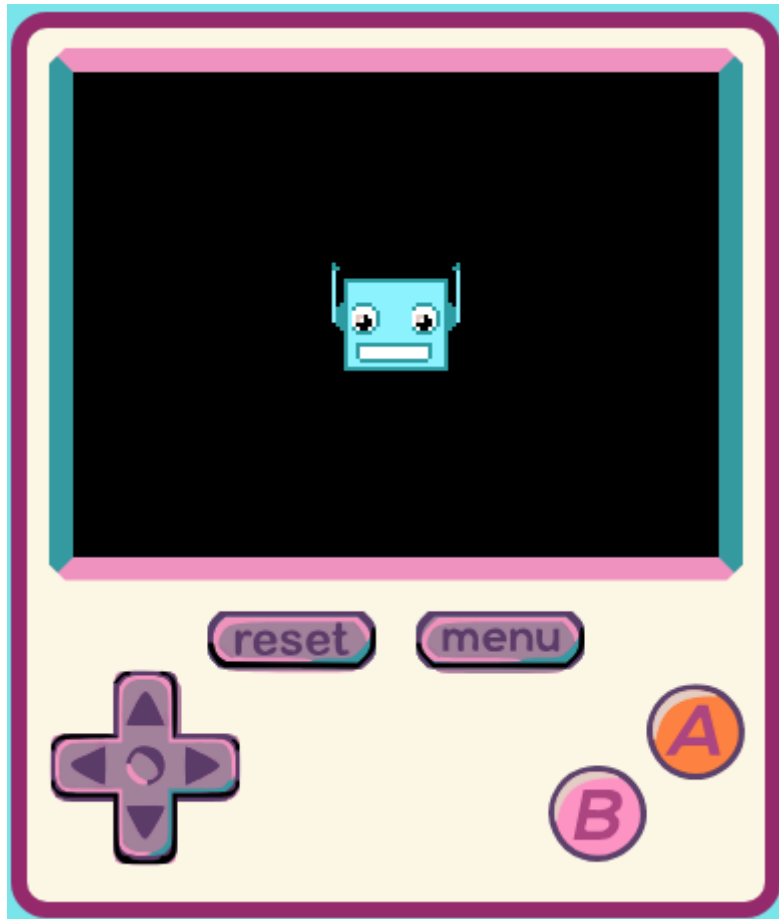




MakeCode Arcade Pixel Art Sprites

Created by John Park



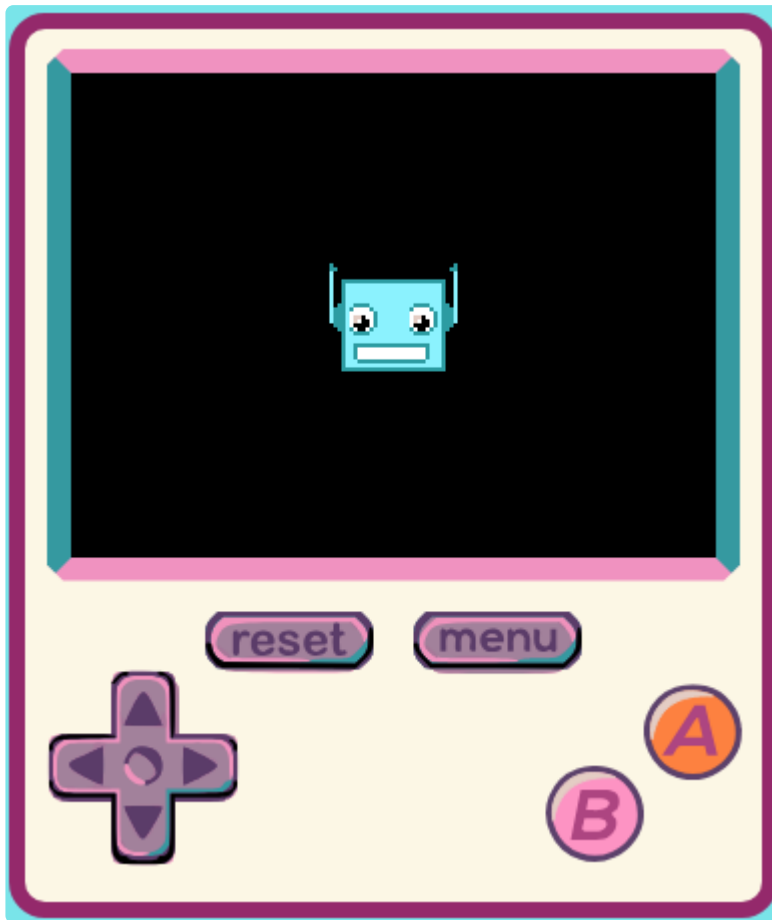
<https://learn.adafruit.com/makecode-arcade-pixel-art-sprites>

Last updated on 2022-02-14 12:40:27 PM EST

Table of Contents

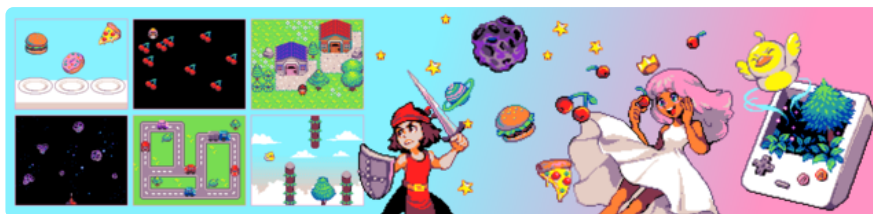
Overview	3
Pixel Art Fundamentals	5
• Pixels	5
• Pixel Art	6
• Software and Resources	7
• Color Palette	8
• The Canvas	8
• Sprite Sheets	12
• Drawing Sprites	12
• Anti-Aliasing	13
Create Sprites in MakeCode Arcade	15
• MakeCode Arcade	15
• mySprite	16
• Sprite Editor	17
• Bigger Ruby	19
• Controls	21
Update the PyBadge/PyGamer Bootloader	21
• PyBadge/PyBadge LC Bootloader	22
• PyGamer Bootloader	22
• Hardware Checks	22
Load a MakeCode Game on PyGamer/PyBadge	23
• Board Definition	24
• Change Board screen	25
• Bootloader Mode	26
• Drag and Drop	26
• Play!	27
Troubleshooting MakeCode Arcade	27
Update the PyBadge Bootloader	28
• Updating Your PyBadge Bootloader	28
• Oh no, I updated MacOS already and I can't see the boot drive!	31

Overview



Let's learn how to make our own pixel art character sprites! Many classic and modern video games use two-dimensional pixel art, a.k.a. sprites, to represent characters. The limitations on size and color palette can unlock worlds of creativity!

We'll cover the fundamentals of pixel art character sprites, as well as learn to make our own using MakeCode Arcade, and even upload our characters and move them around on the PyBadge!



[Microsoft MakeCode Arcade \(https://adafru.it/DD0\)](https://adafru.it/DD0) is a web-based beginner-friendly code editor to create retro arcade games for the web and for microcontrollers.

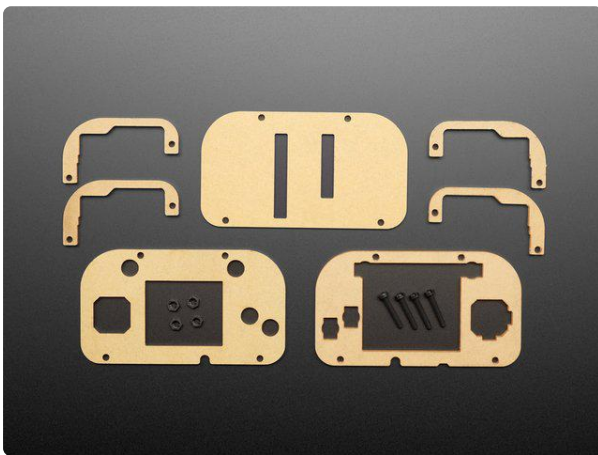
Beta Zone: MakeCode Arcade is still in its beta. It might still have a few rough edges...



[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...

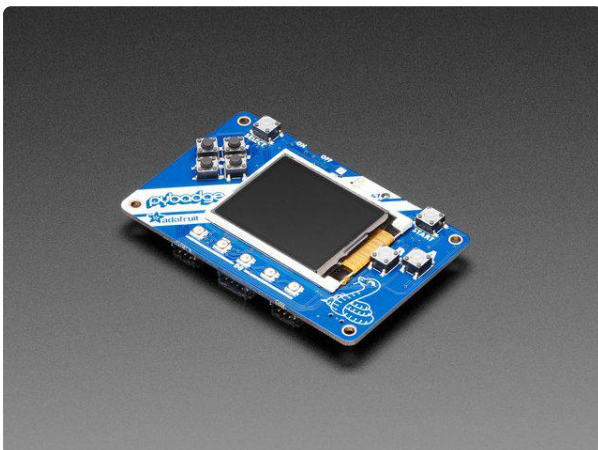
<https://www.adafruit.com/product/4242>



[Adafruit 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...

