MakeCode Arcade: Sparky Invaders

Created by John Park

https://learn.adafruit.com/makecode-arcade-sparky-invaders

Last updated on 2023-08-29 04:07:54 PM EDT
# Table of Contents

## Overview
- Parts

## Create Sparky Invaders in MakeCode Arcade: Setup
- Sparky Invaders Gameplay Goals
- Design Goals
- Start at the End
- Load the Code
- Code Block Comments
- Part I: Setup
- Block by Block
- On Start Loop
- SparkySetup Function
- tileMapSetup Function
- Shots Fired
- Meet the Enemy
- Chip Animation

## MakeCode Arcade: Gameplay
- Part II: Gameplay
- Collisions
- Your Aim is True
- Shot Destroyed
- Sparky’s Been Hit!
- Chips Reach Safety

## 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

Pew pew pew! You can make your own Space Invaders-esque shooting game with an Adafruit twist: Sparky Invaders!

Our own Blue Smoke Monster wants to destroy as many sensitive IC (integrated circuit) chips with his ESD (electrostatic discharge) bolts as possible. Learn how to use MakeCode Arcade to create the gameplay code, graphics, and sounds for this fun, simple game.

Then transfer the game to your Pybadge or Pygamer to play it on the go, hand-held style.

Parts

Adafruit PyGamer Starter Kit
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
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

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

Adafruit Circuit Playground Lanyard
We've got our Circuit Playground friends on lunchboxes,
https://www.adafruit.com/product/3987

Lithium Ion Polymer Battery - 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/2750
Create Sparky Invaders in MakeCode Arcade: Setup

Let's make a game! If you're just getting started making games with MakeCode Arcade, you should check out the fundamentals in these guides first:

- MakeCode Arcade Pixel Art
- MakeCode Arcade Animation
- MakeCode Arcade Platformer Level Design

Once you're ready, come back here to get started with our Sparky Invaders game!

Sparky Invaders Gameplay Goals

Taking a cue from Space Invaders, here are some gameplay goals for Sparky Invaders:

- Arcade-style shooter
- Player character constrained to screen bottom, can move sideways only
- Enemies come in waves from top of screen to bottom
- Player can shoot only one projectile at a time, no spamming the fire button!
- Points are scored for shooting enemies
- Enemy collisions with player character cause loss of one of three lives

Some game elements are beyond the scope of the simple version we'll be creating. For one, we won't have the enemies shooting back!

We'll have some barriers on screen, but won't deal with their gradual destruction from missile damage. These barriers are strong!

No enemy flying saucers will grace our skies either.

**Design Goals**

Rather than a nondescript, mobile gun thing, our game will feature a fun, nefarious character as the player sprite -- Adafruit's own Sparky, the Blue Smoke Monster! Here he is doing what he does best: destroying a delicate integrated circuit (IC) chip.
We'll stick with the space theme, and add a moving starfield, which is easy to do in MakeCode Arcade.

Additionally, we'll style the enemies as DIP-14 IC chips.

Start at the End

Unlike our previous MakeCode Arcade guides in which we built our programs from a blank canvas, we'll start this one at the end, with our finished code. This will allow us to look at how each section works, without building every block from scratch.

Start by launching [MakeCode Arcade beta](https://makecode.microbit.org/) using the Google Chrome web browser. Then, download the MCA_Sparky_Invaders.png file by right-clicking on the image below 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).

This will open the code into the MCA editor.
Code Block Comments
A neat feature of MakeCode is the ability to add comments to your code blocks. This helps you to remember how and why something works, and to explain it to others who may take a look later.

By right-clicking on a block, you can select the Add Comment menu item and then type in your comment.

To show or hide the comment, simply click the word bubble icon on a block to toggle it.

Here’s our game code with comments turned on. You can have a look at it now to get a general idea how it works, then we’ll take a closer look at it section by section.
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!

### Block by Block

Now that we have the code loaded up, we'll take a look at each section to see how it all works.
Part I: Setup

On Start Loop

The on start loop runs one time when the game starts up. Which is to say, all of the blocks contained within it run one time.

First, we create the star field background effect with the start screen effect block.

Then we call two functions, the SparkySetup and tileMapSetup. We’ll have a closer look at both of those in a moment. The reason for using functions in this case is simply to keep the on start block from becoming a mess of too many blocks!

The code on those two functions could live here, but by putting them into functions we create some nice organizational order to sets of blocks that belong together.

Finally, we create a variable called shotFired and set its value to false. This will be used later to set the state when we have Sparky shoot an ESD (electrostatic discharge) projectile. This will allow us to prevent more than one ESD shot from being fired until the previous one has either hit something or gone off the top of the game screen. No spamming the fire button!!
SparkySetup Function

Since this function is called during the on start loop, the code in this block is effectively run just one time at game startup.

In it, we create our player sprite, Sparky, using the set mySprite block, and then renaming mySprite to Sparky in the dropdown menu.

We create simple Sparky pixel art in the sprite editor, at the default size of 16x16.

And, make sure the sprite kind is at the default of Player, since Sparky is our player character sprite.

We'll use the set position block to place Sparky at the bottom center of the screen on x=80, y=108 (the screen origin 0,0 position is the upper left corner).

The move with buttons block is a very quick and easy way to set up controls. By setting the velocity x to 100 and velocity y to 0, only the side to side motion will work.
By adding the set Sparky stay in screen block and flipping the switch to on we can constrain the player to the screen boundary.

The last part of the character setup is to add the set life to 3 block. You can tune that value to greater or fewer lives if you like. This will automatically add three heart icons to the top left of the game screen and we will subtract one from the total each time the character loses a life by being hit by an enemy Chip.

tileMapSetup Function
We'll use the tileMapSetup Function to create the barriers near the bottom of the screen.

