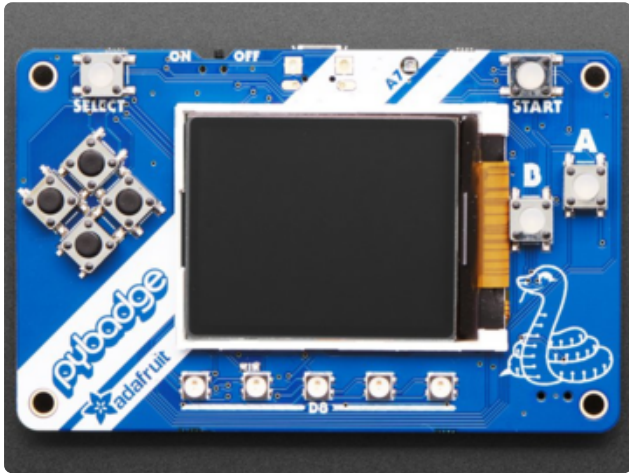
Swap the sprite to the full character  
Have the character exclaim "drat" or "mrrowr" using the **sprite say " "** block  
turn off the flashlight sprite  
pause briefly  
destroy the character sprite  
increase the score  
reset the player's flashlight sprite image and position  
reset the characters to continue gameplay

Finally, when the countdown ends, the game is over and the final score is displayed!  
Have fun trying to get your best score!!

## Update the PyBadge/PyGamer Bootloader

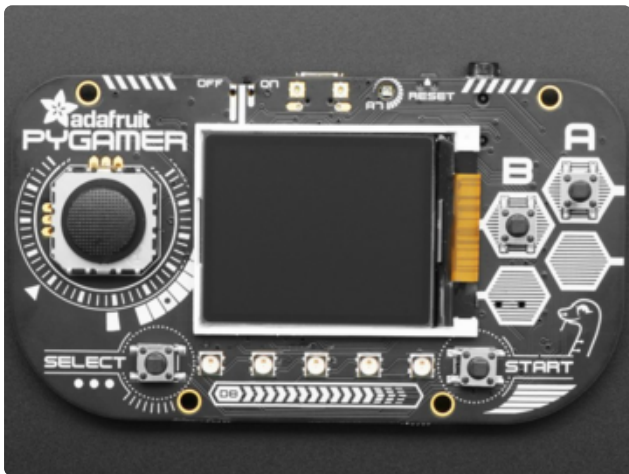
You are at the bleeding edge of handheld, open source, game playing hardware and software, what with your PyBadge/PyBadge LC or PyGamer! Congratulations! It's fun and exciting! It is also changing and improving all the time, so please update your bootloaders before proceeding to put your MakeCode Arcade games on the board!!

Among lots of other reasons, update the bootloader to prevent a problem with MacOS 10.14.4, to fix button problems, and get the thumbstick to work!



## PyBadge/PyBadge LC Bootloader

If you have a **PyBadge** or **PyBadge LC**, please go to this page for instructions on updating the bootloader. (<https://adafru.it/EWI>)



## PyGamer Bootloader

If you have a **PyGamer**, please go to this page for instructions on updating the bootloader. (<https://adafru.it/EWJ>)

A HUUUUUUUGE number of people have problems because they pick a 'charge only' USB cable rather than a "Data/Sync" cable. Make 100% sure you have a good quality syncing cable. Srsly, I can't even express how many times people have nearly given up due to a flakey USB cable! Enter Alert Text...

## Hardware Checks

If, after updating your board's bootloader, you still think you may have a hardware problem, here's a great way to test out all of the functions. From buttons, to the light sensor, thumbstick (PyGamer only), accelerometer (PyGamer and PyBadge only, not the LC), and more, we've got a super nifty set of hardware test .UF2 files you can use.

Click on the link for your board below for more info and a link to the appropriate UF2 file.

