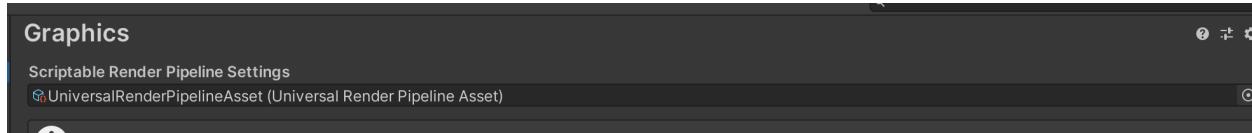


SPRING TERM

WEEK 1: LIGHTING

MONDAY MARCH 29TH 2021:

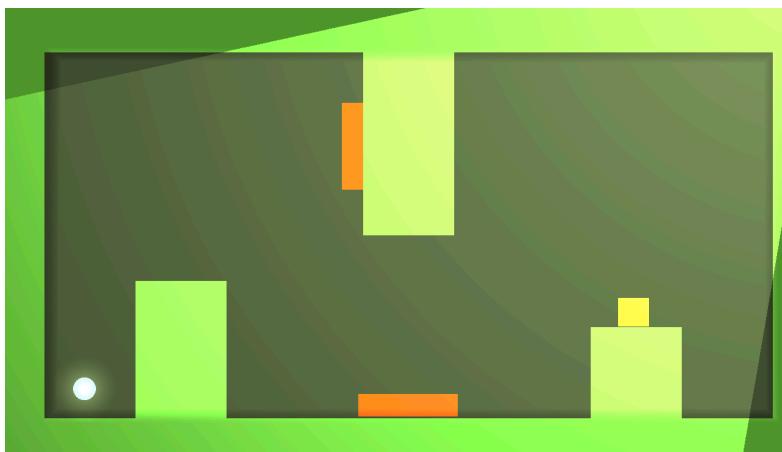
- Researched and read a lot about 2D Lighting
- Created custom 2D Renderer Data objects



- In "Graphics" in the project settings I changed my Render pipeline settings to include my custom 2D Renderer. Before it had none.
- I then replaced all the materials of my game objects (and made it so any new game objects) have this new custom material "Sprite-Lit-Default" instead of just "Sprite-Default" which doesn't register the changes in lighting.

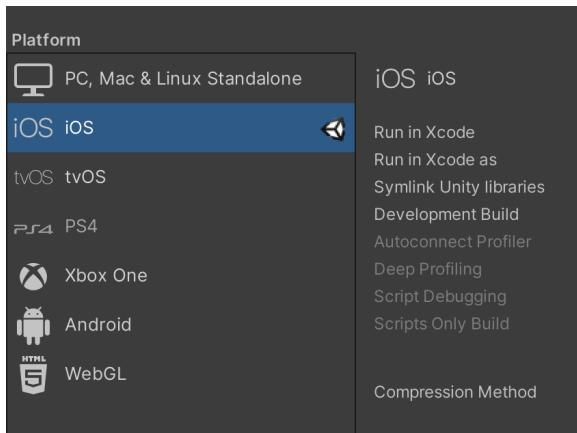
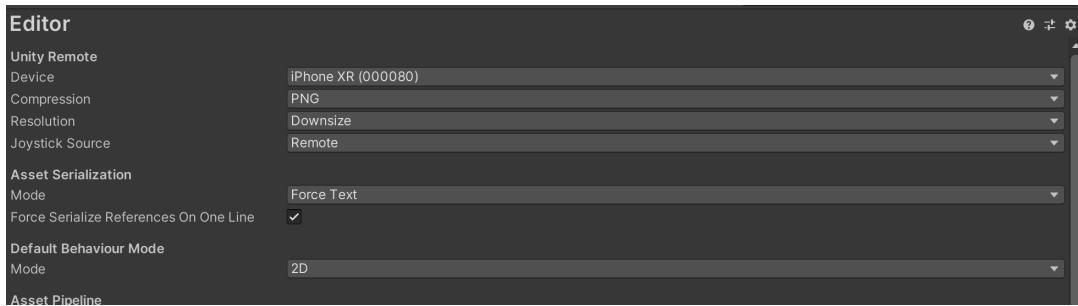
WEDNESDAY MARCH 31ST 2021:

- Experimented with different designs and 2D light ideas
- Added a point light to my ball using a 2D lighting script
- The "new and improved" version:



FRIDAY APRIL 2ND 2021:

- Got it so that my game will preview and upload to my phone using an app called “Unity Remote”
- Instead of previewing the game on my computer, by connecting my phone, and changing some project and build settings, I was able to play and control the game from my phone while it displays on both my phone and computer.



- Encountered a problem where my computer would not build my game. It happened for two reasons:
 - 1) I had too much storage on my computer. Cleared it out during a comp sci meeting (over 25 GB of creative cloud...yikes)
 - 2) I had downloaded a package called “XR Plugin Management” that interfered with the preview to my phone (and to my computer in general). Once I deleted it I was all set.

SUNDAY APRIL 4TH 2021:

- Spent a lot of time researching mobile controls—the accelerometer and the touch inputs. They’re confusing. I’ll get it though.

WEEK 2: MOBILE CONTROLS

MONDAY APRIL 5TH 2021:

- Worked on getting the accelerometer working by creating a vector. It didn't quite work at first until I set the rigidbody velocity equal to the vector multiplied by the x-velocity.
- Included a set up of a basic touch control—it's a little buggy and I think I'm doing it wrong.
- Problem with the accelerometer is—while it does work—it affects my y-value which I need gravity to act upon. I'm thinking of adding a force, but want to try a few ideas out first.

