Go Fishing with Rotary Encoders

Created by John Park

https://learn.adafruit.com/gone-fishing-game

Last updated on 2023-08-29 04:09:45 PM EDT
# Table of Contents

## Overview
- Parts
- Optional

## Adding a Rotary Encoder to PyGamer/PyBadge
- Rotary Encoder Basics
- MakeCode Arcade
- “Crank” in MakeCode Arcade
- Add Extensions
- Example Code
- Fishing Reel Crank

## Fishing Game in MakeCode Arcade
- Gone Fishing
- Load the Code
- Play the Game
- How it Works
- On Start
- Water Layer
- Painting with Code
- Hook and Line
- Spawn Fish
- Casting
- Cranking and Hooking
- Cranking Battle

## Update the PyBadge/PyGamer Bootloader
- PyBadge/PyBadge LC Bootloader
- PyGamer Bootloader
- Hardware Checks

## Load a MakeCode Game on PyGamer/PyBadge
- Board Definition
- Change Board screen
- Bootloader Mode
- Drag and Drop
- Play!

## Troubleshooting MakeCode Arcade
Overview

A rotary encoder (aka "endless encoder") can be used as a controller on the PyBadge and PyGamer in MakeCode Arcade. We can determine the speed and direction of rotation and then apply that to sprite positions or other settings in a game. This fishing game makes use of the "crank" blocks in MakeCode Arcade to reel in the fish!

Parts

Adafruit PyGamer for MakeCode Arcade, CircuitPython or Arduino
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 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, it's the Adafruit PyBadge! We wanted to see how much we...
https://www.adafruit.com/product/4200

Rotary Encoder + Extras
This rotary encoder is the best of the best, it's a high-quality 24-pulse encoder, with detents and a nice feel. It is panel mountable for placement in a box, or you can plug it...
https://www.adafruit.com/product/377

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
Silicone Cover Stranded-Core Wire - 26AWG in Various Colors
Silicone-sheathing wire is super-flexible and soft, and it's also strong! Able to handle up to 200°C and up to 600V, it will do when PVC covered wire wimps out.
We like this...

Optional

You can 3D print a case and spinning reel crank for your PyGamer and rotary encoder.

Filament for 3D Printers in Various Colors and Types
Having a 3D printer without filament is sort of like having a regular printer without paper or ink. And while a lot of printers come with some filament there's a good chance...
https://www.adafruit.com/product/2080

Monoprice Inventor II 3D Printer with Touchscreen and WiFi
The Monoprice Inventor II 3D Printer Touchscreen with WiFi is a perfect entry-level 3D printer with small footprint and reliable performance. It comes equipped with...
https://www.adafruit.com/product/3897
Ultimaker 2+ 3D Printer
The Ultimaker 2+ is one of our favorite 3D printers on the market. It's a well-built open-source compact machine with an excellent UX. Every inch of the...
https://www.adafruit.com/product/2673

Adding a Rotary Encoder to PyGamer/PyBadge

I used the excellent Gaming Handheld Crank () guide by the Ruiz Bros. as a launching off point for building this project. Check it out for lots of details.

For the fishing controller, I connected the PyGamer to the rotary encoder like this:

- Pin A2 connects to the B pin (Green wire)
- Pin A3 connects to the A pin (Blue wire)
- GND connects to the common C pin (Gray wire)
- Pin D9 connects to the Switch pin (Yellow wire)

Since the rotary encoder's switch pin side also uses the common ground, I simply connected it to the C pin over on the encoder side of the part. This prevents you from connecting another wire all the way from the encoder to the board.
I soldered silicone stranded hookup wire to the encoder pins on one end and to 90° angle header pins on the other so I could insert them in the PyGamer's Feather headers without protruding too far from the board.

The additional unused header pins are there simply for mechanical stability of the connections.
With the battery and speaker also plugged in, we're ready to start programming in MakeCode Arcade!

Rotary Encoder in MakeCode Arcade

Rotary Encoder Basics

Rotary encoders are very useful, and seem a bit mysterious sometimes. They can spin endlessly in either direction. And, both the speed at which they're turned and the direction they're being turned can be detected and used by the microcontroller, in this case the PyGamer. But how?!

Inside the switch are two contact points, A & B, and a perforated disc of multiple common ground contact points, C. There's no need for resistors or a voltage pin as we'll use the PyGamer's built-in pull-up resistors automatically in the encoder extension.

As the disc spins, say, clockwise, one of the C pads touches A causing a square wave pulse to go high on the A pin. Then, a short moment later, the C pad touches B causing an pulse on the B pin. The PyGamer is able to determine the speed of the rotation by sensing the period of time between the two pulses.

