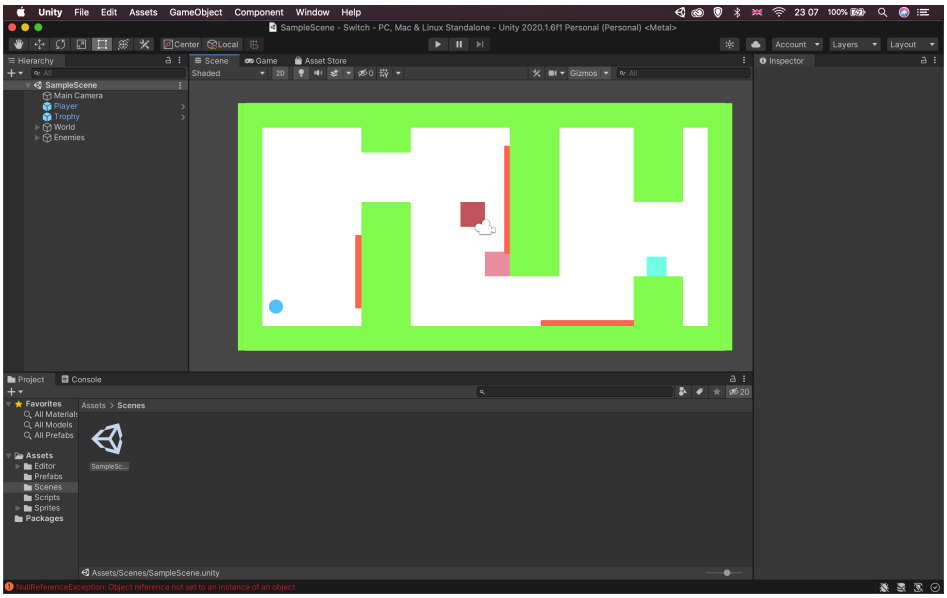
WEEK 6: SWITCH - RECREATED!

MONDAY OCT 26TH 2020:

- I had to restart my old project because my tilemap glitched out (tiles of random pink colors showed up and they wouldn't disappear) — luckily, I was able to have a better fresh start!

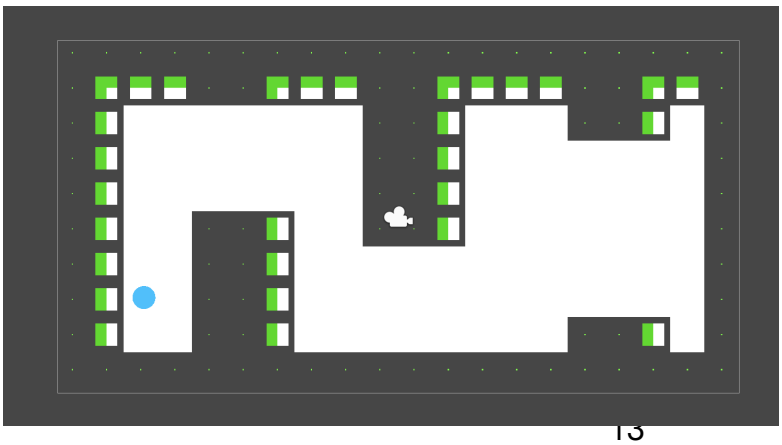


Spent 1 hour trying to fix it....didn't work :(
Total: 1 hour

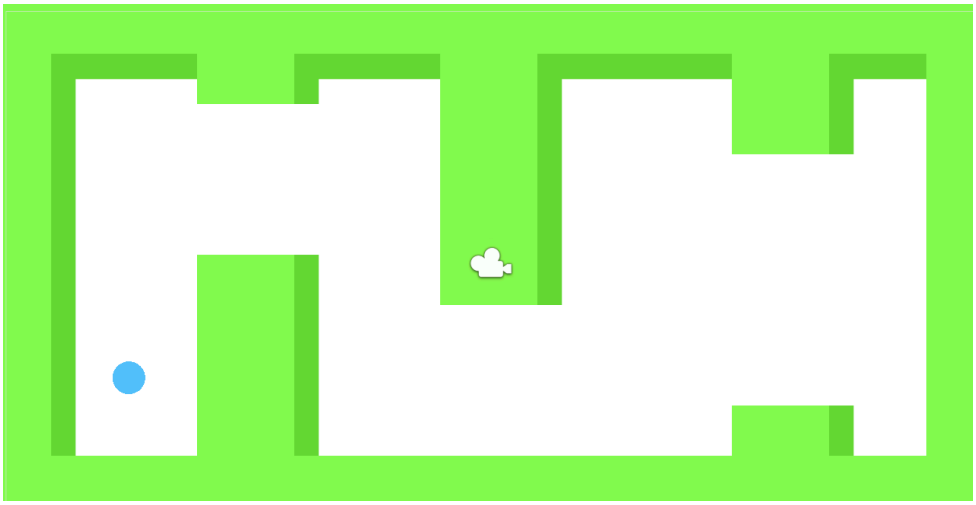
WEDNESDAY OCT 28TH 2020:

- I restarted my Unity project and got to work making a more dynamic background.
- This took A LOT of time to figure out, because every sprite I tried to make (from Unity) smaller than than 4x4 pixels would become super small and disconnected from the other tiles.