WEDNESDAY APRIL 7TH 2021:

- Okay so, x-accelerometer works but still gives trouble. I tried to change my code so that instead of a vector, the accerlerometer is taken as a float value and then multiplied to a vector that just holds my rigidbody position. It kind of made the ball fly all over the place. And it kept centering my ball at the middle of the screen instead of its starting position at the corner.
- I also tried seeing if I could download or edit a package from the package manger labeled "Input" which allows you to alter controls for a game, but the controls are all for PC or a joystick. Think I might have to add a force after all. I'll test run it tomorrow.
- Starting bringing in code from my last game and altering it so it works for this new mobile version.

THURSDAY APRIL 8TH 2021:

- Yep. We're going to test run!
- Can happily conclude that forces do—indeed—work (could've saved a lot of time!)
- Read: <https://learn.unity.com/tutorial/touch-input-for-mobile-scripting> (this will also help me with the main menu—and how the user interacts with it)
- accelerometer works but the ball can't quite find an equilibrium—meaning it's very hard for the user to let the ball stay still (it keeps sliding). Planning to work on this

WEEK 3: BUILDING TO IOS

MONDAY APRIL 12TH 2021

- I started with looking at these three unity manuals:

<https://docs.unity3d.com/Manual/iphone-GettingStarted.html>

<https://docs.unity3d.com/Manual/UnityCloudBuildiOS.html>

<https://docs.unity3d.com/Manual/StructureOfXcodeProject.html>

Honestly, all of which didn't help too much but it was a good read.

I shifted my focus to xcode—because this is what is going to allow me to build the app on my actual phone. I utilised these links:

<https://codewithchris.com/deploy-your-app-on-an-iphone/>

<https://makaka.org/unity-tutorials/test-ios-app-without-developer-account>

<https://stackoverflow.com/questions/39524148/xcode-error-code-signing-is-required-for-product-type-application-in-sdk-ios>

I also looked at a lot of unity forums and stack overflow—but every unity tutorial a user redirected me too never ended up showing up on the Unity page so that was a bummer.

Essentially, every time I tried to build my app I encountered an error with “code signing” and another one that just said there was an error with “exit code 1.” Upon research, I did the following steps in Xcode:

1. Preferences > Accounts > Added Apple ID
2. Added “Apple development” certificate (this isn’t a paid developer subscription)
3. Enabled signing to “Automatically manage signing”
4. Selected my Apple ID as my “Personal Team”
5. Changed the Bundle Identifier from
“com.DefaultCompany.com.unity.template.mobile2D” to
“com.AmethystMcKenzie.com.unity.template.mobile2D”

(which honestly this could be the issue because I’m not sure what this should be)

6. Changed Code Signing to iOS Developer
7. Made sure provision profiles were set to Automatic (because lord know i don’t know how to do this manually)

I also was learning a bunch of stuff like what a Bundle ID is! (basically a domain for your app which uniquely identifies it in the appstore).

This fixed my code signing problem but it did not, however, fix the exit code 1 error, which when I searched up, should essentially have been fixed with my code signing manipulation because it is a code signing issue.....

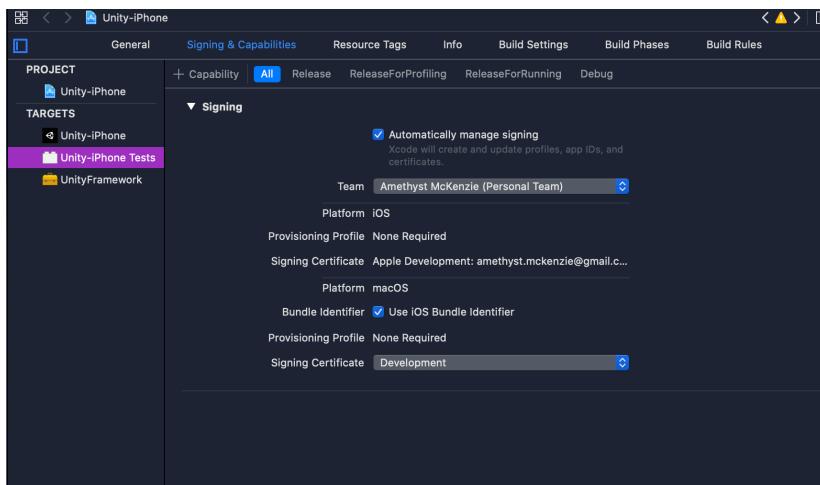
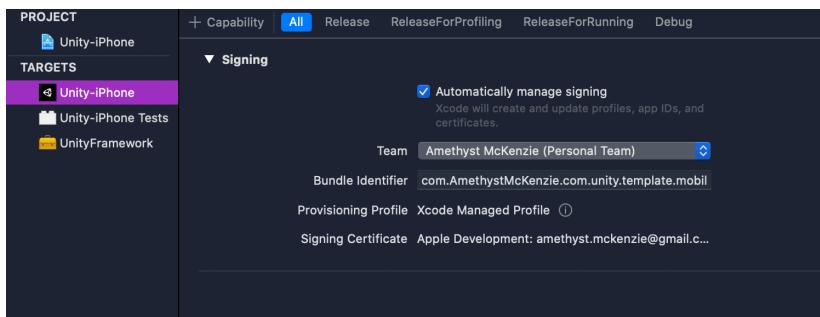
Okay! So maybe I didn't fix the code signing problem. Different approach!