When the knob is turned counterclockwise, the B pulse will come first, followed by the A pulse, so it is this way that the PyGamer know which direction the rotary encoder is turning.
Here is an excellent explainer video from How to Mechatronics:

MakeCode Arcade

MakeCode Arcade is a free Microsoft block programming environment designed specifically to make games, but we can also use it for non-game application development. Learning to use MakeCode is easy & fun.

If you’re not already familiar with the basics of MakeCode Arcade, check out this guide 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 beta.

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

- Pixel Art
- Animation
- Level Design
- Sparky Invaders
- Next Level Game Techniques

For intermediate-level techniques, check out:

- Re-MakeCode the Classics: Arkanoid
- Re-MakeCode the Classics: Py Hunter
- Trash Panda
- Trash Panda 2: Garbage Day
- Marble Labyrinth

Only use the Google Chrome browser with MakeCode!
"Crank" in MakeCode Arcade

Partly inspired by the Playdate () handheld designed by Panic and Teenage Engineering, the Microsoft MakeCode Arcade team developed a set of "crank" blocks to use rotary encoders.

Here's how to add the blocks and use them.

Add Extensions
First, head to MakeCode Arcade beta () editor and click on the setting gear wheel menu and click on Extensions.

From the gallery of extensions, we'll add both controller and feather (you'll need to go to this gallery twice to pick them each individually.)

Now, when we go to the Controller category, we'll see the crank position block and the set crank pinA ___ pinB ___ block are available.

By adding the Feather extension, we are able to select any of the exposed pins on the back of the PyGamer!
Example Code

Here's a very simple program we can create in order to test the rotary encoder in MakeCode Arcade on the PyBadge.

Note how we've set the crank pinA to A3 and pinB to A2, which is how we wired the encoder on the previous page.

This creates a piece of delicious cake as a player sprite, and sets its horizontal position to be driven by the rotary encoder's position.

Upload the code to your PyGamer and give it a try now!
Fishing Reel Crank

Based on the crank design by the Ruiz Bros. found here, I remixed it into a spinning reel style crank to use in our fishing game.

You can download the model here and 3D print it. It prints the arm and handle at the same time, but they aren't touching each other, so the handle is free spinning. The crank is designed to press fit onto the rotary encoder shaft.

I printed the case and followed their build instructions, but decided to leave off the front of the case so I could admire the awesome PyGamer silkscreen!

Want to reverse the direction of the rotary encoder's values? Simply swap the pin definitions in the set crank pins block!
Fishing Game in MakeCode Arcade

Gone Fishing

Let's load the completed version of Gone Fishing and try it out. Download the arcade-Gone_Fishing.png image cartridge file here 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 Gone Fishing .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 the Game

Download the game onto your PyGamer (it won't play in the simulator window in the browser due to the use of external peripherals).

Here's how you play:
Casting

- pull back (down) on the thumbpad to lock the reel
- make a casting motion and shake the PyGamer (you'll hear a tone play)
- release the thumbpad to finish casting

Now, wait for a fish to swim by. You can reel the hook up and down with the crank, and you can move the hook and line side to side a bit randomly by clicking the rotary encoder’s push switch.

Reeling

Once you’ve hooked a fish start reeling it in by rotating the crank forward. You need to reel at a medium pace or you’ll loose the fish! The word "GOOD" will display while you're at a good speed, otherwise "TOO FAST" or "TOO SLOW" will pop up and you'll need to correct your speed. If you crank at the wrong speed for too long, the fish gets away and you loose one of five hearts.

How it Works

Here are the key features of the gameplay mechanic:
On Start

In the setup, we provide the game splash screen as well as the instruction text blocks.

Next, we'll set the crank pins as we did on the previous example. (You remember the one with the cake, right? Yummy.)

set pull pin
Since we loaded the Feather extension, we have access to the IO pins on the board. We can use the rotary encoder button (momentary switch) which we plugged into the PyGamer's D9 pin.

In the Pins category, you'll find the set pull pin __ to down block. Here we've chosen pin D9 and set the pull up resistor direction to up. This means the pin voltage will be normally pulled to high, and when the button is pressed the pin will drop to low. Later we'll see how to read this state of the button press.

Casting Variables

We set up a few variables to keep track of the reel casting motions, as well as the state of a fish being hooked or not. These are used later to determine things such as which sprites to display, location of the hook and line, and more.

Background

We also create a background image sprite for our game screen. Note how the water color has been created with a checkerboard dither pattern of dark blue and light blue, which is done to match the water layer we'll look at next so the two transition seamlessly into each other at the top of the water's edge.
Water Layer

The water layer sprite is also a checkerboard dither pattern, but this time the pattern alternates between light blue pixels and transparent pixels. The overall effect in the game when this sprite is set as the topmost layer in the Z-depth of the screen is that the fish swimming underneath it will appear as if they're under a semi-transparent layer. Since we can't do partial opacity pixels in MakeCode Arcade, this is a useful trick!