Like this:

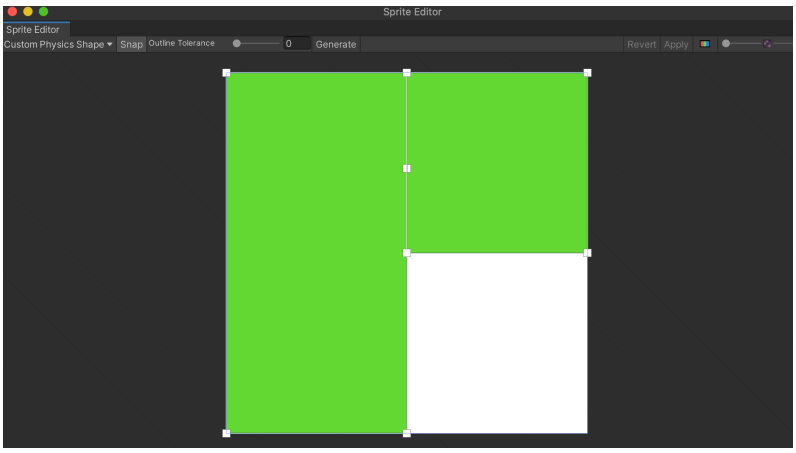


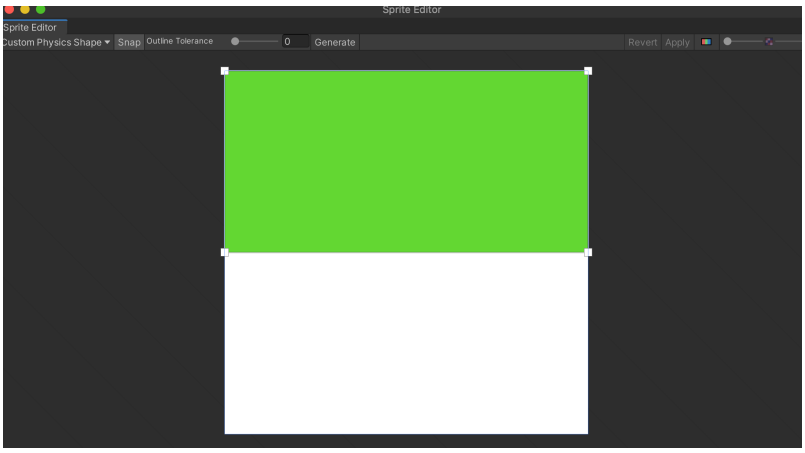
- This took an hour to solve because I actually had to create my own sprites that were a specific size (ex. 64x64) so I could make designs that were more complicated than just a colored square (still not that complicated since I was just trying to make half of the tile one color and half of the tile another color)
- I made my own custom Sprites using an online pixel art editor called Pixilart (https://www.pixilart.com/). I made sprites for the wall shadows (and intersections)
- Thus, I was able to make the image below!! It isn't a design that I did on a software but a direct screenshot from my unity game! I put a tilemap collider around my green walls (+shadows) and altered the physics shape of my shadow sprites in the sprite editor.
- This is because those shadows are actually half a darker green color and half white (the background color) to create the illusion of shadows. I made them sprites because I also want the player to be able to collide with them, like the darker side of a 3D wall.
- This makes the game more dynamic while keeping it 2D.
- I also had to make half of the sprite white (the background color) or else the empty part of the sprite would just be grey by default, and that would create a jagged, and pretty unpleasant design.
-



➡ **Wall Shadows**

- So, by alterating the physics shape in the sprite editor I can make it so that my ball only collides with the darker green part, not the white part. All that's left to do is reset the tilemap collider and now my ball can touch the darker green parts, having the illusion of 3D walls while remaining a 2D game!

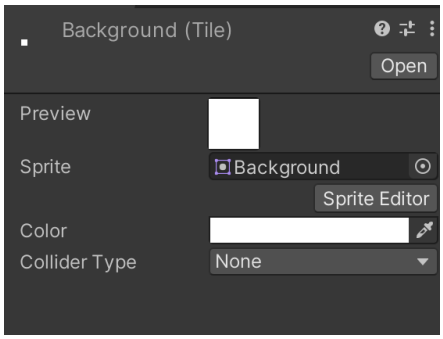


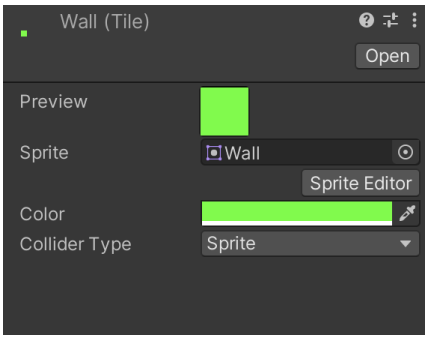


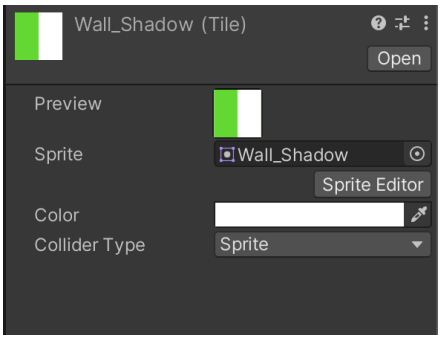
➡**Custom Physics Shape**

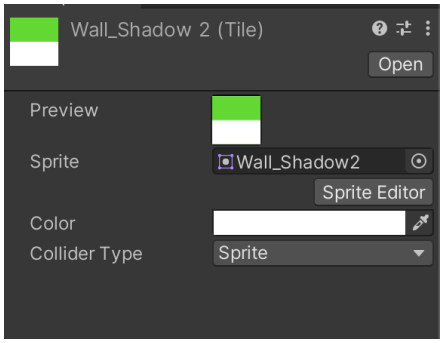
I also had to add a rigibody2D to the tilemap (and make it static so it didn't act like an object that's affected/moved by gravity and other objects running into it) and a composite collider, so I could make only specfic sprites collide-able (if that's a word).

