

Documentation

Table of Contents

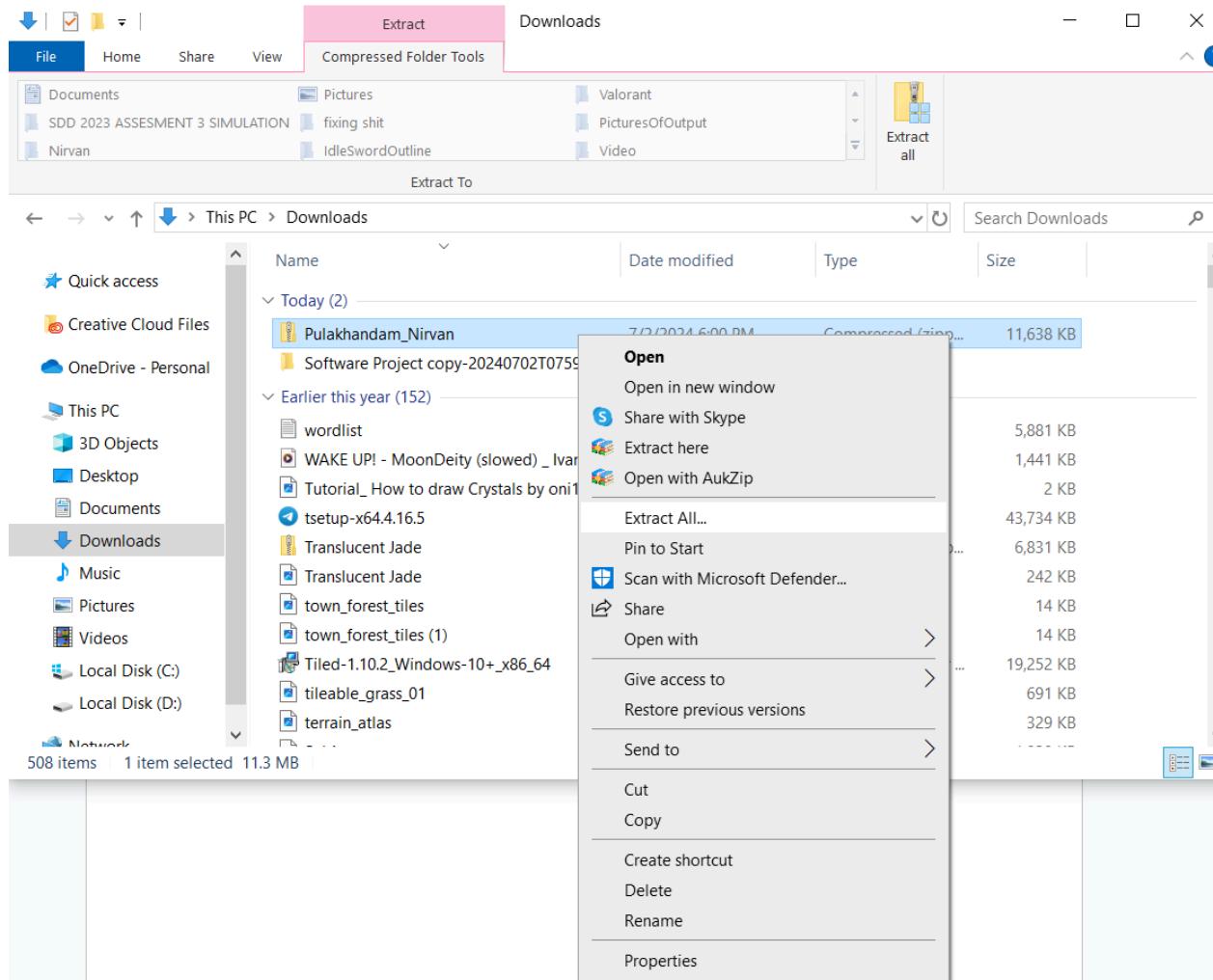
Table of Contents	1
User manual	2
Installation Guide	2
Instructions	5
Testing Report	7
Module Testing	7
Program Testing	24
System Testing	39
Resolution	40
Performance	42
Logbook	44
Evaluation	109
Bibliography	111

User manual

Installation Guide

Step 1: First ensure you have python3 downloaded on your device, then download the zip file labelled Pulakhandam_Nirvan_SDDAss3.zip

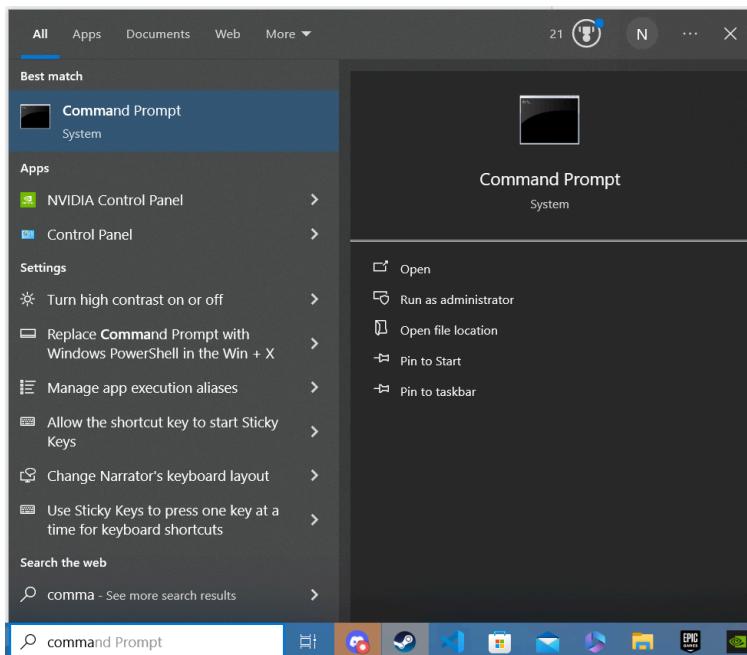
Step 2: Locate the file, unzip the file and then extract all



Step 3: Install the pygame library

Windows Instructions:

- 1) Search for command prompt



- 2) Once in command prompt type in pip install pygame and press enter which will install pygame

```
microsoft Windows [Version 10.0.19045.4529]
c) Microsoft Corporation. All rights reserved.

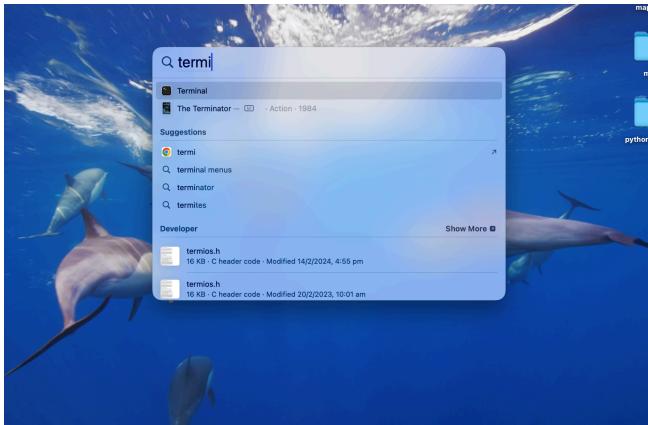
:C:\Users\Nirvan>pip install pygame
Collecting pygame
  Downloading pygame-2.6.0-cp310-cp310-win_amd64.whl (10.7 MB)
    10.7/10.7 MB 4.4 MB/s eta 0:00:00
Installing collected packages: pygame
Successfully installed pygame-2.6.0

[notice] A new release of pip available: 22.3 -> 24.1.1
[notice] To update, run: C:\Users\Nirvan\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.10_qbz5n2fra8p0\python.exe -m pip install --upgrade pip
:C:\Users\Nirvan>
```

The screenshot shows a Microsoft Windows Command Prompt window. The title bar says "Command Prompt". The window displays the command "pip install pygame" being run and its output. It shows the download of the pygame package from version 2.6.0 to 2.6.0, the installation of the package, and a notice about a new pip release. The command prompt prompt is visible at the bottom.

Mac Instructions:

- 1) Search in spotlight for terminal



- 2) Once in terminal type in the line `pip3 install pygame` and press enter to install

A screenshot of a terminal window titled "nirvanpulkhandam -- zsh -- 94x32". The window displays the output of the command "pip3 install pygame". The output shows the package being downloaded from "pygame-2.6.0-cp39-cp39-macosx_11_0_arm64.whl" (12.4 MB) and successfully installed. A warning message at the end of the output suggests upgrading to version 24.1.1. The terminal window has a dark background with light-colored text.

Step 4: Run the file main.py and then use the program as intended

Instructions

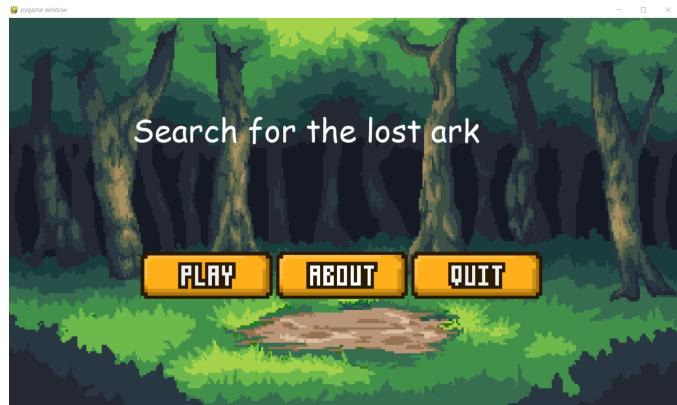
Requirements: Keyboard, mouse/touchpad, suitable device

Outline

This is a simple game which is a rendition of the popular game peggie, this game consists of a world map in which the player can run around in and explore to find levels and then the four levels which are dispersed throughout the map.

Starting the game

The start menus UI is streamlined and simple to use as there are only a few buttons:

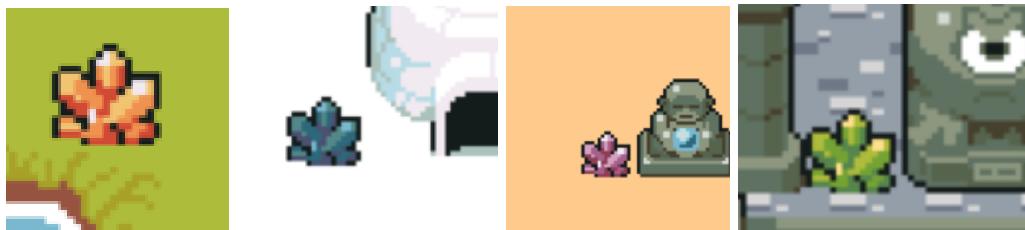


In order to access the instructions within the game the user has to click the about button and to start the game the user needs to press the play button:



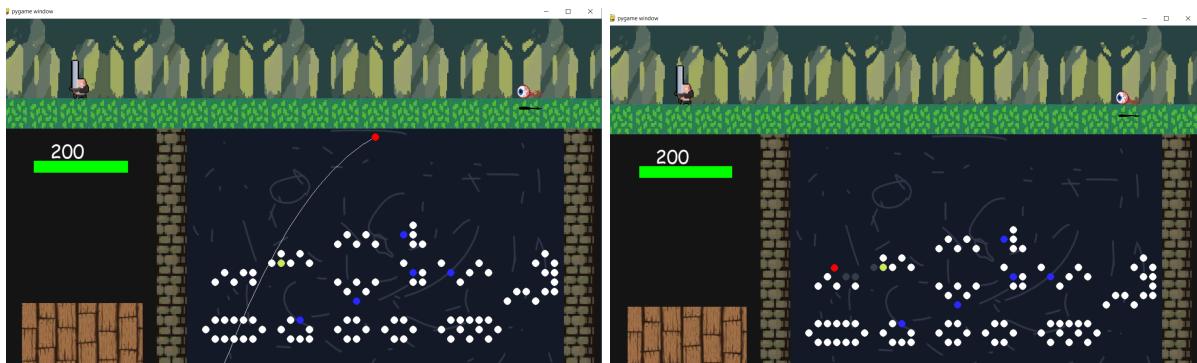
Entering levels

To play the levels within the game the player must travel across the map to the points of interest. Once the player has reached these points they just need to collide with the structure. These points of interest are marked by different colour crystals. In order to progress, the levels must be completed in order going from orange, dark blue, purple and then lastly green. These are the crystals which the player has to look for:



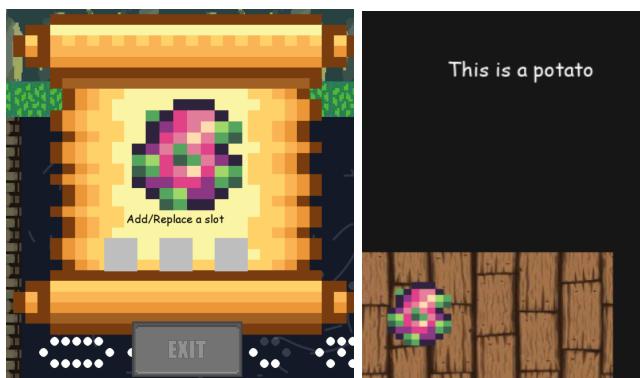
Playing the level

Once the player is in the level they must aim with their mouse and then click to shoot the ball the gameplay itself is not very complex.



Rewards

The player is able to click on one of the three grey squares on the scroll at the end of a level. This will replace the or add a fruit into the side slot. The fruit can then be clicked to be activated however it has no functionality.



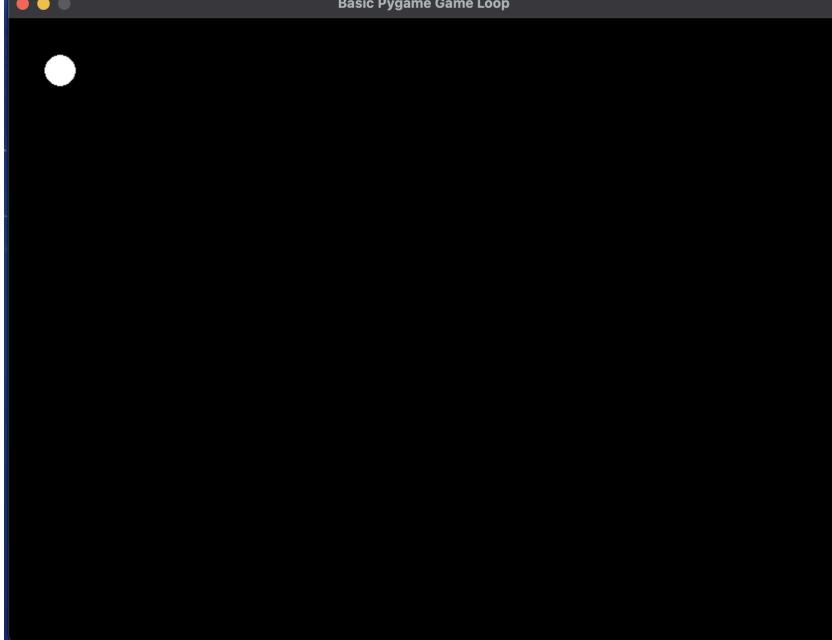
Testing Report

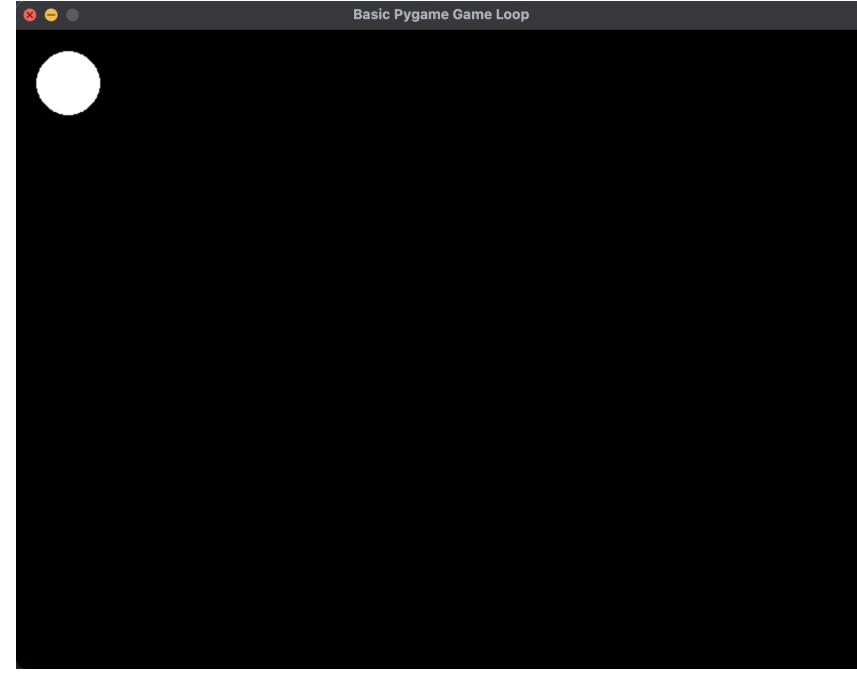
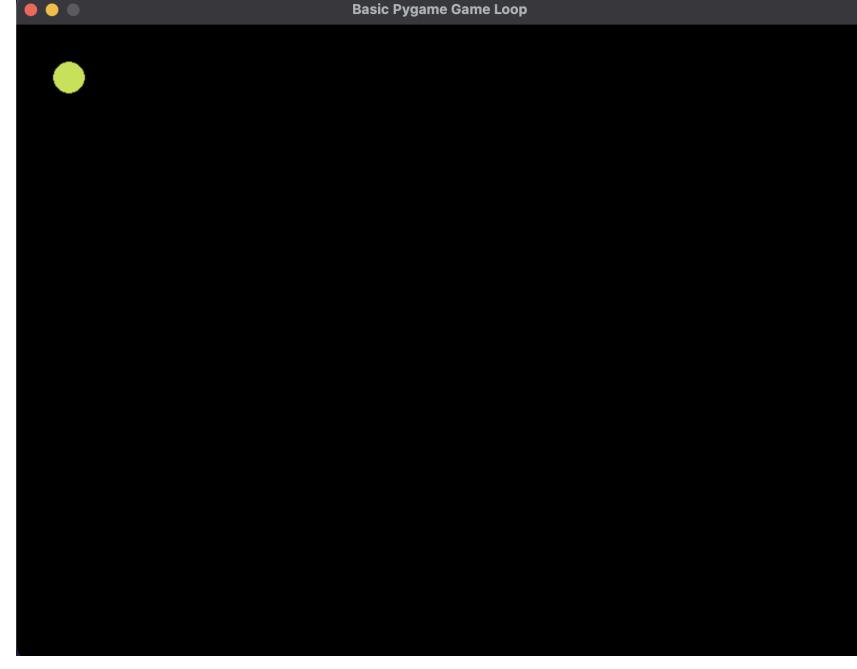
Module Testing

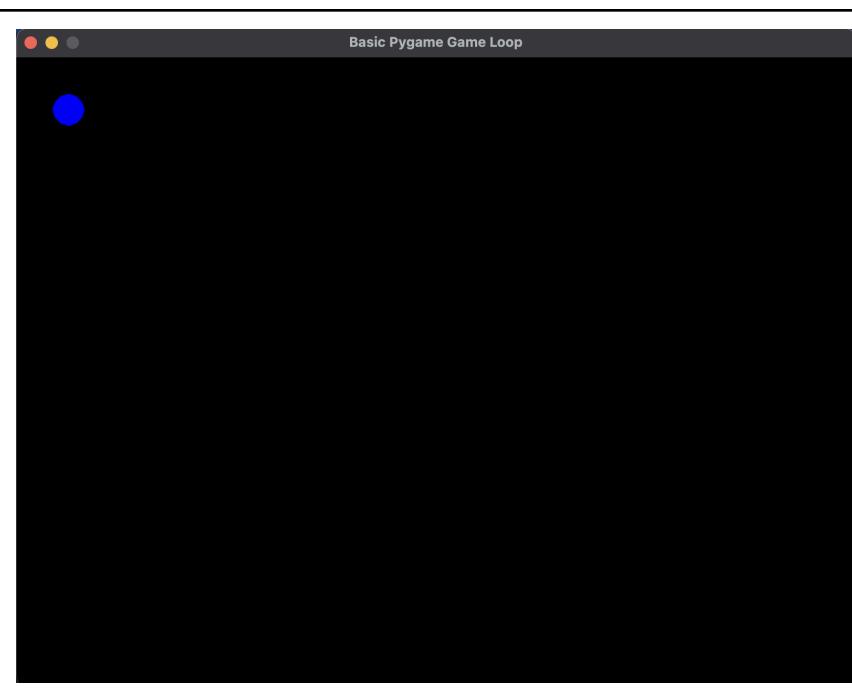
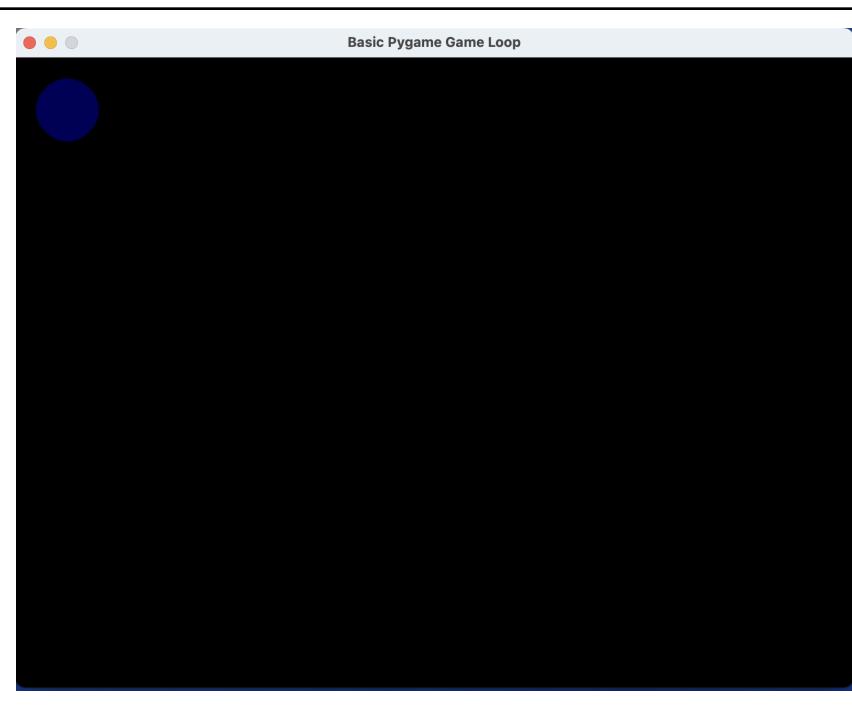
To begin the module testing I first created a new file and labelled it as code tester then I would be able to paste in code that I wanted to test and run the code. I needed to specifically create this new file as to run code I need to create a pygame loop in order to make a screen to display any graphical components, then I tested each class as a separate module:

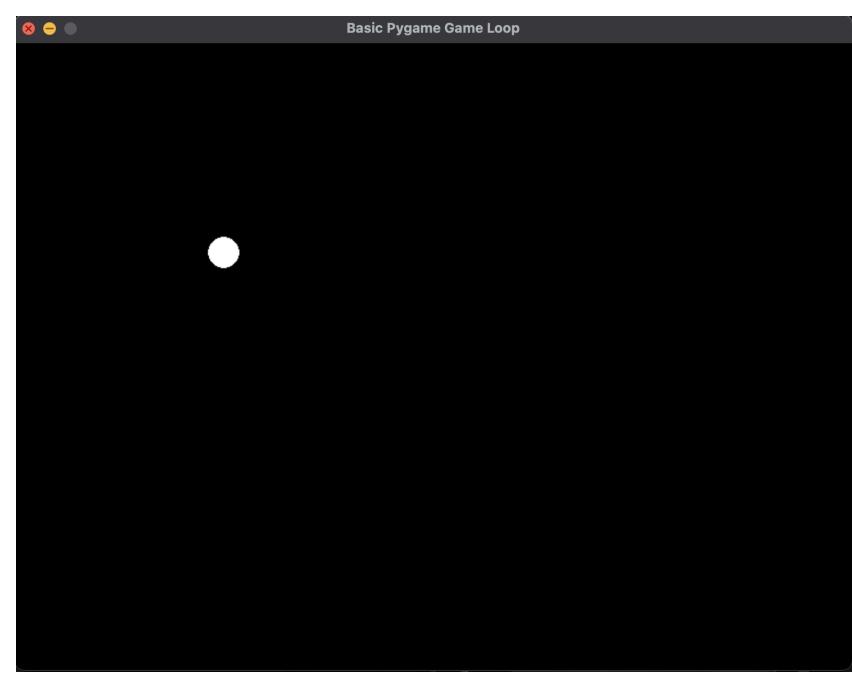
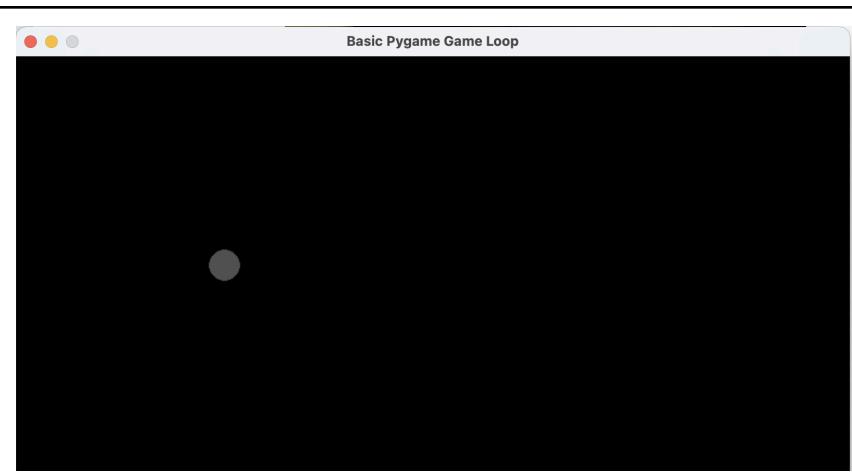
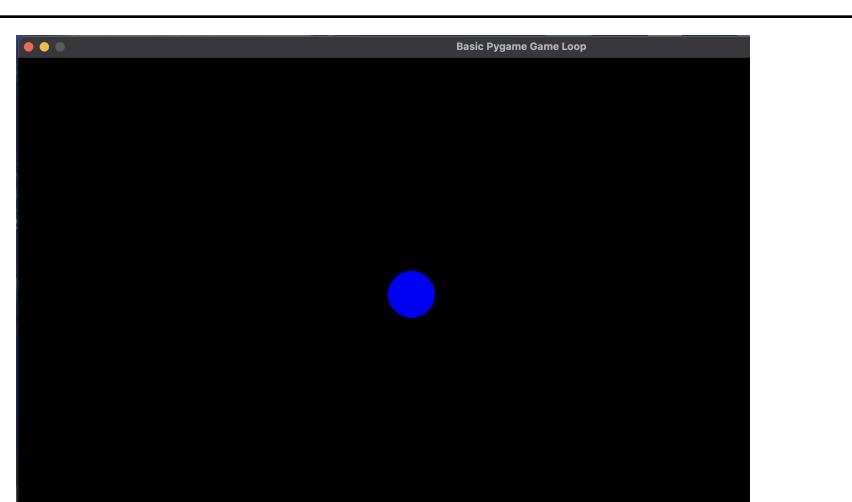
Peg(x, y, n, type)

This class creates a single ball at the x and y coordinates passed in and the radius is dependent on the PEG_RADIUS variable which is set outside of the class as a global variable. The variable n controls the opacity of the ball however if the number is 0 then no opacity is applied to the ball. This class then creates a single sprite at the specified location and which is then added to a sprite class and drawn.

Input	Expected Output	Output
PEG_RADIUS = 15 X = 50 Y = 50 n = 0 type = 'Normal'	This will draw a white ball as that is what the colour of the 'normal' type is. There should be no opacity set on the ball as n is 0. This is drawn at the coordinates (50, 50) which is the top left of the screen.	

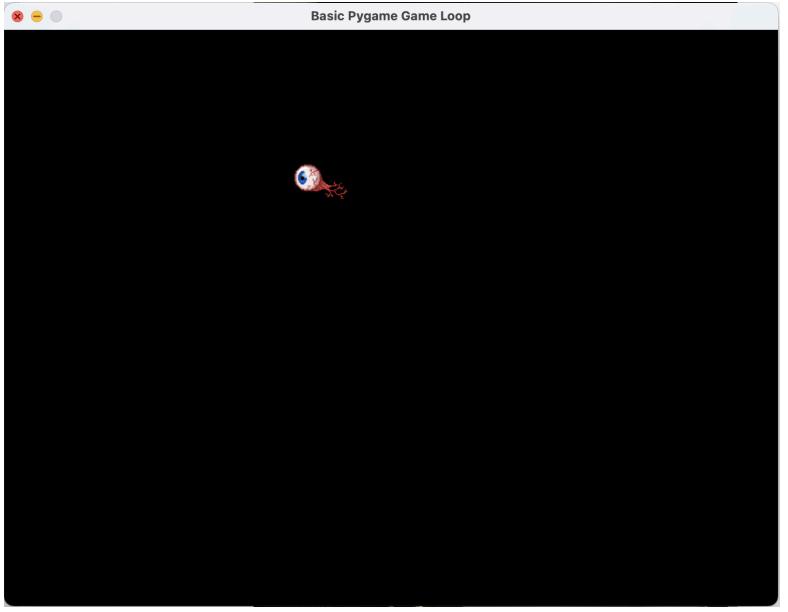
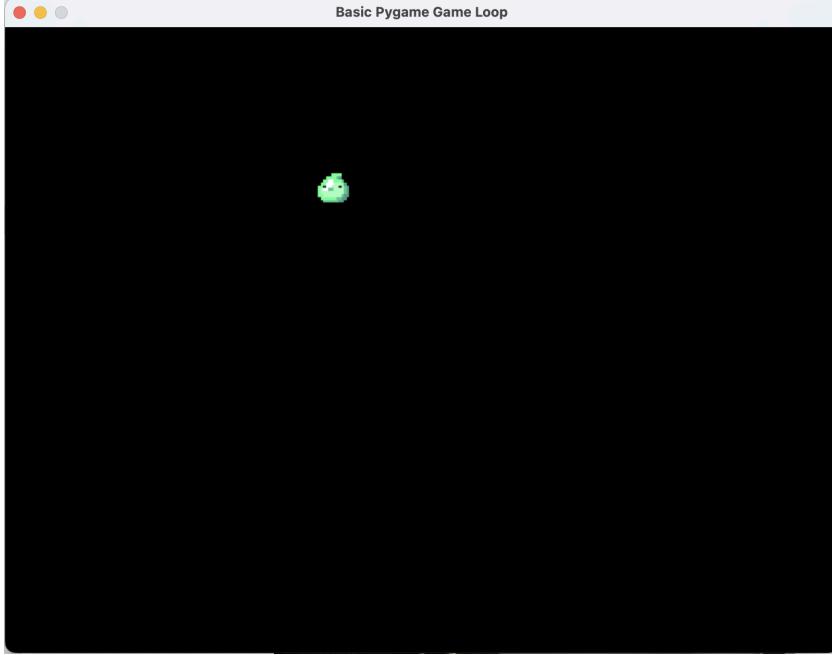
<pre>PEG_RADIUS = 30 X = 50 Y = 50 n = 0 type = 'Normal'</pre>	<p>This should create the same type of ball at the same location except much larger as the radius has been changed from 15 to 30.</p>	
<pre>PEG_RADIUS = 15 X = 50 Y = 50 n = 0 type = 'Reset'</pre>	<p>A ball should be drawn in the same coordinates as the first except it will be a green ball as the type has been set to 'reset' which changes the colour of the ball drawn to green</p>	