When a fish is hooked and reeled in, it will emerge from the water (as it is raised on the Y-axis above the topmost water pixels).

Additionally, as the fish move, they will appear to have a bit of extra secondary motion for free, as the details on the fish move under the blue pixels and then reappear in the transparent spots.

Painting with Code

In order to create the large dithered water layer, rather than paint one pixel at a time, or use an external image editing tool, I simply did some copying and pasting of the code in the JavaScript tab at the top of MakeCode Arcade.

This is what it looks like:

```javascript
let waterLayer = sprites.create(img`

9 .

9 .

9 .

9 .

9 .

9 .

9 .

9 .

9 .

9 .

9 .

9 .

9 .

9 .

9 .

9 .

9 .`, Sprite.resize(128, 128));
```
Hook and Line
The hook and line sprites are made of two separate sprites so we can detect collisions between the fish and the hook, while ignoring the line.

All position settings when casting and reeling will be applied to the hook, while the line will simply follow along. To keep this simple, both sprites are the same dimensions, even though the hook is much smaller than the line vertically.

Spawn Fish
We've used two functions with associated on game update every ____ ms blocks to create our fish with some randomness of spawn rate and velocity.
Casting

In order to cast the line, it's a three step process. The on down button pressed block begins the process by checking that the line isn't already cast, then sets the triggered variable to true.

The prevCrank variable is used to store the current value of the crank position of the rotary encoder, which will be used later to offset the raw value from where we want to place the hook on the screen.

We set the hook and line sprites to blank, as if they're over the player's head and behind them as they prepare to cast the line. We also play a low tone.

When the player shakes the PyGamer, the on shake block flips the thrown variable to true so that the game will then pay attention to the release of the down button. It also plays a high tone.

Finally, when the on down button released block runs it resets the thrown and triggered variables, sets the hook to a random position on the X-axis, makes the sprites visible again, and plays a different tone. It also sets the cast variable true, which is what allows the next important code to run.

If the player wants to try moving the line side to side, they can press the rotary encoder push button in. This will cause the digital read pin D9 block to return a Low or false signal, and then the hook x value will be set to a random position.
Cranking and Hooking

This forever loop is always running. When the game starts, the cast variable is false, so the hook and line are set near the top of the screen.

The hook and line are sent to the bottom of the lake when cast. Then if a fish is hooked, the sprite will change from a swimming fish to a hookedFish sprite. It is set to auto destroy so that it will disappear when successfully reeled in.

When the line has been cast, we set the crankScaling variable to 4 (you can adjust if needed) and is used as a multiplier on the number of encoder rotations needed to move the hook. As the rotary encoder is turned, the crank position is updated and the hook sprite's y position is set to the crank position minus the prevCrank position, divided by the crankScaling value and then an offset of 45 is added to get the hook lower down.

Hooking a Fish

When the hook player sprite overlaps either an orangeFish or yellowFish if one hasn't already been hooked and caught, the original fish sprite is destroyed so the hooked fish sprite can take its place. We also set the fightCounter to 0 which will be used during reeling to determine when the fish or fisherperson has won!
Cranking Battle

Trying to reel in a fish can be hard! To simulate this in the game, we use this game update every 150ms block and check if the player is rotating too fast (more than 10 crank position value changes per 150ms) or too slow (fewer than 5 crank position value changes per 150ms).

Note that these values are set to negative values because of the pin order of the rotary encoder. If the encoder pins were swapped these would be positive value changes.

At each 150ms check, if the change value isn't in the 5-10 range, then the fightCounter is increased by 1. When the fight counter is greater than 30, the fish gets away!

Fish Catch or Escape

We'll use the on destroyed sprite of kind Food to react to a fish either being reeled off the screen or getting away.

First, we'll reset all of the casting checks.

Then, if the fish gotAway we play the sad power down sound, splash the text "That one got away.", and subtract one from the life counter.

If the fish didn't get away, then we increase the score by 1 for orange fish or 2 for the rarer yellow fish.
The beeping sound is played, and then we show the long text congratulating the player on catching the fish, and reporting the fish’s weight from a random number to text conversion block.

Everything is reset and they can start fishing again!

---

**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. ()

---

**PyGamer Bootloader**

If you have a PyGamer, please go to this page for instructions on updating the bootloader. ()

---

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
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
- PyGamer Hardware Check

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](#) 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
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!
Then, click on the Show Code button in the upper left corner. The 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.

Download completed...

Move the .uf2 file to the ARCADE drive to transfer the code into your Arcade.

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 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](#)
- [Adafruit MakeCode Forum](#)
• Microsoft MakeCode Arcade Discord ( -- look for the #arcade channel
• Adafruit MakeCode Discord ( -- look for the #makecode channel

Only use the Google Chrome browser with MakeCode!