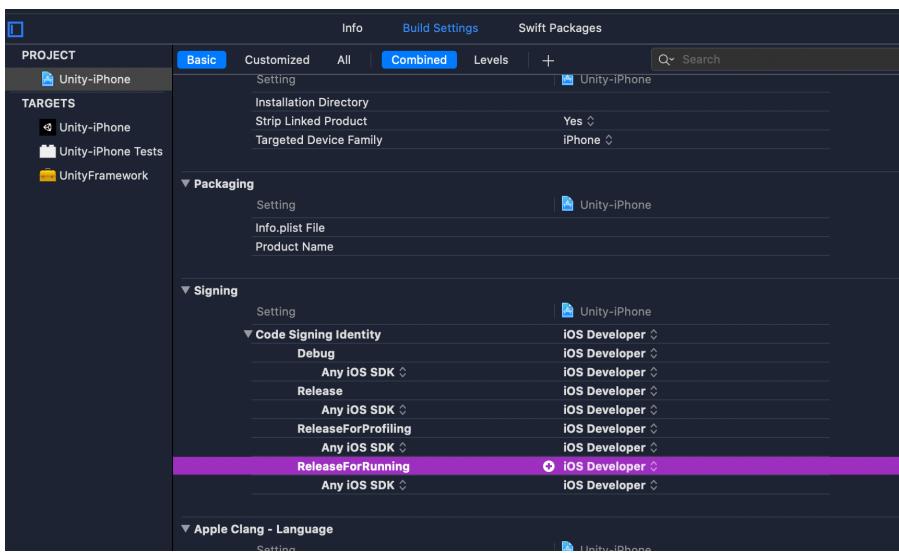
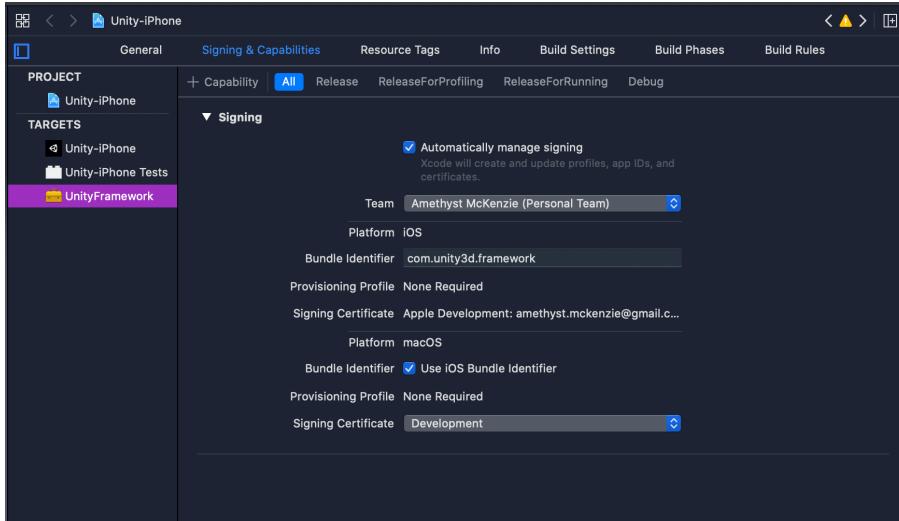
When I went into my keychain settings I discovered that the certificate I was using was "Not Trusted." This took multiple attempts (unsure why) but I managed to get the certificate to be always trusted, but it didn't do much



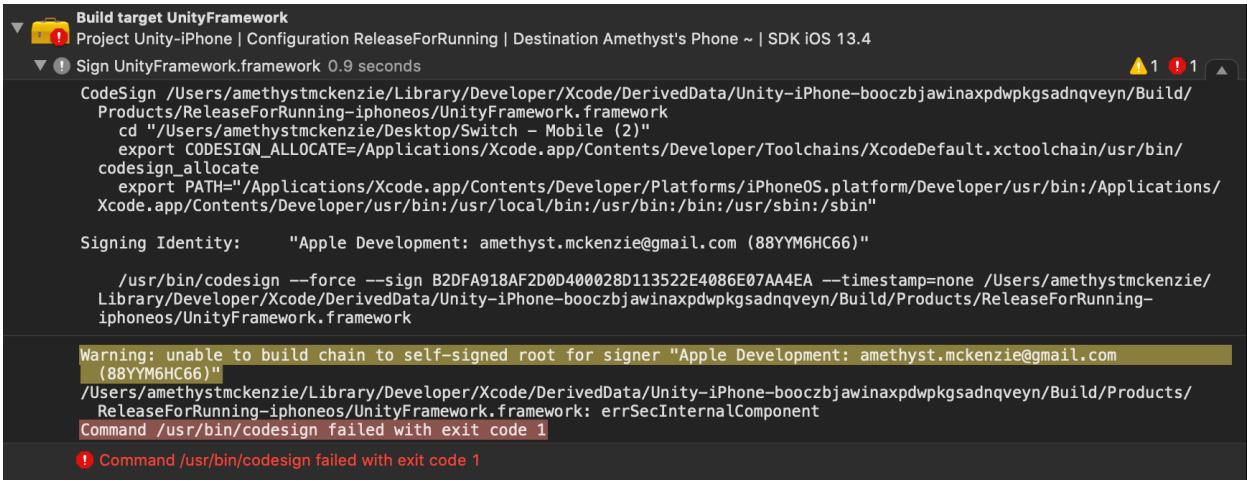
It seems like a lot of people have this problem with Xcode but nobody really has a permanent solution—or most likely I'm doing something terribly wrong.

So far, my Xcode settings look like this:





And this is my error:



```

Build target UnityFramework
Project Unity-iPhone | Configuration ReleaseForRunning | Destination Amethyst's Phone ~ | SDK iOS 13.4
    ▲ 1 ⚠ 1 ▲
    ▾ ! Sign UnityFramework.framework 0.9 seconds
      CodeSign /Users/amethystmckenzie/Library/Developer/Xcode/DerivedData/Unity-iPhone-booczbjwinaxpdwpkgsadnqveyn/Build/Products/ReleaseForRunning-iphoneos/UnityFramework.framework
      cd "/Users/amethystmckenzie/Desktop/Switch - Mobile (2)"
      export CODESIGN_ALLOCATE=/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/codesign_allocate
      export PATH="/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/usr/bin:/Applications/Xcode.app/Contents/Developer/usr/bin:/usr/local/bin:/usr/bin:/usr/sbin:/sbin"
      Signing Identity: "Apple Development: amethyst.mckenzie@gmail.com (88YYM6HC66)"

      /usr/bin/codesign --force --sign B2DFA918AF2D0D400028D113522E4086E07AA4EA --timestamp=none /Users/amethystmckenzie/Library/Developer/Xcode/DerivedData/Unity-iPhone-booczbjwinaxpdwpkgsadnqveyn/Build/Products/ReleaseForRunning-iphoneos/UnityFramework.framework
      Warning: unable to build chain to self-signed root for signer "Apple Development: amethyst.mckenzie@gmail.com (88YYM6HC66)"
      /Users/amethystmckenzie/Library/Developer/Xcode/DerivedData/Unity-iPhone-booczbjwinaxpdwpkgsadnqveyn/Build/Products/ReleaseForRunning-iphoneos/UnityFramework.framework: errSecInternalComponent
      Command /usr/bin/codesign failed with exit code 1
  ! Command /usr/bin/codesign failed with exit code 1