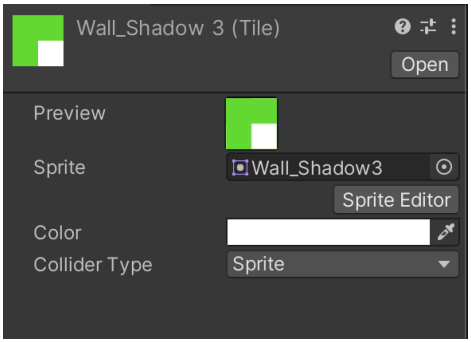
In my tile folder, I then only set the green walls and darker shadows collide type as "Sprite" whereas the white background is set to "None" because I don't want the player to collide with the white.





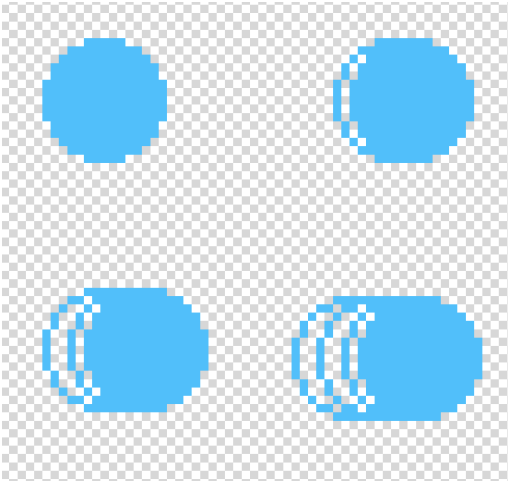






- I was able to find the solution to my problem of having unexplained small tiles by looking back through my Ruby tutorial, as I knew that I was able to make a very intricate tilemap throughout that tutorial—so a solution must have been possible.
- Then, when looking at the dimensions of the sprites in Ruby, and remembering how I had split them up, I was able to come to the conclusion that I had to make my own sprites of my own unique size and add them to a New Tile. Then, I would be able to scale it to the proper shape that I want.

- Tried designing some custom animations but they didn't work out :(



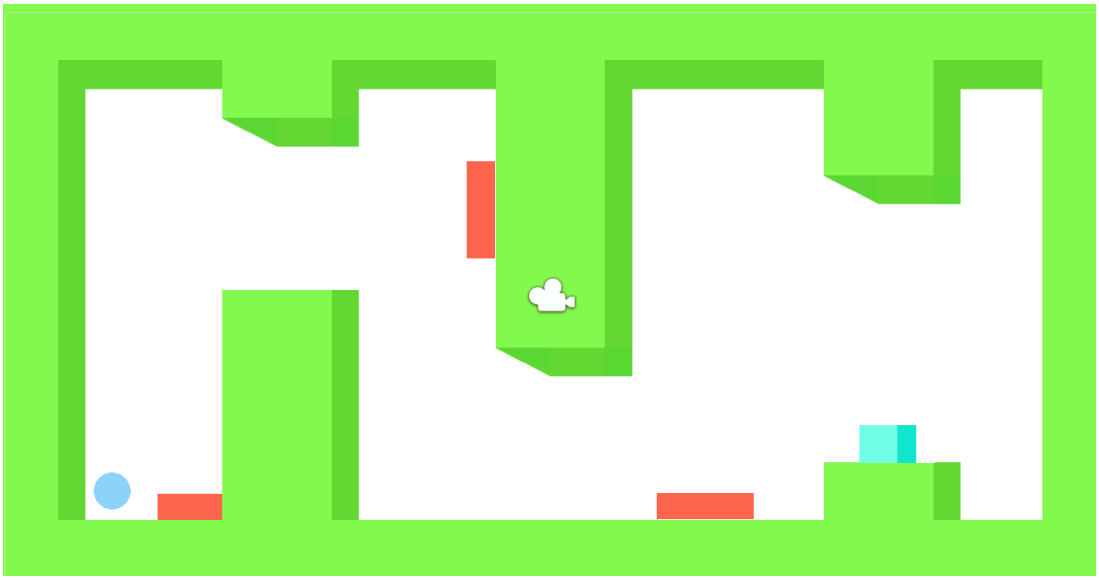
- Added a enemy script and when ball touches an enemy they restart the level. I think I have it so that I can change which level is being restarted accordingly, but I need to research this function a bit more.

```
public void Terminate()
{
    Application.LoadLevel(0); //load the level by its name or index

    //index = the level to load
    //name = the name of the level

    //thoughts:
    //if hits trophy later on, I can increased loadedLevel by 1?
    //and then if death count surpasses a certain number I can set loadedLevel back to the first one.

    }

}
```

Total: 4 hours

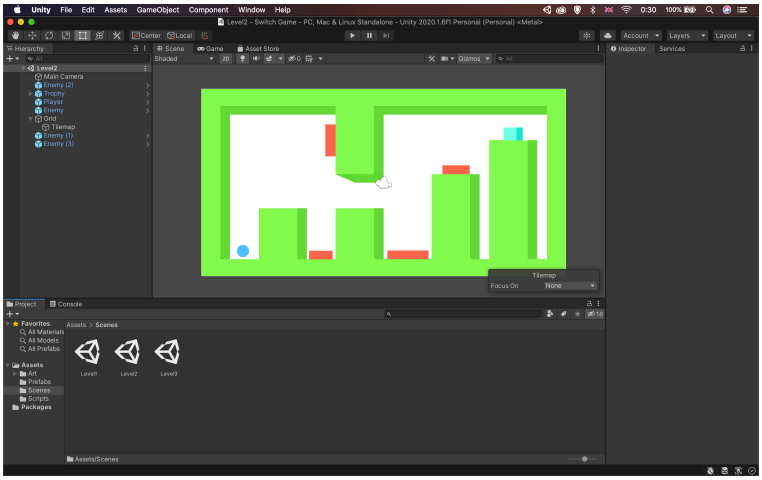
FRIDAY OCTOBER 30TH 2020:

- Worked a bit more on the dynamics of the setting.
- Came up with this (still working on it)



Added trophy function that uses Scene.Management functions, so that player can advance to more levels.