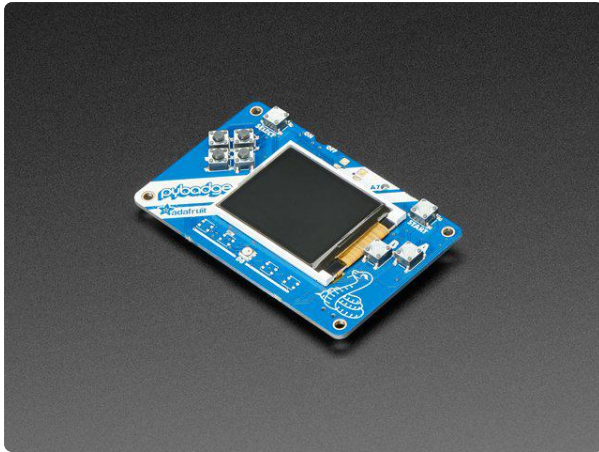
<https://www.adafruit.com/product/4238>



[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...

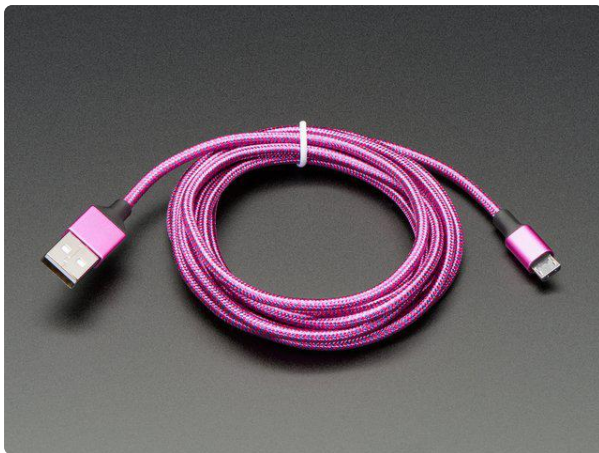
<https://www.adafruit.com/product/4200>



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...

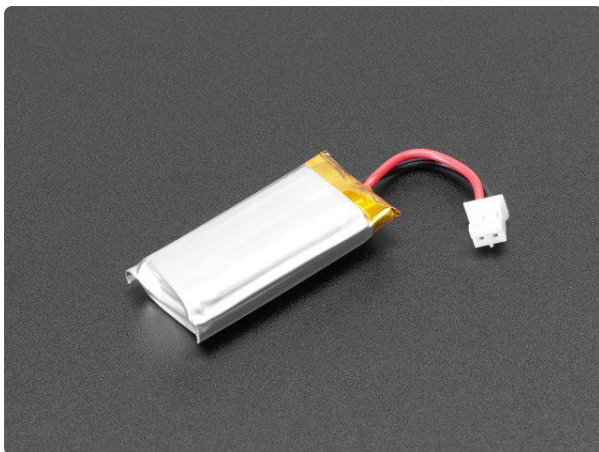
<https://www.adafruit.com/product/3939>



Pink and Purple Braided USB A to Micro B Cable - 2 meter long

This cable is super-fashionable with a woven pink and purple Blinka-like pattern! First let's talk about the cover and over-molding. We got these in custom colors,...

<https://www.adafruit.com/product/4148>



Lithium Ion Polymer Battery Ideal For Feathers - 3.7V 400mAh

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/3898>

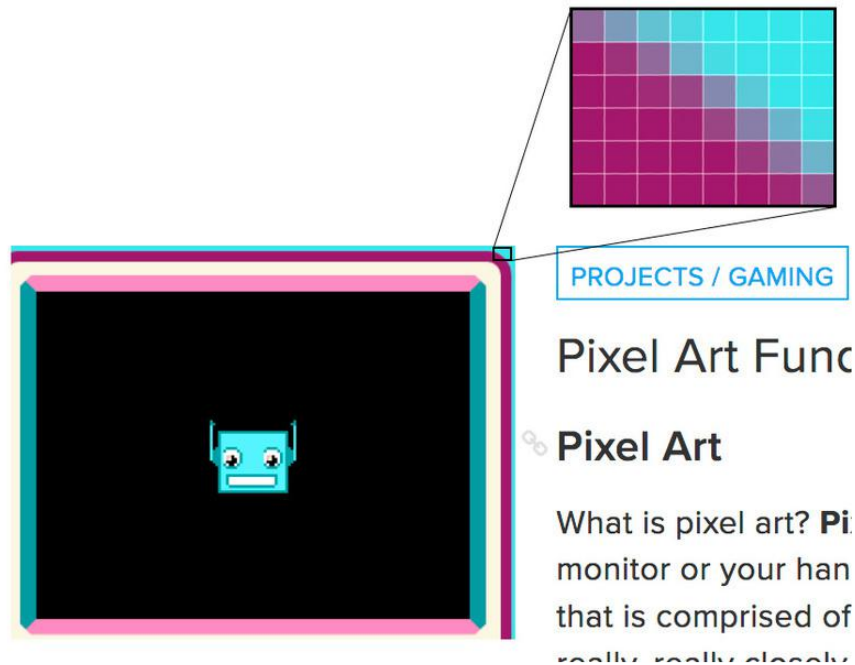
Pixel Art Fundamentals

Pixels

What is pixel art? Pixel is a word for individual “picture elements” on a screen, such as your computer monitor or your handheld gaming device. They are essentially the smallest unique element of a display that is comprised of a single color (this is true of most all displays other than vector scopes). If you look really, really closely at your

screen, you'll see lots of individual pixels that make up the images you normally see when you step back and take it all in.

Here's a closeup of a section of this very webpage so you can see the individual pixel that make up the guide's main graphic.



Pixel Art

For our purposes, pixel art refers to a type of graphics made for use in games, where the individual pieces of art (sometimes referred to as sprites) are small enough to be created by hand, one pixel at a time. This is as opposed to graphics made from pre-rendered or real-time 3D models, or graphics of sufficient complexity and size that it's impractical to create the art at the individual pixel level.

Here, we see pixel art from the game Celeste. The backgrounds and characters are artfully constructed pixel art. And incredibly appealing!



Software and Resources

On the next page we'll use the pixel art editor that's built right into MakeCode Arcade. However, if you'd like to explore other pixel art software, here are some good ones to check out:

- [Piskel](https://adafru.it/ECB) (<https://adafru.it/ECB>) is a free online editor for pixel art and animated sprites
- [Aseprite](https://adafru.it/ECC) (<https://adafru.it/ECC>) is a paid app (\$15) that many professionals use -- there's also a free trial, and it's open source, so you can [build the full version for yourself](https://adafru.it/EEk) (<https://adafru.it/EEk>) if you are inclined.
- [Pixilart](https://adafru.it/ECD) (<https://adafru.it/ECD>) is another very nice, free online editor

These [pixel art tutorials by MiniBoss](https://adafru.it/ECE) (<https://adafru.it/ECE>) are excellent (they created the fantastic artwork for Celeste). In them, they cover both static and animated sprites (and a whole lot more!)



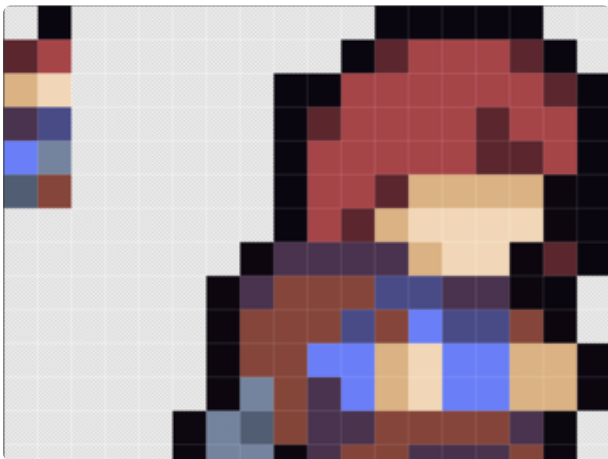
For a deeper dive into creating graphics for the Nintendo Entertainment System (NES) [have a look here](https://adafru.it/ECF) (<https://adafru.it/ECF>).

[This Game Development magazine article on creating retro styled games \(https://adafru.it/ECG\)](https://adafru.it/ECG) has a great look at different retro art styles.