### PyBadge/PyBadge LC Hardware Check

<https://adafru.it/EWK>

### PyGamer Hardware Check

<https://adafru.it/EWL>

Another way to do a hardware check is with the handy, dandy MakeCode Arcade Basic Hardware Test. This was created with MakeCode Arcade and you can use it to check that your d-pad buttons or thumb joystick can move the yellow face around the screen, and that the A and B buttons work to play a sound (just make sure you have a speaker plugged in to the PyGamer first).

You can [open this link \(https://adafru.it/EWP\)](https://adafru.it/EWP) to get to it, or download the UF2 file below and drag it onto your board's USB drive in bootloader mode.

### arcade-Basic-Hardware-Test.UF2

<https://adafru.it/EWQ>



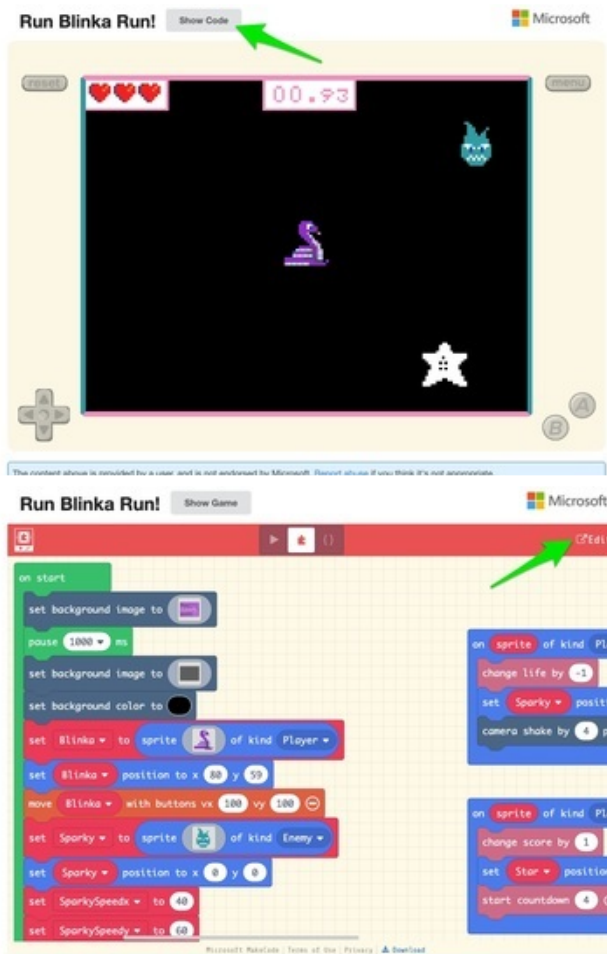
---

## Load a MakeCode Game on PyGamer/PyBadge

Let's load a game! For example, here's a link to **Run, Blinka, Run!** To open the game in the MakeCode Arcade editor, first, click the share link below. This will allow you to play the game in the browser right away.

## Makecode Arcade Game: Run, Blinka, Run!

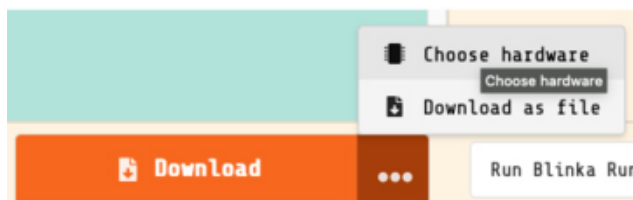
<https://adafru.it/Fqf>



Then, click on the Show Code button in the upper left corner. This shows the code for the game, and by clicking the Edit button in the upper right corner, it'll open into the editor where you can upload it to your PyGamer/PyBadge.

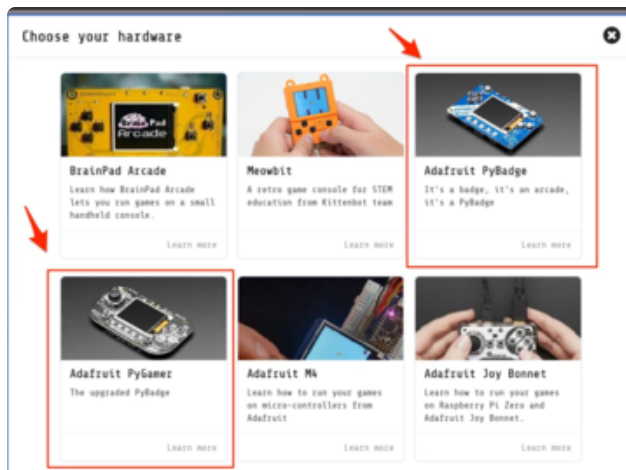
Once you have a game working on the MakeCode Arcade web editor, it's time to download it and flash it onto your board.

