Cosplay Floating LED Fireball with Motion Sensing

Created by Erin St Blaine

https://learn.adafruit.com/cosplay-fireball-prop-with-motion-sensing

Last updated on 2023-08-29 04:37:13 PM EDT
# Table of Contents

## Overview
- Materials
- You'll Also Need

## 3d Printing
- Print Settings
- Customizing the Size

## Code with MakeCode
- Code Walkthrough
- Flame Function
- Forever Loop
- On Shake - the Flare-up Feature

## Make the Flames
- Battery Life
Overview

Complete your cosplay with this Magical Item: a floating fireball that flares up at your command. Whether you're portraying Kael'thas Sunstrider from WoW or trading flaming shots with Super Mario and Luigi, this fireball will push your outfit over the top.

This is a very easy project, with no soldering required. The Circuit Playground board makes motion-sensing programmable lights a snap. Customize your colors and your animation speed with Microsoft MakeCode's drag-and-drop code editor. You'll be stopped for so many photos that you'll never make it to that panel discussion at DragonCon.

This project was inspired by [late-night browsing on Etsy](https()), where a lot of creative makers have similar products. What makes this one special is the ability to customize your own light colors and flame colors, as well as the motion-sensitive flare up!
Materials

Circuit Playground Express
Circuit Playground Express is the next step towards a perfect introduction to electronics and programming. We’ve taken the original Circuit Playground Classic and...
https://www.adafruit.com/product/3333

Adafruit Circuit Playground Express or Bluefruit Enclosure
We’ve got nice cases for many of our beloved boards, but the Circuit Playground Express and
https://www.adafruit.com/product/3915

DIY Ornament Kit - 7cm Diameter
Have you put up with mainstream, uninspiring, low-tech tree ornaments for too long? This season why not deck the halls with codes of holly?This DIY Ornament...
https://www.adafruit.com/product/4035
Lithium Ion Polymer Battery - 3.7v 100mAh
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/1570

Adafruit Micro Lipo - USB Lilon/LiPoly charger
Oh so adorable, this is the tiniest little lipo charger, so handy you can keep it any project box! Its also easy to use. Simply plug in the gold plated contacts into any USB port and a...
https://www.adafruit.com/product/1304

You'll Also Need

- Cellophane gift wrap in your desired colors - get it at the craft or stationary store
- A heat gun or hair dryer
- A 3d Printer (or you can order the part online)
- Glue dots - I used these from Scotch

3d Printing

These rings were designed in TinkerCad. TinkerCad is an easy-to-use 3d design program which is great for beginners, or for anyone who wants to quickly model simple shapes.

If you don't have access to a 3d printer, you can order them from Shapeways here:

- Child Size (15mm)
- Women's Size (22mm)
- Men's Size (28mm)
If you are printing your own, download the files below. I've included all three sizes in the .zip file, so select and print the one that fits you best.

**fireball_holders.zip**

**Print Settings**

I printed in clear PLA, with a raft and supports. I had the most success when I printed the widget lying down on its side.

My clear PLA didn't really come out clear -- it looks really white, and shows up more than I'd like. A flesh-colored filament would probably look better if your character is bare-handed. If your character wears gloves, print in a color to match the gloves.

**Customizing the Size**

If none of these sizes is quite right for you, it's easy to edit the files in TinkerCad to get a perfect fit.

Do not shortcut and simply resize the whole .stl file before printing -- the top end is sized to screw into the Circuit Playground case and if you resize the whole thing, it will no longer screw in tightly.

**Edit in Tinkercad**
Select the size that's closest to the one you want. Click the "ungroup" button in the upper right corner.

Click on the rings to select them. Click the white box in the corner and you'll see measurements appear. Hold the shift key while you drag with your mouse to resize without changing the aspect ratio.

This looks better, but now we're misaligned. This is easy to fix. Drag a box around the rings and the stem to select them both. Click the "align" tool in the upper right, then click the dot in the bottom center to align the rings with the stem.
Once you're happy with sizing and alignment, select the rings and stem and click the "group" button to attach them back together.

Select your resized widget and click the "export" button. Choose "selected shape" and "stl" to download your updated file for printing.

If you want a longer or shorter stem, you can use the same method. Just be careful not to resize the screw tip.
Microsoft's MakeCode editor is a great way to get your Circuit Playground Express to do things without having to muck about in actual code. It's a fun drag-and-drop editor that's easy to use and sneakily teaches you how to write code while you're making pretty things that light up.

This sample code makes a flickering flame effect with red, orange, and yellow. When "shake" is triggered, the lights flare up to a bright yellow, then slowly dim back down to the flicker.

You can download the completed code and get up and running right away, or you can follow along and create your own code, adding your own customizations as you go.

Code Walkthrough

Let's take a look at each of the different blocks to see what's happening. Head to mak
ecode.com to get started.

On Start

Click the Circuit Playground Express and then New Project. You'll see your workspace, with a Circuit Playground on the left, some tabs in the middle and your
workspace on the right. Click the LOOPS tab, and find `on_start`. Drag it into your workspace.

Whatever you put into the `on_start` loop will run once, when you first boot up your Circuit Playground Express.

I've created two variables: `bright` and `on`. The `bright` variable will be used to determine how bright our lights are, and the `on` variable will be used with an on/off switch hooked up to the two onboard buttons.

Brightness will be a changing number between 0 and 255. Our on/off switch will be a toggle, so I've set it to `true` (found under the LOGIC tab).

Flame Function

A function is a container for a block of code. I've made a function for our flickering animation, since this is the most complicated part of the code. Putting it inside a function will keep it together and out of the way of the rest of the code. The FUNCTIONS tab is under ADVANCED. I called my function `flame`.