```

So I've got two options right now:

1. Continue researching and ploughing through to hopefully find a viable solution
2. Figure out how to use testflight..Although I'm not sure if I need a developer account or not for this.

In the meantime I'll have you know that this took an embarrassingly long amount of time for nothing much to be solved so...what else is new!

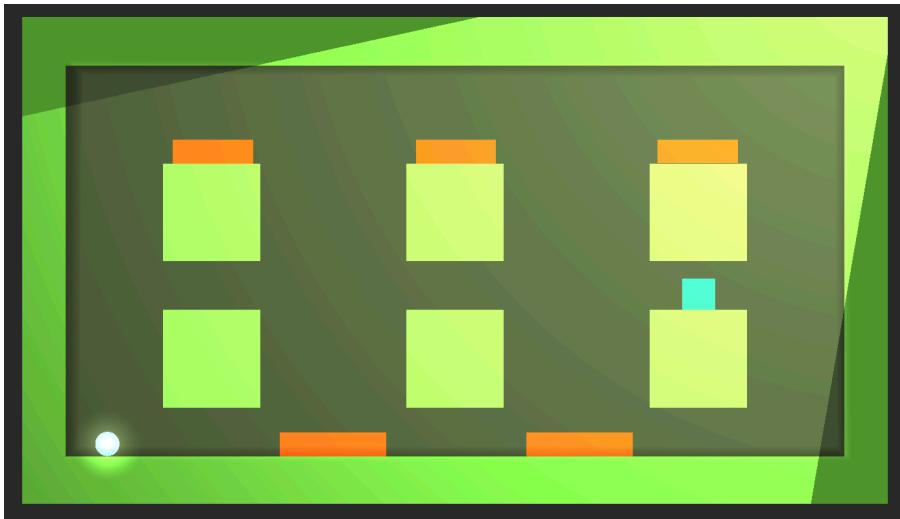
I actually had this problem with Xcode last year as well I realized. I never knew how to do code signing from the beginning! I know because I checked back to see what I did in Xcode last year and how I built my apps but they were built on a simulator on the computer and not my actual phone.

I'm connecting my phone via USB to my computer and I am given the option to build the app to my phone, so it is reading that connection, but the errors prevent it from building.

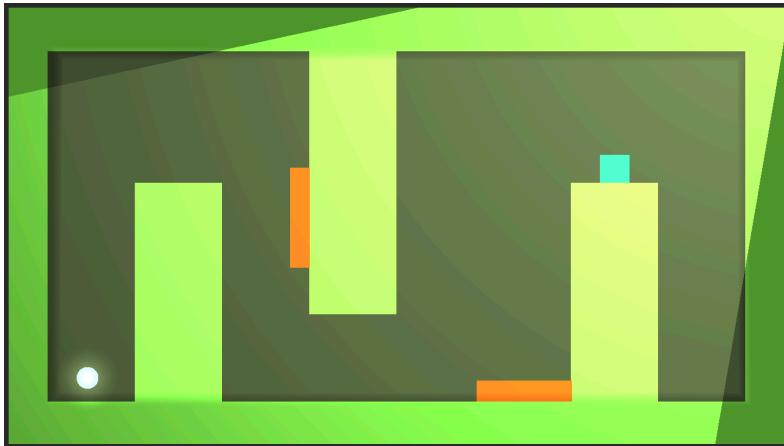
WEEK 4: BACK TO GAME DESIGN

MONDAY APRIL 19TH 2021

- I finally got my touch input working well—I had to create a boolean and also use one of the built-in commands you can use with Touch (called “Phase ended”) which essentially lets the computer know when the user is and isn’t interacting with the specified touch.
- Basically, tapping works like a charm! So does tipping the screen back and forth!
- I also started dividing the force of my Input.acceleration.x so that it wasn’t so quick (because it was little intense). This also helped my “sliding” issue, so it’s a lot easier to stabilise the ball and it can actually remain still now at equilibrium.
- Some level designs were also made:



TUESDAY APRIL 20TH 2021:
More level designs!

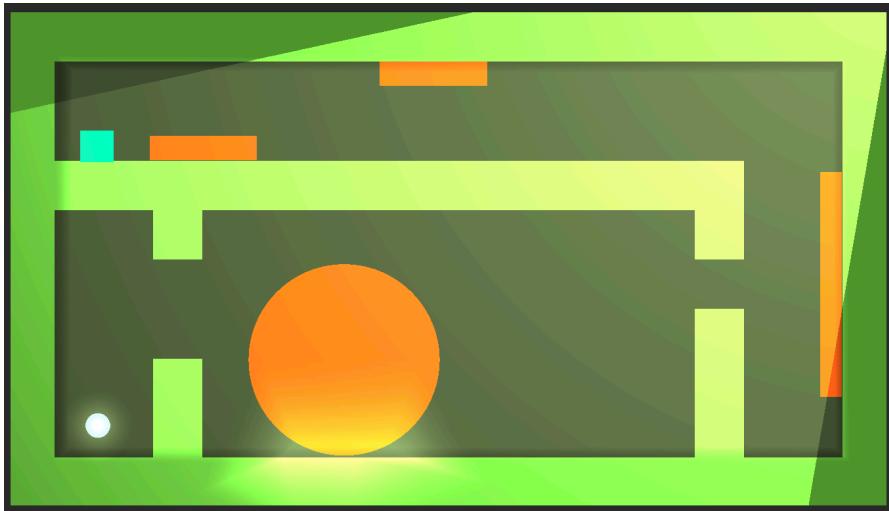


Also researched more about building to IOS. I figured that while I'm trying to figure this out, I'll work more on my game. Also Plan B is just getting the Apple developer account

(because I'm going to get it eventually) so if I'm still struggling by next week I'll buy it and see how I can build it on my phone (and to the app store) from there.