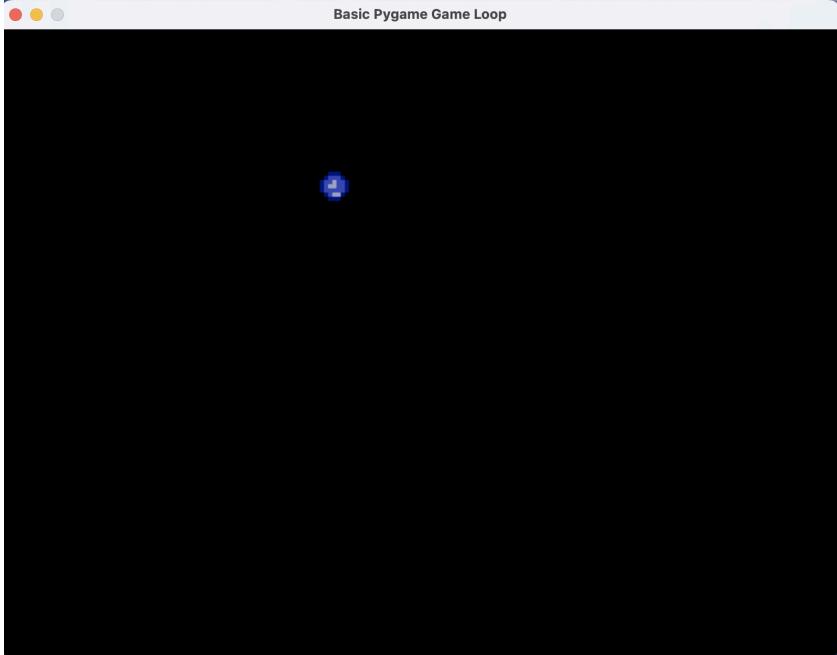
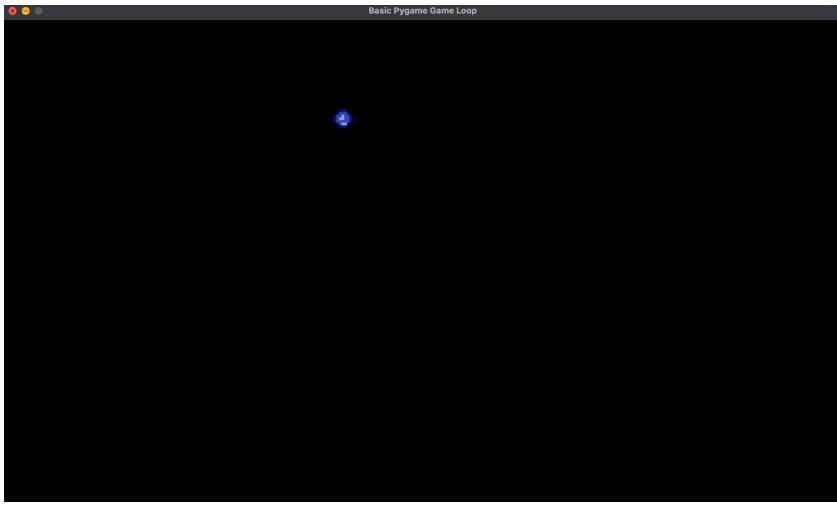
<pre>PEG_RADIUS = 15 X = 50 Y = 50 n = 0 type = 'bomb'</pre>	<p>A ball should be drawn in the same location with the same opacity as the first as no values have been changed, however a blue ball should be drawn as they type is 'bomb' which changes the colour of the ball drawn to blue</p>	
<pre>PEG_RADIUS = 30 X = 50 Y = 50 n = 70 type = 'bomb'</pre>	<p>A blue ball should be drawn as the type is blue, it should be opaque as n is set to 70, the radius is 30 and it should be in the coordinates(50,50)</p>	

<pre>PEG_RADIUS = 15 X = 200 Y = 200 n = 0 type = 'normal'</pre>	<p>This will create a ball with radius 15 at the coordinates (200, 200) as this is the location of x and y and the balls colour will be white as the type passed in is 'normal'</p>	
<pre>PEG_RADIUS = 15 X = 200 Y = 200 n = 80 type = 'normal'</pre>	<p>Creates a white ball with a slight opacity at x any coordinates (200, 200)</p>	
<pre>PEG_RADIUS = 15 X = 500 Y = 300 n = 0 type = 'bomb'</pre>	<p>The will be drawn in the coordinates (500, 300), this will be drawn closer to the centre of the screen with no opacity and will be blue</p>	

`Monster(type, health, level)`

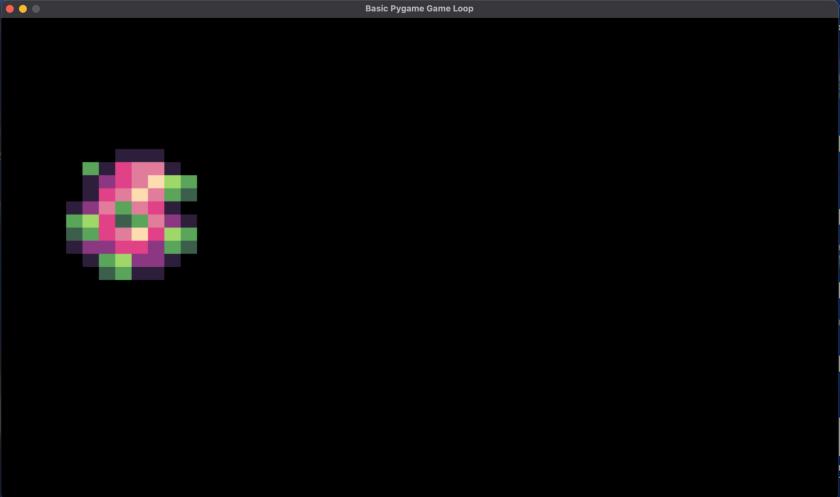
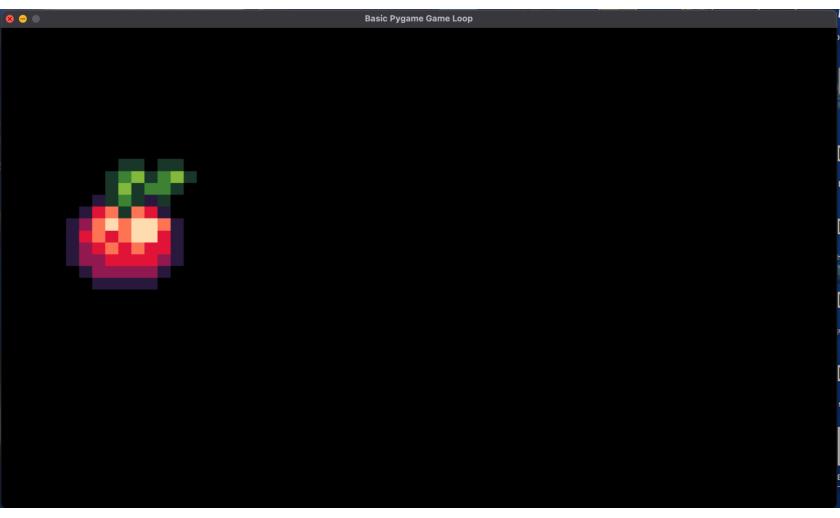
This class creates a monster with the type corresponding to which image will be drawn, the health determines the health of the monster and the level does nothing as it is a feature that was never implemented. Then the draw method `draw_monster(n)`, which is passed in n which is the x location of where the monster will be drawn but is drawn in the same exact y location each time.

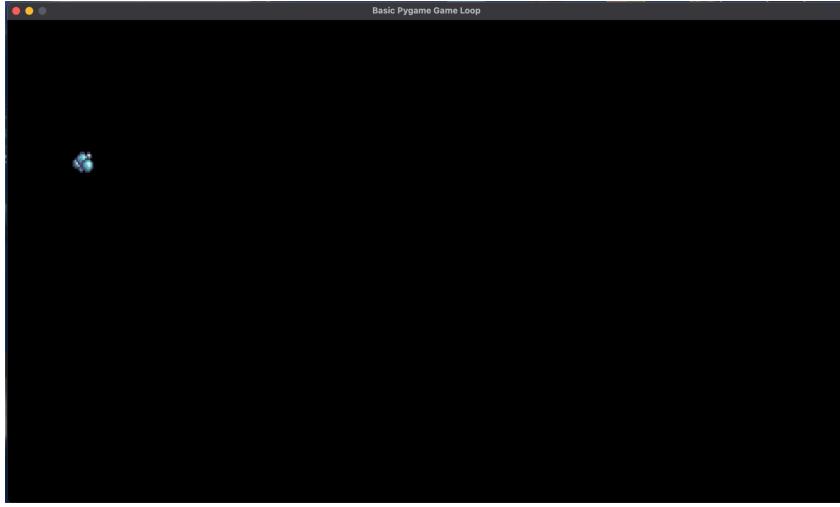
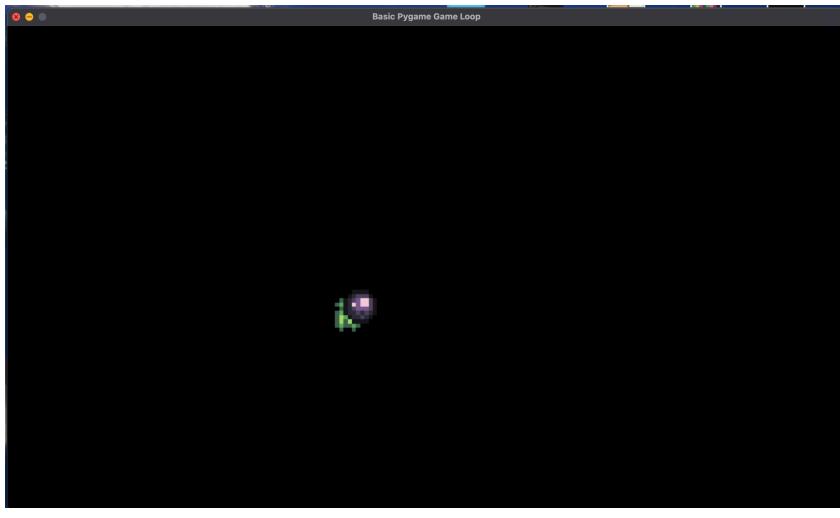
Input	Expected Output	Output
Type = 'eye' Health = 100 n = 300	There will be an eye monster drawn at x location 300	
Type = slime Health = 100 n = 300	There will be a slime monster drawn at x location 300	

Type = orb Health = 100 n = 300	This will create an orb at the x coordinate 300	
Type = orb Health = 100 n = 500	This will create an orb with the x coordinate set to 500	

`DevilFruit(type, x, y, sx, sy)`

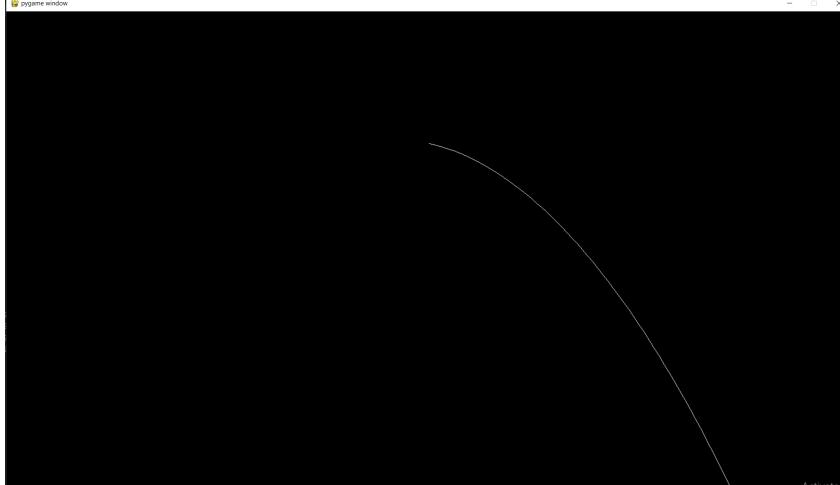
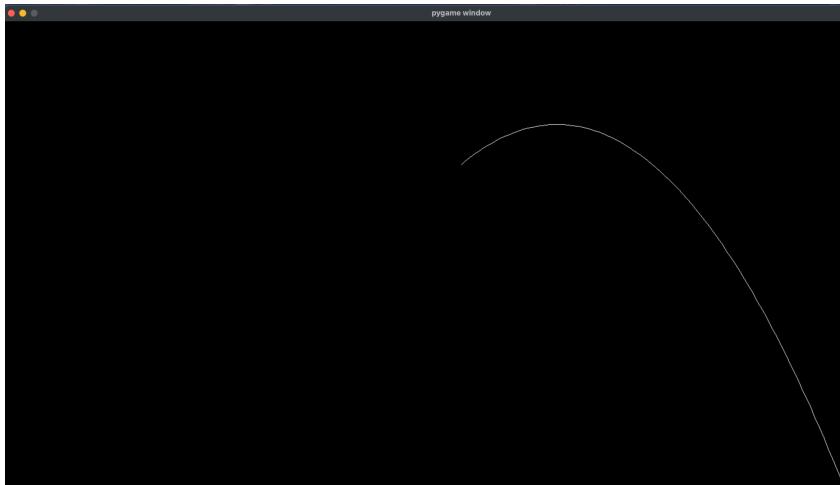
This draws a fruit at the x and y coordinates based on the x and y passed in and scales the fruit according to the sx and sy passed in. The image drawn is dependent on what is passed in for type.

Input	Expected Output	Output
Type = potato X = 100 Y = 200 Sx = 200 sy = 200	There should be a potato drawn at 100 200 with the scale size of 200 x and 200 y	 A screenshot of a Pygame window titled "Basic Pygame Game Loop". It shows a single pixelated potato fruit centered on a black background. The potato has a purple skin with green and yellow spots, and a small brown stem.
Type = apple X = 100 Y = 200 Sx = 200 sy = 200	There should be an apple drawn at 100 200 with the scale size of 200 x and 200 y	 A screenshot of a Pygame window titled "Basic Pygame Game Loop". It shows a single pixelated apple fruit centered on a black background. The apple has a red skin with a yellow core and a green stem with a leaf.

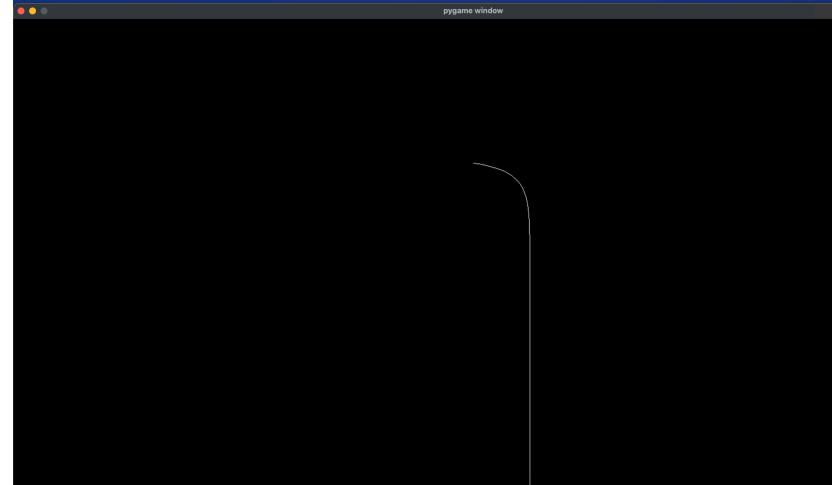
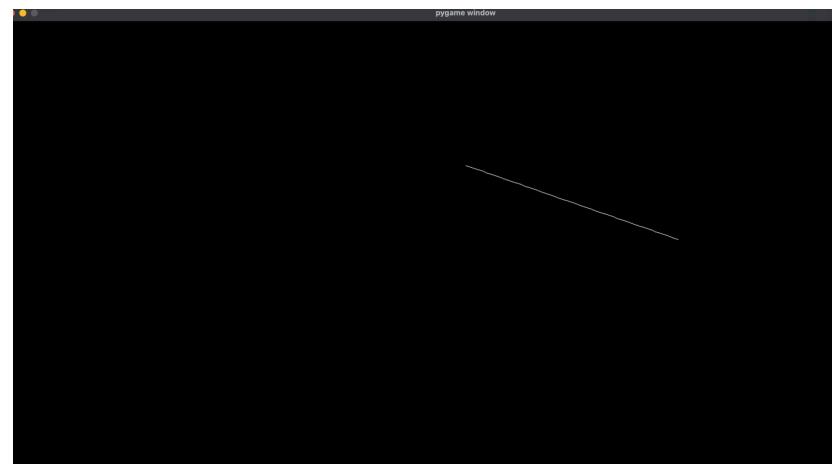
Type = blueberry X = 100 Y = 200 Sx = 32 sy = 32	There should be a blueberry drawn at 100 200 with the scale size of 32 x and 32 y	
Type = plum X = 500 Y = 400 Sx = 64 sy = 64	There should be a plum drawn at 500 400 with the scale size of 64 x and 64 y	

Aim()

This class is passed no variables but has some important variables within the aim class that controls how the trajectory line is drawn, these are gravity, friction and the number of times the for loop is run which controls the length of the trajectory line.

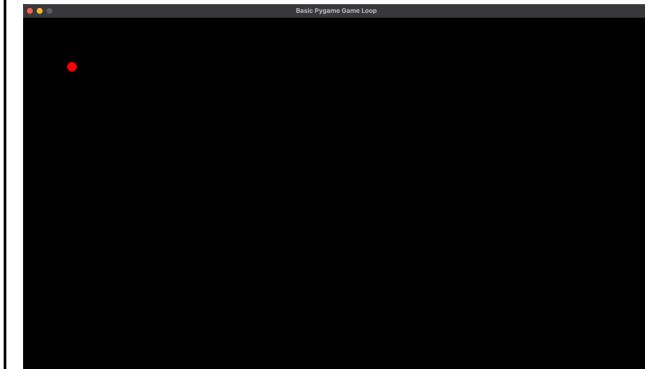
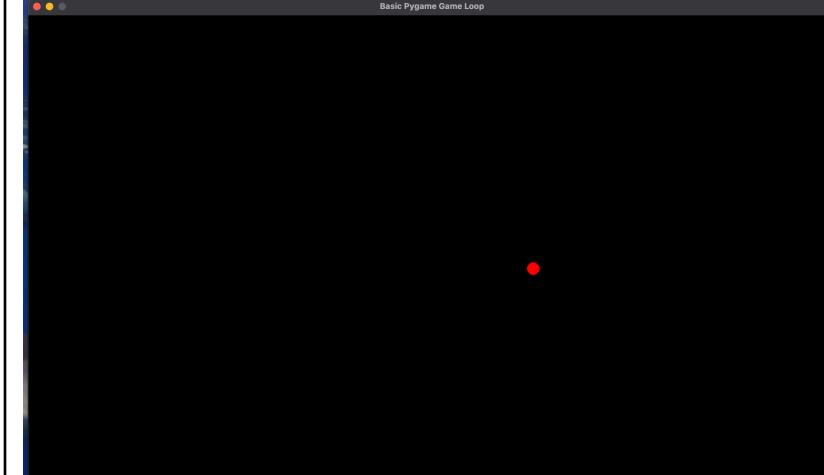
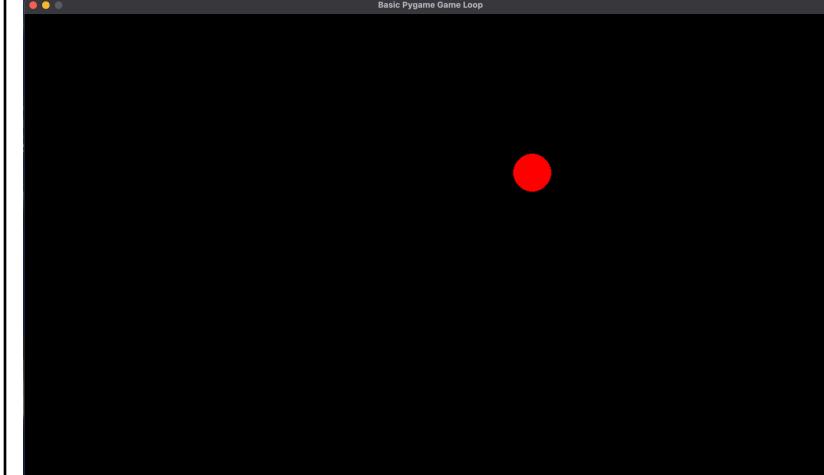
Input	Expected Output	Output
gravity = 0.3 friction = 0.9999 length = 500 cursor = bottom right	This will draw an arc toward the users cursor	
gravity = 0.3 friction = 0.9999 length = 500 cursor = top right	This will draw an arc toward the users cursor but instead the arc goes upward for a Little bit and then goes downward due to gravity which is being added onto the y velocity	

<pre>gravity = 0.8 friction = 0.9999 length = 500 cursor = bottom right</pre>	<p>This will draw a line which curves downward more drastically as the gravity is higher</p>	
<pre>gravity = 0.3 friction = 0.9999 length = 20 cursor = bottom right</pre>	<p>This will draw the same arc however it will be much shorter (and yes I haven't activated windows yet if you look in the corner of the screenshot)</p>	
<pre>gravity = 0.3 friction = 0.8 length = 500 cursor = bottom right</pre>	<p>This should draw a longer line as the length is set to 500 however since the friction is multiplied by the x velocity as the friction is set to a smaller number this causes the x velocity to become a lot smaller a lot faster which causes the line to only curve a slight amount before becoming a straight line moving down</p>	

<pre>gravity = 0.3 friction = 0.9 length = 40 cursor = bottom right</pre>	<p>There should be a slightly more curved line as the friction is 0.9 and it is headed toward the right side of the screen</p>	
<pre>gravity = 0.00005 friction = 0.9999 length =40 cursor = bottom right</pre>	<p>There will be medium length line as the length variable is set to 40 And there should be minimal curve downward as the gravity is set very low which is constantly added onto the y velocity and the line should also go out to the sides a lot as the friction is set very high</p>	

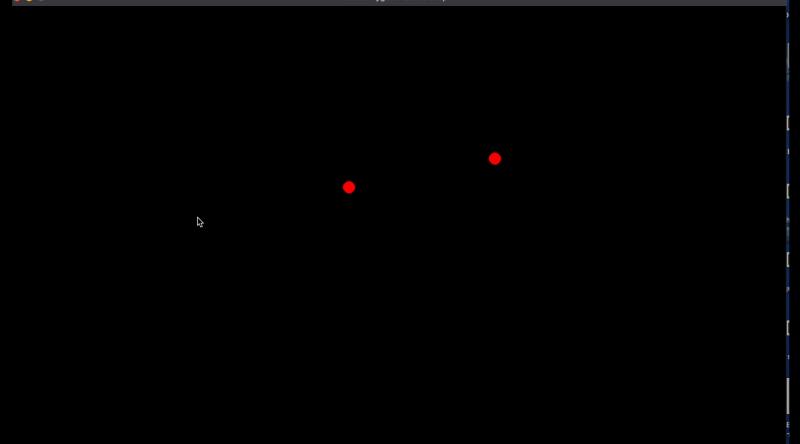
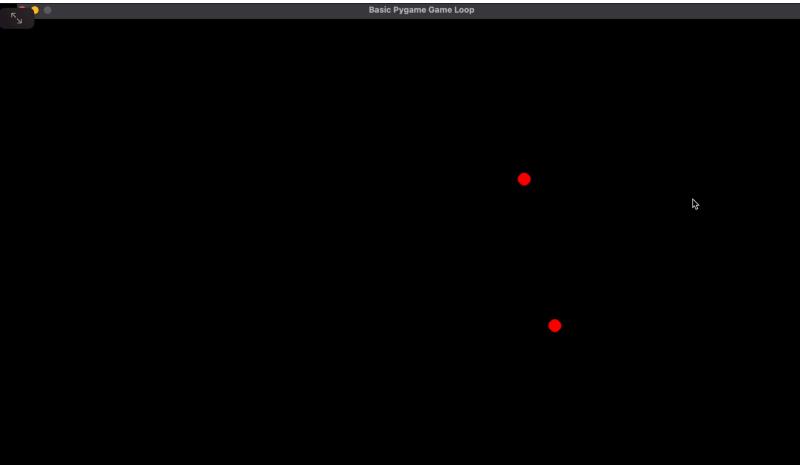
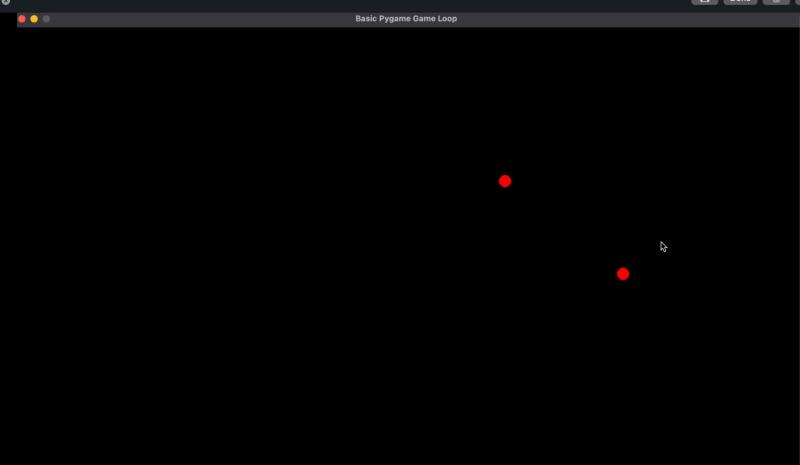
Ball(x, y)

This class draws a ball at the specified x and y locations.

Input	Expected Output	Output
X = 100 Y = 100 ball_radius = 10	The ball will be drawn at the coordinates (100,100)	
X = 800 Y = 400 ball_radius = 10	The ball will be drawn at the coordinates (800,400)	
X = 800 Y = 250 ball_radius = 30	The ball will be drawn at the coordinates (800,250) however it should be bigger since I increased the radius	

```
def shoot()
```

This method takes in the users mouse coordinates and calculates the direction to shoot the ball then the update method continuously moves the ball in the calculated direction

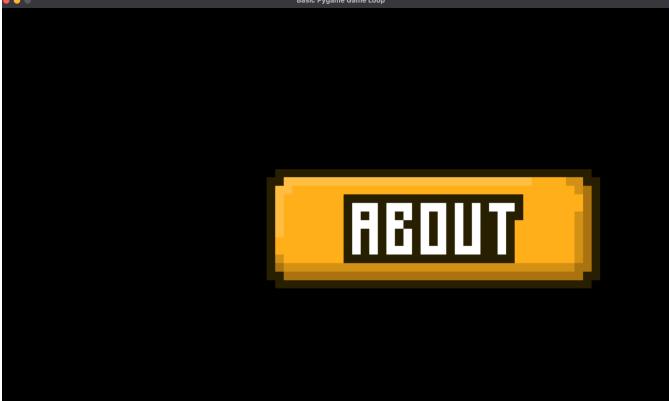
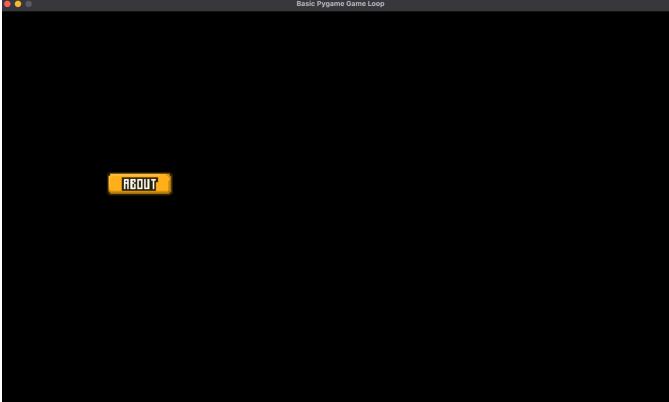
Input	Expected Output	Output
<i>*ball will begin at (800, 250)</i> pos = [300, 300]	The ball should move downward toward the left as the mouse coordinates are less than the balls position	
pos = [900, 500]	The ball should move downward toward the right as the mouses x coordinate is greater than the balls x coordinate	
pos = [1000, 400]	The ball should move downward toward the right as the mouses x coordinate is greater than the balls x coordinate however at a less of a steeper angle since the mouses x coordinate is a lot greater than the balls position	

GameStateManager(currentState)

This class controls which game state is being currently run which is essential for my program. The class has a method called `get_state` which is used to return the current state. Then the method `set_state(state)` can be called to change the state.

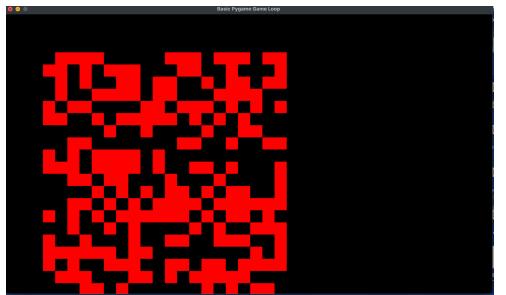
`Button(x, y, img, scale)`

This class takes in an image and then scales the image according to the scale number passed in, then this is blitted onto the screen using the draw method at the coordinates x and y

Input	Expected Output	Output
<code>x = 500 y = 300 img = help4.png Scale = 0.5</code>	A button is drawn at coordinates (500, 300) with a scale of 0.5	 A screenshot of a Pygame window titled "Basic Pygame Game Loop". It shows a single yellow button with a black border and white text that reads "ABOUT". The button is positioned in the center of the black background.
<code>x = 200 y = 300 img = help4.png Scale = 0.1</code>	A button is drawn at coordinates (200, 300) with a scale of 0.1 so it will be smaller than the first button	 A screenshot of a Pygame window titled "Basic Pygame Game Loop". It shows a very small yellow button with a black border and white text that reads "ABOUT". The button is positioned in the center of the black background.
<code>x = 400 y = 400 img = quit2.0.png Scale = 0.6</code>	A button is drawn at coordinates (400, 400) with a scale of 0.6 with a different image this time and should be slightly bigger than the first button since its scale is a larger number	 A screenshot of a Pygame window titled "Basic Pygame Game Loop". It shows a yellow button with a black border and white text that reads "QUIT". The button is positioned in the center of the black background.

Collision(self, new_offset_x, new_offset_y)

This creates a bunch of rectangles depending on if the number in the 2D array is 1134, then it puts the players centre in the middle of the screen if the players rectangle collides with any of the walls then it returns true. I have added a bit of code so that rectangles are drawn so that I can easily see them for module testing.

Input	Expected Output	Output
New_offset_x = 0 New_offset_y = 0 map = randomly generated map of size 20 by 20	It will draw the random map on the screen which will be a random map of size 20 by 20	
New_offset_x = 100 New_offset_y = 100 map = randomly generated map of size 20 by 20	The map will now have an offset of 100 and 100 applied onto the boxes moving the boxes 100 pixels to the right and down 100 pixels	
New_offset_x = 800 New_offset_y = 100 map = different generated map of size 20 by 20	This draws the map with x offset by 800 and y is offset by 100	 <p>First of all the grid I generated wasn't 20 by 20 however this shouldn't have much effect however it appears to not be offset by 800 correctly. This was because I forgot to change the values passed in:</p> 

`Level(display, map, monsters, playerHealth, fruits)`

This class creates the level scene along with the values passed in, the display is just the surface everything will be drawn on, the map is the layout of the pegs, the playeHealth is the amount of health the player has and fruits are the fruits that the player has.

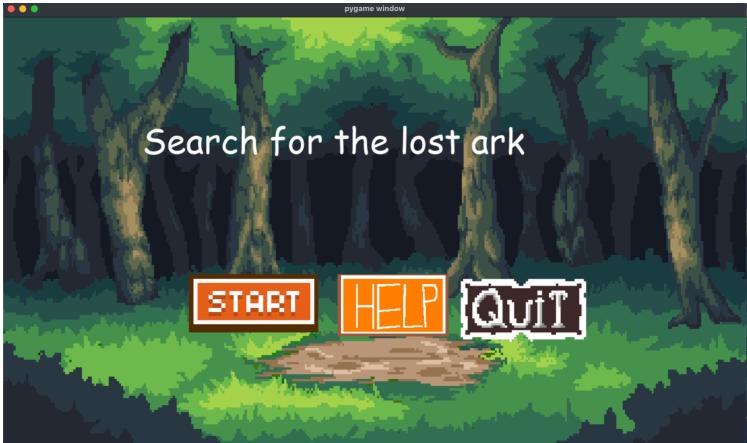
Input	Expected Output	Output
<pre>Map = randomly generated peg map Monsters = ['eye', 'slime', 'eye', 'eye'] playerHealth = 200 Fruits = ['empty', 'empty', 'empty']</pre>	The level should have an enemy in the first position which is an eye, the peg map should be a randomly generated layout, the players health should be 200 and the player should have no fruits	
<pre>Map = randomly generated peg map Monsters = ['orb', 'slime', 'eye', 'eye'] playerHealth = 100 Fruits = [potato, potato, potato]</pre>	The level should have an enemy in the first position which is an orb, the peg map should be a randomly generated layout, the players health should be 100 and the player should 3 potatoes in the fruit slots	
<pre>Map = randomly generated peg map Monsters = ['slime'] playerHealth = 300 Fruits = [apple, apple, apple]</pre>	The level should have an enemy in the first position which is a slime, the peg map should be a randomly generated layout, the players health should be 300 and the player should 3 apples in the fruit slots	
<pre>Map = randomly generated peg map Monsters = ['slime'] playerHealth = 0 Fruits = [apple, apple, apple]</pre>	The scene should change to the game over scene as the health is less than 0	

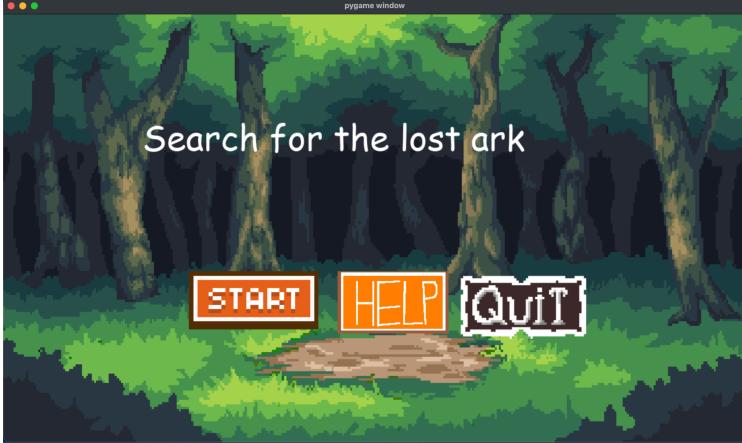
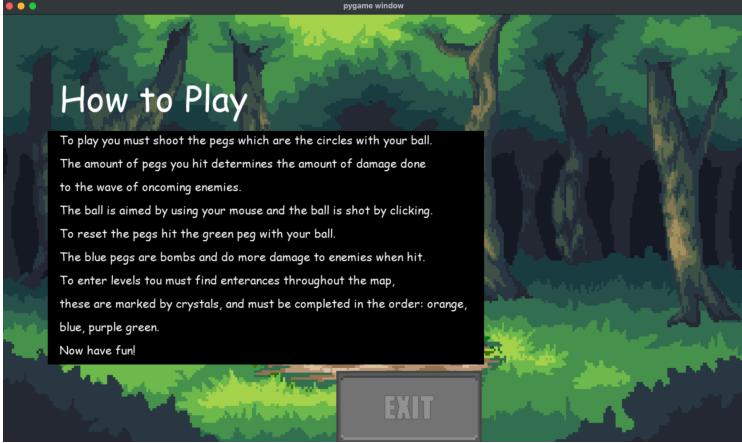
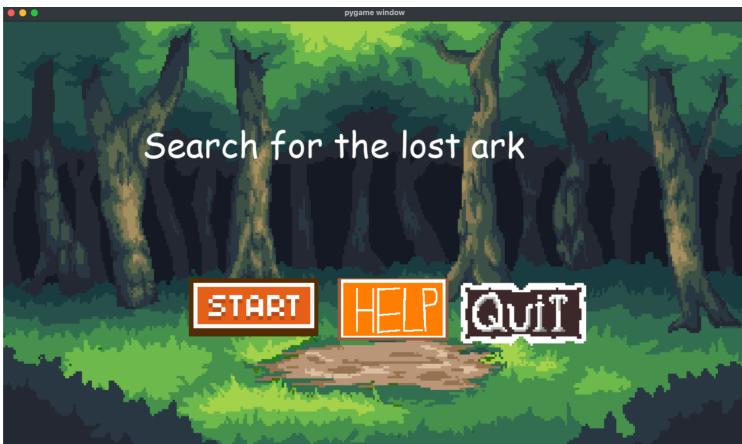
Program Testing

The overall purpose of this program is to have a title screen which provides the user with some basic options, the map in which the player can travel around in and then the levels which have custom layouts and enemies.