There's a lot going on here. The power lies in the `hue, sat, val` block that's embedded in the `set pixel color` block. Both can be found under the LIGHTS tab.

Using the `hue, sat, val` block in conjunction with a `pick random` block allows us to choose specific color ranges and have them flicker at random.

Forever Loop

Whatever we put in the `forever` loop will run over and over, forever. This is sort of the main control area of the code.
Since most of the heavy lifting was done in our `flame` function, we can keep our `for ever` loop pretty simple.

First we'll tell it to reset the brightness to full bright.

This `if/else` block is found under the LOGIC tab. If our fireball is turned on (`if on = true`), the flame function will play. If it's turned off (`if on = false`), all the lights will turn to black (off).

Check your emulator on the left. You should see the lights on the face of the Circuit Playground start to flicker in your chosen colors.

**Changing Your Fireball Colors**

Change your colors in the `flame` function block by changing the numbers in the `hue` area. For example, if you want greens and yellows, make the three `hue` blocks read 60-70, 70-80, and 80-90. You can also change the saturation (`sat`) and make the colors paler or more vibrant. The value (`val`) numbers will change the brightness of the lights. I have mine on full brightness (255).

Experiment here until you get colors you like. The changes will show up on the emulator on the left side of the screen.

You can also change the speed of the flicker by increasing or decreasing the `pause` time.
On/Off Switch

Under the INPUT tab, we can find this block. I've set up a simple toggle using our on variable.

Try it out on the emulator to be sure it works.

For this project, the buttons will be a little bit hard to reach. I don't want to have to futz around and hit the wrong button when I'm trying to quickly turn my fireball on or off. So, I did a copy/paste with this block and changed the button A dropdown to button B. With both blocks present, I can press either button and the lights will toggle on or off.

This is a little user-experience nicety that only takes a second to code, but will save you endless futzing time when you're at an event.

On Shake - the Flare-up Feature

Finally, we want to make the fireball flare up when you activate the motion sensor on the Circuit Playground.

This block is also in the INPUT tab. The first part of the code makes the initial flash, setting all pixels to bright yellow.

Then, the second half of the code makes the yellow fade out until it goes dark, at which time the flame function will kick back in.

You can add a pause block inside your repeat block to slow the fade down. But I like it at this speed. It feels nice and responsive.
Give your code a name (I called mine Fireball) and save it. Click the Download button and a file will download to your computer. Plug your Circuit Playground Express into your USB port and click the reset button. You should see all the lights on the face of your Circuit Playground Express turn green, and a drive will appear on your computer called CPLAYBOOT. If you see a drive called CIRCUITPY, press the reset button a second time to get to CPLAYBOOT.

Drag the downloaded file onto this drive to program the Circuit Playground Express.

Once you've got your lights hooked up, shake the Circuit Playground to see the lights flare up!

If you're having trouble with the download process, head over to the Circuit Playground Express guide for some more detailed instructions and troubleshooting ideas.

Make the Flames

Open up your ornament case. Cut off the hanger bit with a pair of flush cutters.

Drill or melt a 1/4" hole into the center of the dome. If you're drilling, start with a smaller drill bit and work your way up through 2-3 sizes to minimize the risk of breaking the acrylic. If you're melting, use a soldering iron with a tip you don't care about and push it right through the plastic. Make sure your window is open so you don't breathe the fumes.
We'll use glue dots to stick all the pieces together. These things are great. They're not messy, and they have a really strong hold after just a few minutes.

If you mess up and need to get them off, 99% alcohol will help release the stick.

Open up your Circuit Playground case. Find the case bottom (the one with the screw hole in the middle). Stick 3-4 glue dots on the outside of the case around the perimeter.

Press the case into the ornament half, aligning the screw hole with the hole you drilled.
Cut pieces of colored cellophane wrap to around 12" square. I like the look of rougher edges rather than tidy corners here, but don't worry about the shape too much. You'll have time to finesse it later.

Put a few glue dots in the center of your first piece of cellophane.

Wrap the cellophane layer around your ball, sticking the glue dots to the plastic firmly.

Put the top on your ball, then wrap the cellophane around the whole ball and secure it loosely with a rubber band.

Gently heat the cellophane with a heat gun or hair dryer. It will shrink to fit the ball. Don't heat it too much or you may melt holes in it.
Repeat with your second layer of cellophane. I like to put the lighter / more transparent layer on the outside since it give such a lovely depth of color.

Gently melt the tops and corners of the cellophane layers. You can trim off any too-long bits with scissors. It's also fun to cut little flaps in the top and then melt the edges a bit, to give a nice random flame look.

Gently open the top of the cellophane and pull out the top half of the ball. Put a glue dot on your battery, and plug the battery into the Circuit Playground. Place the Circuit Playground inside the bottom of the case, and slip the case top in to click it in place.

Stick the battery someplace out of the way, along the edge of the ball.

If you like, add a little more crumpled cellophane over the Circuit Playground to further diffuse the light for those looking down on the fireball.
Melt a small hole in the cellophane aligned with the screw hole in the bottom of the fireball. Melting works better than cutting here. If you cut the cellophane it's likely to tear, but a melted hole will stay put.

Screw the 3d printed widget into the screw hole in the Circuit Playground case.

Slip your fingers through the rings so the fireball hovers above your hand. Shake it around to be sure everything is secure, and watch the lights flare up!

Reach in and press either of the Circuit Playground buttons to turn the lights off.

Battery Life

Turning the lights off with the buttons will lengthen your battery life, but the buttons do NOT cut power to the Circuit Playground. To turn the fireball fully off, you'll need to unplug the battery. If you just leave the lights in the "off" mode, your battery will slowly drain -- no bueno!
It might not hurt to get a couple extra batteries, and always have your charger on hand.