Color Palette

Pixel art also tends to imply the use of a limited color palette, such as 1-bit (monochrome), 4-bit (16 colors), or 8-bit (256 colors) graphics seen in game consoles of yesteryear. Modern games tend to use 16-bit (65,536 colors) or 24-bit (16,777,216 colors).

Not to say that pixel art with limited color palettes aren't thriving now! But these color restrictions are now often chosen for stylistic reasons, not technological ones when using the latest, most powerful computers and consoles.



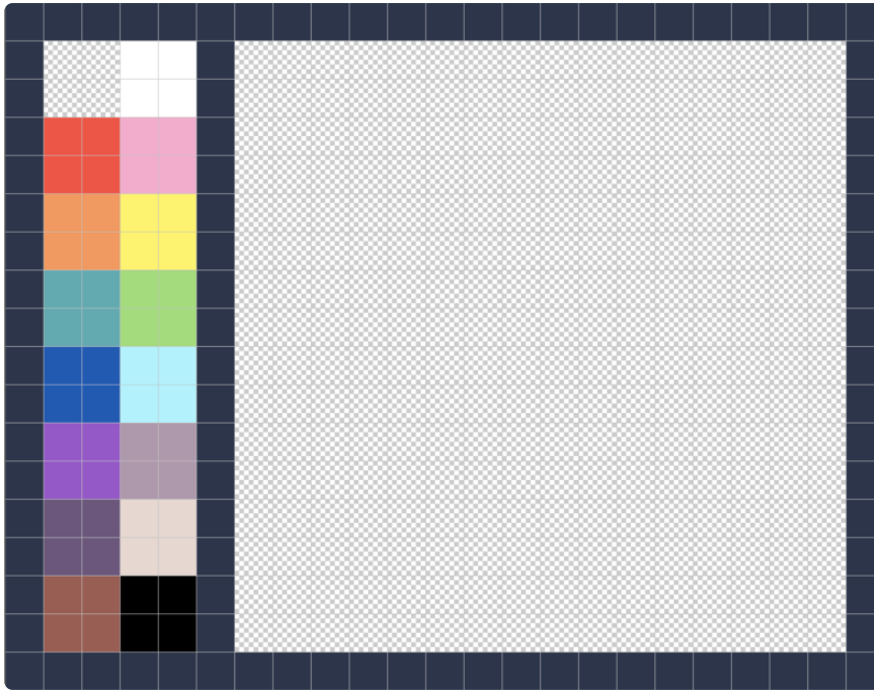
If we look closely at the Celeste character sprite from the scene above, we can see that she is made up of only twelve colors (including transparency as one "color").

On handheld systems, such as the PyBadge and PyGamer, we can create pixel art that will look great on the small screens when compared to the relative busy-ness of higher color depth graphics. (More on aliasing in a bit.)

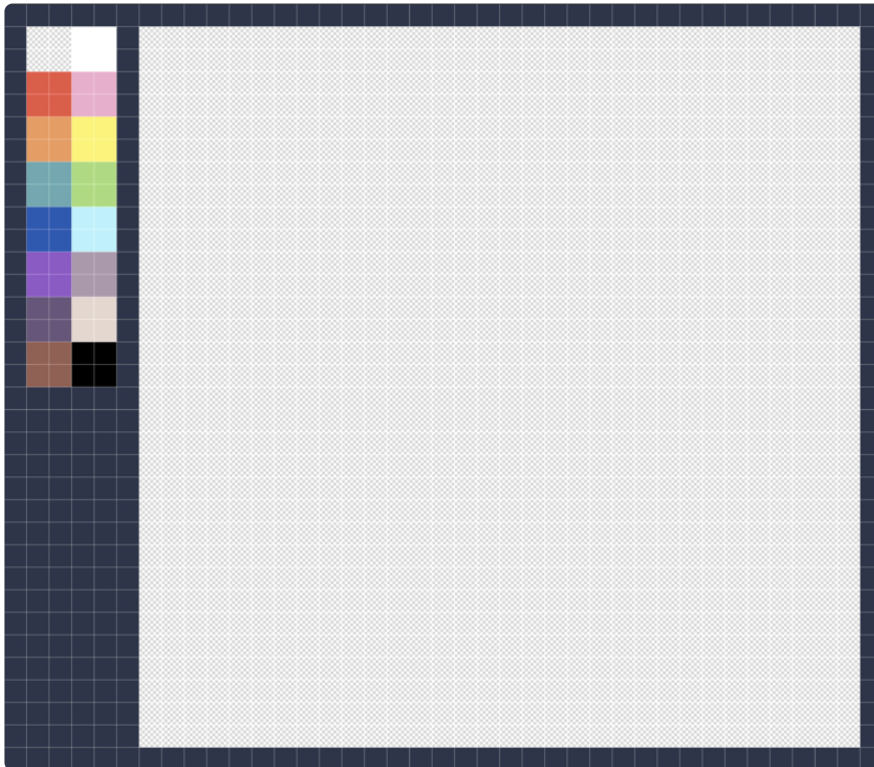
Want to check out some excellent examples of pixel art palettes? [Have a look over here at LOSPEC \(https://adafru.it/EHf\)](https://adafru.it/EHf)!

The Canvas

When creating pixel art characters, one of the first decisions you'll make is the size of the image sprite you're going to make. Here's a typical canvas set up for a 16 pixel wide, by 16 pixel high image.

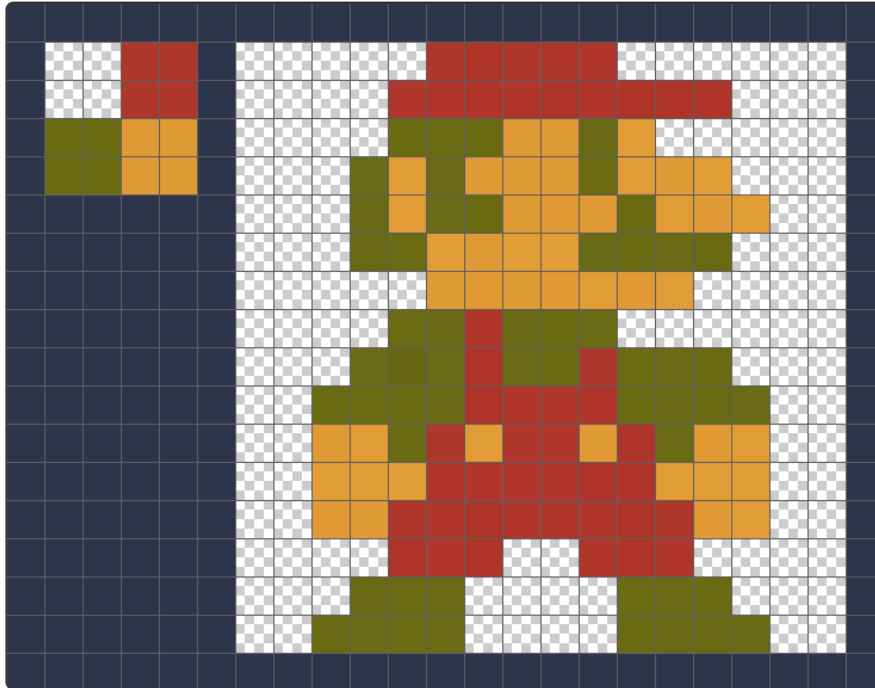


Here's a similar canvas, except this time its set for a 32x32 pixel image.



Each cell of the canvas can have a single pixel drawn into it, or no pixel at all. Cells with no pixel are treated as a transparent part of the image by the game engine.