The first section of my program I will test is the start menu which has three main inputs which are three buttons that can be clicked.

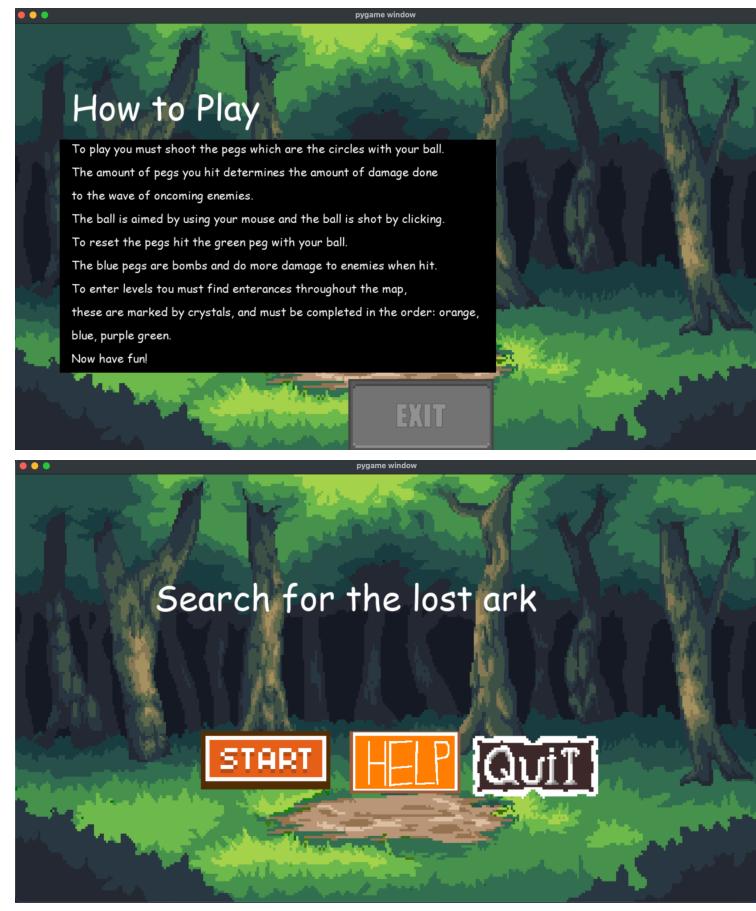
Start Screen

Input	Expected Output	Output
The user clicks the start button using the mouse and is taken to a screen of the map	The user clicks the start button with the mouse and is taken to a screen of the map	 

The user clicks the help button using right-click	The user is taken to the help screen	 
The user right clicks the quit button	The game is quit	 <p>The game then quits</p>

The user clicks the exit button in the menu

The user is taken back to the screen with the start, help and quit button



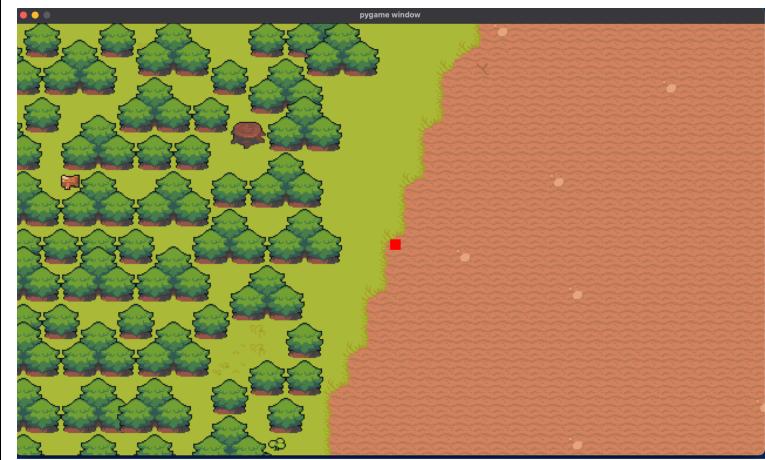
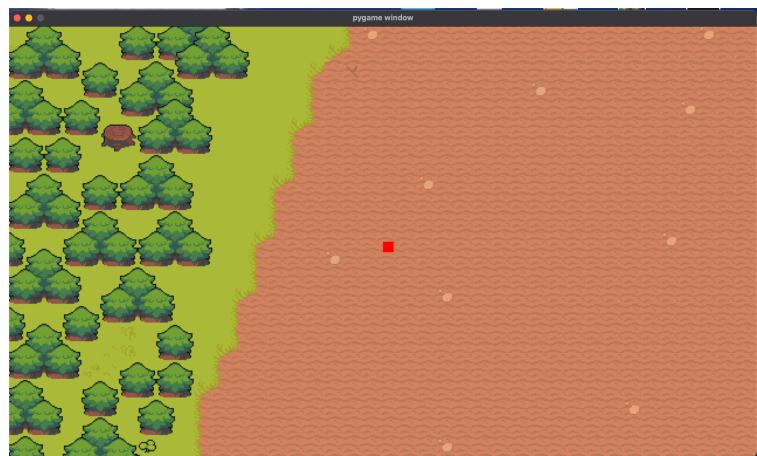
Map and Movement

Input	Expected Output	Output
The user presses w on the keyboard	The user moves up if there is no obstacle in the path and if there is an obstacle then the user doesn't move at all	<p>Case 1:No obstacle</p>  <p>Case 2:Obstacle</p>  <p>No movement happens</p>

The user presses a on the keyboard

The map is offset left if there is no obstacle in the path and if there is an obstacle then no offset is applied to the map and hitboxes

Case 1: No obstacle



Case 2: Obstacle

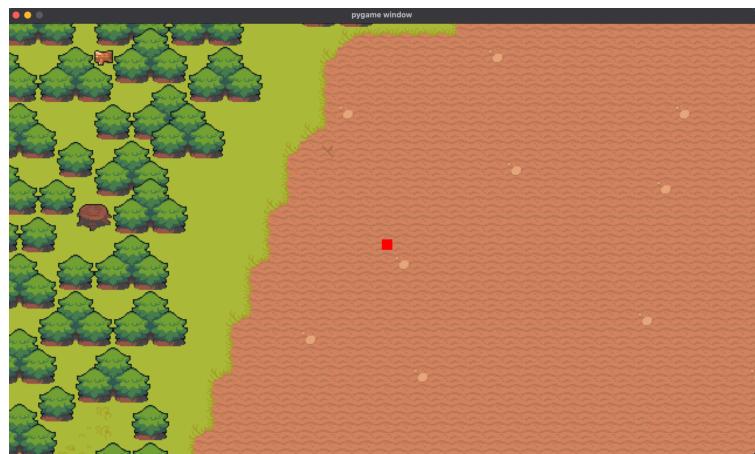


No movement happens

The user presses s on the keyboard

The user moves down if there is no obstacle in the path and if there is an obstacle then the user doesn't move at all

Case 1: No obstacle



Case 2: Obstacle



No movement happens

The user presses d on the keyboard

The user moves right if there is no obstacle in the path and if there is an obstacle then the user doesn't move at all

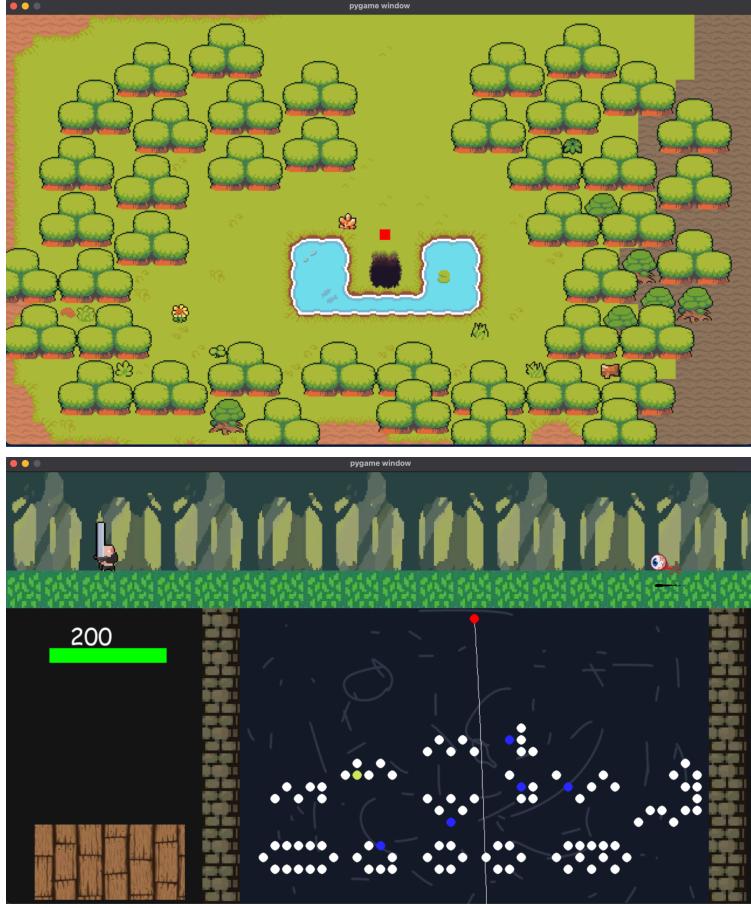
Case 1: No obstacle

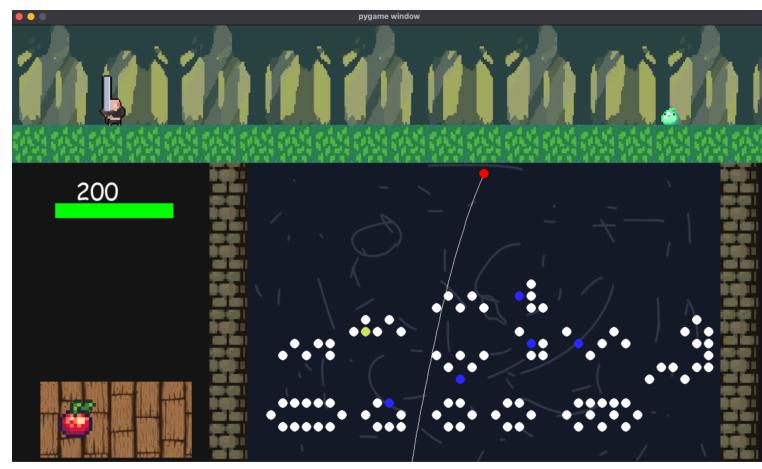


Case 2: Obstacle



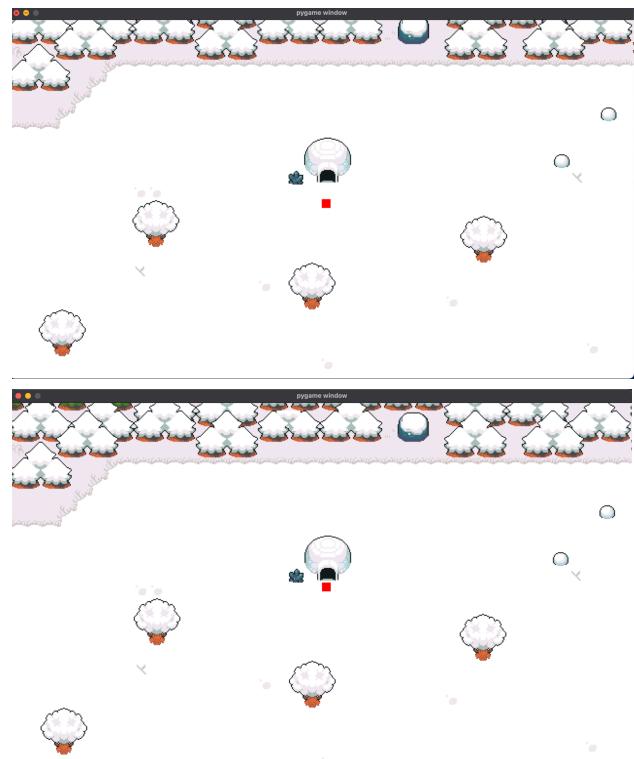
No movement happens

<p>Player collides with first levels rectangle and the collision is then checked with colliderect</p>	<p>The player enters the first level</p>	
<p>The player collides with the second levels rectangle and the collision is then checked with colliderect and the player has completed the first level</p>	<p>The player enters the second level</p>	



The player collides with the second levels rectangle and has not completed the first level

The player should not enter the second level as it is required that the previous level be completed so nothing happens



The player collides with the rock obstacles within the map

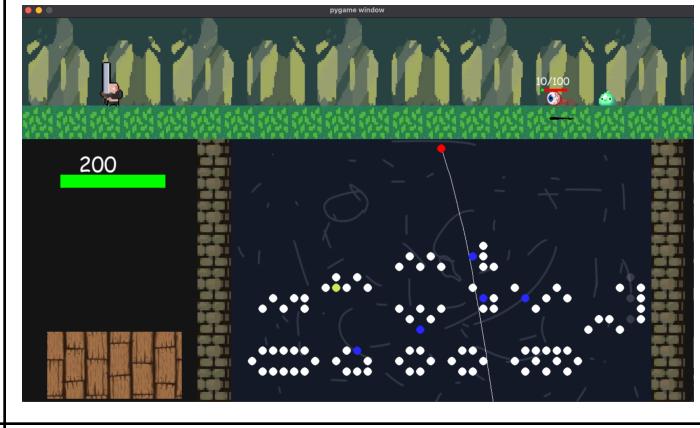
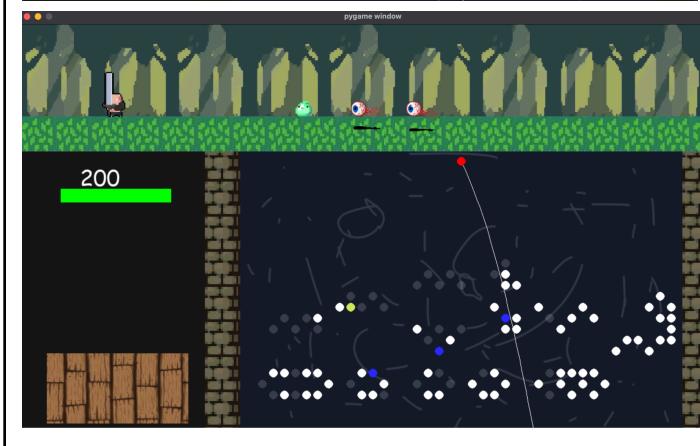
No offset is applied to the map

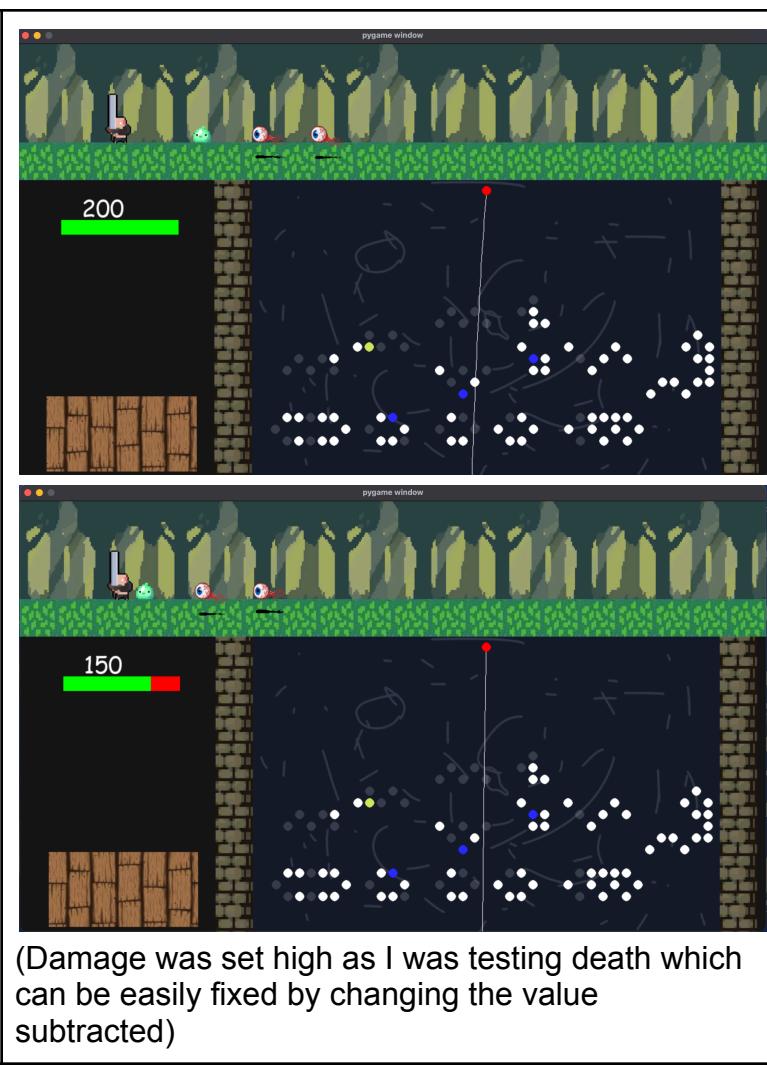
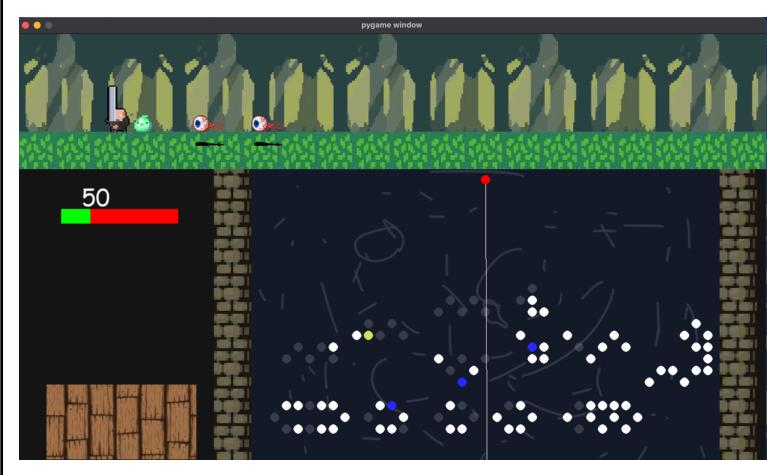


No offset is applied to the map

Level and Fruit Selection

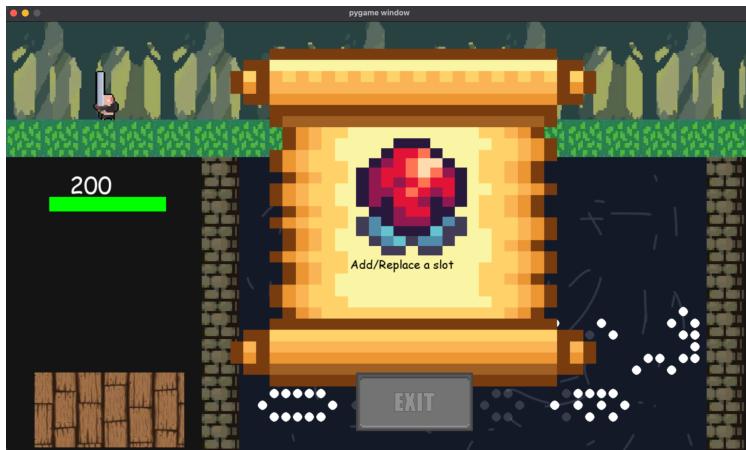
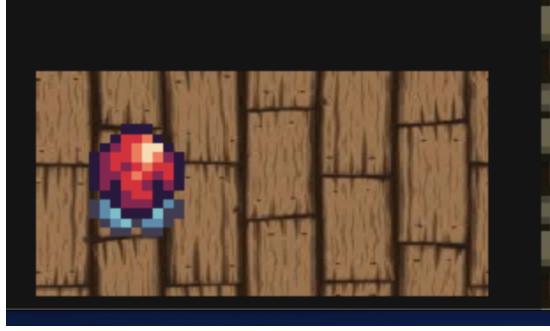
Input	Expected Output	Output
Player right clicks within the play area which is <mouse x pos 330	The ball is shot in the direction of the ball	Get screenshot on windows
The player clicks when the mouse pos is less than 330	The ball is not shot	 Ball is not shot
The player aims above the ball and shoots	The ball won't follow the trajectory and instead bounce off the ceiling	 (Was difficult to get screenshot of ball bouncing off ceiling)

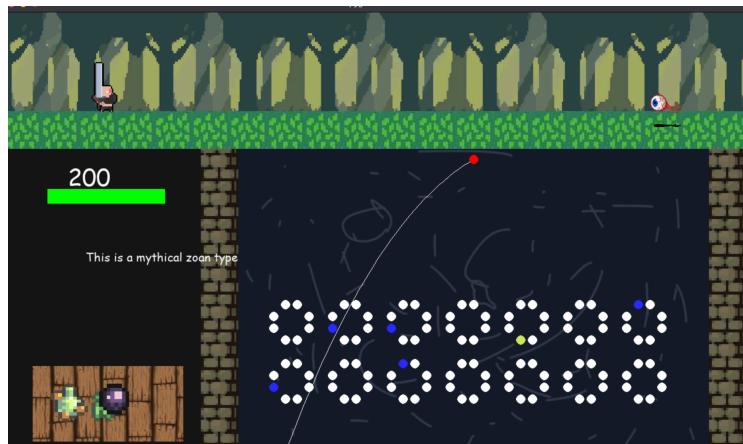
Ball falls of screen and its y position is ≤ 1000	The enemies are moved one position forward	 
The enemies health is less than or equal to 0	The enemy is removed	 

<p>An enemy is in the first position and the balls y-position is less than or equal to 1000</p>	<p>The players health is subtracted by 5</p>	 <p>The top screenshot shows the player's health at 200. There is one enemy in the first position and several balls in the air. The bottom screenshot shows the player's health at 150, indicating that 5 health was subtracted.</p> <p>(Damage was set high as I was testing death which can be easily fixed by changing the value subtracted)</p>
<p>The players health is equal to 0</p>	<p>The gameStateManager is changed to over which runs the Death class</p>	 <p>The screenshot shows the player's health at 50. The health bar is partially red, indicating low health. The player is positioned near some trees and a stone wall.</p>



The monster amount is 0	The if statement is entered which draws the scroll and a random fruit on the screen	The image contains two screenshots of a game. The top screenshot shows a character standing in a room with a scroll on the floor and a red fruit nearby. The bottom screenshot shows the character interacting with the scroll, which now displays a large, colorful fruit icon. A text overlay says 'Add/Replace a slot'. Both screenshots show a green health bar at the bottom left and a map at the bottom right.

The player clicks one of the grey buttons	All of the buttons disappear	
The player clicks one of the grey buttons	The string of the fruit replaces the fruit list in the corresponding position based on which button was clicked e.g. if the first button was clicked then the fruit replaces the first position	
The player clicks a button which corresponds to a position that already contains a fruit	The fruit is replaced by the new fruit	 

The player clicks a fruit	Text appears according to which fruit was clicked	<p>First fruit is clicked:</p>  <p>Second fruit is clicked:</p> 
The player presses the quit button on the game over screen	The game quits	 <p>The game quits</p>

System Testing

For the majority of this project I have been working on a macbook and coding on it, to make my project folder more organised I created a folder called images for all my image assets and a folder called sounds for my sound assets. When I began testing the program in windows I got the following errors:

```
[Running] python -u "c:\Users\Nirvan\Downloads\Software Project copy-20240702T075944Z-001\Software Project copy\main.py"
pygame 2.5.2 (SDL 2.28.3, Python 3.10.11)
Hello from the pygame community. https://www.pygame.org/contribute.html
Traceback (most recent call last):
  File "c:\Users\Nirvan\Downloads\Software Project copy-20240702T075944Z-001\Software Project copy\main.py", line 245, in <module>
    start_img = pygame.image.load('./images/start.png').convert_alpha()
FileNotFoundError: No file './images/start.png' found in working directory 'c:\Users\Nirvan\Downloads\Software Project copy-20240702T075944Z-001'.

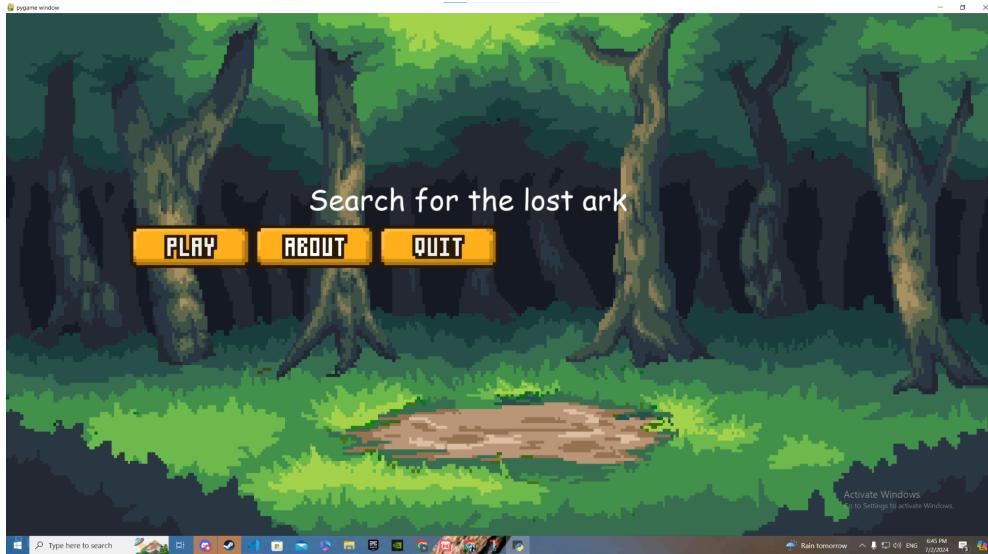
  File "c:\Users\Nirvan\Downloads\Software Project copy-20240702T075944Z-001\Software Project copy\main.py", line 125, in __init__
    self.lvl1 = Level(self.screen, MAPS[0], ['eye', 'slime', 'eye', 'eye'], 200, ['empty', 'empty', 'empty'])
  File "c:\Users\Nirvan\Downloads\Software Project copy-20240702T075944Z-001\Software Project copy\level.py", line 113, in __init__
    self.ball = Ball(800, 250)
  File "c:\Users\Nirvan\Downloads\Software Project copy-20240702T075944Z-001\Software Project copy\level.py", line 973, in collide
    self.collide = pygame.mixer.Sound('./sounds/collide.wav')
FileNotFoundError: No file './sounds/collide.wav' found in working directory 'c:\Users\Nirvan\Downloads\Software Project co
```

This was likely due to the different access paths of files between windows and apple, this was a major issue as I was unable to run the program at all. To fix this issue I had to move all image and sound assets into the main file and comb through the code to remove the ./images/ file access so that it was only searching for images in the main program file. Although this was very annoying and inconvenient It was still better than being unable to run it on windows.

Now upon retesting I tried to do the try and except method which you recommended in class however I only did half and went to test if it was working. To my surprise the program ran just fine, I had no issues with getting images from the folder when I didn't use the os.path.join. This is very weird as I was having issues beforehand. Since there apparently is no need for the try and except section of code I will remove them however if the code crashes when you do attempt to run my program that will be very weird and annoying. ALL OF THIS IS VERY SUSPICIOUS!!!

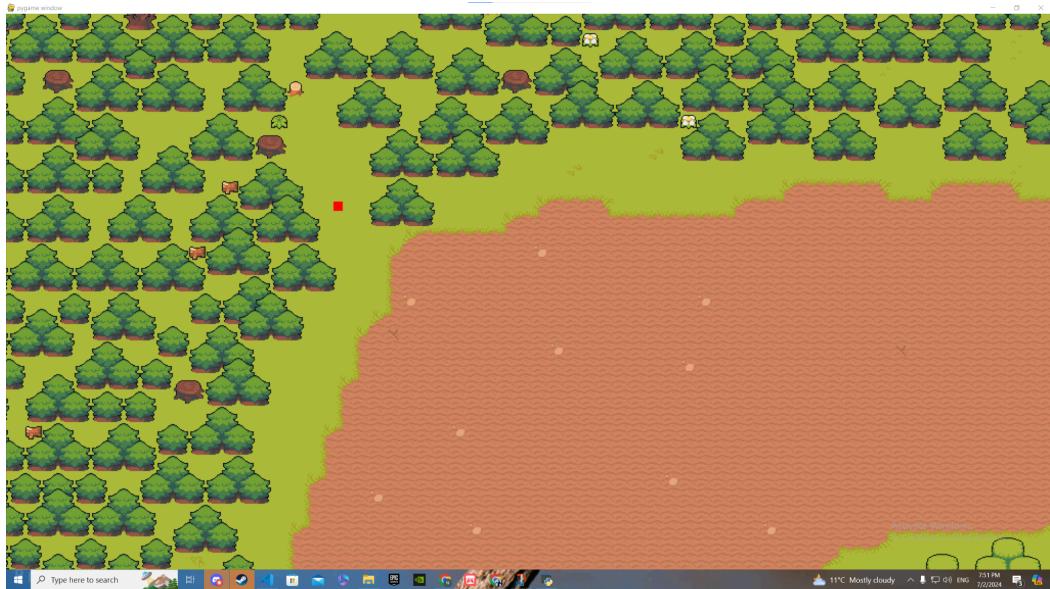
Resolution

Next while running the program on my larger monitor(2048 x 1089 pixels) I checked how screen size would affect the game's UI components and performance. I first checked the start screen when fullscreened.



This was because the buttons were set to 60% of the height on the initial screen which is 1280 x 736 pixels, so when the screen is a lot larger it is still set to 60% of the original screen size which is a lot less than the new screen dimensions causing the buttons to appear much higher up on the screen.

The next thing I checked was the compatibility of the map with the larger screen size. The first thing I observed was that the player appeared to be in the top right corner as it used the original screen size of 1280 x 736 to draw the player in the middle of the screen. The second issue I observed was a significant decrease in performance when it was set to the larger screen size compared to the 1280 x 736 screen size. This however was strange as regardless of screen size all the rectangles for the hitboxes should be drawn so screen size doesn't have any effect on the rectangles of the hitboxes. The only other possible reason for the issue was potentially that having a larger screen size meant more of the map image would have to be drawn on the screen causing the game to appear like it is lagging when the offset is being applied.



The game was also very laggy when moving the player around (screenshots would not have been able to assist in showing this).

In order to fix this issue I decided to remove the portion of code which was allowing my game window to be resizable. Although this would reduce the programs ability to meet the users specific requirements of screen size it would prevent users from experiencing lag when moving around the map and also the UI from being in the wrong place which is an overall better compromise as there is not enough time to figure out the reasons for why the map lagged when full screened.

Performance

The first test of performance was on my windows pc with the following specs:

AMD Ryzen 7 3700X CPU 16GB RAM RTX 3070 ti

The performance worked very well as this was the device I designed a decent portion of my program on. There were a few moments when the ball appeared to be a little bit faster which was likely due to the frame rate not being capped. To fix this I added in a clock tick every 60 seconds to prevent the game from running faster when there is a higher frame rate. Overall the program was able to run very well and had no major issues.

Apple Macbook air m3 cpu 16gb ram

The game performed very well on this computer as this was the main device used when designing the game. The movement around the map was smooth and worked very well. There used to be an issue of a ripple effect caused by screen tearing however this had stopped happening when it came time for system testing. The level performed very well with the ball moving smoothly and no significant performance issues.