The set tile map block determines where any of 15 color indexed tiles are placed on the screen (see this guide for lots more info!) We'll place two green tiles as shown here.

Then, the set tile to sprite with wall block is flipped to the same green color index we used in the set tile map block to indicate the position of the barriers.

In the sprite editor, you can draw out your barrier art.

Then, flip the wall toggle switch to on. Sprites cannot pass through walls, so we can block the enemies and Sparky's shots from passing through.
Shots Fired

Now, we'll set up the fire button control. The on A button pressed loop will run all the code contained within it every time the A button is pressed.

Remember we said in the gameplay goals section how we wanted to mimic Space Invaders in the firing mechanism. Only one shot is allowed on-screen at a time, so no spamming the button!

Here's how we do that. The if...then loop is the first thing to happen when the fire button is pressed. It tests if not shotFired which is a way of saying, is the shotFired variable set to false? When that's the case, it means there is not currently an ESD projectile in the scene, so it's fine to proceed. Otherwise, nothing happens.

To create the shot (when it's allowed) we create a projectile named ESD. Here's the sprite editor for the simple 8x8 pixel projectile.

It is set with a velocity of 0 on vx and -100 on vy which means the projectile will fire straight up at a moderate speed. You can experiment with other values if you like.

We'll also play a little tone pair for each shot, and then set the shotFired variable to true. That means another shot can't be fired until some action causes the variable's state to flip to false, which we'll handle later.

Meet the Enemy

Now, we'll generate the waves of enemies!
We'll use the on game update every 3500 ms loop to create a wave of three Chip enemies every three and a half seconds.

Here we'll first create a projectile for our first Chip with the set to projectile from side with vx vy block. Change the variable name to Chip first.

The reason to make Chip a projectile is that it simplifies the movement behavior setup necessary, vs. a regular character sprite.

Create pixel art for Chip such as the one shown here.

The vx = 2 and vy = 10 values mean that Chip will travel slowly to the right and more quickly downward.

Next, set Chip to kind Enemy from the dropdown menu. This makes it easy to set up the collision behavior later for ESD shots hitting them and other cases.

We want this first of three Chip "projectiles" to be emitted from a randomized spot within the first third of the screen if we break it up into conceptual columns. The set Chip x to block tells the game engine from which position to emit the projectile. By using the pick random 0 to 49 block we limit it to the first "column".

Then, we call the chipAnim function in order to animate the Chip enemy sprite. Those details are contained in the function chipAnim block, which we'll look at next.
You can see that we replicate these blocks two more times, and adjust the x position so that three total enemies are created each time the game update every 3500 ms loop is run.

Chip Animation

We'll create a two frame animation cycle for each Chip enemy so that they wiggle their little DIP legs as they fly! If you're unfamiliar with sprite animation in MakeCode Arcade, have a look at this guide first.

The function chipAnim block contains all we need to animate one Chip enemy. By calling it for each enemy that is generated by the game, we can have them all animate with just these few blocks.

The set anim to create animation of Idle with interval 250 ms block creates the animation group that will contain our frames. Any name for the animation cycle will do, I just picked Idle from the dropdown. You can tune the interval speed to your liking.

Then, add two frames of animation. The first can be the same as the original sprite for Chip, the second simply moves the leg tips (feet?) back a pixel.

attach animation anim to sprite Chip and activate animation Idle on Chip mean that all of the Chip enemies will run this animation cycle at all times.

Now that we have our player sprite, game level map, and enemies set up, we can create the gameplay!
Coding the gameplay is pretty simple, since all of the gameplay pretty much amounts to checking for collisions between things and then adjusting the score, lives, effects, and sounds accordingly.

Note: in order to use the "sprite" or "otherSprite" parameters in the "destroy mySprite" blocks in the next sections, you must drag the parameter from the existing "on sprite of kind Player overlaps otherSprite" block to make a copy of the parameter name.
Your Aim is True
First, let's take care of the ESD shots. They are automatically destroyed when they hit a wall, so nothing special needs to be done for the barriers.

When the shots hit the enemies, we'll know it by using the on sprite of kind Projectile overlaps otherSprite of kind Enemy block.

In here we then destroy with fire effect the Enemy that has been hit (indicated by the otherSprite parameter).

We'll also get rid of the ESD shot with the destroy sprite block, then change the score by 100 points, and play a little tone pair.

Shot Destroyed
If you recall, we set up the fire button so it couldn't be rapid fired, by testing the shotFired variable. Here is where we flip that back to false after the current EDS is destroyed using the on destroyed sprite of kind Projectile block.

The ESD is of type projectile so it automatically disappears when it hits a wall, such as our barriers set in the tile map.
Sparky's Been Hit!
Here's how we deal with an enemy making it past the barriers and crashing into Sparky!

The on sprite of kind Player overlaps otherSprite of kind Enemy block runs its code when the collision occurs.

In it, we destroy othersPrite with ashes effect for 200 to remove the enemy Chip from the screen.

We have Sparky change life by -1 from the three we start with initially.

And then we play a low wobbly tone set.

Chips Reach Safety
If an enemy Chip reaches the safety of a barrier without being hit, we'll have the player lose 50 points, and make the Chip emit an effect. (Remember, Chips are actually projectiles, so they automatically disappear when the collide with walls.)

The on sprite of kind Enemy hits wall with the proper index color from our tile map is used to detect the collision.

We then use change score by -50 and then run the sprite start warm radial effect for 100 ms effect.

Our game is finished! Now, you can play it in the browser, or upload it to a Pybadge or Pygamer!

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 bootloader 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 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

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.

A HUUUUUGGE 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!
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!