

MakeCode Arcade with Raspberry Pi Zero

Created by Michal Moskal



Last updated on 2020-12-11 10:26:03 AM EST

What is MakeCode Arcade?



Microsoft MakeCode Arcade is a **web-based** beginner-friendly code editor to create **retro arcade games** for the web and for microcontrollers. In this guide, you will learn how to assemble your own Arcade hardware from different parts. MakeCode Arcade is [open source on GitHub](#).



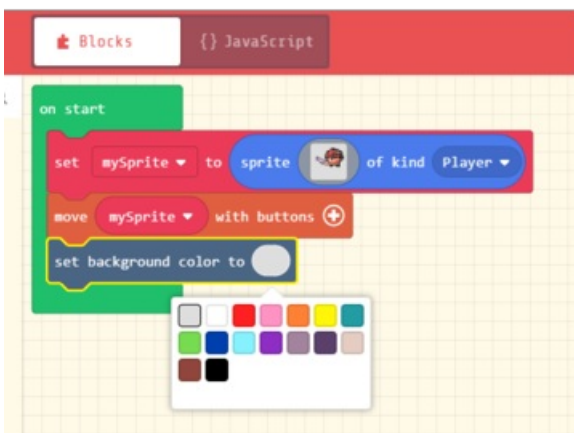
Arcade is for retro 2D sprite-based games.

Arcade supports a 160x120 screen with 16 colors, and has a 2D game engine with sprites, platformer-style collisions, tile maps, particle systems and more!



Arcade is a web-based code editor.

Just like the [MakeCode Circuit Playground Express](#) (<https://adafru.it/wpC>) editor, you can use blocks or JavaScript to build your program in your browser. Nothing to install - it just works!



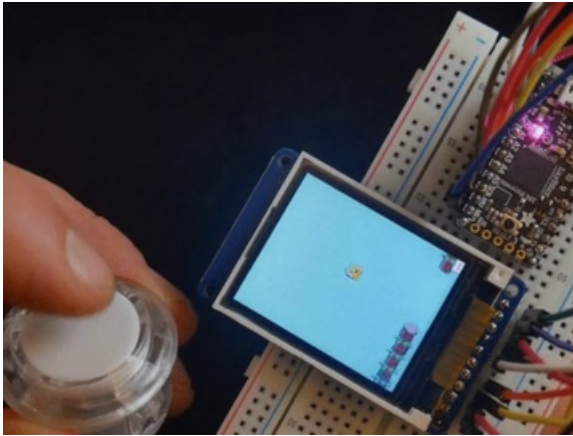
Arcade is friendly to beginners!

Get started with building your own retro games, using the block editor and the many guides. You don't need to be a programmer to get started!

```
1 enum SpriteKind {
2   Player,
3   Projectile,
4   Food,
5   Enemy
6 }
7 enum ActionKind {
8   Walking,
9   Idle,
10  Jumping
11 }
12 let bottomImage: Image = null
13 let projectile: Sprite = null
14 let topImage: Image = null
15 let anim: animation.Animation = null
16 let gap = 0
17 let mySprite: Sprite = null
18 controller.anyButton.onEvent(ControllerButtonEvent.Pressed, function () {
19   mySprite.vy = -100
20   animation.setAction(mySprite, ActionKind.Jumping)
21 })
22 sprites.onOverlap(SpriteKind.Player, SpriteKind.Projectile, function (sprite, otherSprite) {
23   game.over()
24 })
25 scene.setBackgroundColor(9)
26 mySprite = sprites.create(img' --
44 mySprite.y = 300
45 anim = animation.createAnimation(ActionKind.Jumping, 25)
46 anim.addAnimationFrame(img' --
64 anim.addAnimationFrame(img' --
82 anim.addAnimationFrame(img' --
100 anim.addAnimationFrame(img' --
118 anim.addAnimationFrame(img' --
```

Arcade is friendly to seasoned programmers too!

Switch to JavaScript and type away. MakeCode supports [Static TypeScript \(https://adafru.it/DCX\)](https://adafru.it/DCX), a special flavor of JavaScript optimized for compilation on micro-controllers.



Arcade runs on hardware (in particular SAMD51)!

While all games can be shared and played via the web, the fun starts when you compile them for SAMD51-based boards, like the [Adafruit ItsyBitsy M4 Express](https://adafru.it/DCY).

<https://adafru.it/DCY>

<https://adafru.it/DCY>

Setup

This guide will show you how to build a MakeCode Arcade from a Raspberry Pi Zero or Pi 3A+. If you are looking for Adafruit M4 boards, try [this guide](#).



Raspberry Pi Zero setup

- a Raspberry Pi Zero (we won't be using WiFi, but it's fine to buy the Pi Zero W if you want)
- a HDMI to mini-HDMI cable or converter
- a microUSB cable
- a micro SD card (capacity and speed don't matter much)

If you want to get all in a single kit, get the budget packet. If you don't feel like soldering, get the **Pi with headers**.