Apple Macbook Pro 15-INCH Touch Bar

For this performance test, although it sounds like a very good laptop which should be able to handle my program fine, it is actually a very old computer which is probably damaged from being dropped or something along the lines of that as it is slow in performing any task. This should be a good benchmark on how well my program will be able to run on devices.

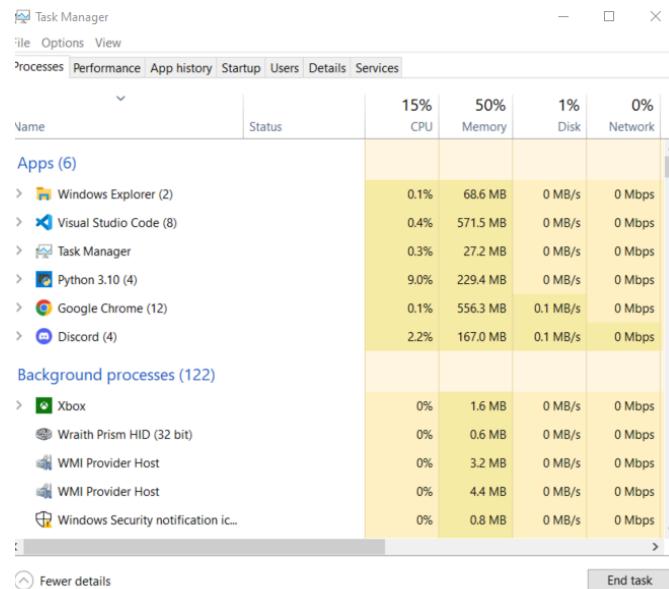
The game ran a decent amount slower than usual when moving around the map and especially within the level where the ball moved a lot slower however it was still playable and functional. The ball should in theory move the same as normal although it appeared a lot slower. This means that on computers which have a lot worse specs the program may have issues with performance. This is mainly due to the drawing of the pegs and the ball's movement.

To run this program it is recommended to have a new model of macbook with at least an m2 chip or a AMD Ryzen 7 3700X CPU or greater as this was what the program was designed in and for. It is also recommended that the computer has at least 8gb of ram as the program didn't require much memory.

These were the memory usage on both macbook and windows:

Spotify Helper (Renderer)	472.6 MB	20	212	874	nirvanpulakhan
Google Chrome Helper (Renderer)	409.9 MB	19	530	50212	nirvanpulakhan
Google Chrome Helper (Render)	392.6 MB	21	285	39198	nirvanpulakhan
Python	320.2 MB	13	452	89025	nirvanpulakhan
Google Chrome Helper (Render)	302.0 MB	21	358	76393	nirvanpulakhan
Code Helper (GPU)	283.6 MB	10	150	1103	nirvanpulakhan
Google Chrome Helper (Render)	274.2 MB	23	382	34185	nirvanpulakhan
Notes (Not Responding)	270.7 MB	258	5,440	53548	nirvanpulakhan
Code Helper (Plugin)	257.6 MB	12	68	86688	nirvanpulakhan

320mb of memory usage on apple

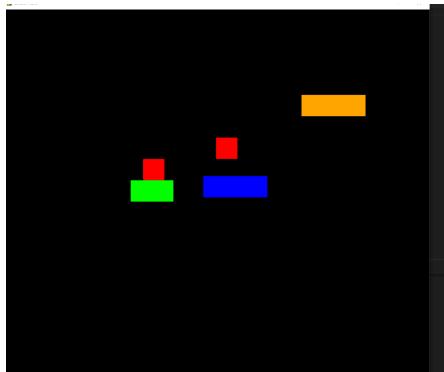


229mb of memory on windows

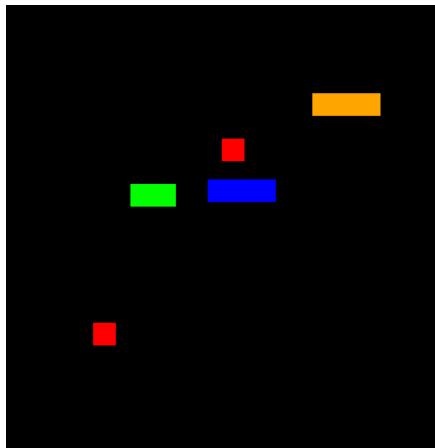
Logbook

Entry 1:

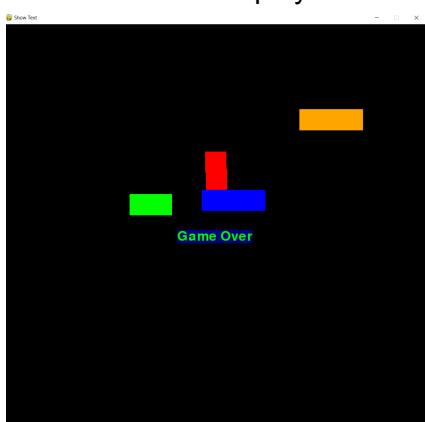
I started testing how pygame worked and creating objects:



Then I constantly subtracted a number from one of the blocks to simulate gravity:



Then another to display text:

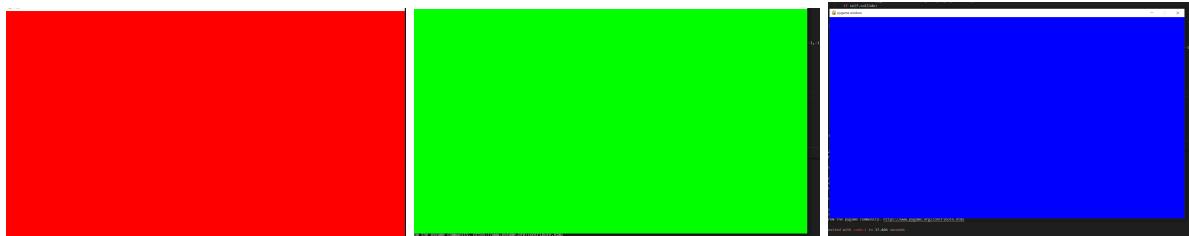


Entry 2:

After I finished messing around with pygame, I first started my program by following along with this tutorial <https://www.youtube.com/watch?v=r0ixaTQxsUI>, It was very helpful in helping me setup the start of my program as I am still very inexperienced at pygame, I was able to set up an initial class which contained the main game loop and then in the main game loop it called the game state manager, this game state manager would then in turn run the current class that it was set to. To set up the classes I had to first create an instance of the class and set it to a variable name. I initially made three classes: **Start**, **Map**, and **Level**. I created a variable for each, then created a dictionary with a different string corresponding to each different class instance. Through this the **gameStateManager** could change which class it was running by changing the string it was using to access the dictionary to a different string within the dictionary.

Finally I could have three screens depending on which string I initially put in the game state manager:

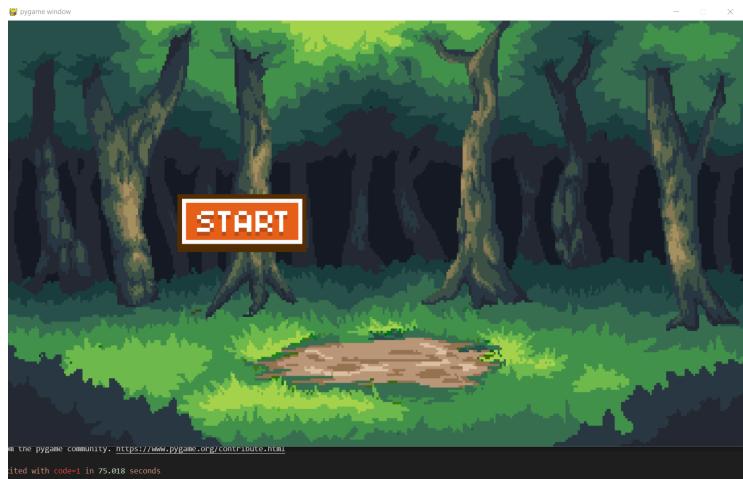
```
self.gameStateManager = GameStateManager('start')
self.start = Start(self.screen, self.gameStateManager)
self.level = Level(self.screen, self.gameStateManager)
self.map = Map(self.screen, self.gameStateManager)
```



For now I have just filled the screen using the `.fill(colour)` pygame function with different colours to be able to differentiate between the different scenes within my game. Each of these classes are passed in the **gameStateManager** so they are able to change what state is currently active this is very helpful as now actions within each class such as colliding with objects can cause the state to be changed

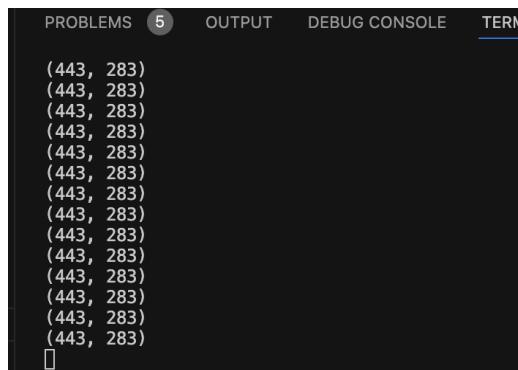
Entry 3:

Next I created another class called **Button**, however instead of creating an instance and adding it to the dictionary for the **gameStateManager** I created it so that I can create an instance of my button class within any of the other classes to create buttons. In the init method of my button class I passed in the x and y position of where I wanted to draw the button, the image itself, and the scale. Then I created a new variable called **self.image** which created an image with the dimensions of the image passed in multiplied by the scale number, then I created a draw method which would blit **self.image** onto the screen at the x and y coordinates passed in:



This was drawn in the **Start** class along with changing the background image to have an initial start screen when the game is first loaded.

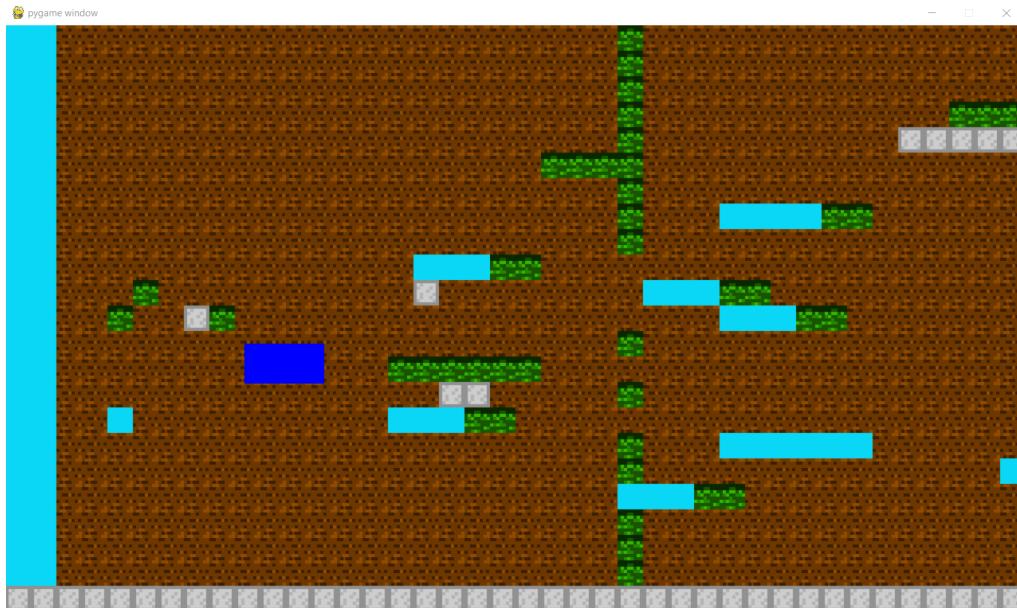
Then through the pygame function **pygame.mouse.get_pos()** I was able to find where the mouse coordinates were on the screen and if the mouse was within the buttons area and the user clicked their mouse I changed the scene to the map scene.

A screenshot of a terminal window with a dark background. At the top, there are tabs labeled "PROBLEMS", "5", "OUTPUT", "DEBUG CONSOLE", and "TERM", with "TERM" being the active tab. Below the tabs, there is a list of coordinates: (443, 283), (443, 283), (443, 283), (443, 283), (443, 283), (443, 283), (443, 283), (443, 283), (443, 283), (443, 283), (443, 283), (443, 283), (443, 283), (443, 283), (443, 283), (443, 283), (443, 283). A small square cursor icon is visible at the bottom left of the terminal window.

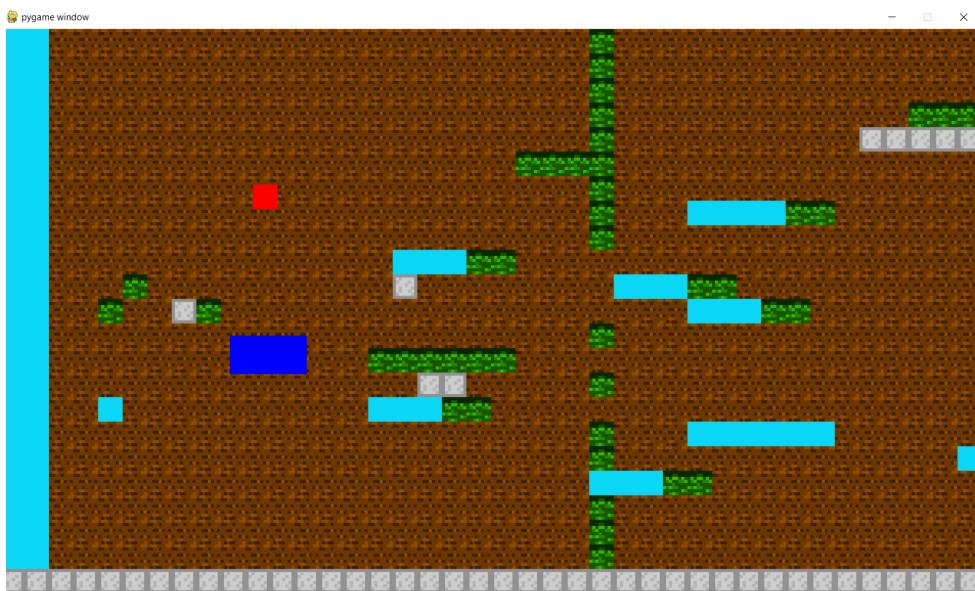
However for some reason the clicking didn't work everywhere in the button and you had to click in a specific location even though the range of mouse coordinates should have been large enough for the button(*if mouse.pos > button.x and mouse.pos < button.x + width*)

Entry 4:

I was starting on the map. I created a 2D matrix which had a -1 to represent an empty space and then a 0 to represent a grass block and a 1 to represent a dirt block. Then I did a for loop and if the value at that position was a 1 or 0 I would multiply its position in the list by 64 as this was the tile size and then I would blit the image on the screen. I got this:



Then I drew a rectangle on the screen and when I pressed wasd on the keyboard I would change the squares position accordingly:



Then next when the square hit the blue rectangle I could check if it collide using `colliderect` in pygame. Then if it did collide I would change the scene to the level using the `gameStateManager`

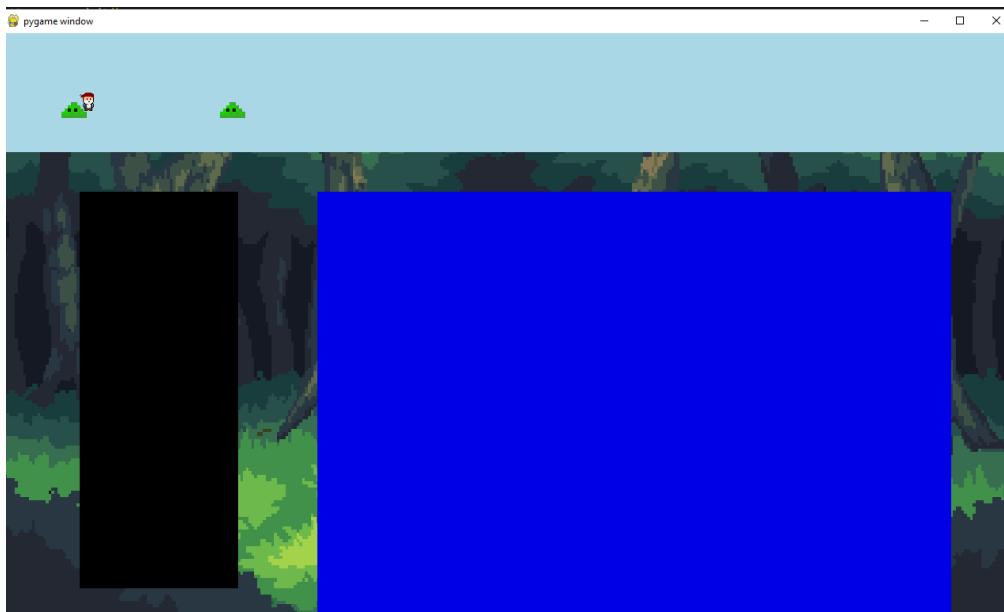
Entry 5:

I created the level by first blitting the forest image onto the background and creating three boxes one for where the players will be one for where upgrades and ammo will go and one for where all the pegs and balls will go and then I blitted a monster and the player onto the screen:



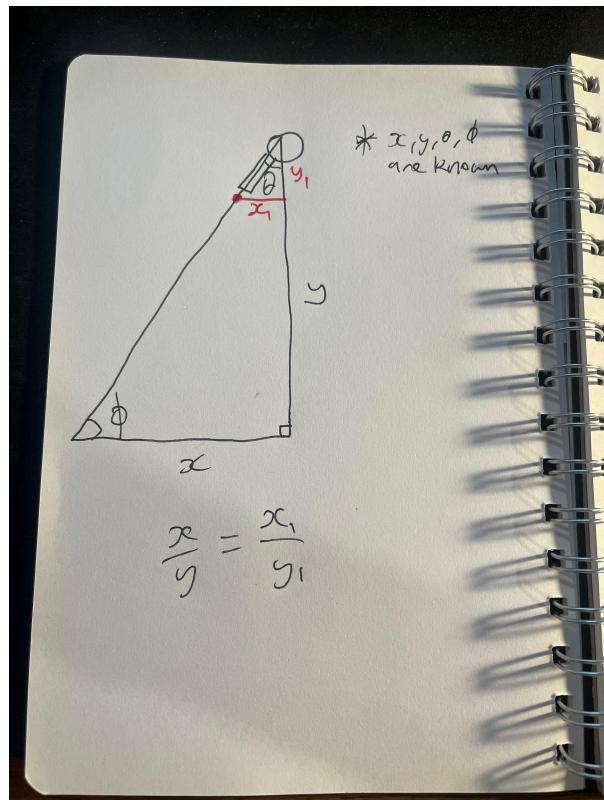
Next I created a list of possible positions for the enemies, this was so I was able to move the enemies to different positions if I wanted. Within this list the first position signified the x location and then the second signified what type of monster it would be e.g number 3 represents a slime.

I was then able to change which positions contained enemies or not:

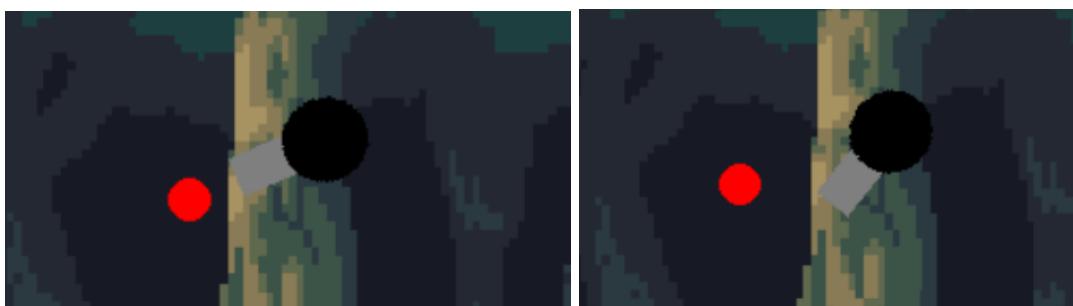


Entry 6:

After I created the cannon I wanted to add in a way to shoot a ball from the cannon. I first wanted the ball to begin by having the ball begin a little bit in front of the muzzle, for this I thought I could use similar triangles to find where the ball should be in front of the cannon as I already calculated the angle that the cannon has been rotated.

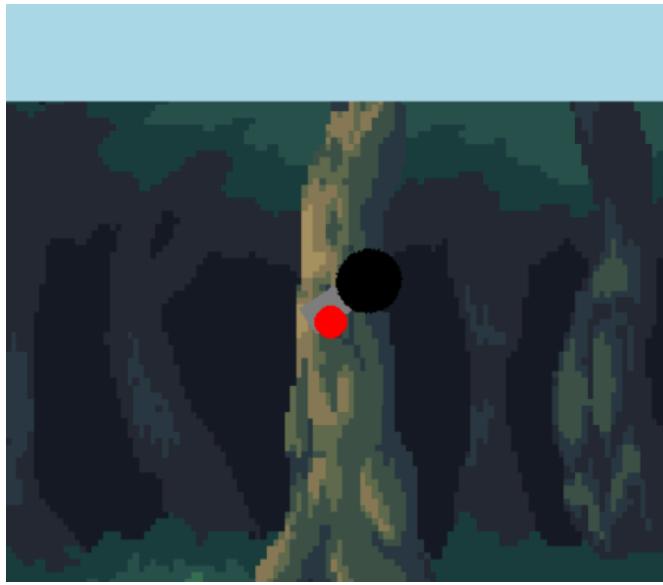


To find the red dot where I want the ball to initially be drawn I can use similar triangles, as I don't know either x_1 or y_1 . I set y_1 as the length of the cannon and then rearrange the equation to find it in terms of x_1 . Then I created a class to draw a ball at the location at the red dot.



As can be seen in the image on the left this does work in being able to draw a ball in front of the cannon, however in the image on the right it does not work for all positions so this has to be reworked. Maybe my maths was wrong.

I then made the ball be placed onto the screen when the player clicks their mouse but as seen above it is not placed correctly:



Then when I click the mouse multiple times:



Yes, I should definitely make it so that only one ball is created at a time.

//2nd approach

I created a new class for the ball that would be shot and have it begin at the same position every time behind the cannon.(I will have to rework cannon rotation to rotate the image around 1 fixed point). This would first create a surface for the ball as it will be easier to have different looking ammo as a surface can have lines drawn on it or images copied onto the surface using blit in the pygame library. However there is a black background on the surface as the surface created is a rectangle. As seen on image below:



I tried to make it transparent using the pygame function `.set_colourkey((0,0,0))` as I have previously done(shown in image below):

```
super().__init__()  
self.imageOG = pygame.Surface((ball_radius*2, ball_rad  
self.image = self.imageOG.set_colorkey(0, 0, 0)  
pygame.draw.circle(self.image, (255, 0, 0), (ball_radius  
self.rect = self.image.get_rect(center=(x, y))
```

But this would not work and give me the following error:

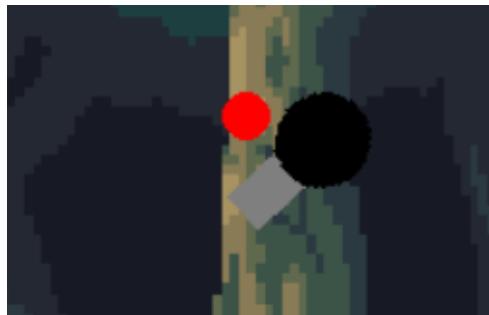
```
File "c:\Users\Nirvan\OneDrive\Desktop\Organisation\level.py", line 247, in __i  
    self.image = self.imageOG.set_colorkey(0, 0, 0)  
TypeError: function takes at most 2 arguments (3 given)
```

(TBH idk what this even means)

I've tried `pygame.Surface.set_alpha()`, but this makes the whole surface transparent as seen in the picture below. I have no idea why this is so annoying to fix.



I finally found a way to fix what I wanted to fix using `pygame.srccalpha`, this is seen in the image below.



Then with this new class for the ball I could create a new instance of it once the first ball had been shot, this would allow me keep creating new balls as I wanted.

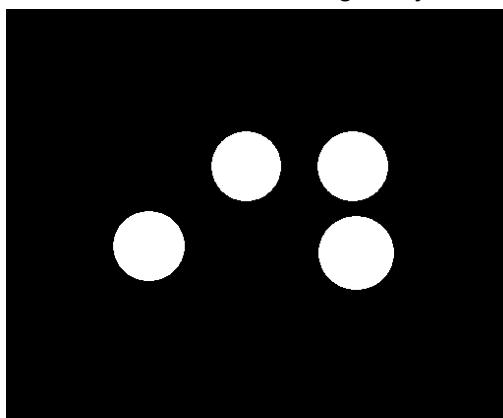
Entry 7:

I created a new class for my pegs. This would be a similar concept to the ball class I just created, I would have my peg class create a single circle to represent a peg. Then I would have a 2D array to signify where the pegs should be placed. This would allow me to create different layouts depending on how I fill the 2D array.

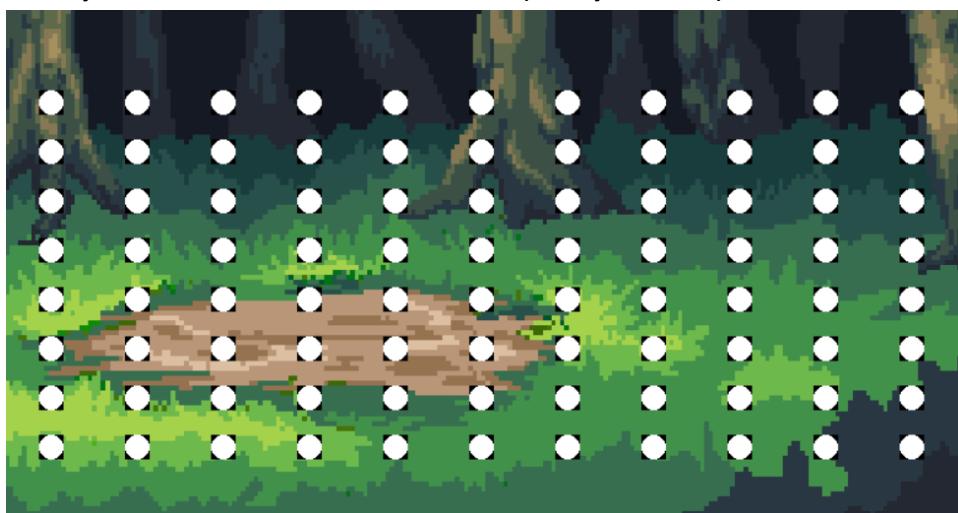
For example the 2D array:

```
[['0','1','1'],
 ['1','0','1']]  
#The one representing a peg and the 0 representing an empty space
```

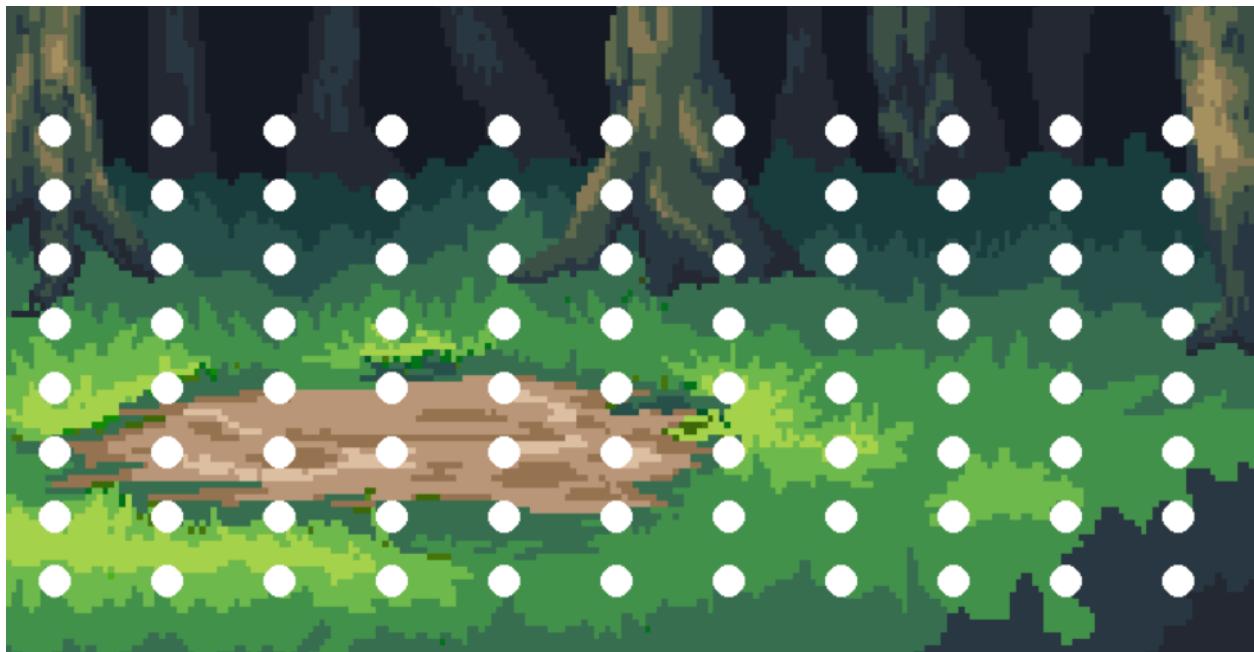
Should create the following array:



Then after creating the peg class I made a for loop that placed a peg and then incremented the x and y coordinates so that it would completely fill the space:



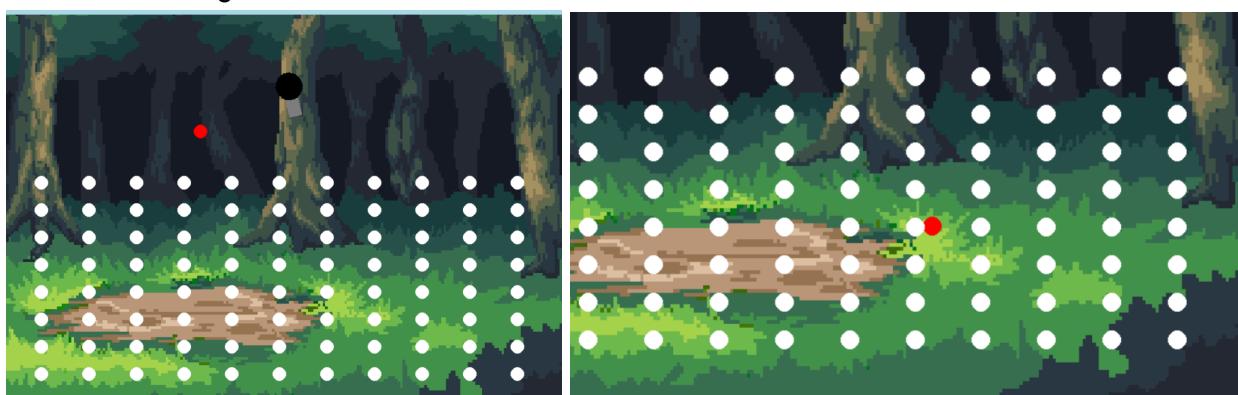
And then after adding the `pygame.srccalpha` I had a transparent background:



Entry 8:

Next I wanted the user to be able to shoot the ball based on the user's coordinates. First I created a ball which moved based on a speed which was added on and a direction. I used this website: <https://www.geeksforgeeks.org/stimulate-bouncing-game-using-pygame/> it was very helpful and I added in the ball colliding with the sides of the wall which was a specific x and y coordinates. This was done by checking the ball's position and if the ball collided with the x or y coordinates which specified the walls then the direction the ball was going was set to a negative value so that the ball bounced off the wall.

The ball bouncing in a different direction:

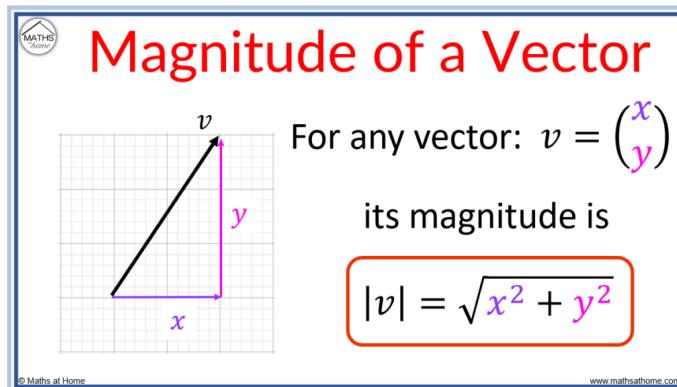


Next I added a variable called friction which was constantly multiplied with the x-velocity to slow it down, this was able to simulate friction.

Next I wanted to shoot the ball in the direction whichever the mouse was pointing in and then create the velocity that the ball should travel in.

To begin first I had to find the direction I did this by creating a list called direction which would supply me with the x and y direction. I subtracted the mouses x pos by 800 as that is where the ball begins and the mouses y pos by 250. The website that helped me was stack overflow:

<https://stackoverflow.com/questions/29059368/add-velocity-to-a-sprite-in-the-direction-of-the-mouse-when-i-push-a-button>. This showed that you had to find the magnitude by square rooting the x direction squared added to the y direction squared.