Added Scenes:

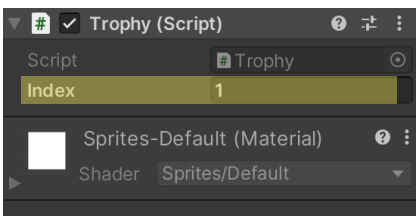


And designed a second level! (Still going to add to it)

Because all these game objects I made in the first level are prefabs I don't have to redo them.

I utilized this with the trophies. I can make the index for each level a public int variable in trophy and simply set which scene I want the trophy of a specific level to go to.

```
public int index; //this way I can make a specific trophy go to a specific level in
//ex. trophy on level1 has index set to 0, trophy on level2 has index set to 1, etc.

// Start is called before the first frame update
void Start()
{

}

// Update is called once per frame
void Update()
{

}


void OnCollisionEnter2D(Collision2D other)
{
    Player control = other.gameObject.GetComponent<Player>();

    if (control != null)
    {
        SceneManager.LoadScene(index); //load specific scene
```



Created a new scene then when to Build Project Settings, and added the new scene there so once I build and run my game, all the scenes will be apart of the final product.

➡ Trying to find a more efficient way to “kill” the ball. Basically, as I have it now, I just reset the levels. However, that makes thigns tricky when I want an explosion particle effect once the ball “dies” (the level will reset before it can play the particle effect).

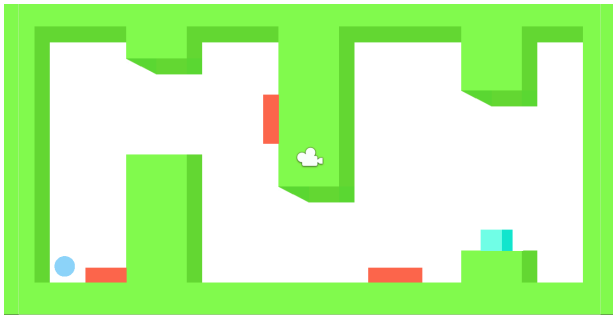
I am currently looking up on the Unity website if there are any functions that allow you to respawn an object, so I don’t have to destroy the ball or reset the level everytime the player dies.

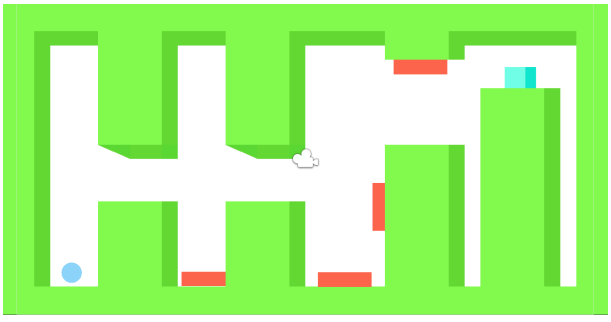
This reset function looked promising at first because it would be helpful when I include the tokens I had in my initial game later on. With this function, I wouldn’t have to set the amount of tokens collected to zero. In processing last year I had to create my own manual reset function that reset the ball’s position and all the items it had collected. Using this function would have prevented that, however, it will cause more problems in the future (such as particle effects, death count, etc.), so I’m going to try and find a way to kill the ball without reseting the scenes.

Total: 3 hours

WEEK 7: SWITCH UPGRADE

MONDAY NOVEMBER 2 2020
- Finished 2 levels
- Added trophy + collisions
- Worked on particle effecs for when the ball dies
- Ran into lots of errors with said particle effect.





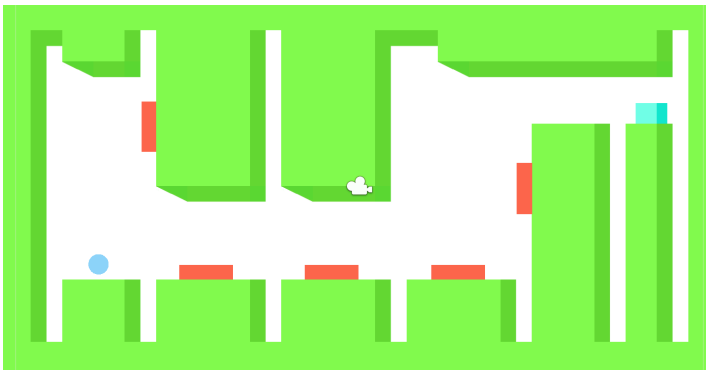
- Particle effects keeps delaying or not showing up.

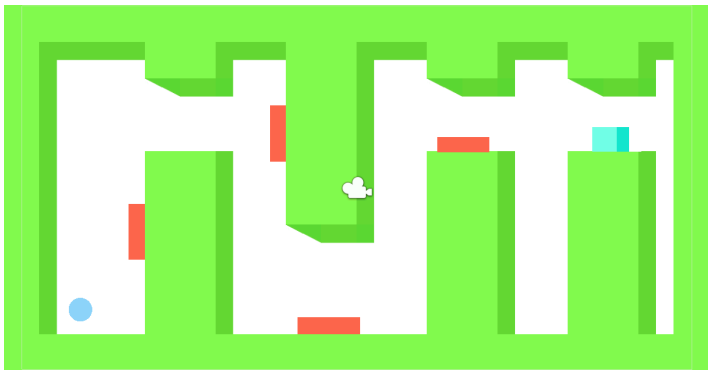
Tried:
- Setting a delay to the effect in case it was running before my game fully loaded

- Reordering the particle system and ball
- Making a new layer for the ball so that (1) the particle system didn't collide with it and (2) maybe would show up all the time. Still doesn't do that.
- Setting the emission rate lower—maybe lag was due to too many particles showing up at once
- Similarily, checked the Max Particle counter. It was 1000. If it was less, maybe that would explain why some particles won't show, but since 1000 is way more than the amount of particles I'm telling to show up, it's not this.
- Checked the effect with the 3D system.
- Preview in both 2D and 3D show up normally.
- Checked script multiple times to make sure it was instatiated correctly (it is).