WEDNESDAY APRIL 21ST 2021:

- worked on my Moving Enemy script, level designs and researched a lot of stuff on shaders and shading in Unity (2D).



- Like lighting, implementing shadows is a whole process on its own. I utilised these links from Unity's youtube channel:
 - <https://www.youtube.com/watch?v=gJMeSkolnw4&list=PLX2vGYjWbI0TPcPOKW6GxwuY18eg7CjKZ>
 - https://www.youtube.com/watch?v=p8H8ZdL_7eA&list=PLX2vGYjWbI0TPcPOKW6GxwuY18eg7CjKZ

Also checked this out: https://blogs.unity3d.com/2021/02/05/introducing-shader-graphs-new-master-stack/?utm_campaign=newnews_global_nurture_2021-global-pe-nt-interm-nurture&utm_content=2021-global-pe-nt-interm-nurture-email-2-new-to-unity&utm_medium=email&utm_source=Eloqua

It's a bit advanced for the design of my game but it was still an interesting read.

I'm planning on working on shading and a menu system over the next week, while updating levels in between. I'll also continue tackling that "building to IOS" problem.

FRIDAY APRIL 23RD 2021:

- Big things today! We (Mr. Healy and I) figured out the problem with building to IOS. Essentially, for some bizarre reason, Xcode created two certificates when making my account, or maybe I accidentally created an extra one, and it was confusing the two. Because the extra certificate isn't in my keychain and isn't valid, I keep getting a code signing error despite the fact that I do have another perfectly valid certificate.
- Basically the computer is using the faulty certificate instead of the good one.

- We have the problem, now we can build towards fixing it!

WEEK 5: IT BUILDS!!

TUESDAY APRIL 27TH 2021:

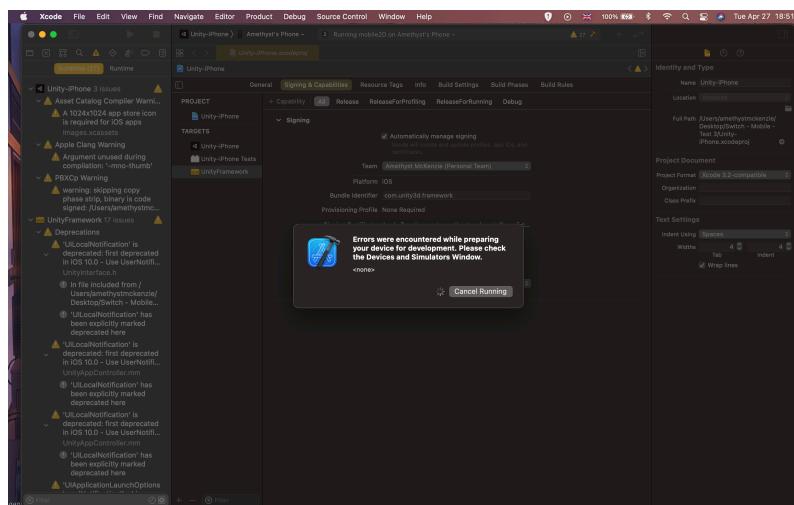
- Okay a lot just happened and it took a lot of time so let me explain—long story short, the app finally *FINALLY* builds to my phone. It's a miracle!!
- Now I need to recount the steps:

So starters, remember that code signing issue? Well turns out that was the least of my problems, and I was able to fix that surprisingly quickly considering my issues from the past two weeks. I'm unsure what exactly it was but when I went into my keychain access I changed my certificate to say “Use System Defaults” under the Trust section instead of “Always Trust”. Also, when I checked xcode (after a couple days of not building), my other faulty certificate that was confusing my computer was gone—though I get more invalid certificates if I, say, change the bundle ID and then try building the app again.

Okay, no more changing the bundle ID!! I was able to find the extra certificates hidden in all my keychains and I deleted it—so that DEFINITELY helped. So, great, it builds now, no more code signing error. But boy oh boy does it not stop there, this is only the beginning.

So, to sum it up, all my software is out of date. Xcode, my laptop, my phone—everything. So I spent a good solid *eight hours* of the day uploading my mac, so I can upload xcode, so I can update my phone, so I can update xcode *again*. Due to my almost impressive lack of storage this took multiple attempts because downloads kept cancelling but we got there.

Once I upgraded my laptop and xcode, I build the game again and I got this error:



What does it mean???? Literally NO ONE knows. People online told me to just “restart my phone,” and I did that twice but it still wouldn’t work. Eventually I decided I should probably upgrade my iPhone to IOS 14.5 so I did that (which took another unnecessary long amount of time) and then xcode had the audacity to say my phone was now incompatible because it only supports IOS 14.4.

Luckily, 2 days ago, Xcode sent out an upgrade so that it can support IOS 14.5, so I begin upgrading (which takes more time). This actually takes multiple attempts because it takes so much storage and I keep running out, so I had to say farewell to a lot of great stuff (I even had to delete Processing...). Now, everything is finally upgraded—no more upgrading.