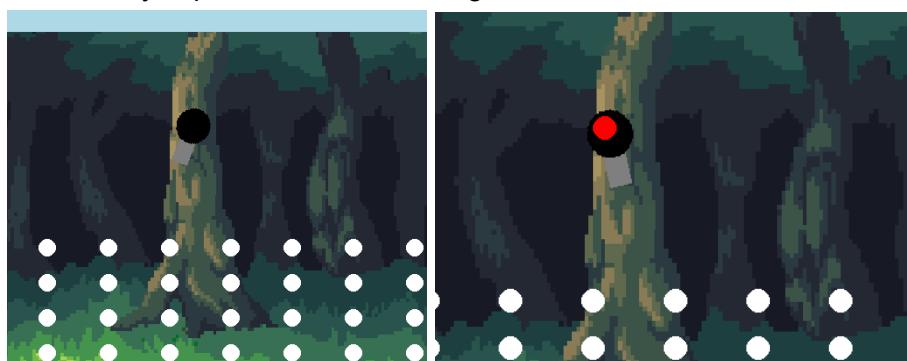
Next I had to divide the x and y direction by the magnitude to get the normalised vector. The importance of normalising a vector is so that its direction is maintained but its length is always 1(I think I don't do physics and have no idea). Then to get the velocity I set the x velocity to the x direction * 10 to give it initial velocity and did the same for the y velocity.

Now I added this all into a method called shoot and had a variable called **self.isShot** set to true when this method was called in my ball class. When the player right clicks the balls shoot method is called, then in the update method of my ball class if the **self.isShot** variable is set to true then it will apply a downward gravity on my ball which would start moving the ball and then the update method would move the ball to the according to the velocity using the **move_ip(x,y)** pygame function.

This all worked relatively well for me to be able to shoot the ball in the correct direction(no screenshot as it was hard to show my cursor pointing in the direction the ball was shot).

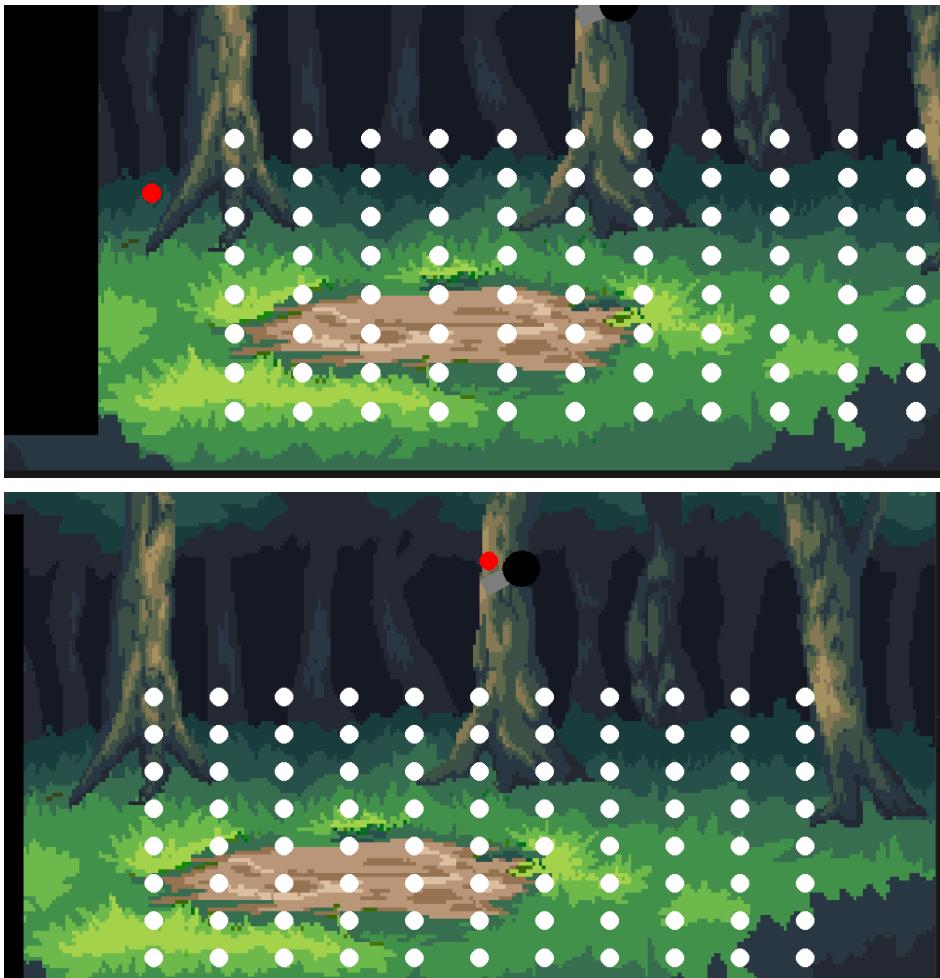
Next I needed the ball to be able to shoot multiple times so I made it when the key Z is pressed a new instance of my ball class is created.

When z key is pressed after shooting:



However there was an issue because the new ball could no longer be shot, this is likely due the old ball still existing so it must be removed.

The ball was now removed when z was pressed however it still wouldn't shoot:



This was a simple fix which I got working, I had used a variable called **self.notShooting**, when this was set to true the balls shoot method could be called but when the if loop was entered **self.notShooting** was set to false, when I pressed z I had to simply set **self.notShooting** back to true and this fixed the issue.

Entry 9:

I am now going to work on my monster creation and movement. I first started by creating a new class called Monster as this should be a better approach and I will be able to create a new instance of my class when I want to. This monster class was initially passed in a string to determine what type of monster along with the monster's health and then the class had a draw function which drew the monster onto the screen.

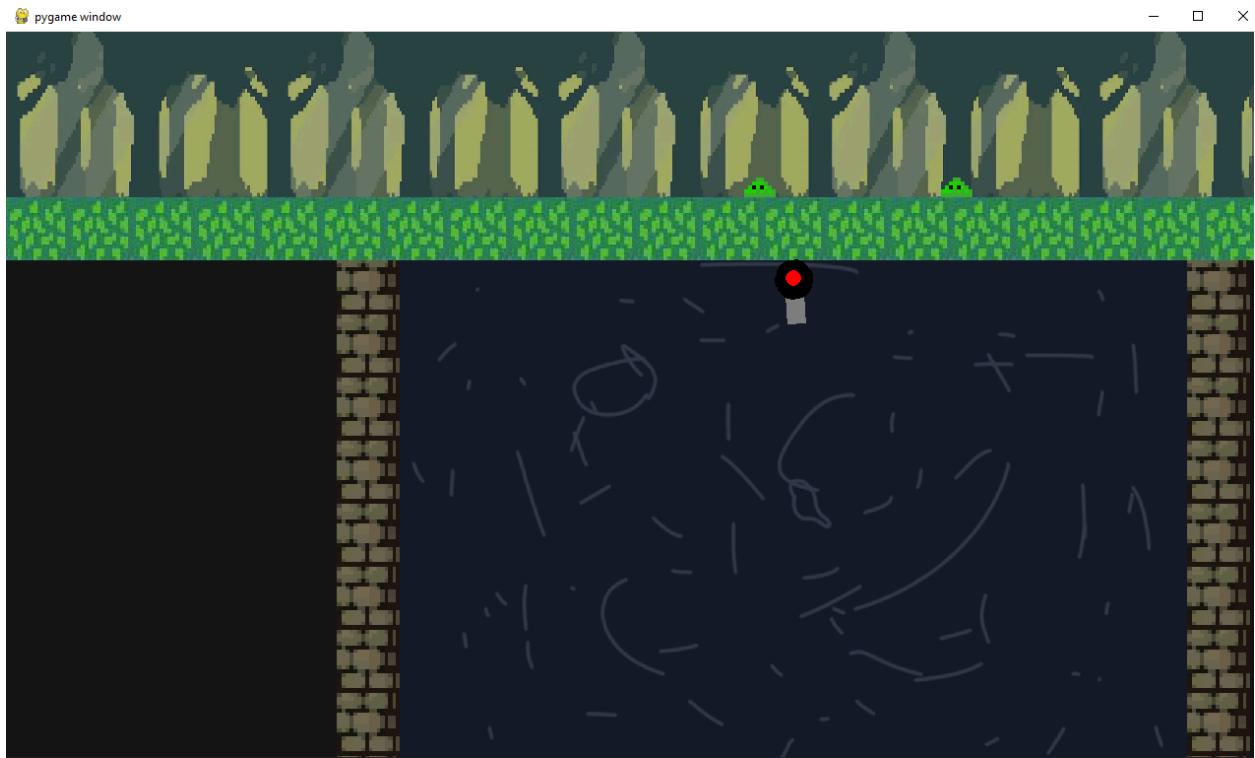
This is me testing by creating two enemies one which is a slime and the other being an orb, these are created at different x location:



Then next I created a series of if statements so that the monsters x location would be subtracted by a certain integer amount so that the enemy moves across the screen. These if statements ensure that the movement only happens after the player has shot the ball.

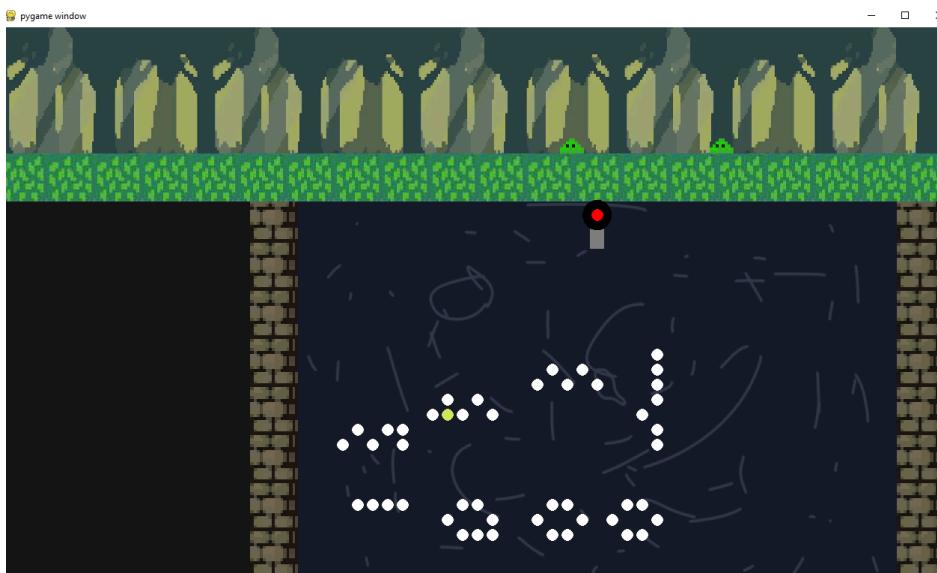
Entry 10:

Next I redesigned the level class so that it looked a lot nicer:



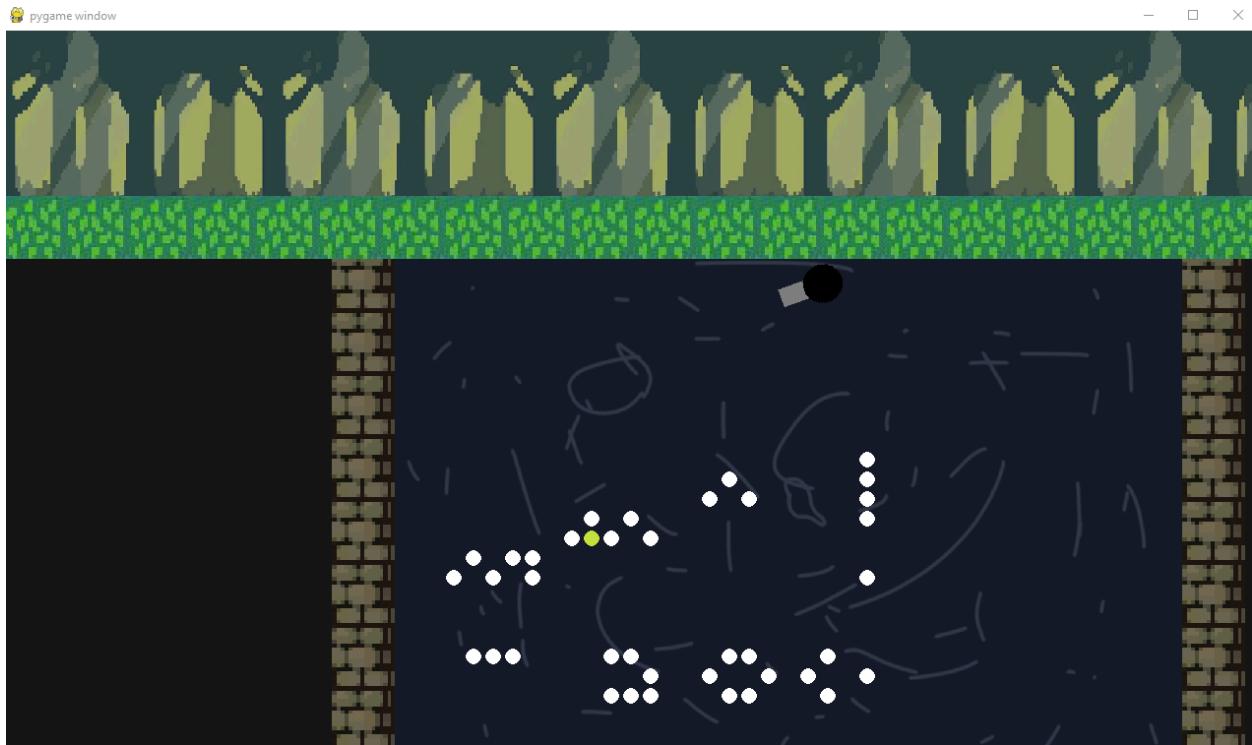
Entry 11:

I created a 2D array which now has 'x's to represent if a peg is there or not and then an 'r' which represents a reset ball. When there is an r a different string is passed into the peg class so a different coloured ball is drawn:



Entry 12:

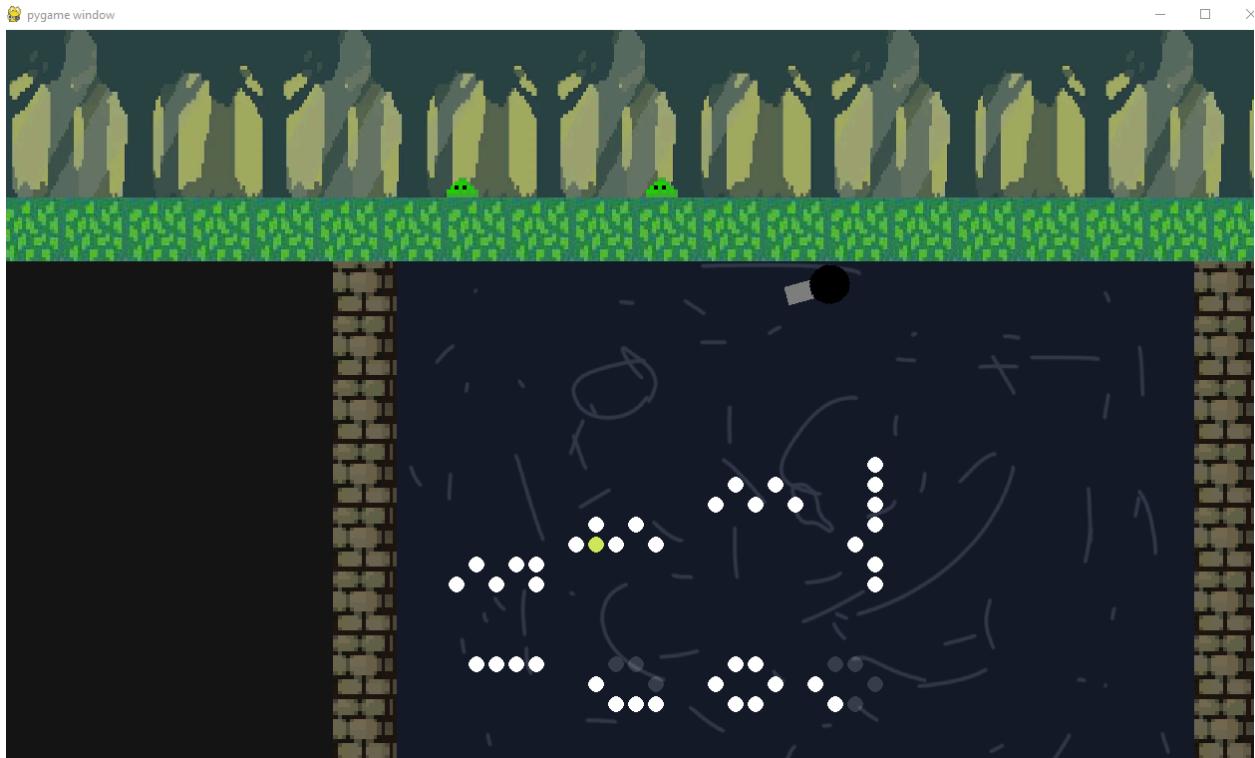
Next I wanted to add collisions so first I created a new method in my ball class, this was called `checkcollision`, this would be passed in the sprite class of all the pegs and then using the pygame function `spritecollide()`, I was able to check if the ball sprite had collided with the pegs if it had then it entered the if statement and set the velocity x and y to negative values and then removed the peg using the `.kill()` function.



This was now working except since it was just setting the velocity to negative values it didn't look very random the bouncing and also caused the ball to bounce back and forth which looked very bad. I tried to think of ways to fix this and how to calculate the angle of the ball and how it would be reflected except most things I tried didn't work. In theory in a perfect system when the ball collides with the peg it should be reflected however this didn't look very good as previously stated.

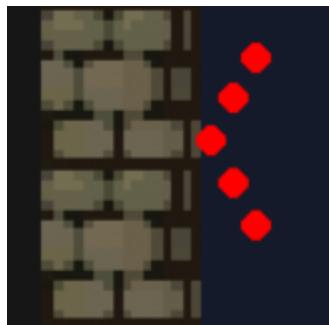
Entry 13:

I wanted to create faded pegs behind the normal pegs so you would be able to easily see how many pegs have already been hit, this was done by creating a new variable that is passed into the peg which for now I just called n, this would set the opacity of the ball if it was not equal to 0 so the balls I didn't want to change I made them pass in 0 for n. Then for the balls I did want to have some opacity I passed in 40:



Entry 14:

The monster movement has been broken after the addition of the shooting mechanisms was updated, this was previously controlled through a complex system of for loops and if statements that made no logical sense. Through changing some of the if statements and the variables that controlled them in an attempt to fix the enemy movement I have created all sorts of bugs such as the enemies moving off screen instantly and the user being able to reload even when the ball was still on the screen and shooting multiple balls.

**Entry 15:**

I Am now deciding to go back to the old idea I first implemented of having a 2D matrix of a fixed amount of possible positions that any monster can be at. This will be significantly different from the first list that had 3 elements, 2 for location and one for the type of monster. This new version will have one value for the x value and the other value as an object of my monster class that I created. If I am able to successfully get this to work I will attempt to have a method of my monster class that will smoothly move it from one position in the list to the next.

I have now been trying for a little while to loop through the position list and check if the space is empty signified by a character 'x', if it isn't empty the monster that is in that position will be drawn on the screen. This has not been working and the reason was because I was looping through the wrong list...

Great now it works!

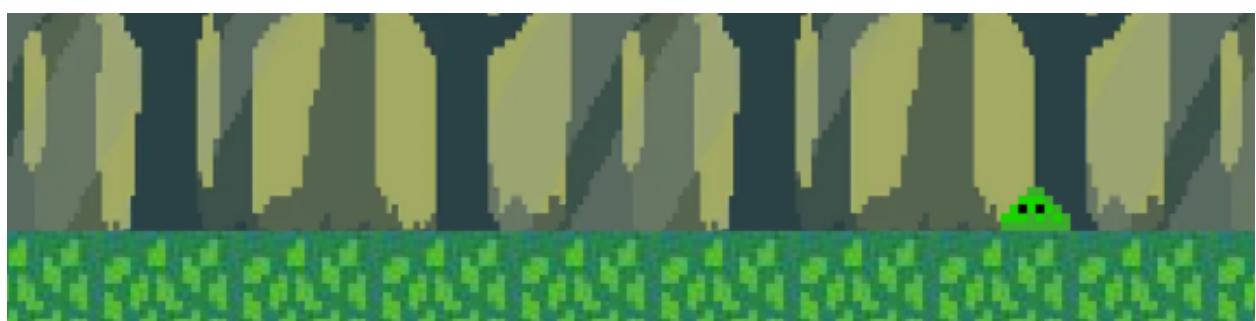
Depending on where in the list the monster is it will be draw at a different location:

Case 1:



```
self.positions = [[200, 'x'], [300, 'x'], [400, 'x'], [500, self.monster_list[0]], [600, 'x'], [700, 'x'], [800, 'x']]
```

Case 2:



```
self.positions = [[200, 'x'], [300, 'x'], [400, 'x'], [500, 'x'], [600, 'x'], [700, 'x'], [800, self.monster_list[0]]]
```

I have now made the monsters move forward however when the monster leaves the first position I want the second monster in the list to enter the scene, this only happens when the first monster moves 2 places. The program throws the following error when I attempt to change what part of the list is being checked:

```
File "c:\Users\Nirvan\OneDrive\Desktop\or
  self.states[self.gameStateManager.get_s
File "c:\Users\Nirvan\OneDrive\Desktop\or
  if self.positions[7][1] == 'x':
dexError: list index out of range
```

Even though there is clearly 7 lists in the main list:

```
= [[200, 'x'], [300, 'x'], [400, 'x'], [500, 'x'], [600, 'x'], [700, 'x'], [800, 'x']]
```

This makes no sense.

I have now fixed this issue and am able to move a slime across the screen by changing its position in the list. However it looks very snappy and doesn't seem like they're moving but more like teleporting. I will now attempt to implement this.

This has been a nightmare and I now have a strong dislike for coding, for whatever reason my brain could not figure out how to move the slime across the screen smoothly between point A and B. This is very frustrating. I have tried making a new monster class that draws a surface and image and then looping until the position of the enemy is equal to the position in the list. This didn't work and I undid the changes as it was causing issues, I tried a system of two if statements in an if loop that I was sure would work where if `int(counter) == 1` it moves the slime forward one pixel, then once that if statement is entered once it can't be entered again until the second if statement is entered(`int(counter) == 2`), the `int(number)` python function works in the way that if `counter = 0.4` then `int(counter) = 0`, so then the number would need to be incremented multiple times before a slime moved forward, thus making it appear that the slime is moving. I thought this would work but sadly it broke my program :(

Entry 16:

I am working on the collision system. I wanted to create a better system. I was trying to think of how I would calculate how to reflect the pegs using angles and then the balls normal, however I have come up with a better solution to detect collisions and made a new method called `collisionGoat()` as I spent a while trying to think of a solution to this problem. I am able to get the pegs centre x and y positions and then I am also able to get my balls centre x and y positions. I subtract my pegs x centre coordinates from my balls x centre coordinates and do the same for the y values. Through this I am able to determine the direction my ball is from the peg it has just hit. Then I am able to use the same calculations for when I shot the ball to create a new direction. So basically instead of bouncing off pegs I instead shoot the ball in a new direction each time it collides with a peg.



(Pictures aren't very helpful here)

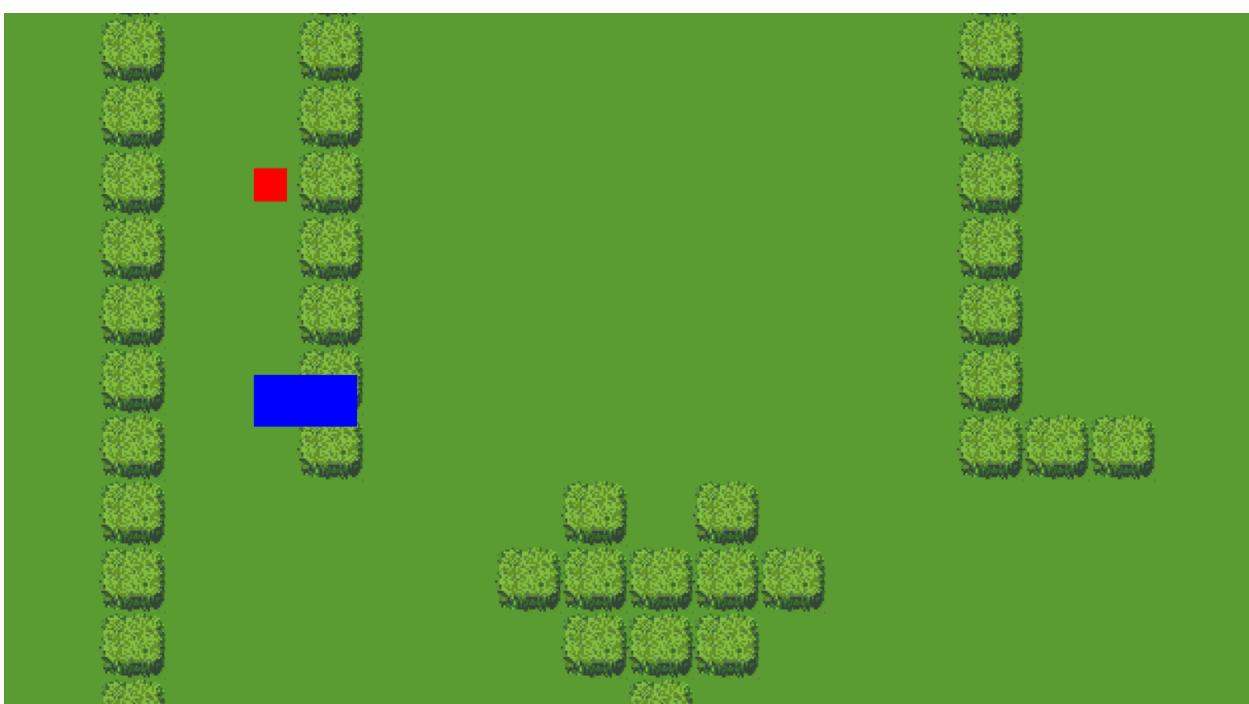
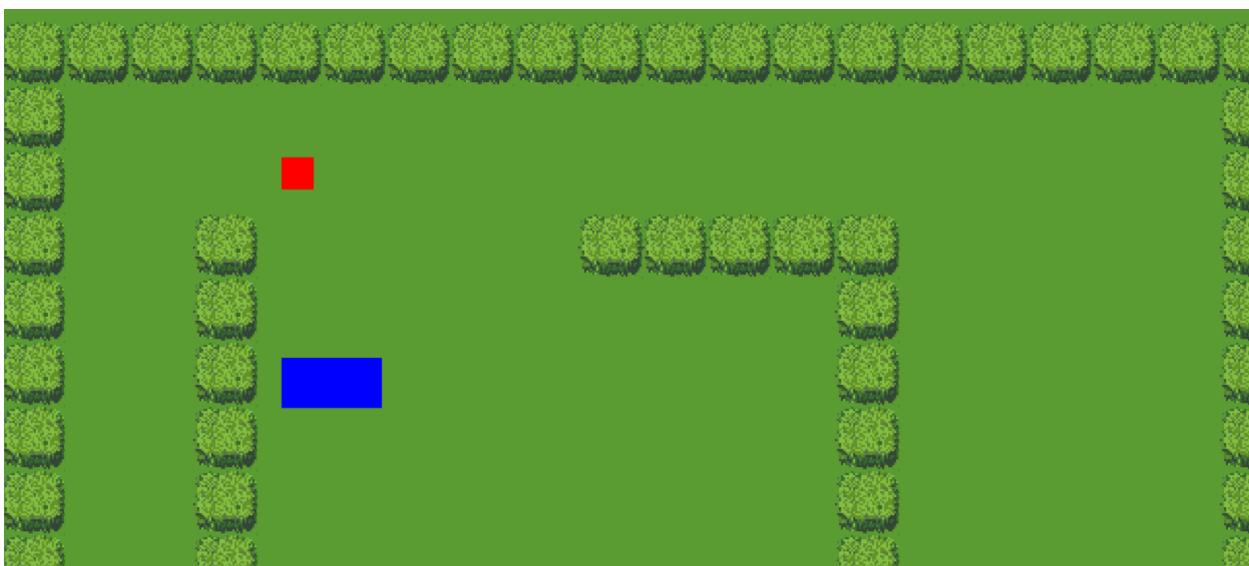
The main issue now was that there was no longer any friction so the ball looked like it kept bouncing forever as long as it kept hitting pegs. To fix this issue I created a variable called `self.reflect_friction`, this number was constantly multiplied by the friction variable then multiplied to the new direction so that each time the ball was shot off in a new direction it would be shot off with a little less initial speed.

The other issue was that if the ball collided with the ball exactly on top of it, then since it didn't take into account the previous velocity the ball would lose all of its momentum and just bounce up and down until it fell off the map. For this issue I tried by adding a number to the direction if the balls x coordinate was equal to the pegs x coordinate however that didn't fix this issue.