Total: 4 hours

WEDNESDAY NOVEMBER 4TH 2020:
- Particle effects still don't work but that's okay.
- Looked up on the Unity website and forums if this was normal. One person had a similar problem but it was never fixed properly.
- Moved on to making more levels and brainstorming a better idea to kill the player as my current way of doing so is inefficient.



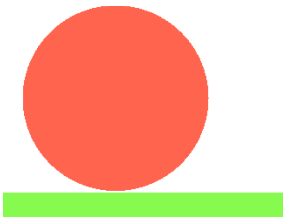


- Finished Two more levels
- Created a trophy function so that player could go in between levels
- Made the index of each level a public variable so that I can control which level the trophy takes you to within each level.
- Tried to get Particle effects working.

Total: 3 hours

FRIDAY NOVEMBER 6TH 2020:
- Created a MovingEnemy function for the boss of the "Green level".
- Created an object with a Rigidbody2D and a Box Collider and set the y and z variables to freeze rotation (so that it wouldn't move around)
- Also set it as a trigger so that the player wouldn't be able to push the enemy around.

- Applied this to all the other static enemies as well.
- This is because I am sharing the Enemy class with both the static and moving enemies. Since they both kill the player in the same way, except one just moves, I can create a moving enemy class that lets the enemy moves back and forth for certain durations, while still killing the player like any other enemy would.
- This follows the design I made last year in processing.



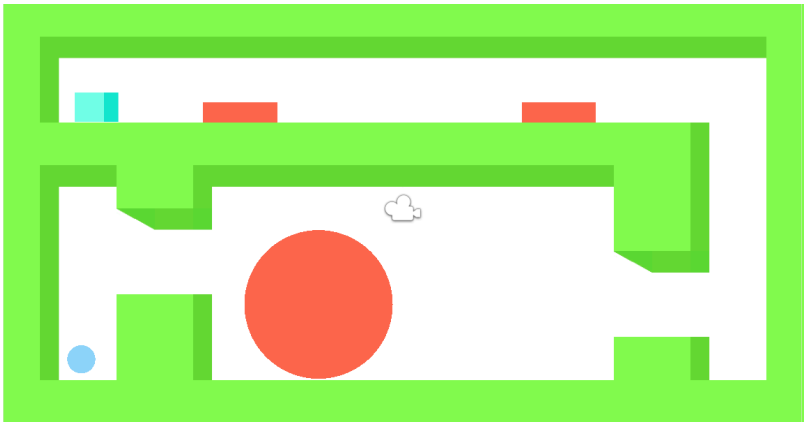
- Started brainstorming for the Token Class (for the next group of levels)
- Tried to get particle effects working (didn't work :(...)

Total: 3 hours 45 mins

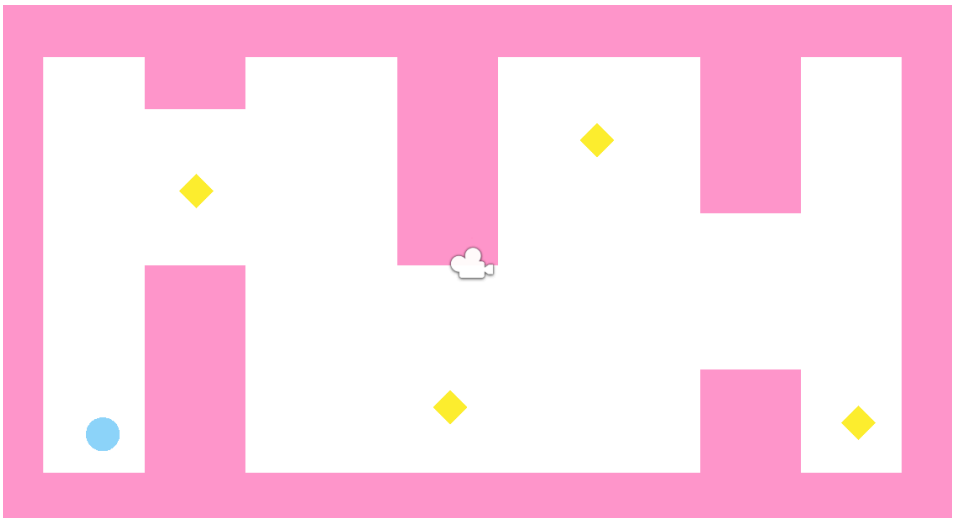
WEEK 8: FINISHING FUNDAMENTALS

- The goal for this week is to finish as many fundamental of the game as possible

MONDAY NOVEMBER 9TH 2020:
- Finalised the last Green level:



- Coded the Token Class and attached it to the token prefab.
- Created a public function within the player Class that increases the amount of tokens you have by 1, which gets called in the Token class everytime you collect one.
- The tokens are set to "Is trigger" as well so the ball can't move it around, even though it gets destroyed after being collected.