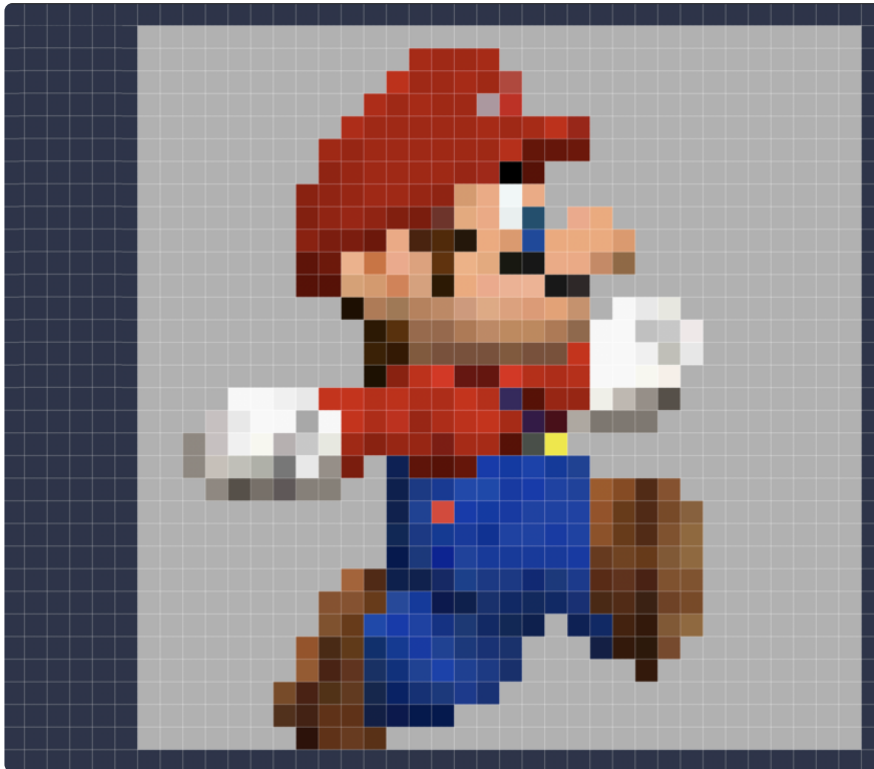
First, we have Mario from the NES Super Mario Bros. game He's made in a 16 x 16 pixel canvas and has a four color palette (one color being transparency).

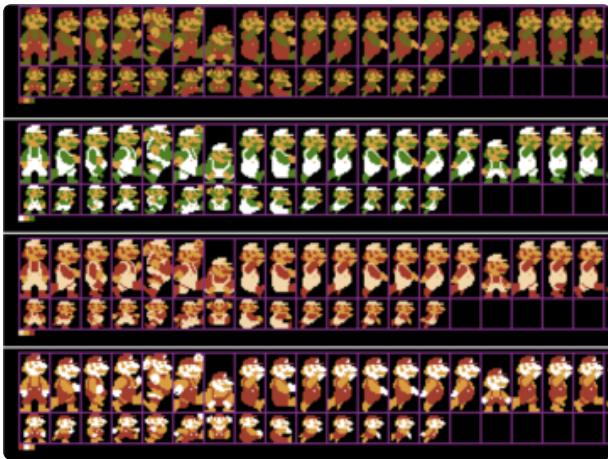


Next we have a 32x32 pixel Mario from SNES Super Mario World. He's now sporting a 16-color palette!



Finally, here's a 32x32 pixel sprite from New Super Mario Bros. for Nintendo DS. The sprite looks like it was not hand made, pixel by pixel, so much as rendered from a 3D model, since the shading is so complex. The color palette is virtually unrestricted, as evidenced by the huge number of different shades of blue, red, Mario's skin-tone, and more.





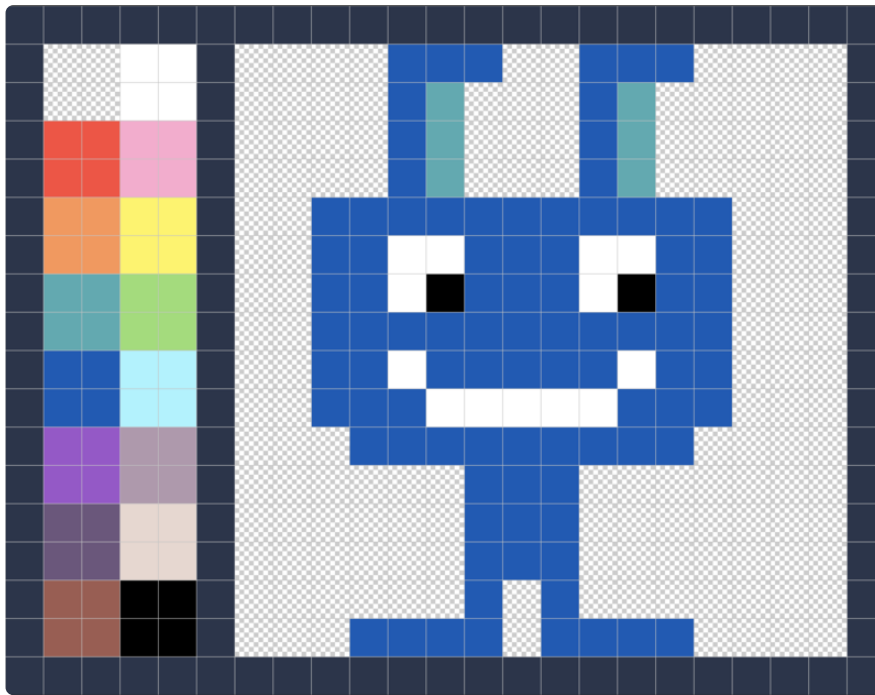
Sprite Sheets

If you'd like to look at examples of pixel art from existing games, search online for "sprite sheets". For example, "Mario sprite sheets" turns up images such as these, which are pulled directly from the in-game resources:



Drawing Sprites

In this example, we see a palette of 16 colors (the first one isn't a color, it allows you to make a transparent pixel) and then we have a grid of 16 x 16 pixels for the art. You can pick a color and then use the pencil tool, for example, to click and place pixels where you like.



Here's a very nice example from the [MiniBoss Studio tutorials \(https://adafru.it/ECE\)](https://adafru.it/ECE). In this sprite art progression for a shovel, note how the limited color palette is still enough to

- first, create the basic form of the object
- then, "shade" it (add dimension) using additional colors
- finally, to round off the appearance of the corners using anti-aliasing techniques

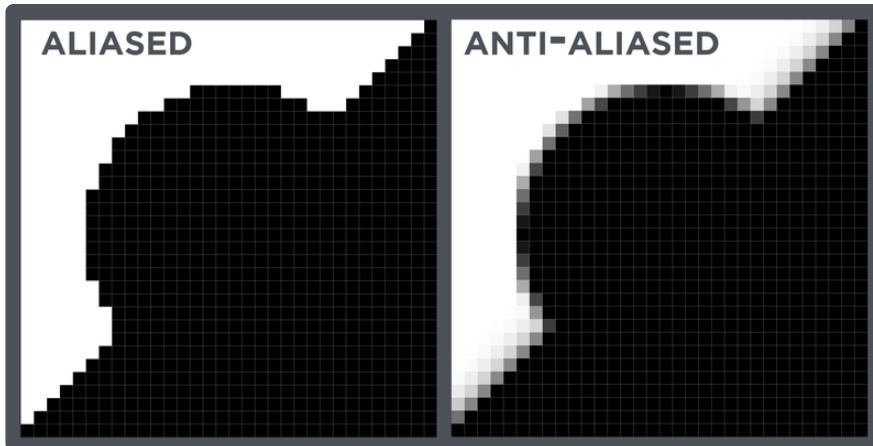


Anti-Aliasing

Aliasing is the name given to the rough, stair-stepping effect that's seen when pixels are used to create curved or diagonal lines. While we can't change the basic problem of square (or rectangular on some screens) pixels being used to represent curved or diagonal shapes, we can lessen the effect by using a technique known as anti-aliasing.

Typically, anti-aliasing is achieved by adding medium contrast pixels at the problem areas. In other words, an aliased black line on a white background can be anti-aliased by the addition of grey pixels to help fool the eye.