Entry 17:

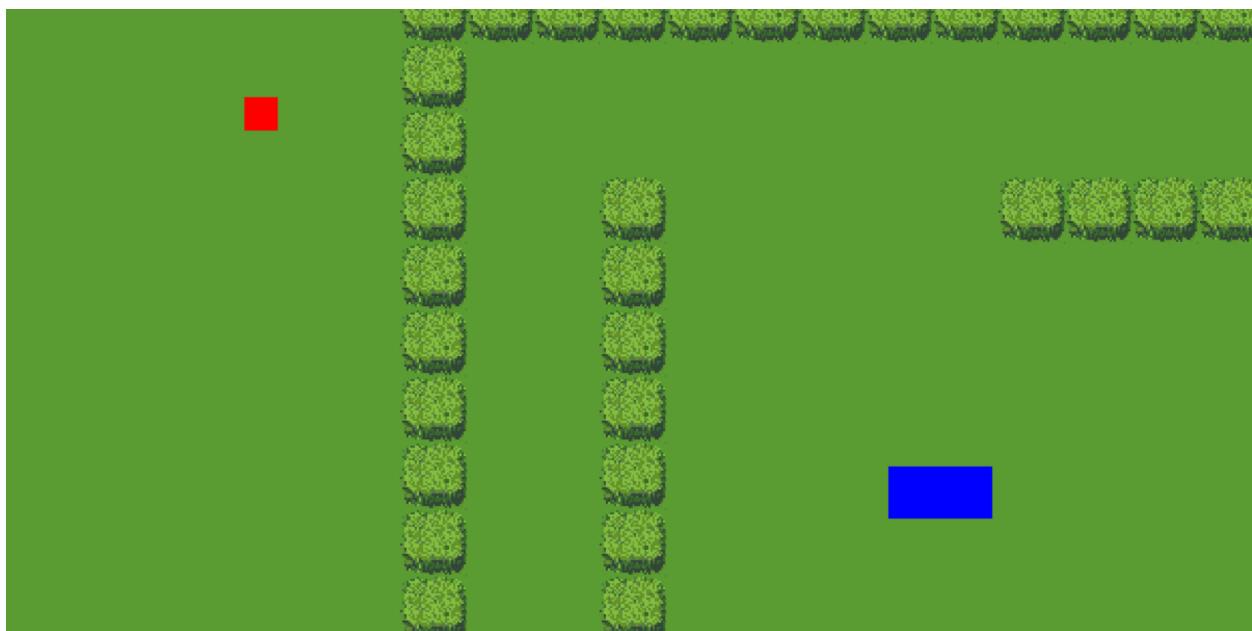
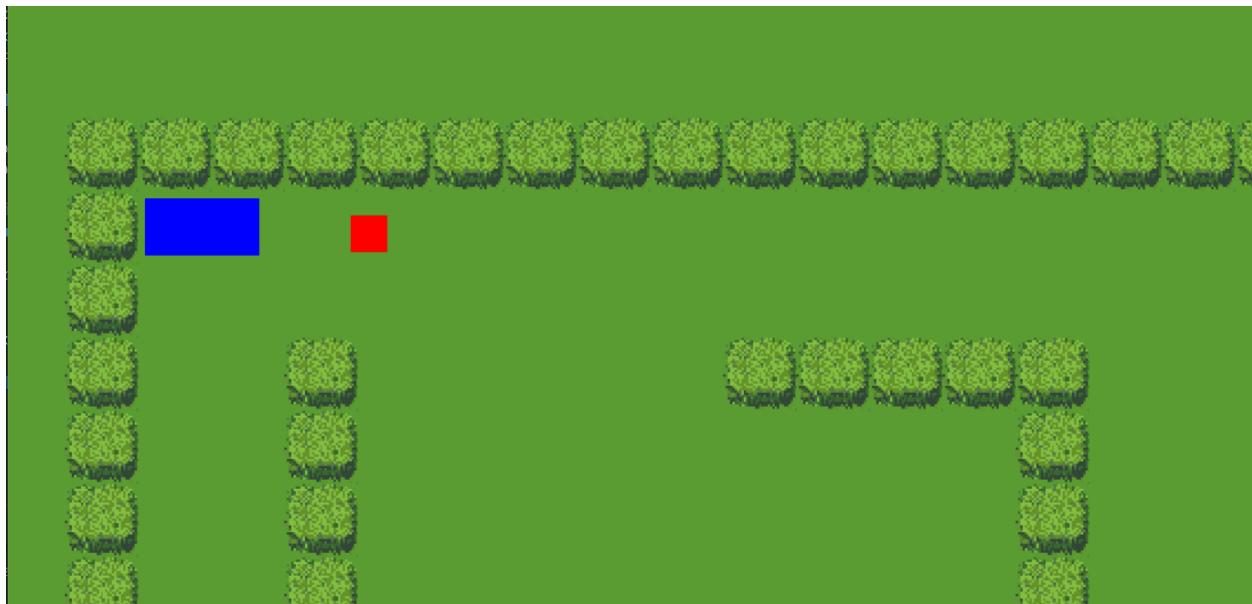
//Creating the map

I have scrapped the old design but used the same concept of having a 2d array containing single characters for certain tiles and looping through the array to draw each tile. I have now also changed the tile size to 64 pixels to reduce lag from looping through and drawing all of the tiles. Now instead of moving the player when the keys are pressed I have now made a variable called offset, when the player moves in a direction a value will be subtracted or added to the offset and then then the offset is added to the location of all things in the map to be drawn ultimately making it seem as if the player is moving around,

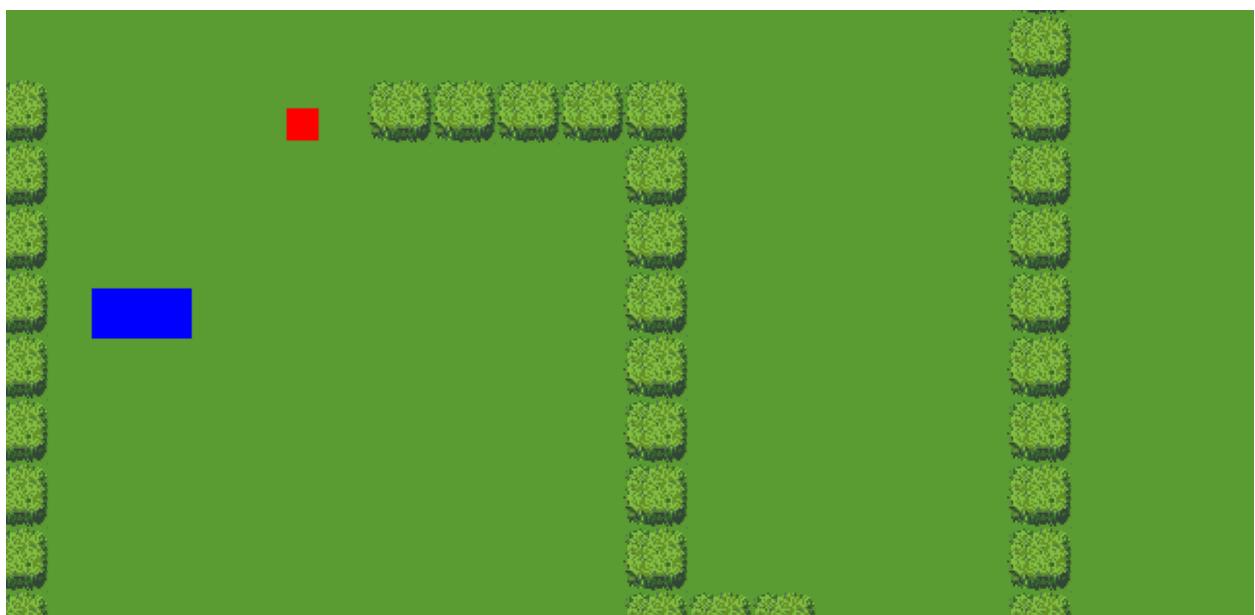
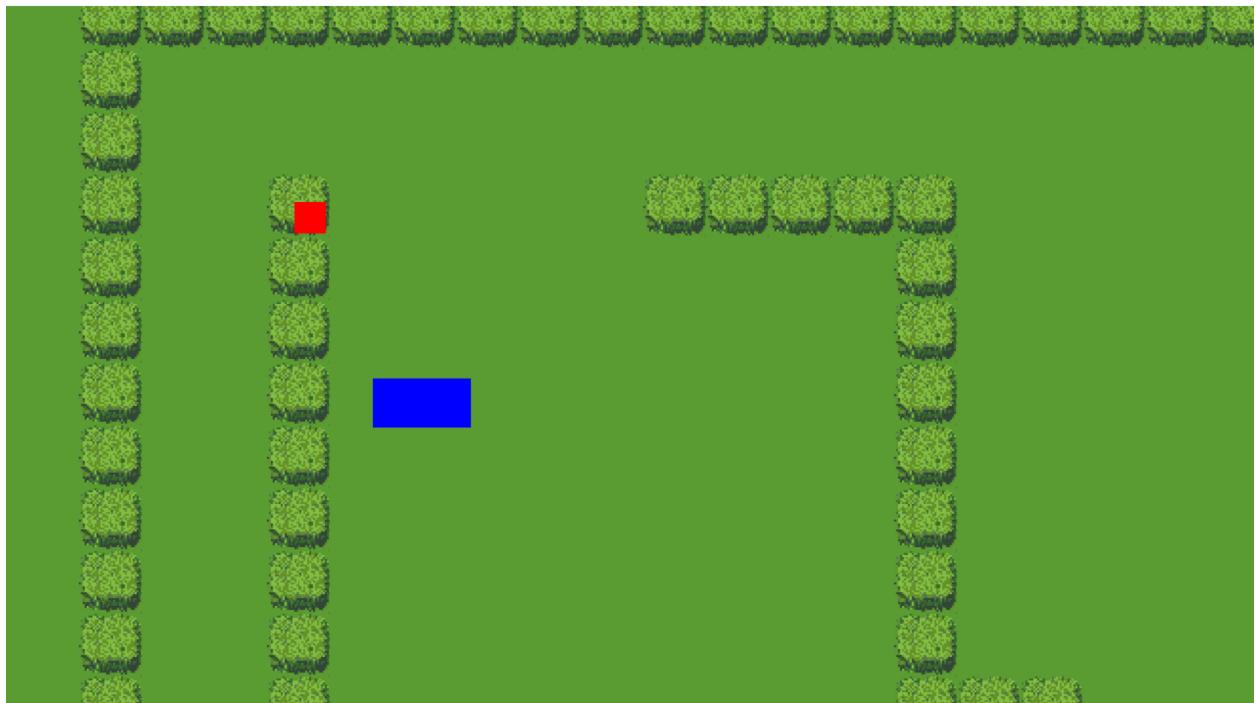


The only thing I struggled with was adding an offset to the blue box which is used to enter the level... so now you can't actually enter the level from the map, this in theory sounds easy enough to fix.

Okay that actually wasn't too bad to fix except now it looks like the box is trying to run away from me when I press my keys.



Okay now it works. I had the offset being added to another variable called self.x and self.y which was also getting added when the player was moving causing it to move in a weird way.

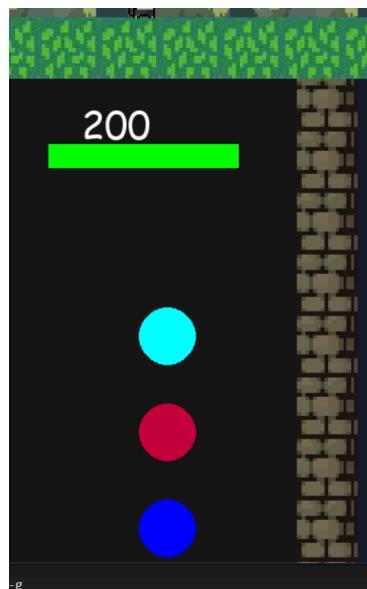


Entry 18:

Ammo:

I am going to add an ammo system which will display the amount of ammo that the person has on the side column of the game. This will show the type and order of the ammo. I will also of course have to fix the function that shoots the ball to make different types of ammo with different properties depending on the ammo available(I pray the level class will be finished soon; my class is already reaching 1000 lines...). Hopefully adding in the ammo won't be too hard.

I have added in a new class called ammo which takes in coordinates and then a string to determine the type of ammo to be drawn on the screen, they are then passed to the class that creates a surface and draws it at the coordinates passed in:

**Entry 19:**

I have been trying to add in a custom death animation since animation features are my favourite aspects of making this game. However the system I used to initially make my first sprite animation worked just not well... I have now been trying to make another separate method of my monster class function that creates a custom death animation once called however my code is stupid and it doesn't work. I have created a for loop that continuously calls my death animation and then calls another method called stop_normal() to cancel the normal animation so it doesn't happen at the same time, then once tube death animation is done the main game loop calls a different method of my monster class to allow the normal animation to proceed again except when I run the code nothing happens, I have now images since nothing happens :(

Quick side note: I was looking through my code and saw this if statement...

```
if 1==1: |  
    if self.positions  
        #print(self.p  
        if self.positions  
            self.
```

I really need to clean up my code

Yes I actually implemented the different animation by calling a death animation method which changes an if statement that controls which animation to play however sadly it broke everything else, the slime magically comes back to life, there is no health bar and they die regardless of if a peg is hit or not.

This is image of the beginning and end of death animation:



I've given up, I'm just going to clean my code up. It is too messy to deal with and it is making me go crazy. I have reached 1130 lines of code in the level class alone.

Entry 20:

So I tried to clean up the code and it was a whole mess I think I am just going to roll back on an old version that I have in my google drive, luckily it was only the sound and exit screen in the other version which isn't worth the effort fixing the file for as I can re code those features through reading through my previous entries.

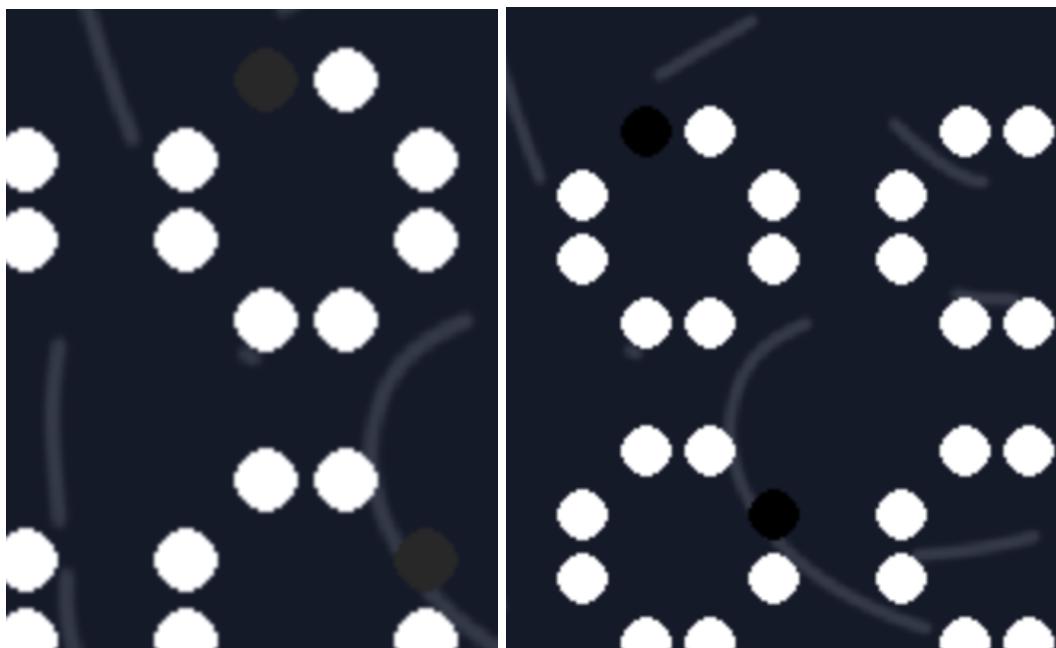
I have now been refocusing my time on working and building the map, before once the level was done you could exit back to the map but you would be back at the old map but you would be touching the rectangle that would take you into the game so you would instantly be taken back to the game. Using an if statement with a boolean that turns true once it has collided now

the if statement that takes you into the level is only entered once so after a level is completed you cannot go back to the same level.

```
self.level2collide = pygame.Rect.colliderect(self.player, self.level2)
if self.level2collide and self.level2completed == False:
    #level2 = Level
    #level2.run
    self.gameStateManager.create_level(2)
    self.level2completed = True
```

Entry 21:

I re added in the bomb feature that lay dormant in my code, in the peg layout map, I put in a 'b', so when the for loop goes through the map and sees a 'b', it creates a peg and passes in the string 'bomb' when creating an instance of my peg class, in the peg class if the string passed in == 'bomb' then it will create a black ball, then in the collisionongoat method of my ball class if the peg it collides with is bomb it will add 60 damage to the damage play an explosion sound and shoot the ball off with very high velocity, however when it hits other pegs it instantly loses that speed as the way that it bounces of other pegs doesn't take into account it's previous velocity which is annoying I may fix this if I have time, if I don't have time it is a feature.



Another issue is that initially the colour looks faded and only when a peg reset is hit it turns a proper black colour, another issue is that I want to have bombs only be hit once so when a peg reset is hit they don't come back, except when I made a for loop to check if the bomb had been hit already when the pegs were being reset it crashed my program.

Interesting thing to note is that the addition of the bomb significantly changed the gameplay as before it was a struggle to kill all 4 slimes before they dealt half your health in damage. Now you can easily kill them before they get close to you.

Entry 22:

I am now going to yet again work on the movement of the enemies, it has been something I worked on very early and yet struggled to wrap my head around I have tried at least three different methods of implementing it and all always felt off and not good (movement: 3; me: 0)

Initially I had a list that had all the positions of enemies and moved them along, then I changed it to moving the enemies by having positions specific to them, then finally a combination of the two, but now I will try to have the positions not controlled by a list and complicated if statements which check whether it can move or not.

Entry 23:

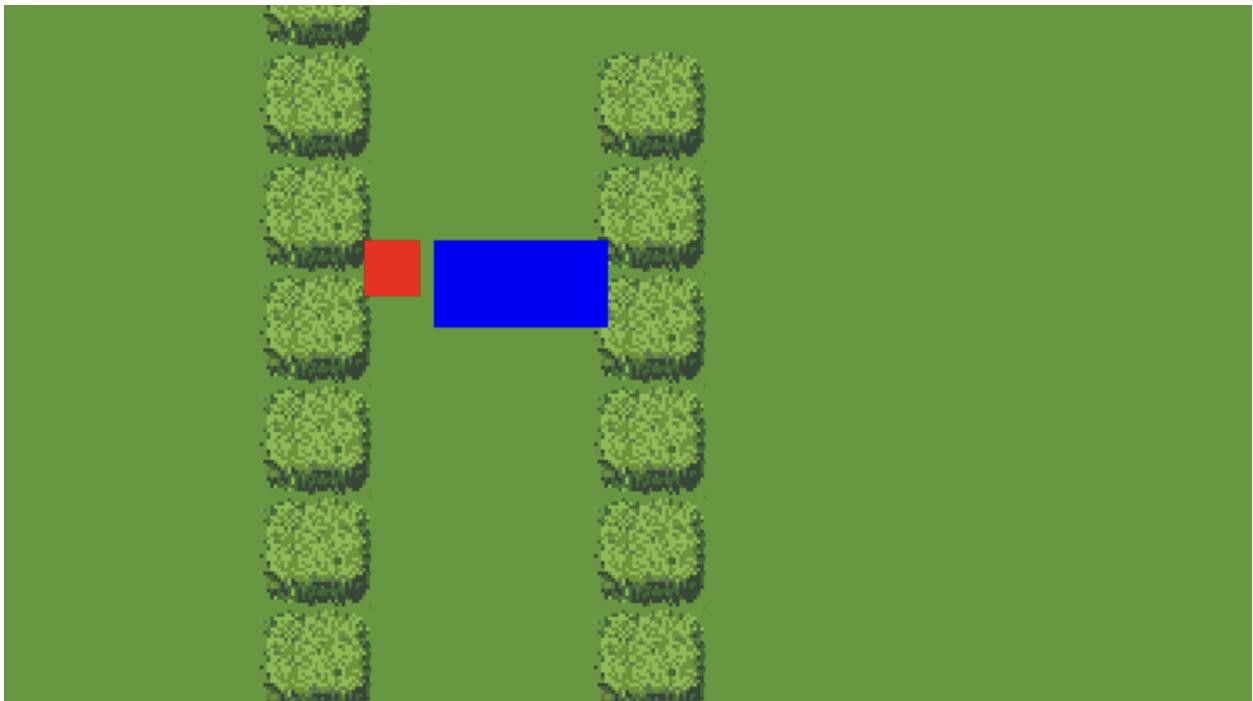
Okay I gave up on the previous entry about monster movement and am going to work on other stuff.

I have found a major flaw in my program which I hadn't considered before, I created a level class which is then called by a gamestate manager, I intended the a new instance to be created every time the level class was to be called and being passed the different enemies for each level,

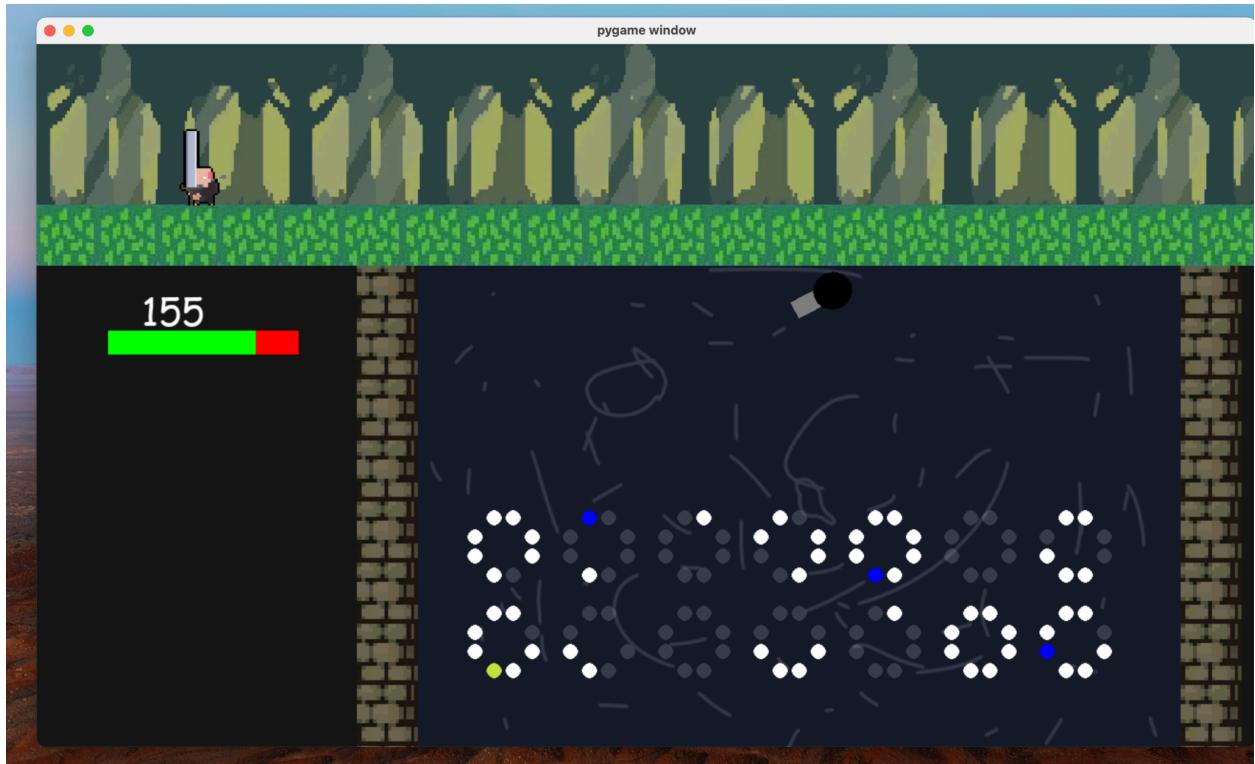


After the level ends a scroll appears which shows upgrades which is a feature that hasn't been added in yet, and then an exit button that takes you back to the map.

When you are back in the map you can then go to another blue rectangle which enter you into the level class.



Then when the player runs into it, the player class is run again in the main loop however when this happens it goes back to running the old level class which is not good.



I attempted to make a way to make a new instance of the class inside the game state manager so that when the level class was called again it would create a new instance but this would only show a black screen as the run method of my level class was not being run.

I am now going to try to make a class which is a level selector and will create a new instance of my class when it is called so that multiple levels will be run without requiring me to make multiple level classes by copying the level class into separate files which would be dumb, inefficient and a waste of storage.

Hooray! The feature actually worked which I was worried would be very difficult to implement as I had made the gamestatemanager very early on and forgot how it worked. I created a new class called **LevelSelector**, this then initially created separate instances of my level class and append it to my loop then a variable called **self.lvlnum** controlled which level in the list was being run in by the **LevelSelector** class, then I created a new method in my level class called **exit_code()**, which is a pretty bad name in retrospect, anyways this would return a boolean to say if the level has been completed, if the boolean returned is true then the variable controlling the which level in the list is incremented by 1 and then the '**map**', class is called by the **gameStateManager**, this then returned the player back to the world map and has successfully changed which level will be run next to prevent the player from replaying a level which has already been completed by them.

Entry 24:

I added in a variable that is passed to the init function of my level class called map, then in the levelselector class I passed in a position in an array of different maps to give each level of the game a different peg layout. Now let's see if the code runs.

Okay the code runs correctly:

Level 1:



Level 2:



Entry 25:

I wanted to have the ball spawned in once the user had shot the ball and the enemies had moved across one position, I first tried creating an instance of my ball class when they initial ball was deleted, however this only worked on the first shot but not on the second or third, instead I created a variable called self.killed to check if the first ball was shot and removed (I named this function self.killed because of the inbuilt pygame function .kill() which is used to remove sprites from sprite lists), once the variable is set to True after killing the first ball a new ball will be drawn and self.killed will be set to false again.

Entry 26:

I added in a new monster called 'eye' except the images are in facing the wrong way, to fix this I simply need to use the

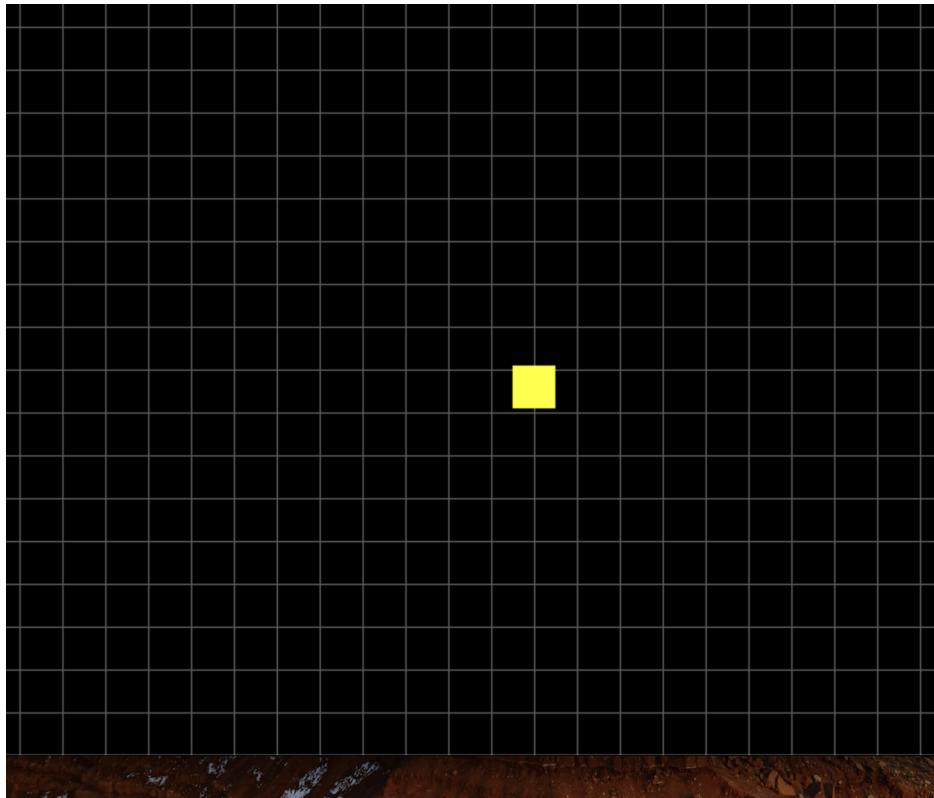
Entry 27:

I have created a new file in my program so I can attempt to recreate the map with better camera tracking as the player always appears in the top left corner of the screen as the camera isn't tracking the player but an offset is applied to everything.

I started by creating a grid on the screen by drawing lines on the screen.

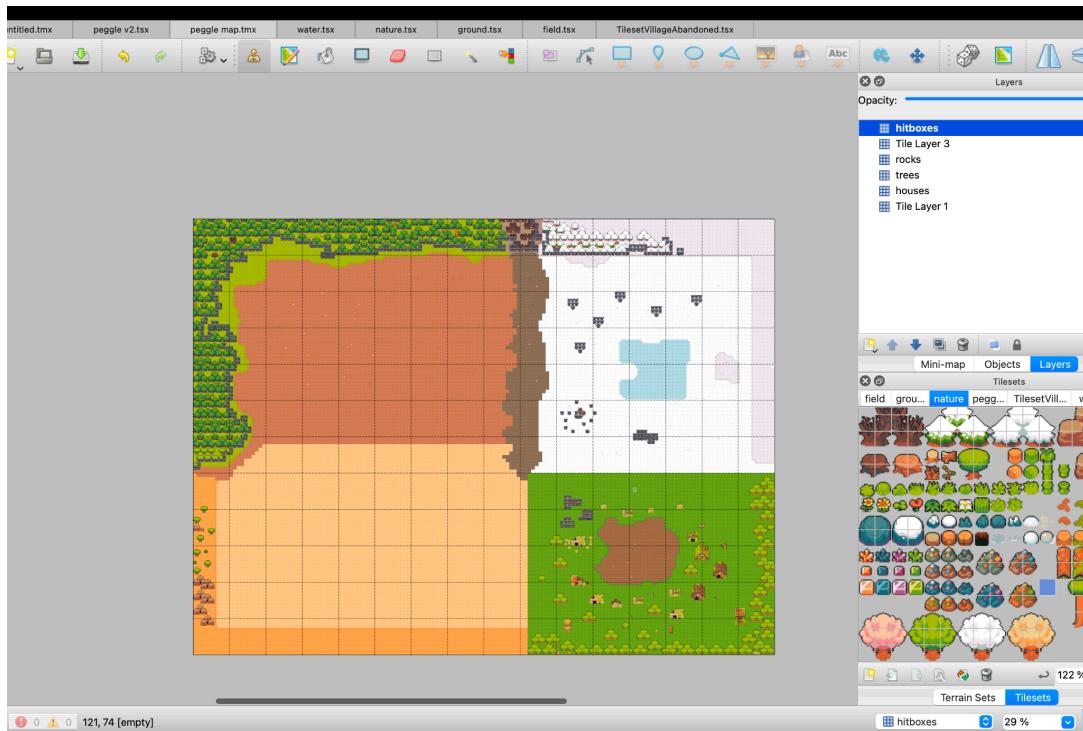


Then I created a player class that would be drawn on the screen at the location specified, then a surface is created at those coordinates, it is also passed in the sprite classes which the player will be added to the sprite classes draw and update method is called to allow the player to be drawn on the screen. The player's position is moved by using keyboard inputs of wasd, and using the move method of my player class, this takes is passed in an x and y value which is added to the x and y coordinates.



Entry 28:

I have scrapped the old tile map system I was working on as I was not able to create multiple tiles which could be collided with when I was trying to add them to the sprite group, instead now I found a software called tiled, this software allows you to import tilesets, create a canvas to place them on and export it as a png, this could then be blitted onto my game so I would not have to waste time redrawing each tile which would most likely improve performance, I found free tilesets on opengameart.org, imported them into tiled and then created a simple map as my level design skills are trash,

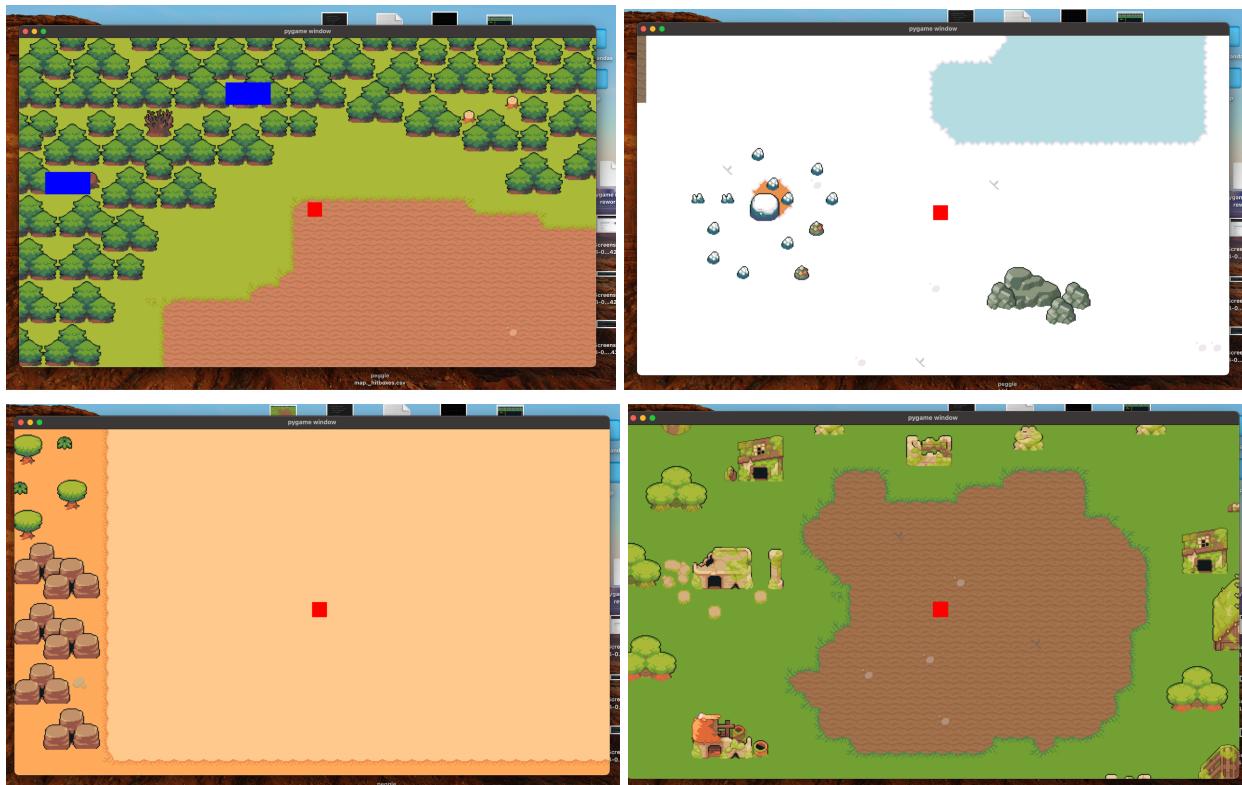


I created multiple layers as seen on the left of the screenshot above. This created layers of trees and other objects to make images to be layered on one another. Then I exported it to my game and blitted it on the screen.



As seen above the old tiles from the old map system were still displayed on the screen above the map, and they moved the same amount when an offset was applied to both the old tiles and the png of the map, this means that I should be able to create a hit box map which creates boxes on where the player won't be able to get past. The map image will be displayed so that the hit boxes won't be able to be seen and give the illusion of an image of the map which contains hit boxes.

These are the different areas in the map without the bushes:



Entry 29:

I created a csv file which contained the locations of where I wanted the hit boxes to be, then I opened and read the csv file.

```
self.scroll = [0,0]

self.hitboxes.add(Hitbox(100,100))

with open('hitboxes.csv', 'r') as csvfile:

    csv_reader = csv.reader(csvfile)

    for row in csv_reader:

        for col in row:
            #print(col)

            if col == -1:
                # print('oooooooo')
                pass

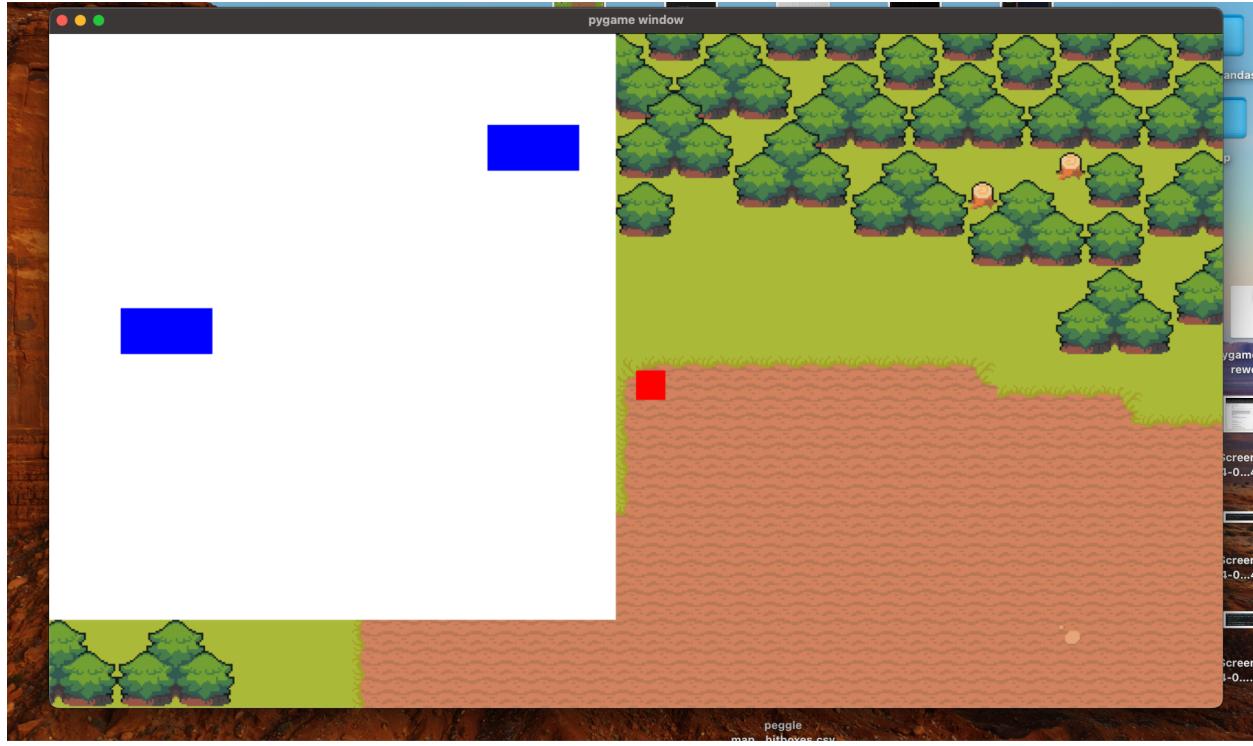
self.layout = [
```

However even when col was equal to -1 it didn't print my debug statement and I wasn't able to create an instance of my hitbox class for each time that it was equal to 1134, which was the number in the csv file that signified the location of the hitbox.