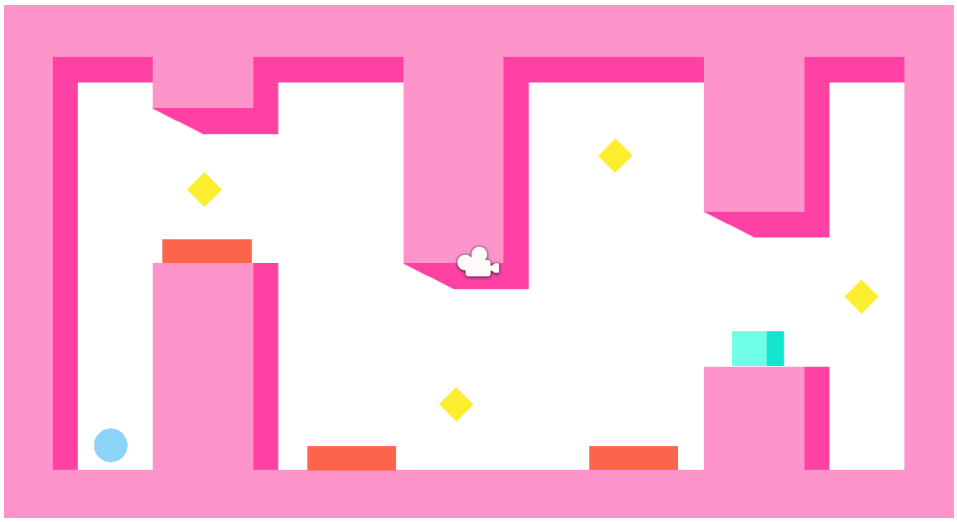
- Created basic template in Unity of the Pink Levels, where the new component of tokens is introduced.

Time: 2 hours 30 mins

- Changed it so that a new level is loaded in the player's class. This is so, when I need the player to collect a certain amount of tokens before proceeding, I can access the amount of tokens the player has and compare it to the amount of coins needed.
- Currently running into an issue where each time a coin is collected it increases the token_num by 2. I think this is because the software is percieving the destroying of the gameobject as the player collecting a token.

- Added a respawn function. Now my death function is just a Destroy(gameObject).
- Added a LevelManger Class so that I can change the respawn point in any level.

Time: 1 hour 30 mins



Finished designing first pink level. Had to make new pink sprites and edit their custom physics shape but since I've done this before it went along a lot faster.

Time: 30 mins

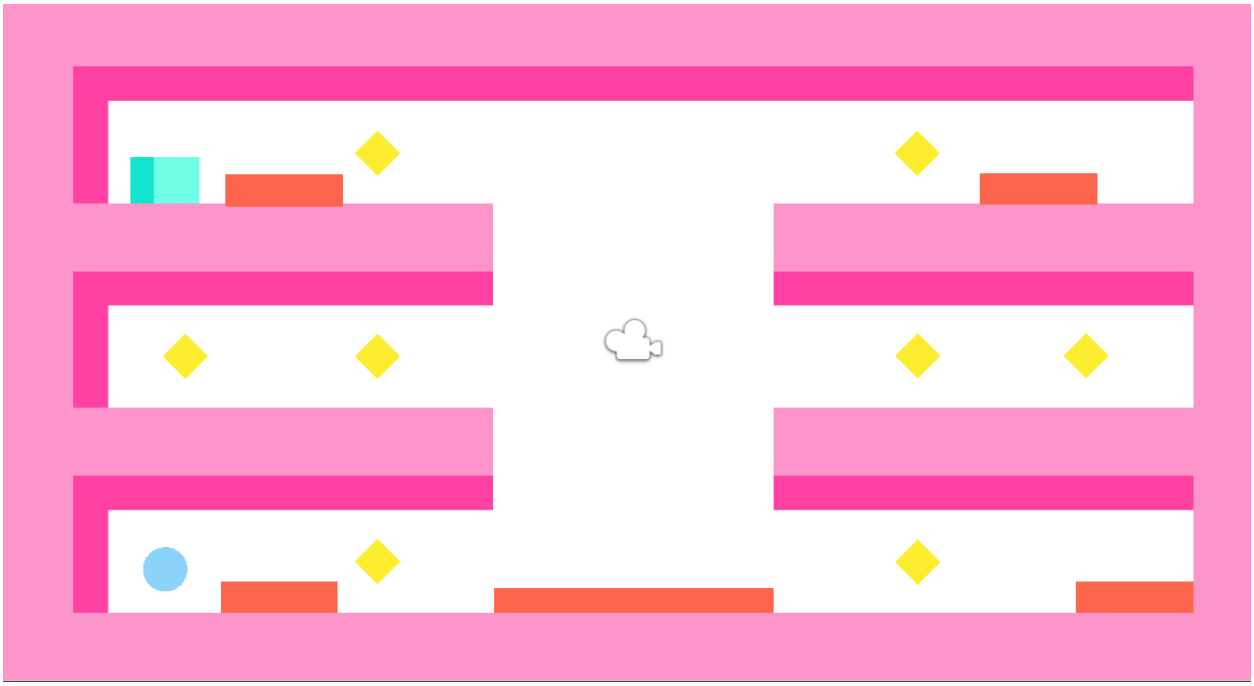
Total Time: 4 hours 30 mins

FRIDAY NOVEMBER 13TH 2020:

Changed it so that all global variables and functions (basically the organization of the level) is in the Level Manager Class.

This class will now handle the total amount of tokens collected, the respawning of the ball, the death count, and the "respawning" of the tokens.

Also began design for future levels:



Changed all my enemies and basically evertyhing the ball can collide with to "Triggers" because I don't actually want the ball to be able to move them around. I noticed this problem when interacting with my Moving Enemy. As I want it to share the enemy class, making it a trigger allows the ball to be killed by it without pushing it around.

Also decided to change this for my trophy function because I don't need the player physically bumping into the object. It can pass through if it wants as the appropiate functions are called immediately after collision.