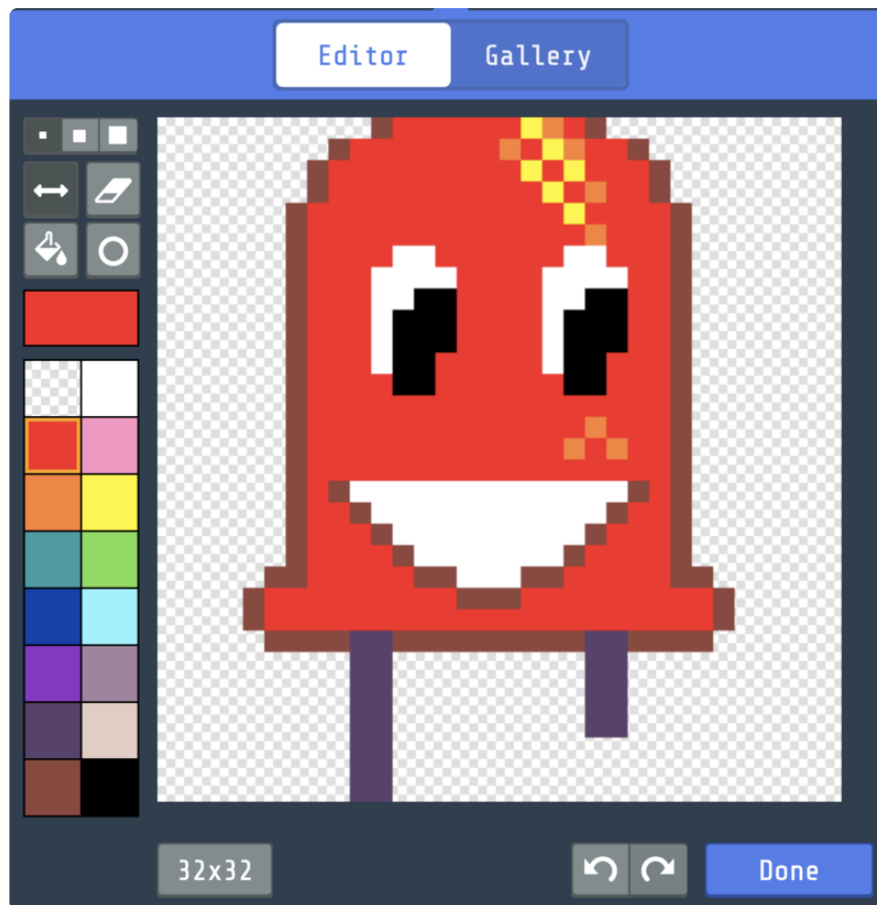
In this example, we start with two colors, black and white. To fix the aliasing, we add pixels in four shades of gray.



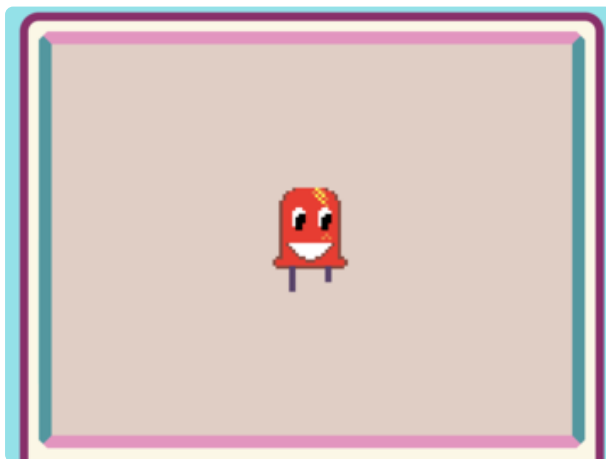
This can look a bit strange and fuzzy up close, but look at these same images at actual scale to see how effective the anti-aliasing is at removing the "jaggies".



Create Sprites in MakeCode Arcade



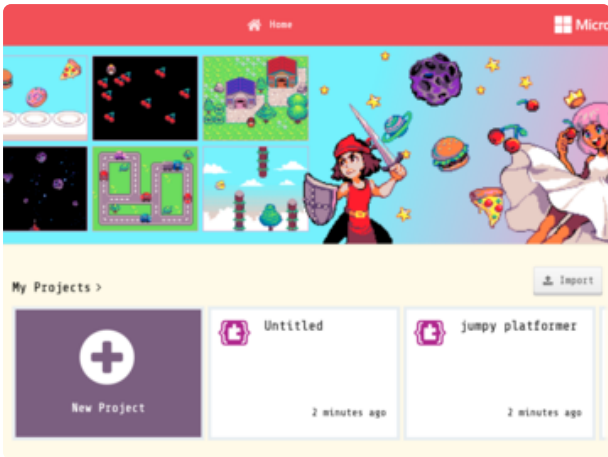
Now that we understand the fundamentals of pixel art, let's make our own character sprites inside of MakeCode Arcade!



Now that we understand the fundamentals of pixel art, let's make our own character sprites inside of MakeCode Arcade!

MakeCode Arcade

First, open a new Chrome browser window (Chrome works best) and go to [MakeCode Arcade \(https://adafru.it/DD0\)](https://adafru.it/DD0).



Next, click on the New Project button.

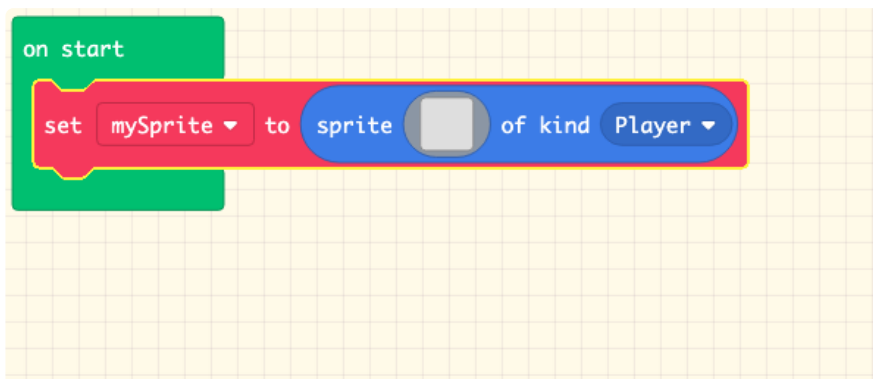


This will take you to the MakeCode Arcade Editor.



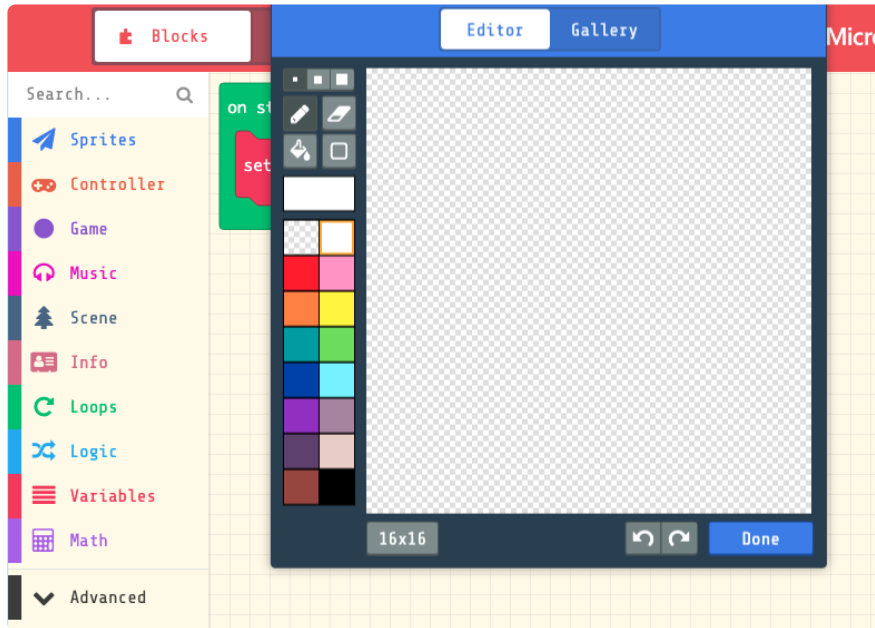
mySprite

Let's create a character sprite. In the Sprites category, click on the set mySprite to sprite of kind Player block and drag it onto the canvas.



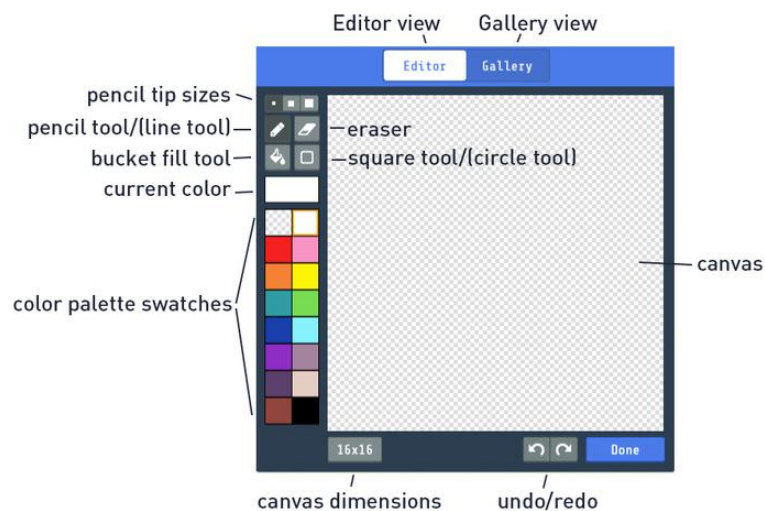
Sprite Editor

This creates a sprite object that you'll be able to control in the game. But how do you actually make the pixel art for it? Click on the gray box to the right of the word `sprite` in the block you just added and MakeCode Arcade will launch the built in sprite editor!