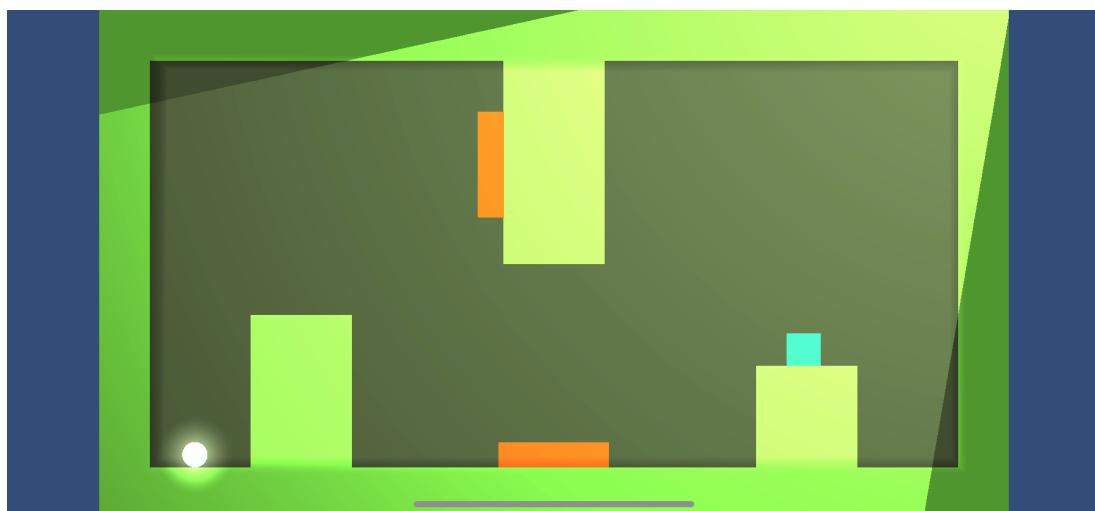
I build the game—it doesn’t work—it says: “Linker command failed with exit code 1” and beneath that it says it can’t find a certain phone library that it needs. I probably deleted it in my rampage to get free storage, sorry not sorry.

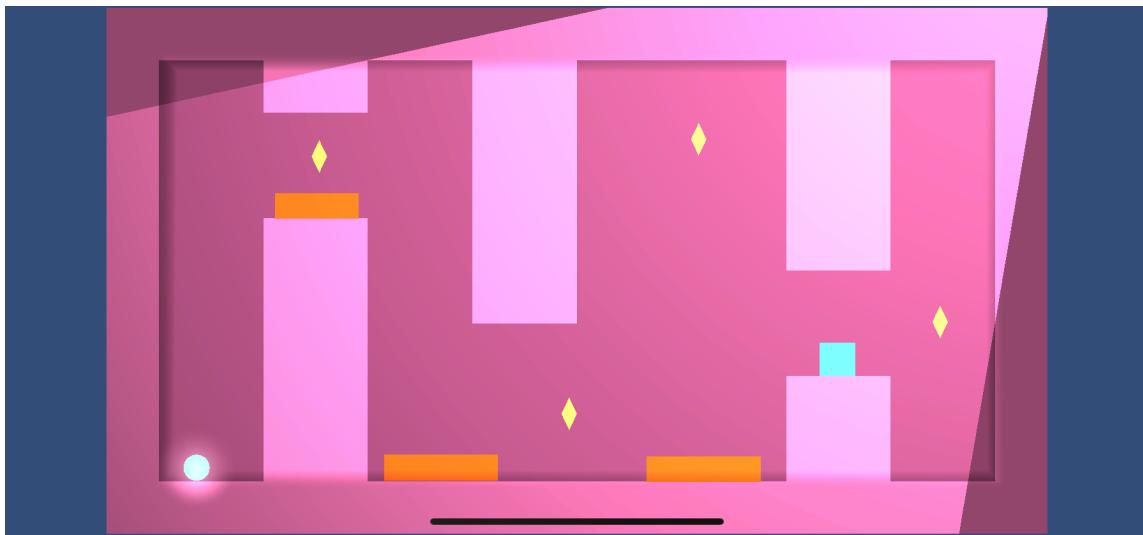
I just re-build the game from unity, get all my stuff back, get a new folder, and run it through xcode.

I build the game, it works—yes—it is finally on my phone and I click the app and a message pops up basically saying that my phone doesn’t trust my certificate, which is quite disrespectful might I add, and I have to go figure out a whole new thing. Luckily, this isn’t actually as time consuming. When I pressed cancel to stop the build upon getting that message it automatically opened a section in my settings titled “Device Management”. You get there by going Settings > General > Device Management. I clicked Trust, and voila! It is now trusted.

I open my game, and it is literally the most satisfying, beautiful, crisp image I have ever seen—I am rejoicing in the heavens, I have achieved my purpose in life.

So yeah, here are some photos:





The thing that seems to be a problem is the dimensions, and I quickly found out that that was because I wasn't using the right preview in my game. By comparing the two images, I was able to find the right dimensions and now I'm doing some slight redesign, nothing too crazy!

I feel like my accelerator controls are a little weird. It works perfectly fine but it just feels a little too resistant, so I'm going to ease up on the number that's dividing the force of the accelerometer.

Overall: Feeling very satisfied!

HOW TO BUILD A UNITY APP USING XCODE TO YOUR PHONE: (A tutorial by yours truly)
Things you must have before building an app using Xcode: Apple ID/Account, an iOS device, a developer certificate (that should be auto-generated if you link your Apple ID to Xcode. If you haven't done this, go to Xcode -> Preferences [or press command and then the comma button] and click Accounts. Add your apple ID using the plus icon).