Total Time: 2 hours

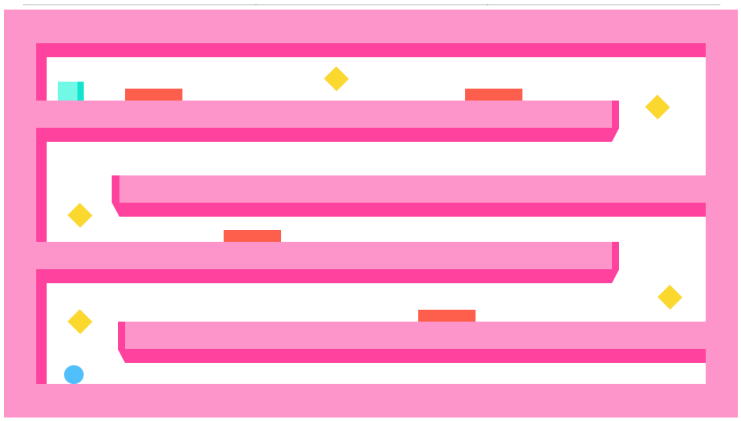
WEEK 9: PROGRESS

MONDAY NOVEMBER 16TH 2020:

Brought back the "restart level" funciton as a temporary solution to make the coins respawn when the player dies.

Encountered an error where I can't access my tile palette anymore—I believe it's because I have to update Unity. However, I don't necessarily want to update Unity in the middle of my project as it may change code/functions I already implemented.

In the meantime, I've been designing levels and editing code.



I've been thinking about gradually making the walls of levels thinner so that more can be added to it, and the difficulty will thus increase. I'm even thinking on making the levels "taller" and adding a cinema camera to the ball so it can track the player's movement up the more complex levels.

Currently can't really start working on this idea because my tile pallete refuses to open :/ and I'm not sure how to manually update Unity. Normally a notification would pop up so I'm waiting for that to happen again.

Also added a line of code that makes the tokens rotate!

WINTER TERM

WEEK 1: DETAILS

TUESDAY DECEMBER 1ST 2020:

Here's a little plan of what I want to do over these next few weeks:
- cinemachine
- timer
- 2D Lighting
- Global Variable
- Homescreens / pause screens

For the homescreens and 2D Lighting I am mainly focusing on research, reading the two sources I found at the beginning of the year, and watching official Unity Tutorials.

For the cinemachine and timer I'll still need to research a bit, however, I have some prior experience with Ruby's 2D Adventure. I'm trying to find a way to make a global death count, but Unity doesn't really deal with global variables so I need to find some loop holes.

The last thing I want to do, around the last week, is just start on the yellow levels and implement the timer.

THURSDAY DECEMBER 3RD 2020:
- Added a cinemachine camera to level
- Added confiners
- Follows Ball
- Problem: Once the ball dies, it doesn't follow it anymore because it won't follow the clone that is created.
- Solution: I think that once I get global variables implemented, I will be able to reset levels without resetting the death count. This will fix all my issues of the tokens respawning as well.

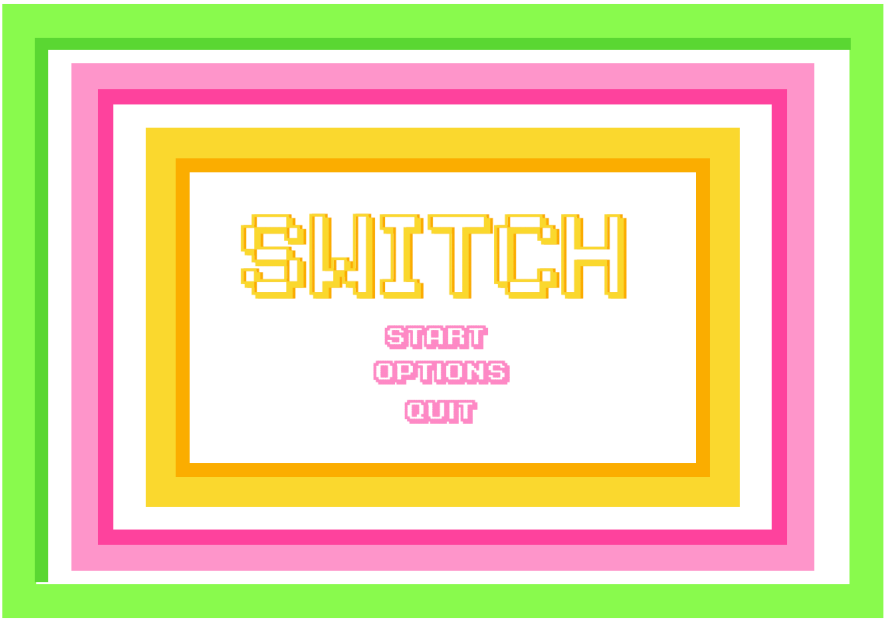
WEEK 2: RESEARCH

WEDNESDAY DECEMBER 9TH 2020:
- Added more gradually difficult levels
- Did it so that the cinemachine goes horiztonally, then vertically, then both, so that the levels increase in difficulty
- Tried to get global variable working (1 hour)

- Read and researched a lot about 2D lights (even tried implementing some but they were confusing)
- Once global variables are working I'll be able to reset individual levels. So that function is completed. But I'm still working on making the death count separate and an overarching count.