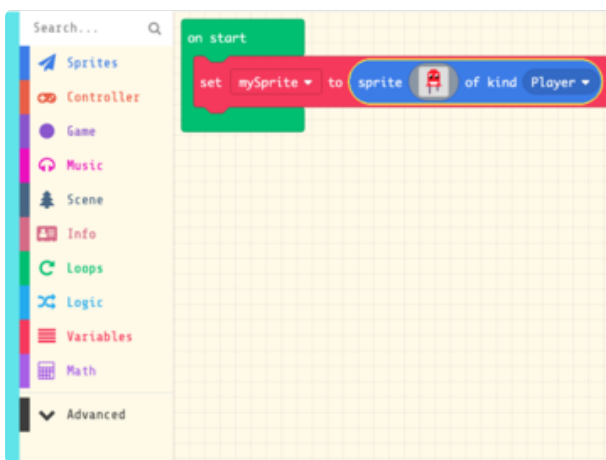
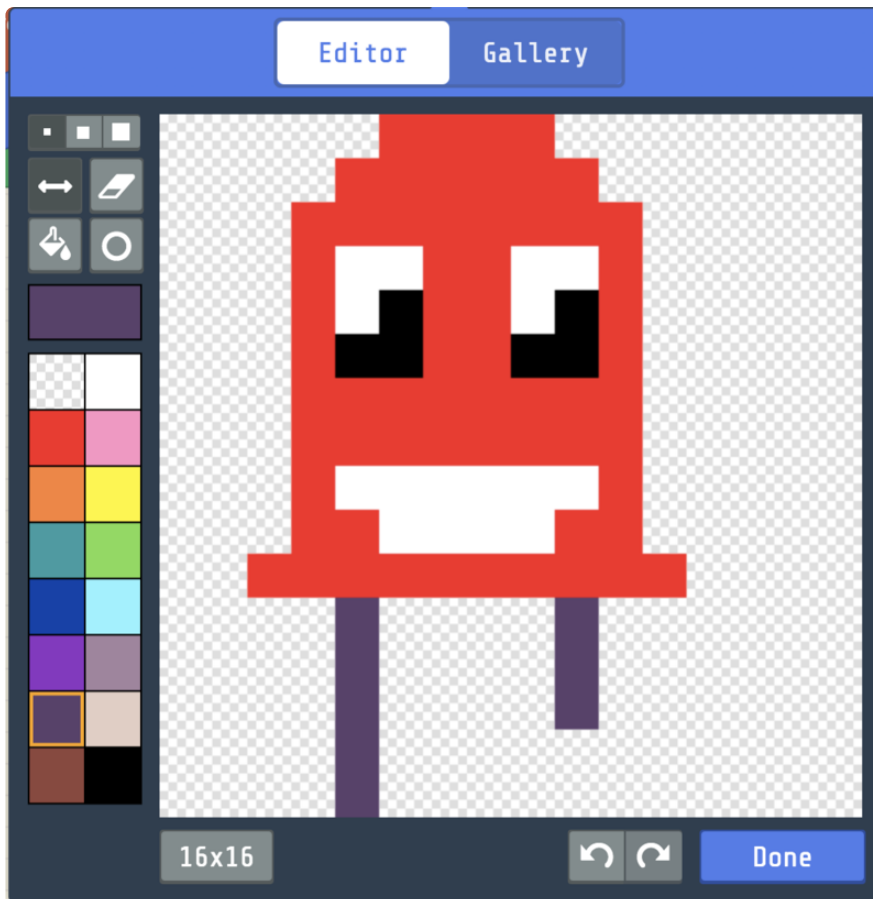
The default canvas size for your sprite is 16x16 pixels, as shown on the bottom of the sprite editor. You can click that number to cycle between a few presets. If you do, keep clicking until you get back to the 16x16 setting.

Next, let's have a tour of the editor.



These are the tools and interface elements of the sprite editor:

- Canvas is where you draw your sprite
- Editor and Gallery buttons allow you to switch between editing sprites and selecting from a set of pre-made sprites
- Pencil tip sizes allow you to switch between 1x1, 3x3, and 5x5 pixel tip sizes. These also apply to the eraser, and square/circle tools
- Pencil tool is used for drawing pixels in the canvas with the currently selected color. Just click or click-drag to use! It has an alternate behavior if you press the L on your keyboard -- it turns into a straight line tool
- Eraser is used to remove pixels from the canvas
- Bucket fill tool will flood fill an area with a color. The boundary of the fill is defined by the color of the pixel under the tool when the mouse button is released
- Square tool creates a rectangle defined by the click-drag-release points of the mouse cursor. The stroke width of the shape is determined by the pencil tip sizes. Press the C key on your keyboard to create a circle instead. Note: circles work best on larger canvas sizes
- Current color indicates the color you've got picked
- Color palette swatches are the colors you can use in your sprite. Click one to make it current. Note: choosing the transparent color allows you to erase pixels with the pencil/line, bucket, and square/circle tools
- Canvas dimensions determines how many pixels you'll be able to use for your sprite. Click this button to cycle between presets
- Undo/redo allow you to go back in time and fix mistakes! Or, change your mind and go back to the future!!
- Done press this button when you're finished and are ready to go back to the main MakeCode Arcade program



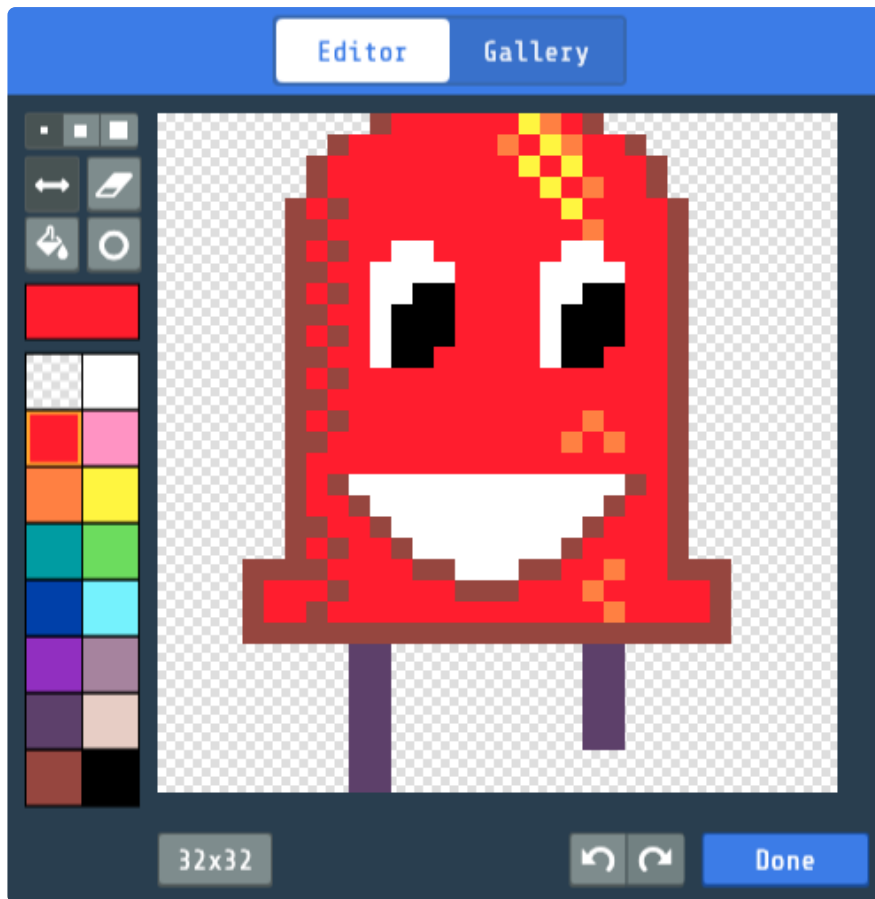
Try out the sprite editor now and create your own version of Ruby the red LED.

This sprite was created using the pencil tool and four colors. You can try out other tools, such as the paint bucket to flood fill Ruby to a different color! Or maybe the line tool to add some stylish stripes.

When finished, press Done. MakeCode Arcade will load your player sprite into the simulator.

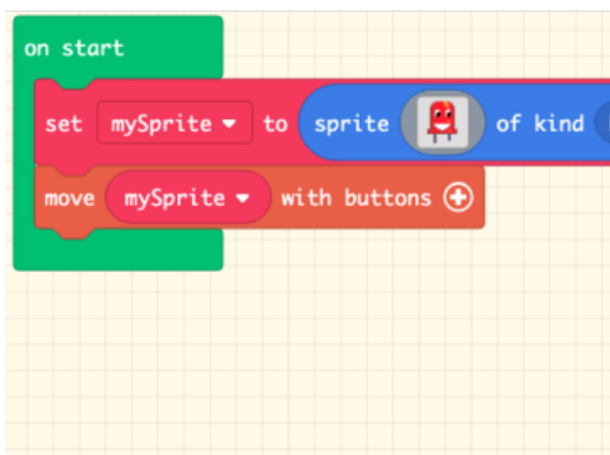
Bigger Ruby

This is looking a bit small for our character -- let's return to the sprite editor and make a 32x32 pixel version instead!