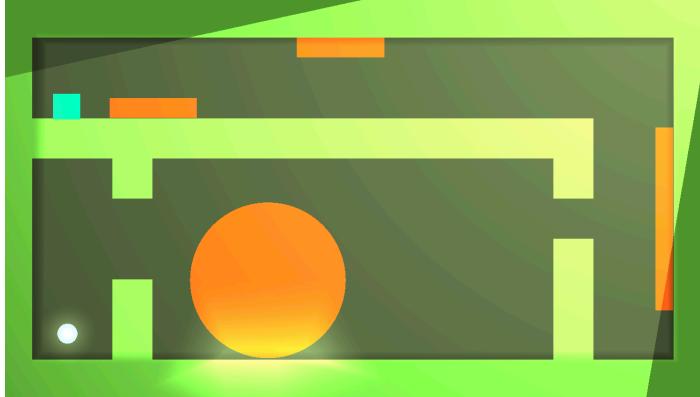
- Open your project and then select your project file within Xcode.
- Near the top of your screen you should see a tab that lists "General," "Signing & Capabilities," "Resource Tags," "Info," "Build Settings," "Build Phases," etc..
- Go to "Signing & Capabilities" and click "Automatically manage signing"
- In the "Team" section, you should select your Apple ID (for a personal team) or your Developer team.
- Since you are just building to your phone, at this stage your bundle identifier can be anything you want, just make sure it is unique to your project.
- You will see three tabs underneath the "Targets" label on the left of your Xcode project —"Unity-iPhone," "Unity-Phone Tests," and "UnityFramework." Click each one, while still under the Signing & Capabilities tab, and click automatically manage signing and select your team again.
- Plug your device into your computer and build your game.

- The app should show up on your phone! If you haven't built an app to your phone before you will get a privacy message stating that your phone is not trusted.
- Once this message pops up, and you click cancel, an option in your settings app will appear.
- Go to Settings > General > Profiles & Device Management, then choose the trust option.
- Run your app again and you will be able to test it on your phone.

WEEK 6: FINISHING UP

TUESDAY MAY 4TH 2021:

- Spent most of the time finishing levels and fitting them to my device's screen.



WEDNESDAY MAY 5TH 2021:

- Finished all pink levels
- Started working on yellow levels—implemented timer and began designing.



- Trying to finish in a week so I can focus on submitting to the AppStore, as it is a long and tedious process.
- Finished MainMenu screen with working buttons/transitions.

SATURDAY MAY 8TH 2021:

- Added background music and sound effects.
- Added my AudioSource to my Global Manager so that it will continue on to each level but won't duplicate after you die on the first level.
- Re-Attempted particle effects, but decided they actually don't look that great with my game. Tried redesigning it but wasn't satisfied with results.
- Finished all basic coding elements of the game.

WEEK 7: PREPARATION

TUESDAY MAY 11TH 2021:

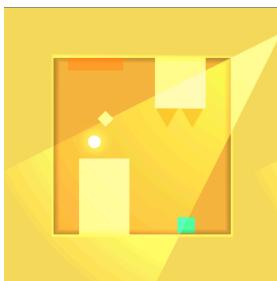
- Implemented Pause Screen.
- Designed pause and play buttons and added animations so that button changes when pressed.
- Finished designing all levels!
- Started filling out information on the App Connect account (like descriptions, tags, URLs etc.)

WEDNESDAY MAY 12TH 2021:

- Did a LOT of testing. Went through each level and fixed a few errors in my Level Manger Game Objects.
- Designed App Icon and started working on all the screenshot/UI dimensions for promotion of the game.



App Icon #1

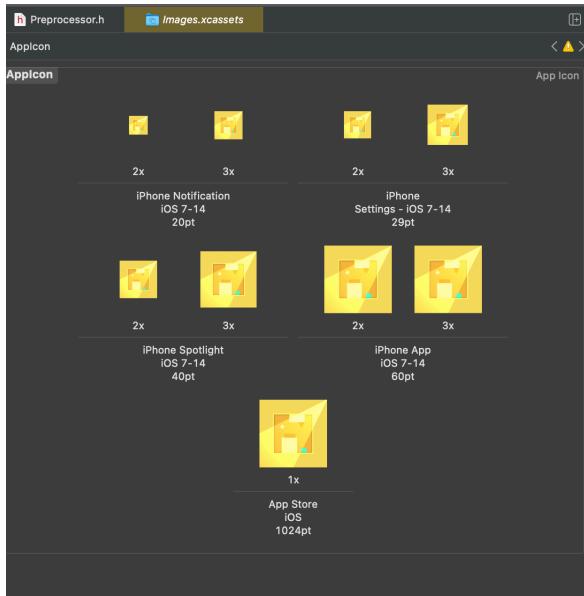


App Icon #2

- Asked my friends which icon they preferred.

THURSDAY MAY 13TH 2021:

- App Icon #1 won!
- Started preparing my app for a build to the appstore so that friends can test it on TestFlight. This includes adding the right dimensions of all my App Icons to my XCode Project and making sure everything looks good.



- Finished Settings Page (which features how to play and credits to the background music).

FRIDAY MAY 14TH 2021:

- Uploaded my first build to App Connect so that it can be run on TestFlight!
- I did this by selecting “Archive” on my Xcode project and then going to my Organizer and selecting “Distribute App.”
- Here I specified for it to distribute onto “App Store Connect,” that it should be “Uploaded” and “Include bitcode iOS content” and “Upload app’s symbols to receive symbolicated reports from Apple.” I continued to click “Automatically manage signing.” Finally I selected “Upload” and waited for it to show up as a build on my TestFlight tab within Apple Connect.
- I received an email stating that it was missing push notifications. By default, Xcode allows push notifications, and since it was initialized but I wasn’t utilizing them anywhere, Apple sent me a warning. Trying to figure out how to solve this.
- Later tonight I uploaded another build, with a few minor modifications (designing issues)

SATURDAY MAY 15TH 2021:

- Uploaded a third build with even more modifications. I didn’t credit the background music properly so after my meeting with Mr. Healy (yesterday) I redid my settings page to credit the sounds correctly.