1 x [Raspberry Pi Zero W Budget Pack - Includes Pi Zero W](#)

All you need on the Raspberry Pi Zero side

Add to Cart

1 x [Raspberry Pi Zero WH \(Zero W with Headers\)](#)

Great combo with the Bonnet

Add to Cart

1 x [Raspberry Pi Zero - Version 1.3](#)

For the experts, add headers, power cables, HDMI cables and converters as needed!

Add to Cart



Raspberry Pi 3A+ setup

You can use Pi 3A+ instead of Pi Zero. It always comes with headers and you don't need a HDMI to mini-HDMI cable. On the other hand, you will need USB-A to USB-A cable.

1 x [Raspberry Pi Model 3 A+](#)

Pi 3A+

Add to Cart



Controls

MakeCode Arcade uses 7 buttons. There are many options for buttons:

- **Adafruit Joy Bonnet** and likely a plastic case for Pi Zero
- **Arcade Cabinet**
- random buttons and joystick that you wire up together

You can also get **Joy Bonnet Pack with Pi Zero** which includes many of the components.

1 x [Adafruit Joy Bonnet for Raspberry Pi](#)

(optional) Easy pluggable buttons!

Add to Cart

1 x [Joy Bonnet Pack for Raspberry Pi Zero - Includes Pi Zero W](#)

A Pi Zero and Bonnet ready to go!

Out of
Stock

1 x [Raspberry Pi Arcade Cabinet Pack](#)

Can't beat the feeling of real arcade buttons

Out of
Stock

1 x [Arcade Button - 30mm Translucent Blue](#)

Just one of the many available buttons!

Add to Cart

Firmware



Firmware

If your SD card comes with files on it, delete them, or just format the SD card as FAT32. **Do not** format as ExFAT. Then download the ZIP file below, unzip it, and copy all files to the root directory of the SD card.

After this, if your SD card drive is named `D:`, then there should be file `D:\9.0.3.gz`, and if your SD card is called `/Volumes/NO NAME`, then you should have `/Volumes/NO NAME/9.0.3.gz`. There should **not** be an `arcade` folder on the SD card. The ZIP below above contains binaries built from <https://github.com/microsoft/uf2-linux> and <https://github.com/microsoft/pxt-arcade-cabinet-menu>

If you want to use Raspberry Pi Zero with Joy Bonnet, use the first file below. If you're building your own arcade with your own buttons and Pi Zero, use the second file.

If you want to use Raspberry Pi 3A+ (other models like 3B or 4 will not work), use the third file. It's setup for custom input by default, but if you want to use it with Joy Bonnet, just copy `arcade.cfg` file from the first zip over to your SD card.

<https://adafru.it/H4B>

<https://adafru.it/H4B>

<https://adafru.it/H4C>

<https://adafru.it/H4C>

<https://adafru.it/HAp>

<https://adafru.it/HAp>



Joy Bonnet

If you're using the Joy Bonnet, then congratulations, you're done!

Just connect the Pi using HDMI cable to a screen, and using the MicroUSB cable to the computer. After a few seconds you should see a menu much like the one below. Move to the next section to download games to your Arcade.



Custom Inputs

If you want to use a big joystick and buttons, it's quite easy!

Using standard connection

When you load up the `arcade-cardboard.zip` on your Pi, it will show you pinout to follow. You would generally connect buttons one by one, and see which one you have connected by looking at the screen. Then you know which cable you've got, and you move it to the right pin.

Once you're done connecting everything, press ABABAB and the configuration program will be replaced by the game menu program. This will be persisted after reset.

Using custom connections

The other option is to connect the buttons as you like and then tell your Pi where the buttons are connected. This is done by editing `arcade.cfg` file on the SD card.

If you have the Arcade Cabinet Pack, follow the [assembly guide](#). If you plug in the buttons the way it's described in there, you will need the following in the `arcade.cfg` file:

```
BTN_A=4
BTN_B=25
BTN_RESET=11
BTN_EXIT=5
BTN_MENU=16
BTN_LEFT=10
BTN_RIGHT=22
BTN_UP=23
BTN_DOWN=27
```

If you are building a custom player arcade cabinet (1 or 2 players),

```
BTN_RESET=4
BTN_EXIT=3
BTN_MENU=2
BTN_A=26
BTN_B=19
BTN_LEFT=13
BTN_UP=6
BTN_RIGHT=5
BTN_DOWN=0
BTN_A2=11
BTN_B2=9
BTN_LEFT2=10
BTN_UP2=22
BTN_RIGHT2=27
BTN_DOWN2=17
```

Feel free to connect them differently, just [follow these instructions](#) and update the file with the right GPIO numbers.

Downloading Games



Adding games

To add new games head to <https://arcade.makecode.com> !