In this version, not only is Ruby larger, but I've used some extra colors to add a bit of dimension and bling. The checkerboard technique is a type of dithering used to add a type of color mix that isn't possible with such a limited color palette.





Controls

That's looking better! Next, let's add some controls so we can move Ruby around the screen.

In the Controller category, pick the move mySprite with buttons block and add it to the on start loop.

Now, you can use the d-pad buttons on the simulator (or the arrow keys on your keyboard) to move Ruby around!



Update the PyBadge/PyGamer Bootloader

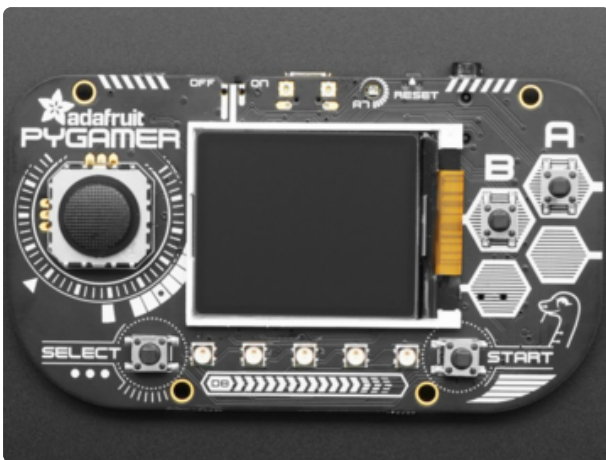
You are at the bleeding edge of handheld, open source, game playing hardware and software, what with your PyBadge/PyBadge LC or PyGamer! Congratulations! It's fun and exciting! It is also changing and improving all the time, so please update your bootloaders before proceeding to put your MakeCode Arcade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

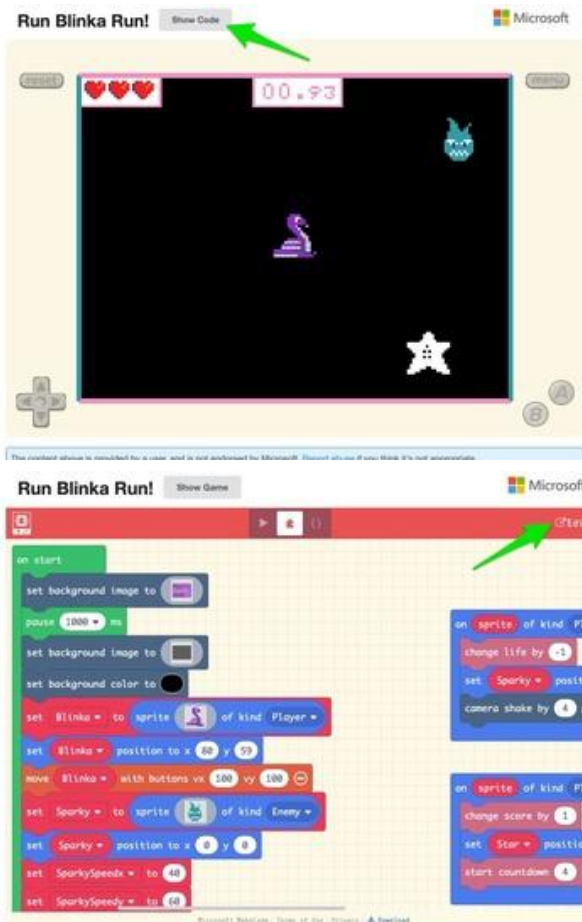


Load a MakeCode Game on PyGamer/PyBadge

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!

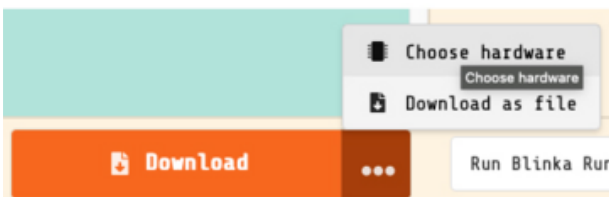
<https://adafru.it/Fqf>



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.

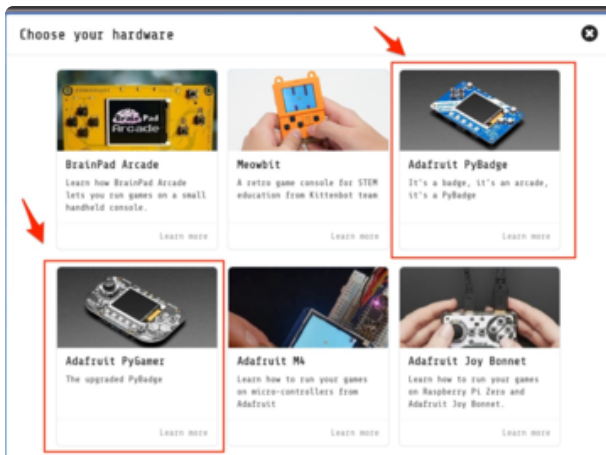
Once you have a game working on the MakeCode Arcade web editor, it's time to download it and flash it onto your board.

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



Board Definition

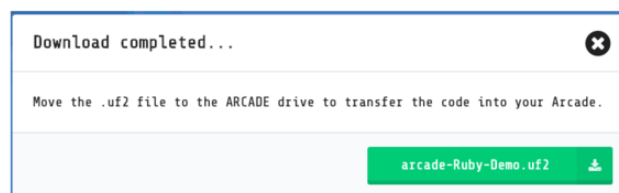
In order to load a game made in MakeCode Arcade onto the PyBadge, first choose the proper board definition inside of MakeCode. Click the ellipsis (...) next to DOWNLOAD and then the Choose Hardware item.



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.

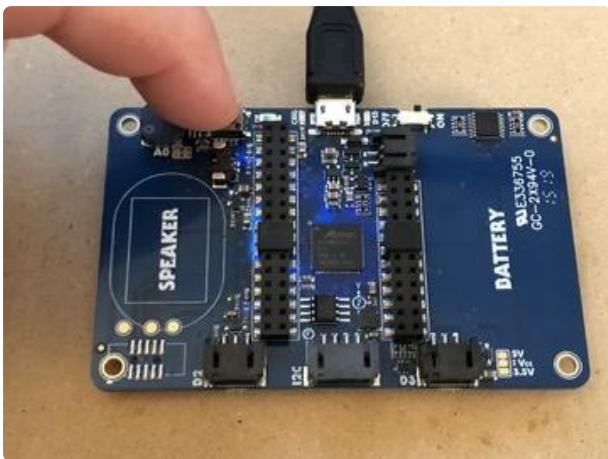


A HUUUUUUUGE number of people have problems because they pick a 'charge only' USB cable rather than a "Data/Sync" cable. Make 100% sure you have a good quality syncing cable. Srsly, I can't even express how many times people have nearly given up due to a flakey USB cable!



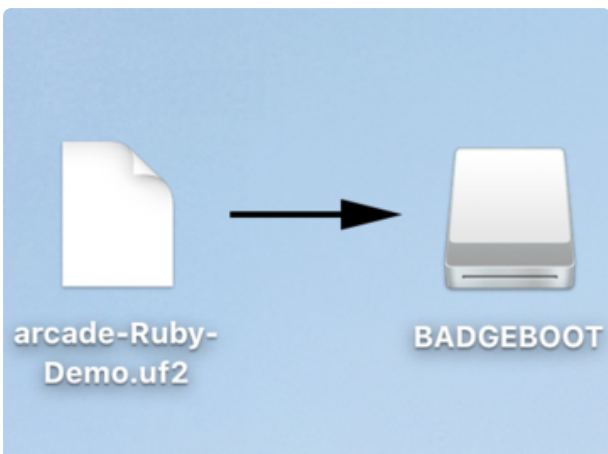
Bootloader Mode

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.