THURSDAY DECEMBER 10TH 2020:
- Read a lot more about 2D lighting and watched an official unity tutorial going over it
- youtube.com/watch?v=F5I8vP90EvU&t=189s
- I think 2D lighting works differently with tilemaps though so I'll have to research that as well.
- Found some sound effects to use
- Designed some more levels

SUNDAY DECEMBER 13TH 2020:
- Got the global manager working! (global death count works now)
- Finished the time manager
- Finished drafts for Yellow Levels
- Started research on menu screens and designed my first draft of one:

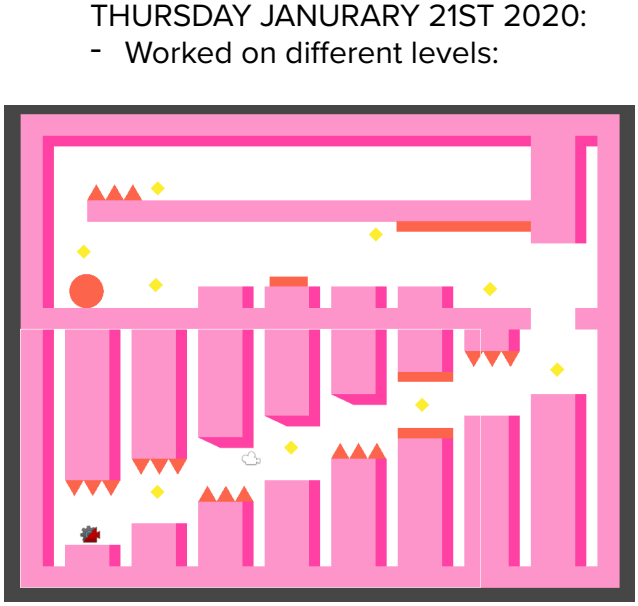


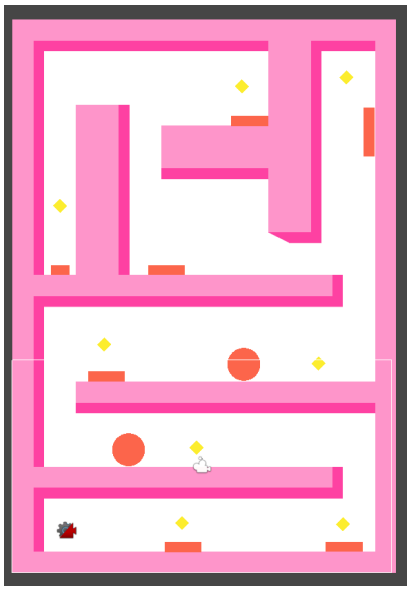
WEEK 3: FINALIZING FUNCTIONALITY

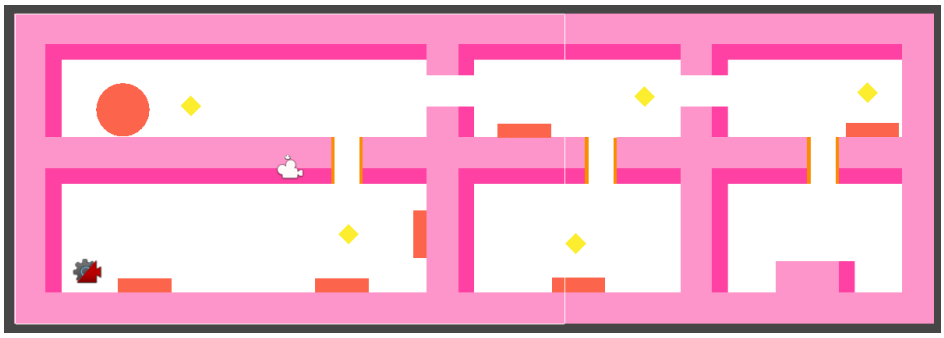
MONDAY JANURARY 18TH 2020:
- Got trophy to change colour depending on if you've collected all the tokens or not

- Researched more on menus in Unity
- Cleaned up my code (it was really messy and full of commented out functions)
- Tried to fix some collision glitches (where the ball would move abruptly against corners)

THURSDAY JANURARY 21ST 2020:
- Worked on different levels:







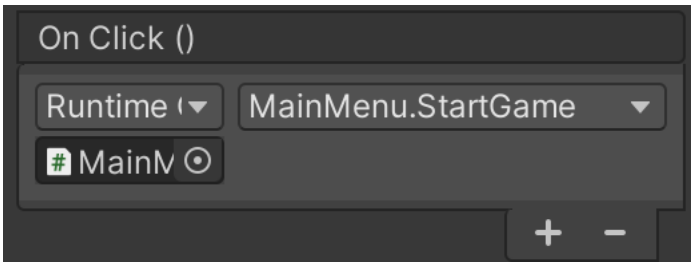
- got the spikes working, but they were overly sensitive and had weird collision behaviour.
- Fixed the preview aspect ratio of each scene. Though, when I build and run the game it still appears to zoomed in, unsure why.

WEEK 4: MAIN MENU

MONDAY JANURARY 25TH 2020:
- Created MainMenu Script

- Stripped away the draft menu screen and started working on the functionality of the screen
- Added a button component to the text I displayed on the screen
- Created text that changes color when mouse hovers over it, and then when it is clicked by the mouse
- Created text that animates (scales bigger) once hovered over
- Also, fixed the sensitivity of the spikes (the box colliders were a bit off)

WEDNESDAY JANURARY 27TH 2020:
- Got the button to actually go on the first level by using the OnClick() function.
- This was strange because I had to add the script of my MainMenu to an empty object and then drag that object (not the script) into the available section or else the program didn't recognise the function I had created in MainMenu. Essentially, using this method I can create separate functions within my MainMenu script that allow the player to navigate different screens without having to make a bunch of scripts.



MOUSE HOVER MOUSE PRESSED

- Fixed issue of aspect ration (in Thursday meeting), I just had to change some project settings (deselect all the other aspect ratios except the one I wanted 16:9, and then set the fullscreen mode to Windowed)