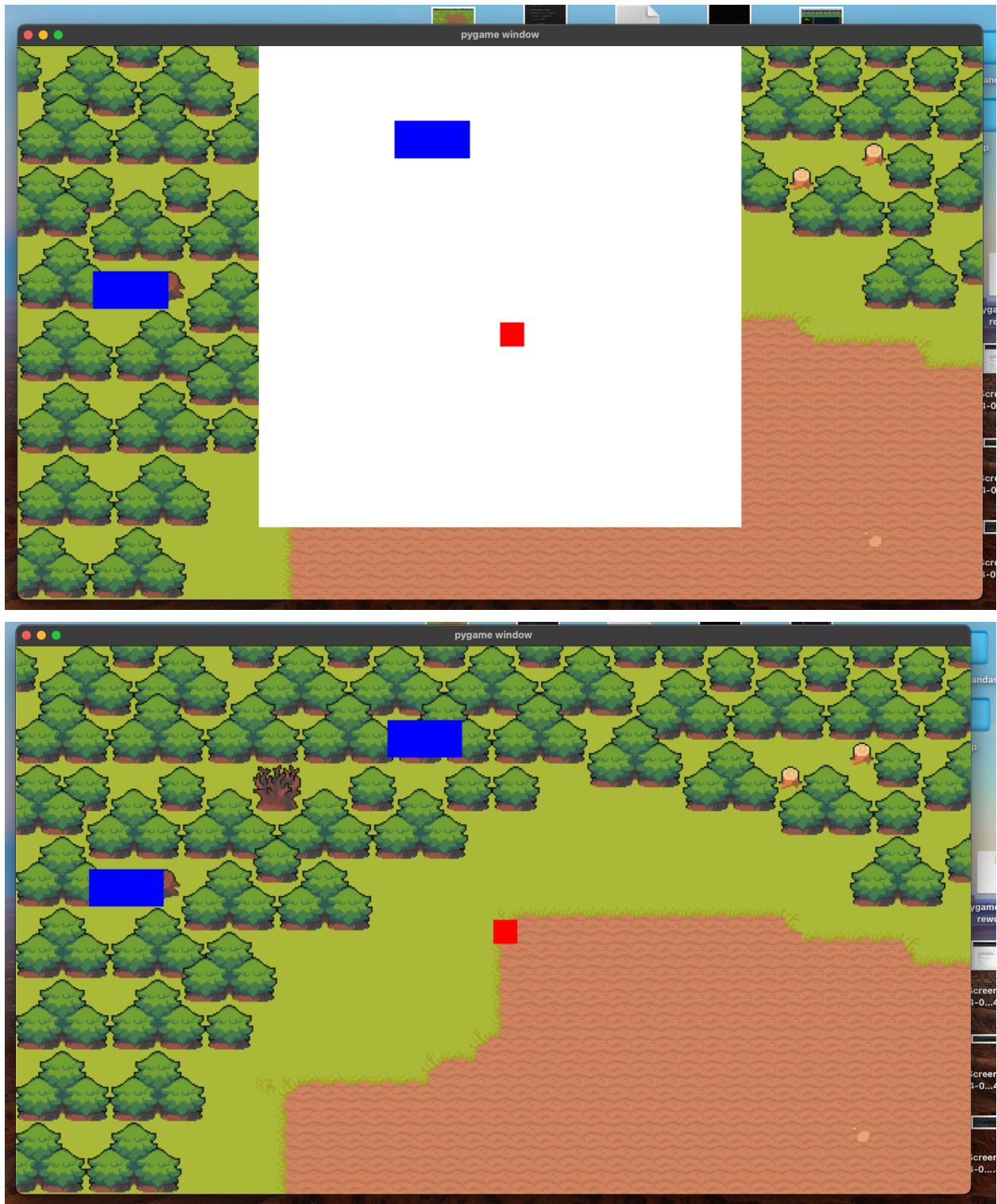
Instead to fix this I first started experimenting with using a list containing letters with an x signifying the placement of the hit box.

I then first needed to create the hitbox class which was passed in the location of where it was required to be drawn, then it created a surface and then filled the surface in white. This is very similar to many of my classes which just use a surface and are passed in coordinates on where to be drawn.

Then I created a list which contained only x's to create a solid block, as seen below.



Then I tried applying an offset to the block, this wasn't initially working as I first tried to loop through the sprite class containing the hitboxes and calling a method of each one to add an offset to its x and y coordinates, this however caused the value to be continuously added causing the box to move left and right rapidly.



Instead to fix this I loop through the sprite class and kill each hitbox (this is using .kill() pygame function), then I redrew the hitboxes with a new offset applied. For some reason this caused the hit boxes to move more than compared to the background image:



To fix this I divided the offset by a specific number until it seemed to move the same amount as the background image which was 31.9999:

```
    row in range(len(self.layout)):

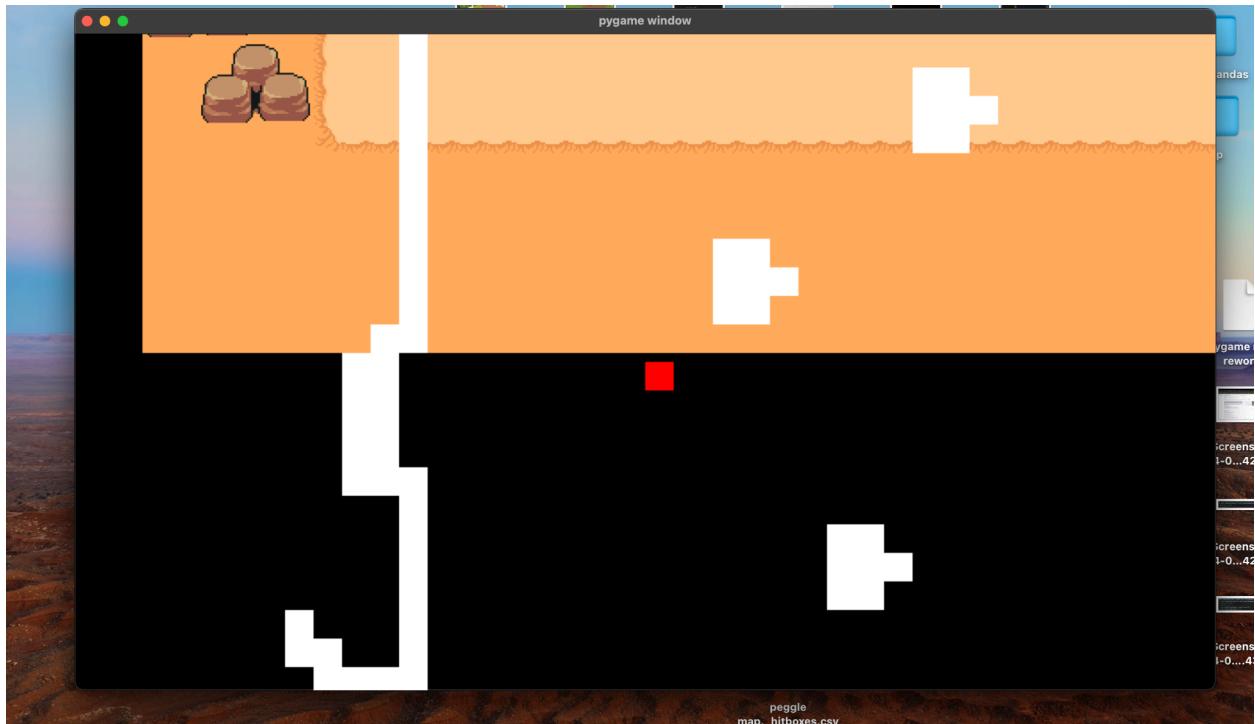
        for col in range(len(self.layout[row])):
            if self.layout[row][col] == 'x':
                self.hitboxes.add(Hitbox(row+self.offset[0]/31.9999,col+self.offset[1]/31.9999))
                #print('created hitbox')
```

Then finally I used the map of the proper hit boxes to draw the hit boxes. This was looking very promising and I was very glad that It had worked:

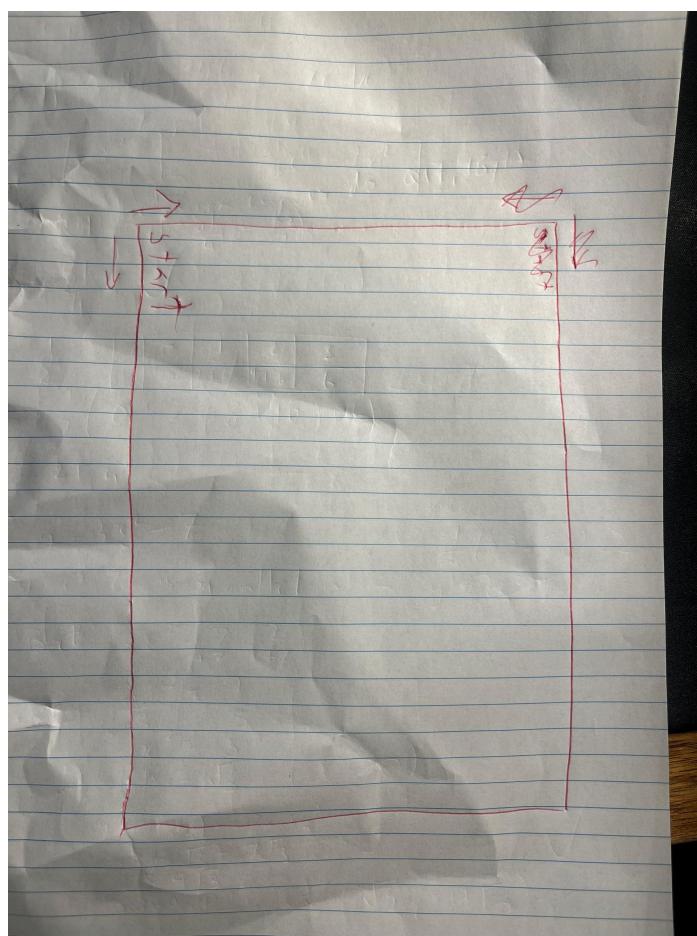


I was very happy that I got it to work until I realised that it was drawing the map sideways...

This has to be the dumbest way for it to mess up...



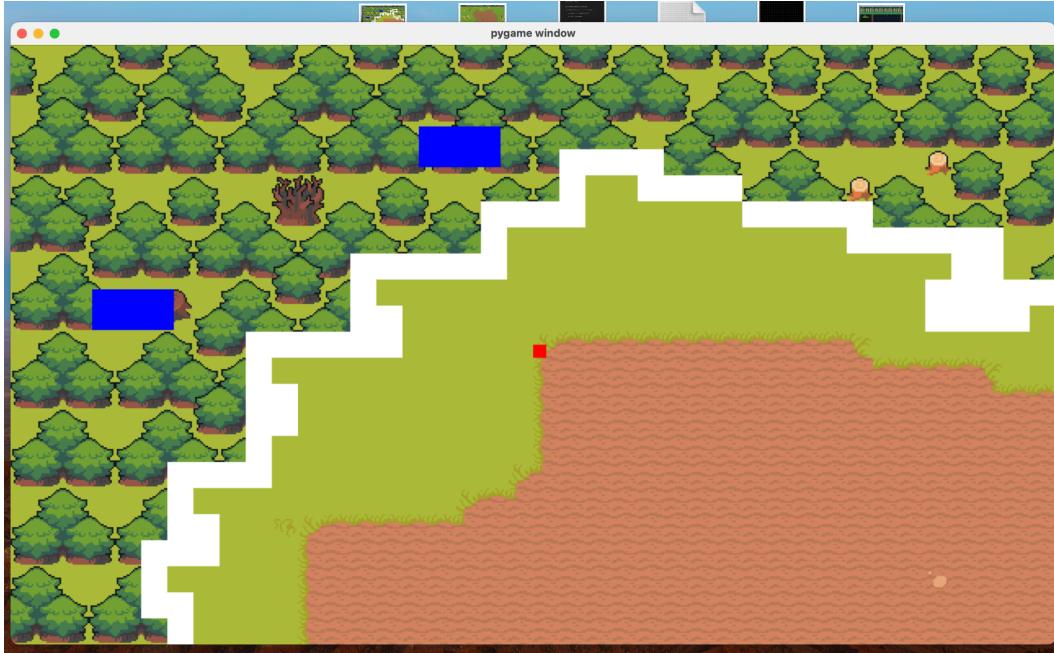
The map is being draw like this:



The map had been flipped and rotated 90 degrees from how it should be displayed so I will now have to read the list by going down the rows first and drawing all of them and then going across the columns as seen above in the very bad diagram I made.

Entry 30:

I have figured out the issue I was drawing the box at `self.hitboxes.add(Hitbox(row,col))` the row represents the y-value, I go down the rows as I loop through the list, this means that when I draw the hitbox I need to draw it with row as the y-value and col as the x-value. Once I had done this It fixed the way that the boxes were drawn:



This is now being drawn in the correct positions and I will have to create a system to check for if the player has collided with the boxes and stop the relevant offset from being applied.

First of all I created 4 if statements, one to check each side of the box so if the player hits the left side of the box a positive offset is added as a negative offset is being applied by the player to move left, then I repeated it for all sides of the boxes. Then I used the `colliderect()`, pygame command to see if it was colliding, however I wasn't able to tell if it was colliding from the bottom, top, left or right. To fix this I compared the location of the player's position to the box's position. However when testing this was still not correctly detecting if the player was colliding from the bottom or the sides, this is because I was getting the boxes position from the corner, to fix this I added 32(the boxes size) so if the players x-position was greater than the boxes position + 32 then it was colliding from the right. I used this logic to check for left and right and it did work however when I added in the collision for top and bottom it broke the code. This was because when I collided from the bottom instead of the player not being able to move the player would move right instead as the code thought the player was colliding from the right so it would change the offset accordingly. This was very bad as it sometimes caused the player to clip through the barrier and then be stuck:



This may be caused by the boxes which are offset slightly more than they should while the player is moving(it was difficult to screenshot this), so that when the player stops moving and collides with the bottom of a block they are also technically colliding from the right or left as they have clipped inside the block which would add or subtract the offset for the left or right accordingly causing the player to move to either the left or right and move outside the wall.

Entry 31:

I have given up on fixing the barrier around the map as I was having too many issues and wasting a lot of time(I feel I am starting to run out of it).



I am implementing health this will display the health of the enemy out of its max health.

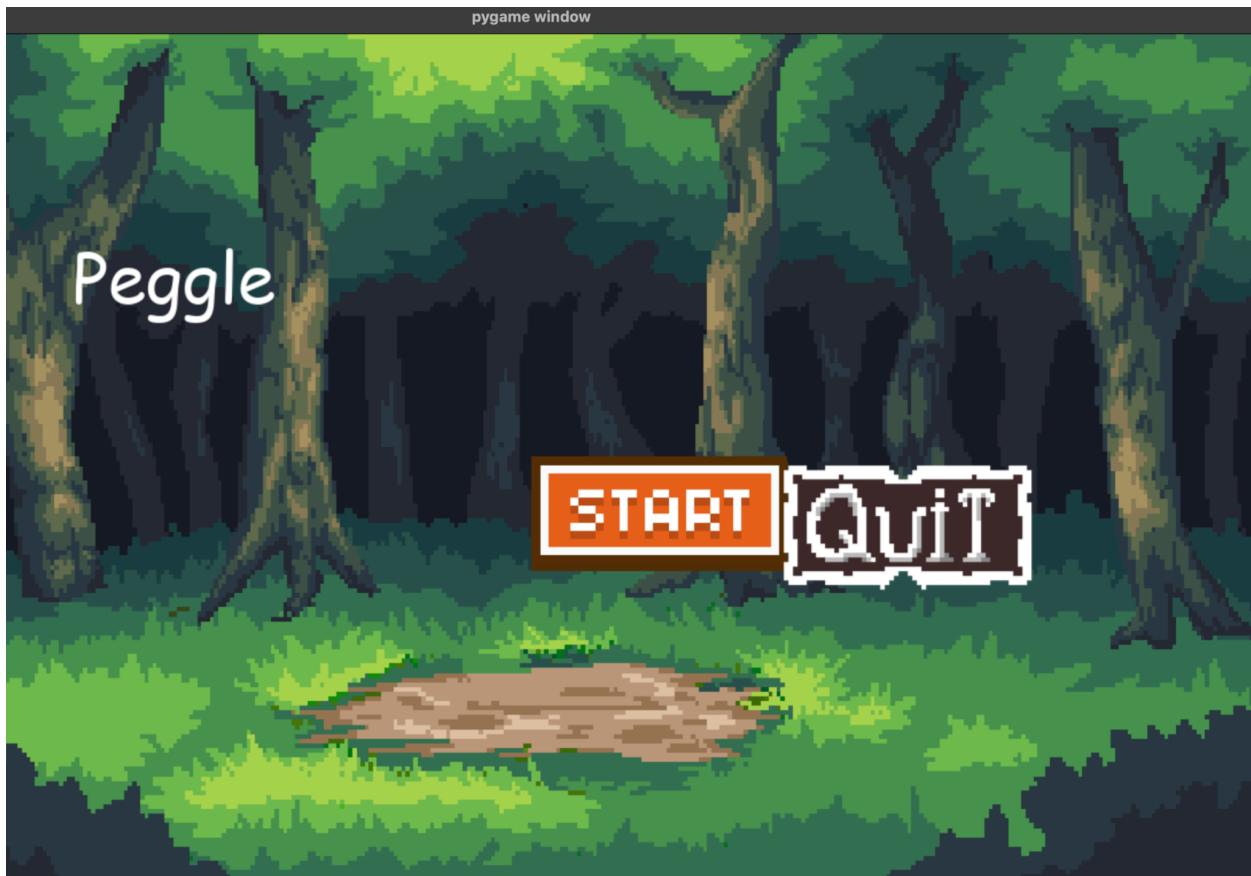


This was fairly straightforward to implement, first I converted the health passed in into a string and called it myString, then I created a test surface and then made the current enemy health into a string along with the original health stored in myString. This was then blitted onto the screen, I just needed to resize the text and place it higher above the enemy:

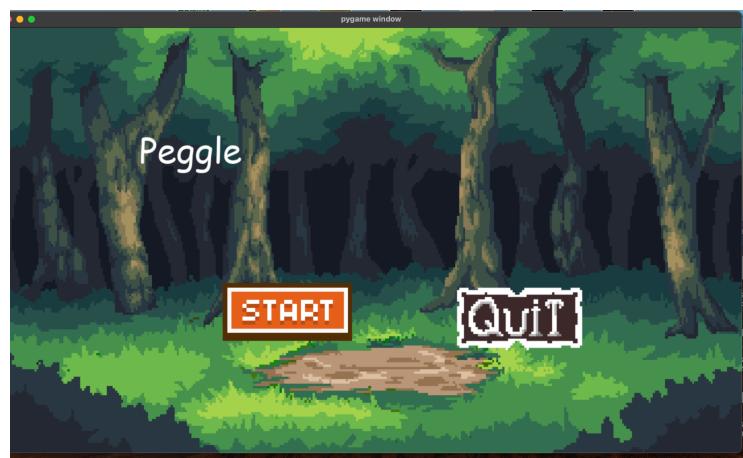


Entry 32:

I am fixing up the start menu, the buttons are off centred and don't stay in a similar position when you full screen, one is drawn at the width divided by two and the other is drawn at the width divided by 2 plus 200 pixels,



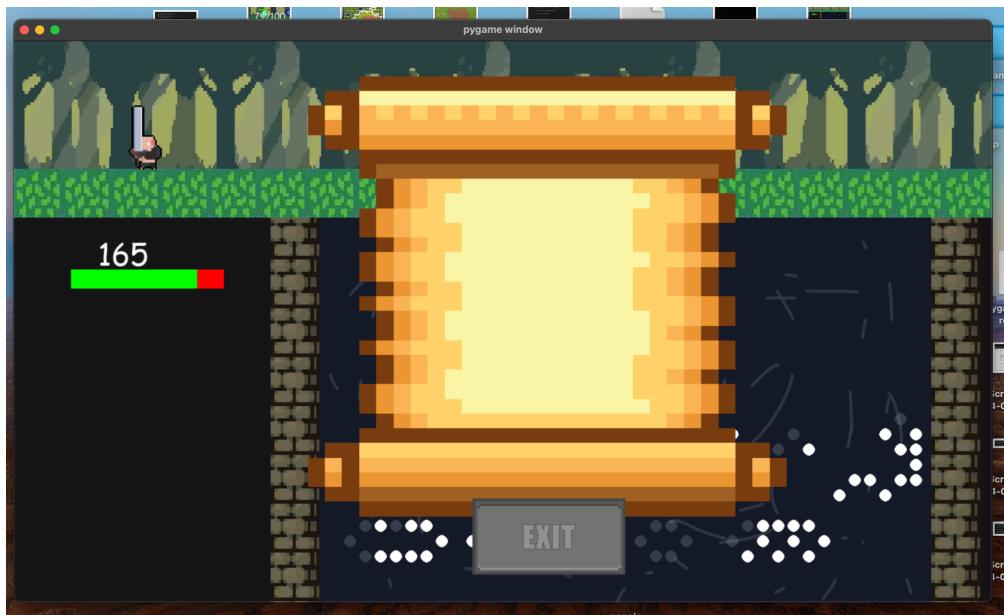
The reason it moves so much when fullscreened is due to the fact that pixels are being added on and those 200 pixels take up a significantly larger portion of the screen when it isn't fullscreened, to fix this I multiplied the widths by a decimal(e.g. 0.8), this was able to control what percentage of the screen to draw this button:



Entry 33:

I wanted to be able to have the players health be carried across the different levels, as right now when a new instance of my level class is created the player has 200 health, when the player completes a level and the second level class is run the player is back to full health. The way I thought of implementing this was by creating a list with the first level, this level would then be run. Then when the player presses the exit button self.exit and self.playerHealth are returned, if self.exit is true, then a new instance of my level class is created and appended to my list of levels with the values returned from the first level being passed in to it, and then run.

First level where the player has 165 health remaining:



The second level where the player begins with 165 health:



Entry 34:

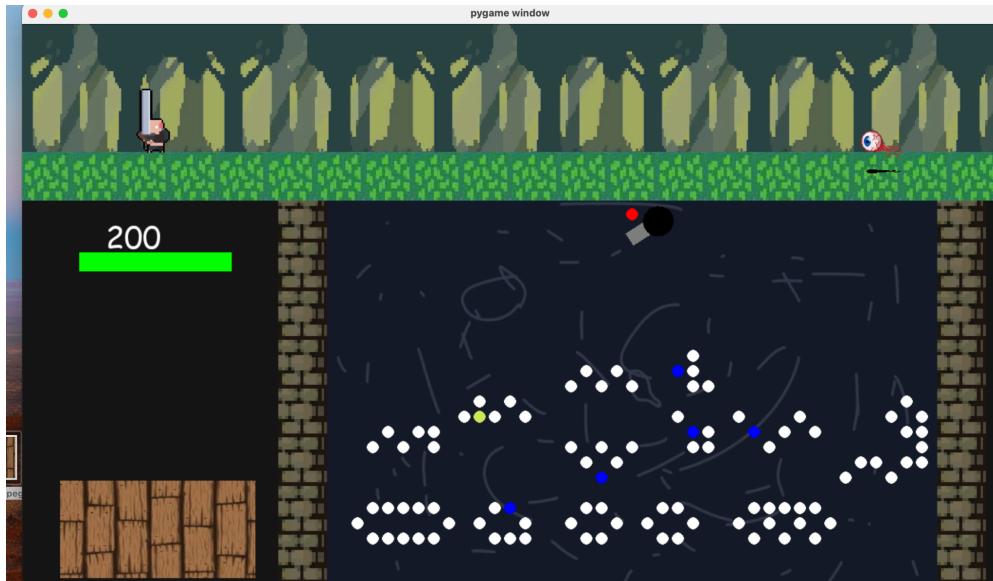
A while back I was trying to add in an ammo system which keeps track of the amount of ammo and types, I am now adding something kind of similar, it will be a fruit system, these fruits give special abilities and each fruit can be used once per each level, the user gets a maximum of 3 fruits and after each level has the option to switch them out I will be using these fruits which I found on opengameart.org:



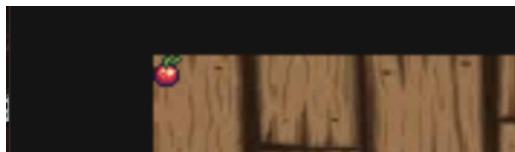
These have a similar concept to one piece which has fruits that give special abilities which was my inspiration for this idea(you should watch one piece):



I first found an asset of a wooden block, this was then blitted onto the screen, this will be where the players 3 devil fruits will be displayed:

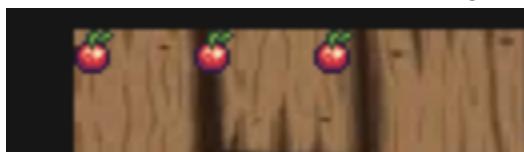


Next I created a class called Devilfruit, then I created a list called fruits, which contains a string representing the different fruits, then I created three instances of my Devilfruit class and pass in the different strings contained in the fruit list, then finally I added all of the new objects that I created to a list. Next I created a method called draw in my devil fruit class.



The fruit was then drawn but only one since in the fruits list I had apple in the first position, empty in the second and pineapple in the last.

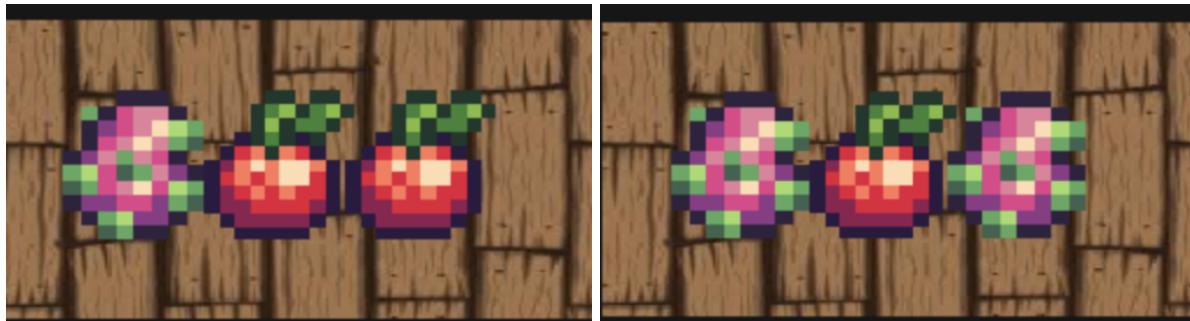
When I put all three fruits to apple I got:



And then finally once I rescaled them I got:



Then I added different fruit types and got this output depending on the order of the list:



Now that I have implemented some basic visuals through the devil fruit class and am able to add more fruits when I need them. I now wanted to make the player be able to click the fruits and use them to gain abilities. First I got the rect of the image of the fruit and then used a in built python function called colliderpoint, I passed in the user's mouse coordinates and the rect of the image, then used a print statement to check if it worked I got the following output:

```
nirvanpulakhandam@Nirvans-Laptop:~/Pygame$ pygame 2.5.2 (SDL 2.28.3, Python 3.11.3)
Hello from the pygame community. https://www.pygame.org/
2024-06-18 17:18:35.223 Python 3.11.3 on macos
macOS versions of this application are not supported.
pygame 2.5.2 (SDL 2.28.3, Python 3.11.3)
Hello from the pygame community. https://www.pygame.org/
2024-06-18 17:18:35.223 Python 3.11.3 on macos
macOS versions of this application are not supported.
pygame 2.5.2 (SDL 2.28.3, Python 3.11.3)
Hello from the pygame community. https://www.pygame.org/
2024-06-18 17:18:35.223 Python 3.11.3 on macos
macOS versions of this application are not supported.
pygame 2.5.2 (SDL 2.28.3, Python 3.11.3)
Hello from the pygame community. https://www.pygame.org/
2024-06-18 17:18:35.223 Python 3.11.3 on macos
macOS versions of this application are not supported.
pygame 2.5.2 (SDL 2.28.3, Python 3.11.3)
Hello from the pygame community. https://www.pygame.org/
2024-06-18 17:18:35.223 Python 3.11.3 on macos
macOS versions of this application are not supported.
pygame 2.5.2 (SDL 2.28.3, Python 3.11.3)
Hello from the pygame community. https://www.pygame.org/
2024-06-18 17:18:35.223 Python 3.11.3 on macos
macOS versions of this application are not supported.
```

This means that the collisions is not working when my mouse hits the image of the fruit

When I printed out the image rect I got:

```
<rect(0, 0, 64, 64)>
```

This is most probably getting the dimensions only but not the location of the image

This may be due to since it just an image that is drawn on the screen so getting its rect won't be able to find its coordinates, instead to fix this I created a surface called image, then I blitted the apple image on it and set its rect position to the position passed into the function and got this output:

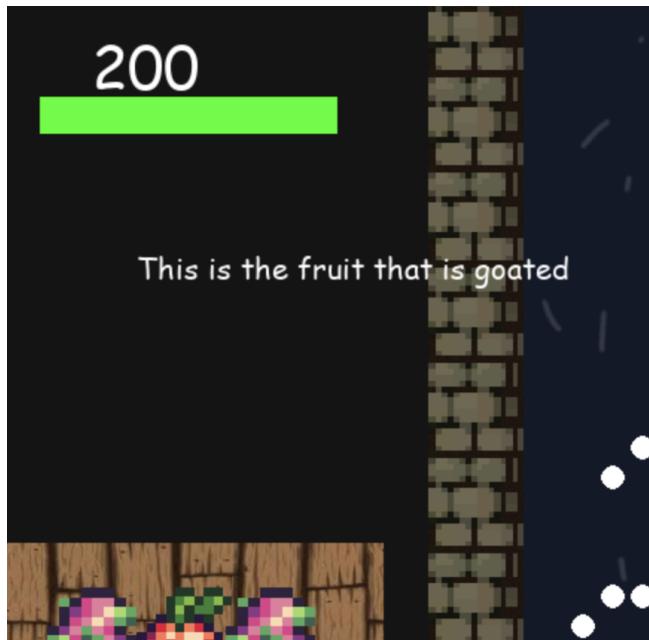
```
self.levels[self.lvlNum].run()
File "/Users/nirvanpulkhandam/Downloads/Software Project/cleanedcode.py", line 304, in run
    fruit.draw()
File "/Users/nirvanpulkhandam/Downloads/Software Project/cleanedcode.py", line 630, in draw
    self.image = pygame.surface((32,32))
TypeError: 'module' object is not callable
nirvanpulkhandam@Nirvans-Laptop Software Project % 
```

I have no idea what this means tbh and I kinda thought it was gonna work, I could find no solutions on my own and explanations about module object not being callable made no sense, instead I undid the changes and found another solution, I loaded in an image of the apple then got the rect of it which would get me the dimensions of the image as seen in the print statement above, then I set the topleft of the rect to the coordinates passed in on where to draw the image, then this rect was blitted onto the screen, this now was able to display the shape but still when I tried to check for collisions it wasn't working then when I attempted to print the mouse coordinates, they kept printing the same location regardless of how I moved my mouse which was strange:

This is when I realised how dumb I was being, I defined the mouse position in the init method of my Devilfruit class. this meant that it wasn't updated and only got the mouse position once.

