

For the third assignment, I will be looking to re-create the classic game Snake using the raylib framework.

My initial goal is just to make a version that works using mouse or keyboard controls. If I am able to, I'd like to also make it so the snake pulsates to the beat of a song. Nothing fancy like the pulse, beat, or song changing, but just a single track that it goes with, to add some uniqueness. However that will only be added if I feel I have time once the core game is done.

First, let's deconstruct Snake

- **Play area**

- The game is played on a largely empty square, with only 4 walls at the edge of the screen to designate boundaries. Some versions of the game may introduce a wall or block to mix things up, but for now we'll keep it simple.

- **Player**

- The player controls a snake, which generally starts as 1 - 3 circles/squares to represent the snake. They either can move in a limited manner (up, down, left, or right) or they can move in a more freeform, mouse/joystick manner.
 - For my version I'm going to attempt both versions, one simplified where you can move using the arrow/WASD keys, and one where the snake follows the mouse cursor

- **Goal**

- The goal of snake is to make the snake as long as possible. To do this, pellets are spawned one at a time at random points on the map. Only one pellet will remain on the map until captured, and then a new one will respawn.
 - As pellets are captured, the snake will grow one unit in length (circle/square)

- **Challenge**

- When a player hits the wall or runs into part of the snake, the game is over and a score is shown.
 - As the snake grows, players will have to think about how they navigate around the map to ensure they won't box themselves in with their tail or run into a wall.
 - I may look to add a random column that gets added in eventually, but we'll see.

- **Score**

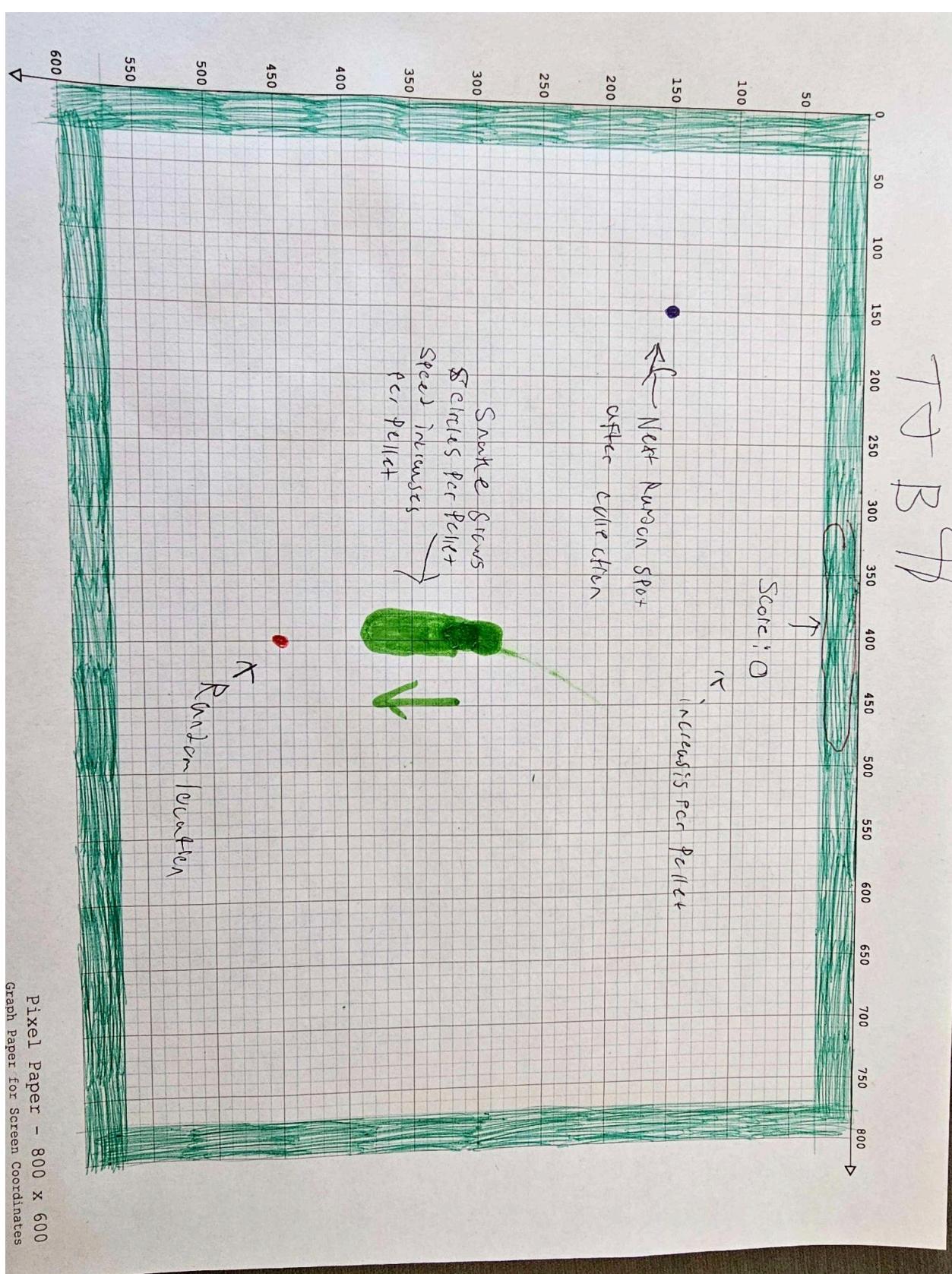
- Snake generally doesn't have a win condition, tho it technically is possible to fill the whole screen with a snake before it eats its own tail.
 - Players will aim to get the highest score, by collecting as many pellets as possible.