Please only use the Google Chrome browser with MakeCode! It has WebUSB support and seems to work best



## Board Definition

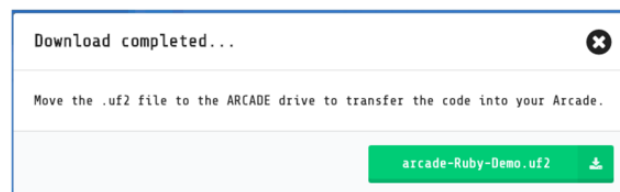
In order to load a game made in MakeCode Arcade onto the PyBadge, first choose the proper board definition inside of MakeCode. Click the ellipsis (...) next to DOWNLOAD and then the **Choose Hardware** item.



## Change Board screen

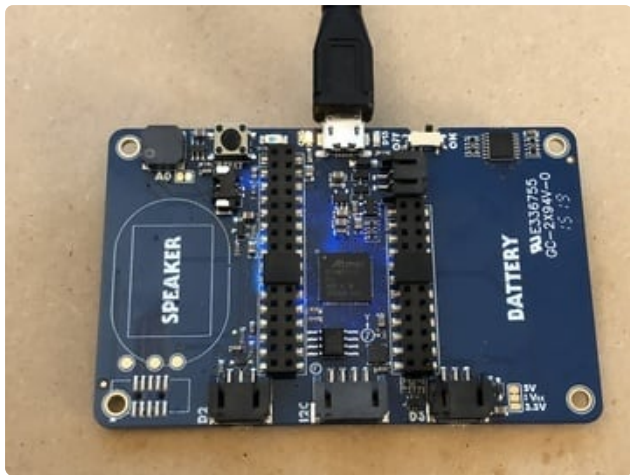
Click on the image of your board, either the PyBadge/PyBadge LC or the PyGamer

This will cause the game .uf2 file for your particular board to be saved to your hard drive. You only need to do this the first time you use a new board. Thereafter you can simply click the **Download** button on the MakeCode Arcade editor page.



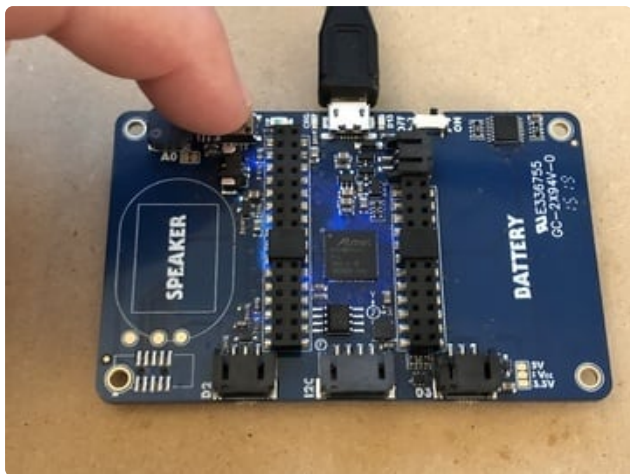
A HUUUUUUGE number of people have problems because they pick a 'charge only' USB cable rather than a "Data/Sync" cable. Make 100% sure you have a good quality syncing cable. Srsly, I can't even express how many times people have nearly given up due to a flakey USB cable!





## Bootloader Mode

Now, we'll put the board into bootloader mode so we can drag on the saved .uf2 file. On the back side of the board you'll see a reset button at the top. Make sure the board is plugged into your computer via USB with a USB micro B to A data cable. Also, be sure the board is turned on.