Once you have created your game,

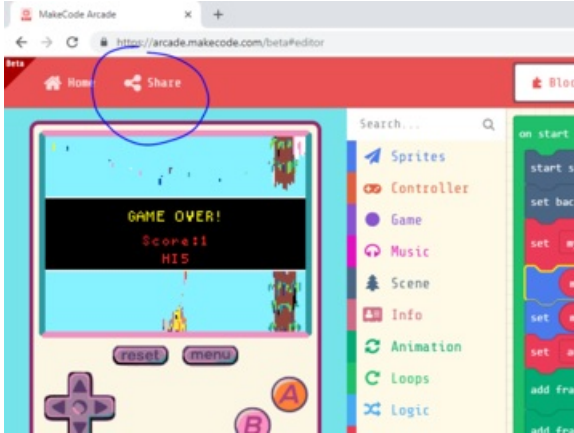
- Press the **Download** button and say you don't want to pair anything if asked. You should then be given options to select type of board.
- Select Raspberry Pi (it will have the image of the menu above). This should create a file named `arcacde-something.uf2` .
- Copy it to **ARCADE** drive that should be visible on the computer to which your Pi is connected with the USB cable.

If the **ARCADE** drive doesn't show up, make sure the USB cable is plugged into the USB port on the Pi that is closer to the HDMI port. The other one is power-only. If using a case, make sure it's plugged all the way in.

Sharing Games

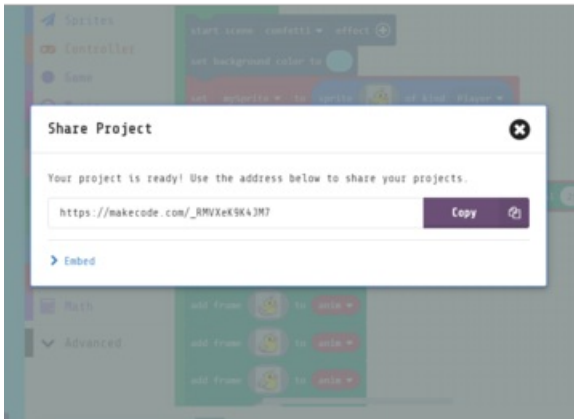
There are plenty of ways to share games with your friends, family and anyone else who you think might enjoy something you've created!

Let's quickly go over a few of your options.



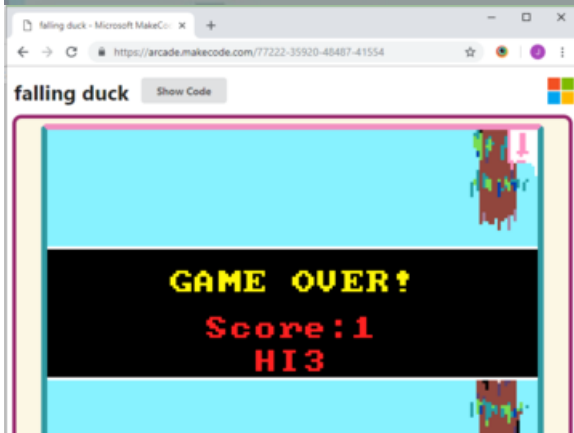
Online page

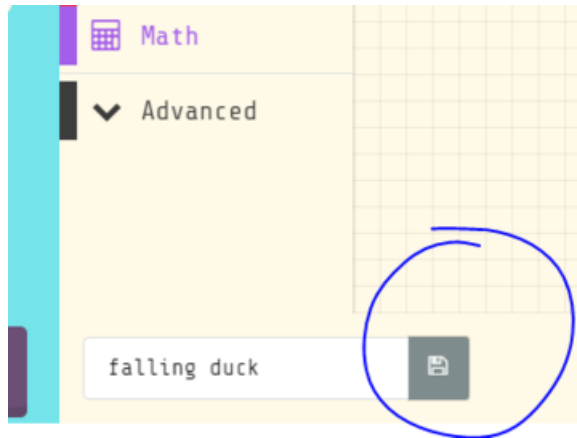
Click on the **Share** button and follow the steps to upload a version of your game online. It will generate a URL for you.



Anyone with the generated URL will be able to play in a full screen mode, or edit your game! They'll also be able to download it onto their own Arcade if they have one.

This page supports oEmbed and other various embedding options.





Save into an image

Click the disk icon will save your code into a PNG image. You can import the image back or simply drag and drop it into the editor to reload the code.

The image below contains the **falling duck** game.



falling duck
arcade.makecode.com

□ You will literally want to download the image itself - its a polyglot file that contains data within its pixels!

Downloaded .UF2 files

Downloaded .uf2 files also contain the game code. You can import them or drag and drop them into the editor to reload the code.

GitHub repo

You can also use GitHub repositories to store and share your code. This is particularly useful if you end up building a lot of code or if you want to share an extension with other users. Read all about getting started with GitHub at <https://makecode.com/extensions/github-authoring> .