Alright, that's the basics of snake. Now onto coding and pen & paper notes!

Pen & Paper Notes

Game Development Fundamentals

Game Process Doc



Game Development Fundamentals

Game Process Doc

Wkref!

After ~~too~~ doing research to find ways down, I got it working

The only problem now is my snake grows very, very slowly

$O \rightarrow O \rightarrow O \rightarrow O$

Current way ↗

$O \rightarrow O \rightarrow O \rightarrow O$

Desired way ↗

I think my solution may involve frame rate, so going to investigate having my speed match my frames

OK! So I got the snake to grow... better but not exactly what I wanted. So now I have a new problem

My snake growth works, but since its player movement, when the snake stops, all the previous parts merge to one

Collision, I forgot to take notes but I got both snake collision & wall collision to work a few kinks I need to iron out

1. - Game breaks once player gets above 20 pts & moves back

L Flashes game over screen for a millisecond

lochs

2. - Pellets not randomizing after restart

Game Development Fundamentals

Game Process Doc

Snake Ideation

Left to Do

D - Collision

D - Duplication

Finished

■ - Score

■ - Pellets

■ - Walls

■ - Snake

■ - Snake Control

~~Duplication~~

Currently trying to do an array of Locations so the Snake lags behind.

- How do I only have
the array update slowly?

Fairly an issue where my Duplication
If statement isn't firing...

Ok so I got the Pellet collection to work again but now
I'm struggling with getting Duplication

I tried an array with no luck as it kept updating every +
the same

~~Array, List <vector>~~

This worked...ish
Now my snake has turned
into a poor snake.

Switch case
instead of if/else

Post Mortem

So first I hope my paper notes are enough, I'm not much of a drawing person and I find that when I'm thinking things over I tend to forget about writing notes down, so there are some gaps for sure!

There are two big takeaways I have from working on the project.

1. **Scope Changes:** I originally wanted to try and have the snake pulsate to the beat of music to add some flair to it. After the last week and a half of trying to put this together, I've had to adjust the scope. I've attempted to add audio into the game, but I don't have the brain capacity to figure out the pulsating before the assignment is due. So this be one of my first instances of having to reduce scope due to constraints.
2. **Tutorials help... ish:** So there were a few key moments I was stumped with my code, specifically how to use List<> and the continuous movement. To help with this I was doing digging online using Google, YouTube, and ChatGPT to explain the concepts to me. None of these was a perfect solution or offered an easy drag-and-drop method (not that I would do that!) but they each helped me understand, somewhat, the problem I'm facing and potential solutions. This let me Frankenstein my own solution together.

This project was definitely more of a beast than I thought it would be, tho part of residual exhaustion from Design Week. I'm still happy with how my version of Snake turned out, and while I don't think I'll be rushing to use Raylib again (apart from the next assignment) it did provide some interesting understanding into the more "elbow grease" method of development... tho I think I'll stick with game engines!