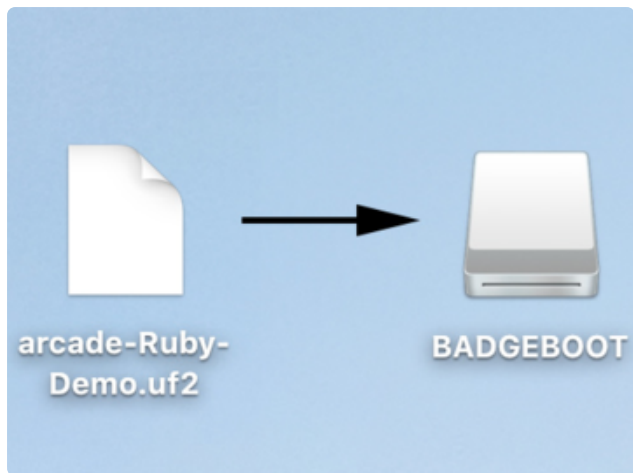


Then, press the reset button. This will initiate bootloader mode.



When the board is in bootloader mode you'll see a screen similar to this one show up.



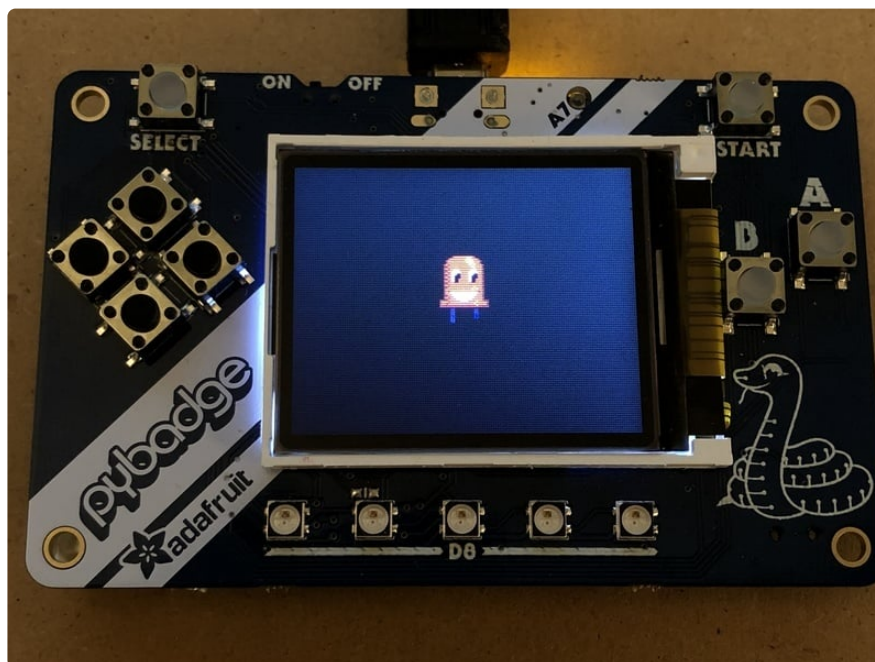


## Drag and Drop

Now that the board is in bootloader mode, you should see a **BADGEBOOT** drive show up on your computer as a USB flash drive. Simply drag the arcade game .uf2 file onto the drive.

## Play!

That's all there is to it! Once the file is copied over the board will restart and launch the game!



Keep an eye on [Adafruit.com](https://adafruit.com) for additional game related content.

---

## Troubleshooting MakeCode Arcade

If you run into trouble with MakeCode Arcade, here are some resources for getting help:

- [Microsoft MakeCode Arcade Forum \(https://adafru.it/EXI\)](https://adafru.it/EXI)
- [Adafruit MakeCode Forum \(https://adafru.it/EXJ\)](https://adafru.it/EXJ)

- [Microsoft MakeCode Arcade Discord \(https://adafru.it/EXK\)](https://adafru.it/EXK) -- look for the #arcade channel
- [Adafruit MakeCode Discord \(\)](#) -- look for the #makecode channel

Only use the Google Chrome browser with MakeCode!