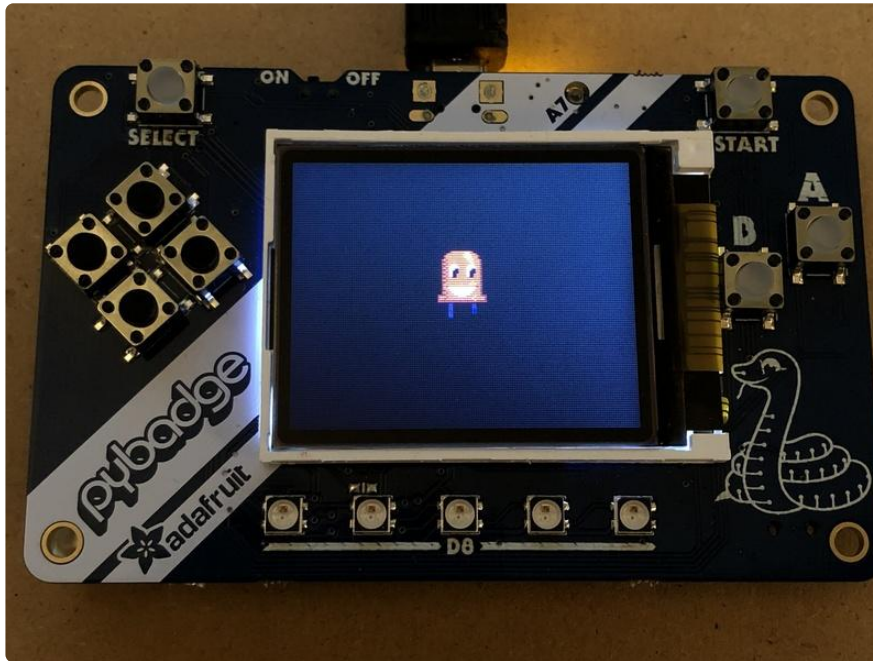


Drag and Drop

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!

Update the PyBadge Bootloader

Update the PyBadge Bootloader to prevent a problem with MacOS 10.14.4 and to fix button problems, and to prevent occasional damage to the loaded program.

Your PyBadge may need its bootloader updated for several reasons.

Bootloaders earlier than v3.9.0 do not protect against a rare problem in which part of internal flash is erased on startup.

Starting with MacOS 10.14.4, Apple changed how USB devices are recognized on certain Macs. This caused a timing problem with boards that were loaded with a MakeCode program, preventing the `PYBADGEBOOT` drive from appearing. Also the A and B buttons will be reversed in MakeCode if your bootloader is too old.

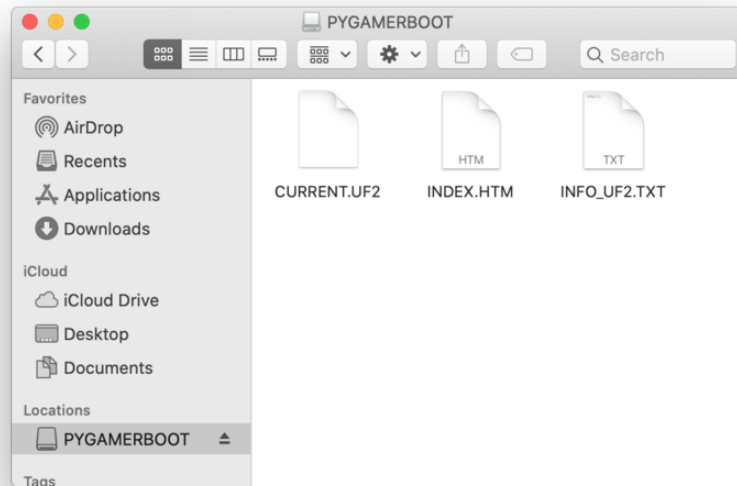
A HUUUUUUUGE number of people have problems because they pick a 'charge only' USB cable rather than a "Data/Sync" cable. Make 100% sure you have a good quality syncing cable. Srsly, I can't even express how many times people have nearly given up due to a flakey USB cable!

Updating Your PyBadge Bootloader

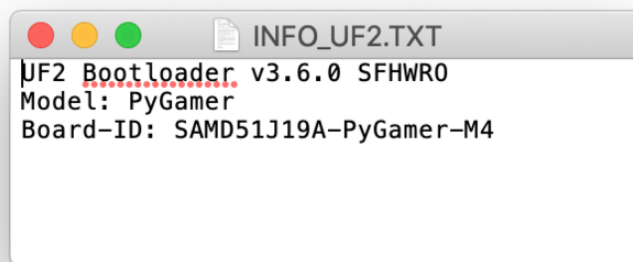
To see if you need to update your bootloader, get the UF2 boot drive to appear on your board. If you're running MakeCode, click the reset button once. If you're running CircuitPython or an Arduino program, double-click the reset button.

If the boot drive is named `ARCADE-D51` or `BADGEBOOT`, you definitely need to upgrade. If you see `PYBADGEBOOT`, click the `PYBADGEBOOT` drive in the Finder and then double-click the `INFO_UF2.TXT` file to see what's inside.

(The screenshots below are for `PYGAMERBOOT`, but `PYBADGEBOOT` will be similar.)



The bootloader version is listed in **INFO_UF2.TXT**. In this example, the version is V3.6.0.



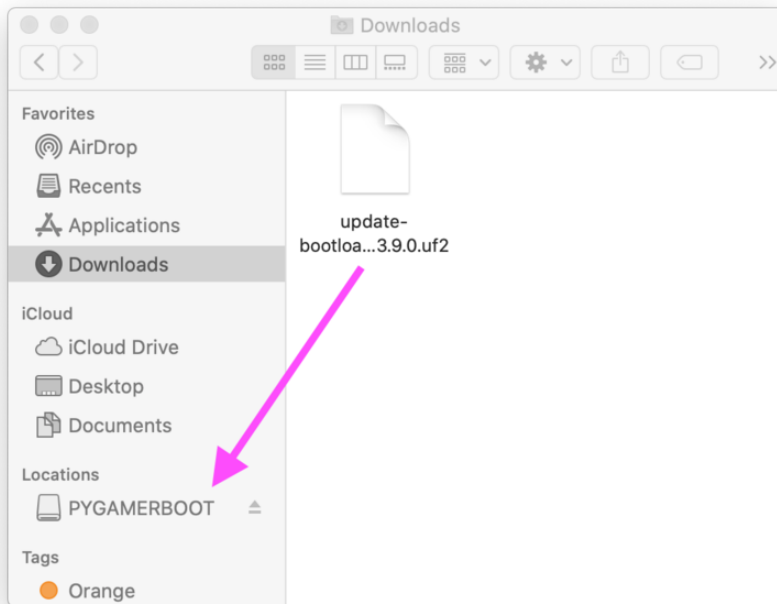
If the bootloader version you see is older than v3.9.0, you need to update. For instance, the bootloader above needs to be upgraded.

Download the latest version of the PyBadge bootloader updater on the circuitpython.org download page for the board:

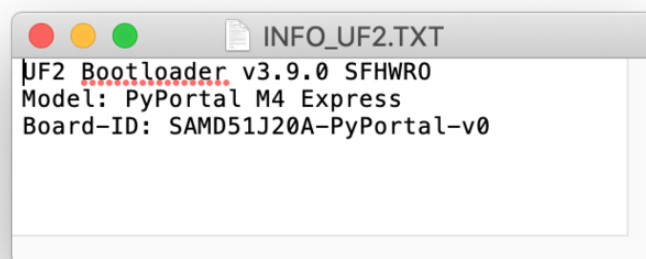
Latest PyBadge and PyBadge LC
downloads

<https://adafru.it/EF4>

The bootloader updater will be named `update-bootloader-arcade_pybadge-vN.N.N.uf2` with whatever version. Drag that file from your `Downloads` folder onto the bootloader drive: `ARCADE-D5`, `BADGEBOOT`, or `PYBADGEBOOT`.



After you drag the updater onto the boot drive, the red LED on the board will flicker as the bootloader is updated. The NeoPixels will flash and turn green again. A few seconds later, `PYBADGEBOOT` will appear in the Finder. After that, you can click on `PYBADGEBOOT` and double-click `INFO_UF2.TXT` again to confirm you've updated the bootloader.



After you update the bootloader, you will need to reload CircuitPython whatever other user program you are using.

Latest CircuitPython for PyBadge
and PyBadge LC

<https://adafru.it/KIC>

Oh no, I updated MacOS already and I can't see the boot drive!

If your Mac has already been updated to MacOS 10.14.4 and now you can't see a boot drive in the Finder, you need to find another computer that will work. Not all upgraded Macs will fail to show the boot drive: older ones can work. Or find a Mac that hasn't been upgraded yet. Any Windows 10 or Linux computer should work for upgrading your bootloader. Windows 7 computers [will need drivers installed \(https://adafru.it/Bf7\)](https://adafru.it/Bf7), but then can work.