Then after an hour of trying I finally saw the print messages I had been looking for:

Next I made it that when the mouse is colliding with the fruit and the mouse is clicked next is displayed on the screen:

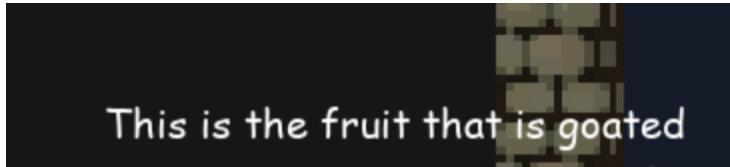


However when multiple fruits are clicked then the text overlays itself as they aren't undrawn:

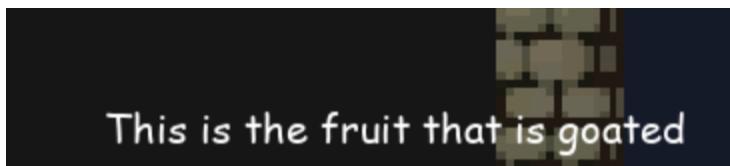


To fix this I made a method called activated() this returned active, if a fruit was active then all other fruits would call the method deactivate which hides their text:

When first fruit is clicked:



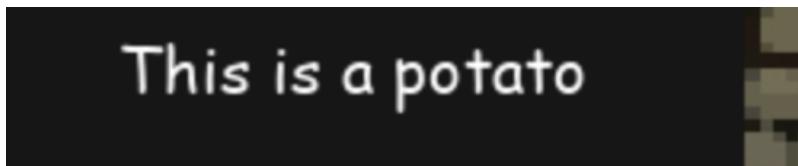
When second fruit is clicked:



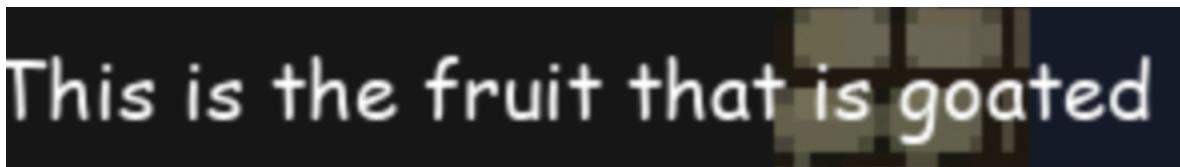
The potato text appears behind the fruit then disappears when the potato isn't being clicked

This is likely due to the apple remaining active which I dont know why.

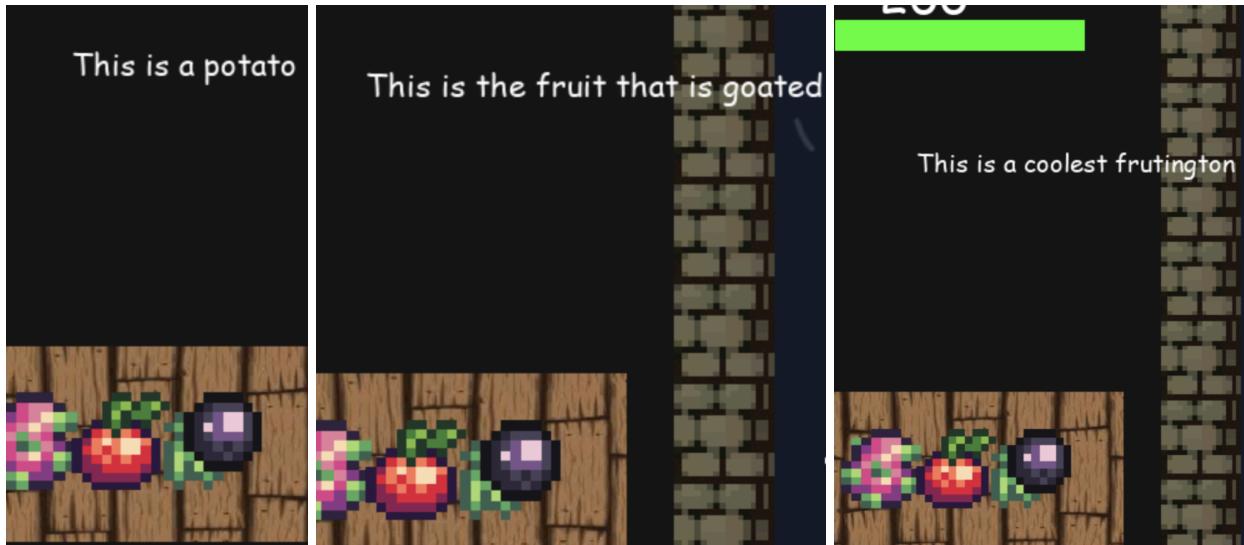
When potato is clicked first the text appears:



When apple is clicked afterwards, the text appears as it should but when you try click on potato again it doesn't work:

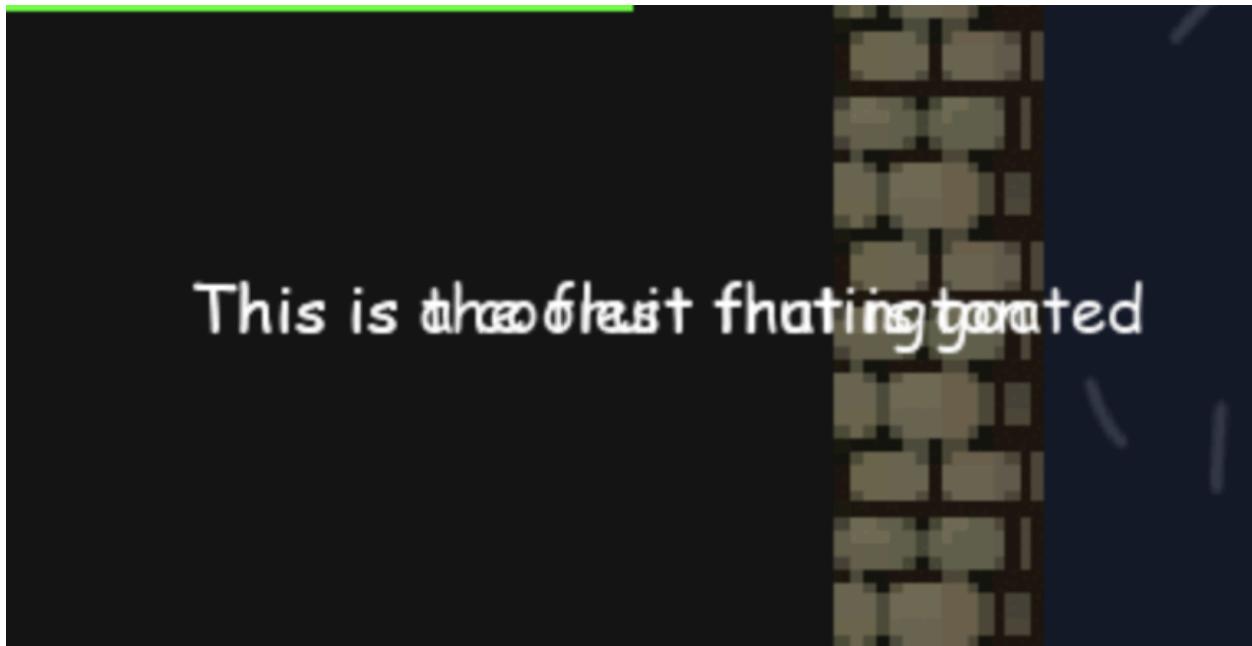


This is fixed by making it that when the fruit is clicked a variable called active is set to True, then it is returned, when the variable is returned, if the first fruit returns true then a method called deactivate() is called for the other two fruits which removes their text. Then when the draw method is called again active is set to False, this is to stop only the first fruit being checked from always being active the following output is:



(Ignore the text I was using to test this 🙏😭)

Next I wanted to make it that the changing fruit only works when the player hasn't shot the ball, I just made it that the text changing doesn't work when the player has shot the ball but this messed up the text:



Instead to fix this I created a function to check for if the player has clicked a fruit, called `get_click()`, this does the same as when it was inside the draw function but instead when the player has shot the ball the `get_click()` function is no longer run.

I wanted to display a random fruit on the scroll that appears after completing the level, this then can be swapped out for another fruit if the user wants to. To begin with this I added two more variables that are passed into the Devilfruit function, this was `sx` and `sy` (scale x and scale y), then when all the monsters had been killed, I created a new object of my Devilfruit class and passed in the correct position and increased the scale size:

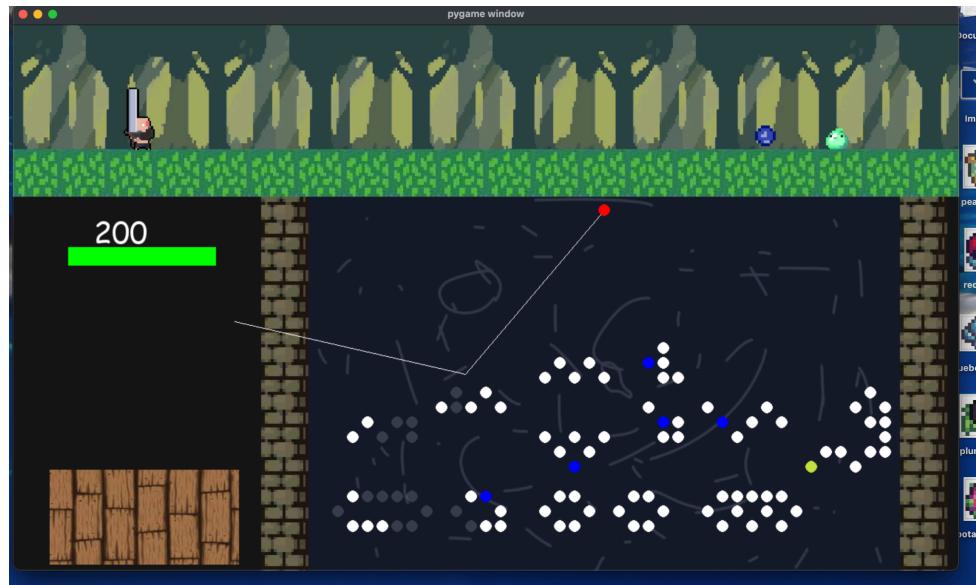


Entry 35:

I want to add a system for me to be able to aim where the ball will go and a line will be drawn, to add this I first created a sprite class with the correct `super.init()` and inheritance of `pygame.sprite.Sprite` for this to be added to sprite groups. In the init method I gave this class information about the balls starting location, then created a surface the size of the screen for the line to be drawn on, and got its rect, using `get_rect()` Then I created an update method which drew a line from the ball to the mouse's position:



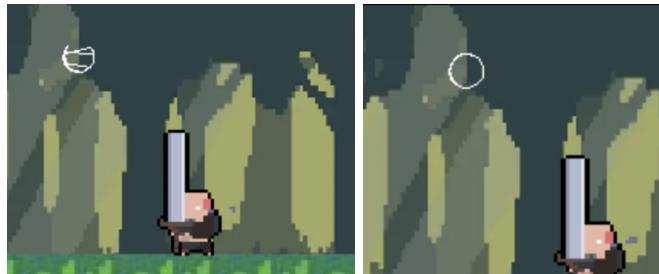
Then I found that you can draw multiple lines by passing a list of all the locations of each line thanks to the tutorial <https://www.youtube.com/watch?v=qLKuFPWDHjU>, the pygame function requires a list of coordinates and then connects them on the screen, this is me testing it out:



Now I should be able to create a list of points that a ball would travel through according to the friction and gravity, I first created an empty list called points, then I used the same technique of getting the balls direction when being shot and put it into my update method, this in theory should allow me to keep adding the position + the velocity to the points list and then adding the gravity and friction to the velocity, however when I run this code it did not run, I have no idea why because I thought it would but I will do a print statement to see what the **self.points** list looks like:

I got this chaotic mess I think something messed up because all the points alternate between **47.69554195180687** and **93.030548217498**

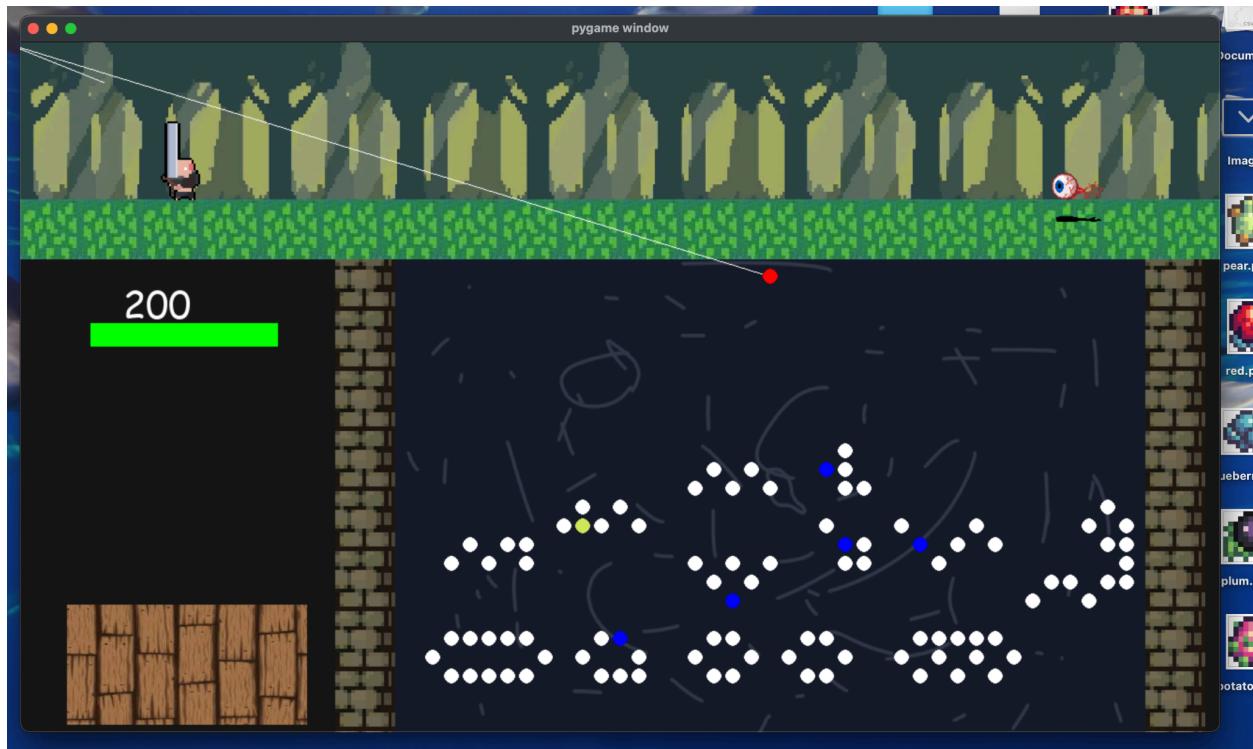
Beside significantly lagging my game and causing it to slow down, the lines were actually being drawn and my mouse was actually influencing the way that they were being drawn:



Next, I used the python function `.clear()`, to clear the points list so that the trajectory isn't drawn all over the screen and I got a singular dot that followed my mouse(very hard to see):



Next I used the `.copy()` function to create a copy of the position list before I appended it to the points and this significantly helped(the line also moved slightly with my mouse):



I am going to do the same for the velocity list to see if it fixes the issue:



The output was very similar and not much changed

HOOOOORAYYYYY!!!! I got it to work. Firstly, I took out the loop in which I created my points into a different function called **calculate_trajectory()**. Then I called it in the update function, this fixed the issue because when the update function was repeatedly called it would change the velocity which was being used to calculate and draw the trajectory line, then by having a separate function which is called It was able to draw the trajectory and then redraw the trajectory when the function was called again(I think at least), anyway these are the outputs:

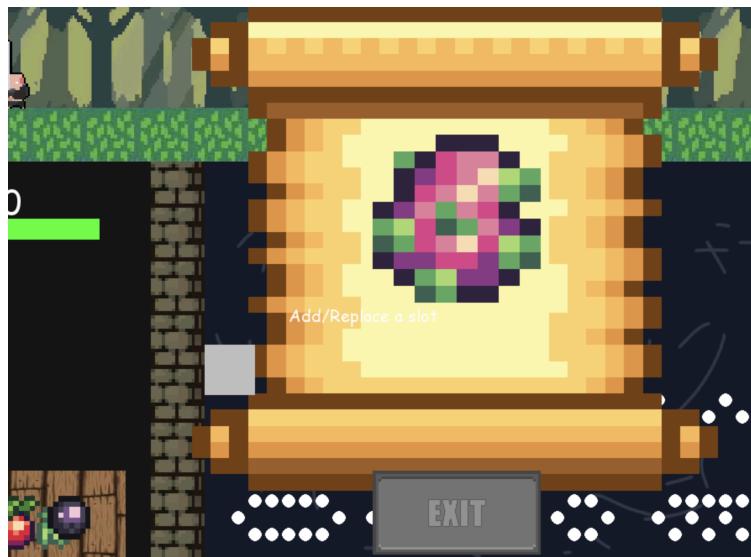


Then finally I created a variable called **self.shooting** and then created a function called **start_shooting()**, and **stop_shooting()**, these would set self.shooting to True and False respectively, then I called the **stop_shooting()** function when I shot the ball to stop drawing the trajectory.

Now I can finally officially remove that dumb cannon which I made in ms paint and was one of the first things I coded in this game. It didn't even point in the correct direction and was rotated around the corner of the image.

Entry 36:

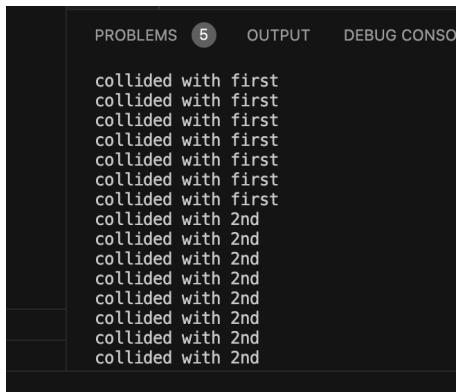
I want to add buttons to be able to replace fruits within my game, I first started by drawing a rectangle on the scroll and adding some text(it is currently barrel visible):



I then added in 3 more:



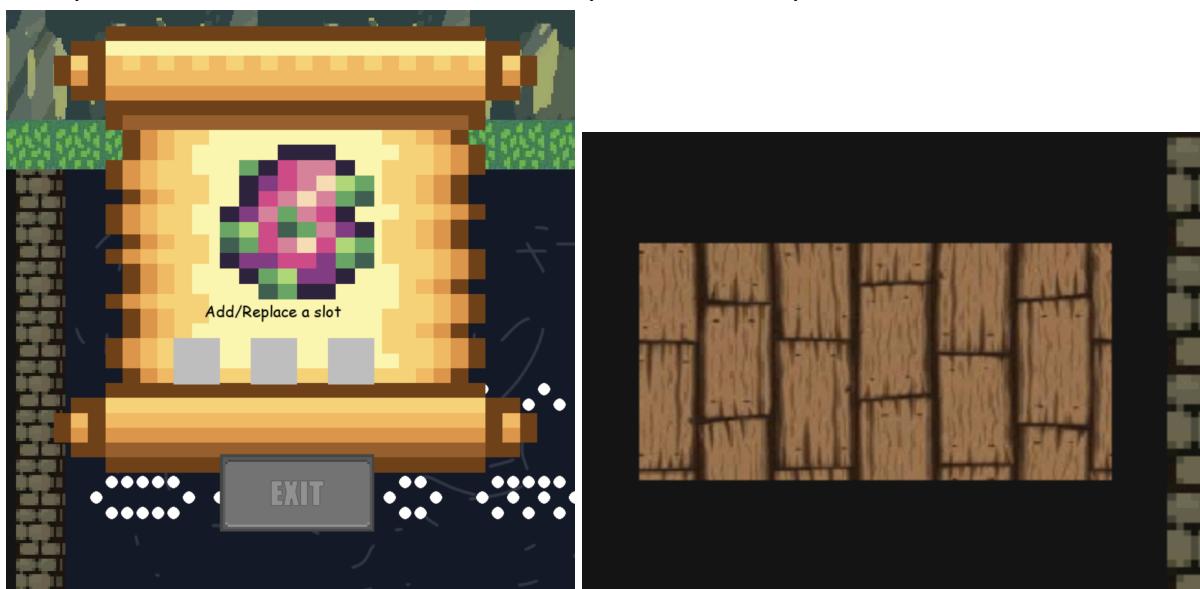
Then I checked for collisions between the mouse and the rectangles and used a print statement to test if it was working:



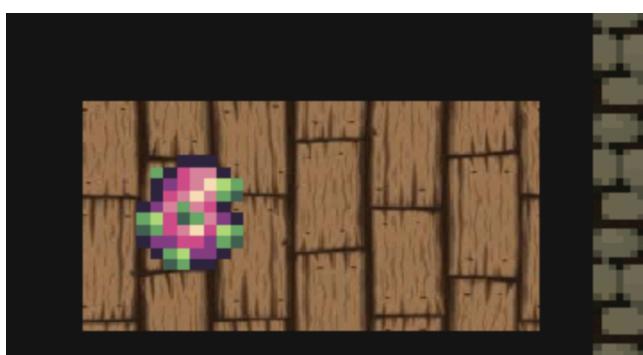
PROBLEMS 5 OUTPUT DEBUG CONSOLE

```
collided with first
collided with 2nd
```

Then I made it that when the first box is clicked the first position of the fruits list is replaced with the option above it so that the next level you enter your fruits are updated currently in the example the user has no fruits and has been presented the option to add the fruit:



When the user clicks the first slot and presses exit to the map and goes into the next level their fruits look like this:



So now the user is able to successfully add fruits to their inventory area, now I have added in the fruit system I just need to add some functionality to them.

Entry 37:

I started with creating a new file for my map just so I can have a blank starting place to rethink on how I am going to implement the hitboxes as the image part is fine. I'm going to go for a similar system. Firstly I created a nested for loop to read in the map list. Then I created a rectangle at each position in the list where the number was 1134 multiplied by the tile size which was 32. Then I had the **new_offset_x** and **new_offset_y**, this would change the location of where all the rectangles would be drawn. Then I drew the player in the centre of the screen and if the player collided with any of the boxes I would return true. Then in the main loop when I press any keys it would have a variable called new offset x and y passed into the collision method, this would then check if the player has collided and if the player has then the opposite offset is applied e.g. if the player is moving left and an offset is being added then if it is colliding an offset will be subtracted to cancel out the offsets. Then after the player has pressed the keys the offset is set to the new offset x and y. (screenshots would not really assist here as the player will just be standing still)

Entry 38:

I started redesigning the map so it looked a little bit nicer. This was through placing different types/ arrangements of objects then exporting the map again as a png. Then in my code I replaced which png was being called.

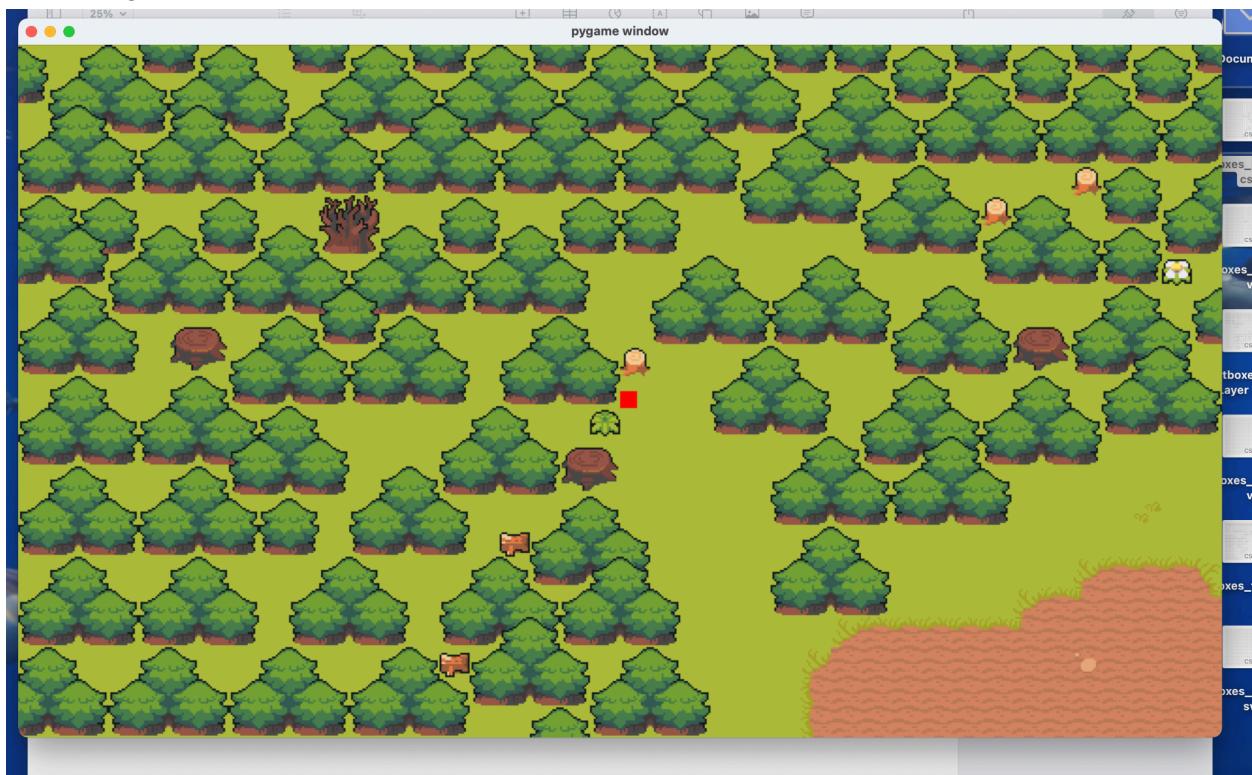
Then I had to replace the hitboxes for the new map(the grey tile represents the hitbox):



However when I ran the code my player began in the forest:



After fixing the map I had this:



Then I exported the hitboxes as a csv:

Then using the csv I created a very large list which was used to create the hitboxes in the map code.

Entry 39:

Next I fixed the level selector as, yes it was creating a new instance with the health passed into it, the level layouts were always the same.

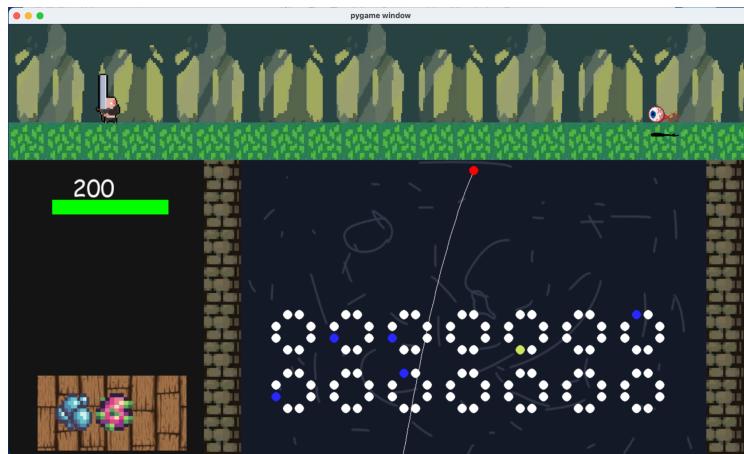
I did this by creating variables called first and second (as for now I'm only testing with two), then when the if statement is first checked for if first is false it enters the if statement and created the level. Now when you enter each level there are custom enemies

Entry 40:

There was a massive issue I ran into just as I was about to start my testing report, when I entered into different levels while they did have different layouts and enemies the health and fruits were not being passed in correctly I would end a level with one fruit from the first level and add it to the second slot however when I entered the second level it was not there:



This was very frustrating as I couldn't see why the fruits and health were not being passed in correctly, I first thought that maybe it was due to it passing in the same stuff from the first level so I hard coded it so that the 3rd level was definitely getting the fruit passed in from the second level however this did not fix it. It only worked when there was a single level being appended but when I had a single level being appended I wasn't able to have custom enemies and layouts for each level. It took a while to come up with a solution but instead of having different levels appended and using a if statement to decide which level to appended instead I had only one level being appended but the list containing the layouts and enemies would be increment instead this finally fixed my issues:



Entry 41:

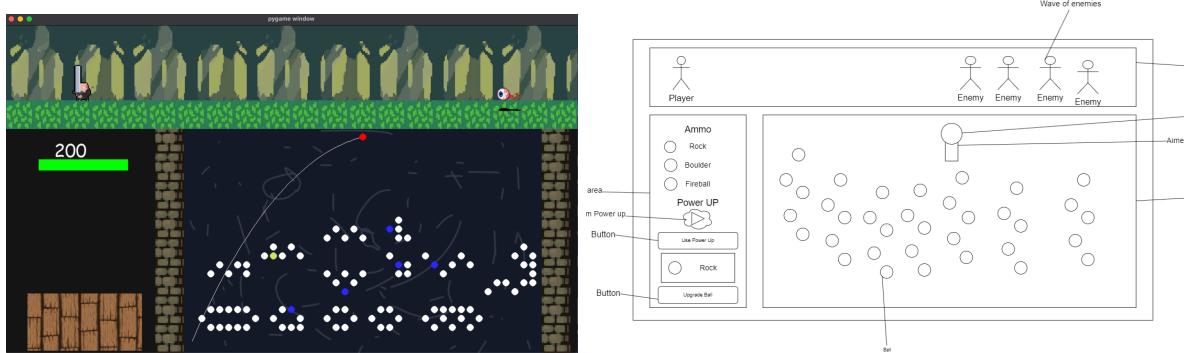
I had to make a last minute fix as I was going through the testing phase of my program. I remembered an old issue I forgot to fix was that sometimes the ball would just bounce up and down without moving side to side. To fix this issue I created a variable called `self.checkdir`, this would do the normal calculation of the direction that the ball would be sent in. Then i created an if statement and set it that if the x direction was 0 and the ball's velocity was either greater than 0.1 or less than -0.2 then this would mean that the ball was moving with a greater velocity and if the x direction calculated was 0 then it shouldn't set the ball's velocity to 0, instead I made it set the x velocity to a negative value of whatever it currently was and this fixed the issue.

Entry 42:

I wanted the entries to end on an even number.

Evaluation

Overall this was a successful assignment as my program was able to achieve most of my goals. I feel quite confident and satisfied with the final product for my project. My program was a game built in python using the pygame library and I was required to utilise many graphical elements as well as the users keyboard and mouse clicks in order to create an interactive game. My game ended up looking quite similar to my original plan. However my overall program lacked being customisable.



Features Implemented:

Start Menu:

- This had the main functionality of buttons that were able to take u to the instructions as well as a start and quit button

Map:

- This had a png of the map made in the software tiled and the player was able to appear to move by applying an offset to the image
- This had walls which were able to prevent the player from walking outside the map as well as through objects within the map such as house by checking collisions
- Had areas which would enter the player into levels which had to be found by exploring the map

Level:

- The level had the main mechanics of being able to deal damage based on the number of pegs hit
- Enemies moved toward the player and then dealt damage once close enough
- The ball was able to be shot in the direction of the players mouse and a line was drawn so the player could see where the ball would go
- The level had collisions between the peg and the ball

Issues/Bugs

The main bugs which I encountered was compatibility issues, when resizing my game window a lot of ui and graphical elements would just get moved more into the corner as I didn't implement proper scaling of graphical elements depending on the screen size early on which made it a lot more difficult to implement later on when most of the program had been completed. There were not many bugs that I encountered beside this.

Missing Features

During this project I believe I managed my time as effectively as I could as I implemented all the core features that I wanted and followed the gantt chart relatively well beside finishing the program a few weeks before the due date. However there were many features that I wanted to add as I created my game but ran out of time as well as being sick the weeks leading up to the due date. The features I wanted to add were firstly functionality of the fruits, currently they don't actually do anything. I had originally intended for them to work as power ups but I ran out of time to give them functionality. Then the next feature missing was an image for my player, this wasn't a big concern for me as I was unable to find a good tile set for the player however if I had more time with this project I could have created my own player using pixel art. The last feature that I wanted to add was that the offset not be applied when the players got close to the edge of the map as currently when the player gets close to the edge of the map they can see the black screen behind the map. This makes the game look slightly worse and it is quite irritating for me.

Overall I am satisfied with the final product, I'm not the best programmer so I definitely struggled at times and the final program is nowhere near perfect however for the time available I completed this task to the best of my ability.

Bibliography

Royalty-Free Sound Effects

Company: Envato Elements

Link:<https://elements.envato.com/sound-effects/game-sounds>

Accessed:28/6/24 was last time it was accessed

How was this used:

- I used this resource to be able to have sound effects within my game for example the bomb sound effect

Youtube tutorial

Author: Coding with Sphere

Link:<https://www.youtube.com/watch?v=r0ixaTQxsUI>

Accessed:N/A

How was this used:

- I used this tutorial in assisting me in creating a game state manager

Youtube tutorial

Author: Coding with Russ

Link:https://www.youtube.com/watch?v=G8MYGDf_9ho

Accessed:N/A

How was this used:

- I used this to teach me the fundamentals of how classes work
- I also used this to assist me in making a button class
- I used the free button assets but eventually replaced them

Stack Overflow

Author:TestZombie

Link:<https://stackoverflow.com/questions/29059368/add-velocity-to-a-sprite-in-the-direction-of-the-mouse-when-i-push-a-button>

Accessed:N/A

How was this used:

- I used the maths from the example he provided to figure out how to normalise a vector

The fundamentals

Author:GeekForGeeks

Link:<https://www.geeksforgeeks.org/how-to-draw-rectangle-in-pygame/>

Accessed:N/A

How was this used:

- I used this page and many other similar websites posted by geek for geeks, this taught me the basic fundamentals of pygame such as teaching me how to draw a rectangle