- Added my screenshots to the iPhone Display section. Finished all the information for my App on the Apple Connect account.
- Created my own gmail account and twitter for the app
- Fixed the “push notification” warning I was getting from Apple. In the “Preprocessor.h” file underneath Classes in my Unity-iPhone project there is a line of code that states: “#define UNITY_USES_REMOTE_NOTIFICATIONS 1.” I changed the “1” to a “0” so it is now: “#define UNITY_USES_REMOTE_NOTIFICATIONS 0.” Simple fix!

SUNDAY MAY 16TH 2021:

- I made a website so that I could make a page for my privacy policy.
- Designed the website using WIX. Made a home screen and contact page, with animations.
- Finished my privacy policy
- Put all them links into the Apple Connect account!
- Sent out a bunch of TestFlight invitations to my friends so they can test my game :)

WEEK 8: SUBMITTING TO APPSTORE!

MONDAY MAY 17TH 2021:

- Still having errors with my privacy policy. I believe an admin (someone with higher privileges on the Suffield account—like Mr. Healy) has to review all the privacy stuff and confirm that my app is not collecting any data.
- In the meantime, Mr. Healy ran my game and my app isn’t compatible for all screen sizes—so going to work on that!
- Fixed it so that my game works for all screen sizes (took a load of time). The process was made easier through an asset on the Unity store called “Auto Letterbox.”
- Re-did all my constraints (basically I “locked” each UI element to a certain place on the screen so no matter what device you use it will always be in that location in regards to the overall screen dimensions). Now, all UI elements (and the game itself) gets scaled down to fit different iOS screens.
- Sent out a bunch more TestFlight invitations.

TUESDAY MAY 18TH 2021:

- Built all the changes into the beautiful version that is “1.0.4”
- My App Icon was a little blurry so I built another version—now version “1.0.5” is superior. I love her.
- Put all my screenshots in for 5.5” displays (now that my device is compatible for all screens)
- Sent out probably my final TestFlight invitations before submitting to the AppStore. (Just need my privacy data confirmed by an admin—last step!!)

WEDNESDAY MAY 19TH 2021:

- SUBMITTED MY APP TO THE APPSTORE!!!!!!!!!

HOW TO SUBMIT AN APP TO THE APPSTORE: (another tutorial by yours truly)
Things needed prior to this tutorial: Apple Developer Account (Apple Store Connect). Patience.

- Open the Xcode project you want to upload. Archive it. Go to Window > Organizer to view your Archives.
- Here you will distribute your App to your Developer Account. Click “Distribute App”
- Specify that you want to Upload your App to App Store Connect
- Click continue through the next two pages that ask you to “Include bitcode iOS content” and “Automatically manage signing.”
- Select Upload
- You will now find your build underneath TestFlight > Builds > iOS in your Apple Connect account. It will take a few minutes to process.
- Once processed, you may need to manage a few things involving US Export laws. If your app (like my app) does not violate any laws or include any encryption just select “No” to any of the prompted questions. If you do, then make sure to specify what it is that your app does, and what encryption it includes.
- Fill out all the information, details, screenshots etc. of your app underneath the App Store tab in your Apple Connect account.
- You’ll need a Support URL and Marketing URL—put links to any social media or websites you have in this section.
- Scroll down and you will see a section that says “Build,” upload the build you want to submit for review in this section. You should select a build that has already been Archived and uploaded to your App Connect Account (as explained above).
- In the left-side tab you will see a section called “App Privacy” underneath General. This is important.
- Link your privacy policy in the area that says “Privacy Policy URL.”
- If you are an admin, click the “Get Started” button and specify what data (if any) your app collects/stores. Then select “Publish.” If you are not the admin of this developer account, or do not have admin privileges, ask the admin of the account to do this for you.
- Submit for review!

WEEK 9: REFLECTION

TUESDAY MAY 25TH 2021:

- So, my app is submitted and I have completed all the core goals I set out for myself at the beginning of the year!

A refresher of those goals:

CORE GOALS

Completed one major Unity tutorial by the end of the Fall Term

Completed my own mini project with Unity

Become comfortable with C#

Learn how to work with light and animations in Unity!

Submit the app to the public!

I even completed a lot of my reach goals! Like working with lights and animation, and creating a beautiful interface + graphics for my game.

REACH GOALS

Understand how to make really beautiful graphics in a 2D game

Experiment with soft-body physics

Incorporate art into my code to achieve a stunning interface

Create my own soundtrack

Smooth animations!

I'm excited to experiment with soft-body physics in the future (will be a challenge for sure) and even work on my own soundtracks for new games. I also would love to work more with Unity animations, and become more comfortable with them.

As for the future of Switch, I still have a lot of updates and features I want to implement. A whole list in fact.

Next updates:

- number levels
- Save which level you are on and allow user to go back to menu
- Level select page
- tutorial page
- build for android!
- pause screen lets you go back to main menu
- Instead of "go back to menu" in settings page put "return to menu"
- When u switch gravity change direction of light. + more visual indicators

Which I'm very eager to start working on.

This whole process has been so so rewarding and insanely enjoyable, albeit stressful at times (but in a good way!). It was amazing to be able to turn a simple idea into a fully-fleshed out project, and have a tangible result at the end of all the hard work, time and

effort I put into each assignment throughout the year, and especially these last few months (fully feel like Switch is my child). I learnt so much this year through research and just powering through problems, and I have a great understanding of C# and Unity, which is one of the biggest things I wanted to gain by the end of the year. I was also able to practice and experiment with graphic design, and I believe I have grown as an artist, which I did not expect coming into this.

Overall, I am very pleased and humbled by this process. Coding and designing an app from scratch is no easy feat, and submitting/building through Apple took a lot more work than I initially anticipated, coding aside. Between trying to find the right method to build my app to my phone and going through all the legal requirements Apple expects of you in order to actually submit your app for review, I definitely appreciate all those games I see on the AppStore a lot more.

I'm very excited for what the future of switch, and my computer science endeavours, hold!