Final Project

Design Document

By Pierce Whitaker for CSC221 Final Project

[Introduction 2](#_Toc192066654)

[Project Functionality 2](#_Toc192066655)

[Design Process 2](#_Toc192066656)

[Project Development 2](#_Toc192066657)

[Pseudocode 2](#_Toc192066658)

[Flowchart 2](#_Toc192066659)

[UML Diagram 2](#_Toc192066660)

[Requirements 2](#_Toc192066661)

## Introduction

### Project Functionality

Here is where you should put the general workings of your game. Think about how it works (movement, points per collection, how you place items, etc.). Think about the design decisions made (can the snake eat a fruit with parts of its body that is not the head? Does a fruit spawn on Pac-man and instantly get it? Can the game run forever?)

I am making the classic 2D snake game. The game is simple, like others I want to have a grid space in a border where the snake can freely move around in using the controls WASD. The main objective being to collect the fruits that randomly generate around the map. The snake can only collect fruits with its head as its body grows from each fruit collected. Another key component of the game is that the snake can not pass through itself, so running into its own body would end the game. Lastly its important that I implement a key rule that you cannot backtrack on the same key. So if I were to go right using the D key, I cannot use the A key to go directly backwards.  
  
**Design Decisions**

So far, I have decided on a 800x800 pixel window, making the application easy to see. I also paired it with a white background as I prefer it.   
  
For the snake I made it a dark green color, just to change it up and not give it the typical rgb standard green. It has a cell size of thirty. Which paired with a decent sized play field can make for some very long games. The snake is set in the same position with each play but I hope to change that in the near future.   
  
The movement aspect so far is kind of choppy, and I hope to fix that a little as I go. I also decided on using WASD instead of the typical arrow keys just because I use a 60% keyboard when doing my work at home.

Lastly, the border is something I’m still working on. I have it set as a 700x700 bit rectangle so the player can see it clearly, apart from just the window. It instantly closes the game when hit by the snake for now.

### Design Process

Here is where you will talk about how you created your final project. This should include the reasoning behind the design decisions previously discussed (they `why` of your designs). Additionally, include any hiccups/issues, parts that went well, and parts you enjoyed in making your final project.

From the beginning I used a decent portion of the code from the demo, altering it to fit the needs of the snake game. Another issue I experienced was with the border, not really sure on how to impliement the colliderect from the demo code so I just used a less than or greater than statement to track collisions on the border. That wasn’t the only thing I did differently from the demo at the beginning as I went with cells instead to fit in with the grid.   
  
Some issues I have ran into during the coding process were a lot of errors. Mainly doing all my work in IDLE, while do-able, I found it a lot easier when opened with visual studio code. The reason why I had so many errors were because my functions werent indented correctly and didn’t match up with their parent. Secondly, I had issues with my idenitfiers. They were very inconsistent as I found myself typing them in wrong. Being the so creative coder during the start of the project I decided to type them in all different ways! So some would be capitalized, some would be lowercase, and some would even have their first letter capitalized just because I felt like it I guess. It ended up being a mess for me to correct, and I never fully straightened them out due to the process it would take.

## Project Development

### Pseudocode

This is for your pseudocode. Please provide it and explain it.  
  
// Intialize Pygame   
Set Width to 800

Set Height to 800  
Set cell\_size to 30

Make window size to (Width, Height)  
Set window title to “Snake Game”

// colors  
Set Red to RGB Tuple (255, 0, 0)

Set Green to RGB Tuple (0, 255, 0)

Set DARK\_GREEN to RGB Tuple (0, 150, 0)  
Set Black to RGB Tuple (0, 0, 0)  
Set White to RGB Tuple (255, 255, 255)  
  
// game border

Set border\_rect to rangle at (50, 50) with bit of 700x700  
Set border\_thickness to 8  
  
// snake head

Set player\_pos to rectangle to (90, 90) with size 30 x 30  
Set direct to (1, 0) // moves to the right  
  
// game loop  
Set running to True  
WHILE running  
 //control game speed  
 delay 80 miliseconds

//clear screen

fill screen with white color  
 //user input

FOR each event in event queue  
 IF event type is QUIT

Set the running to False  
 ELSE IF event type is keydown  
 IF key is W and direction is not (0,1)

Set direction to (0,-1) //Moving up  
 ELSE IF key is S and direction is not (0,-1)  
 Set direction to (0,1) //Moving down

ELSE IF key is A and direction is not(1,0)  
 Set direction to (-1,0) //Moving left  
 ELSE IF key is D and direction is not (-1,0)  
 Set direction to (1,0) //Moving right  
  
//snake head movement  
Add direction [0]\* cell\_size to player\_pos.x  
Add direction[1]\* cell\_size to player\_pos.y  
  
//check for border collision

### Flowchart

This is for your flowchart***.*** Please provide the design you based your algorithm on in the form of a flowchart as discussed in the course.

### UML Diagram

This is for your UML diagram***.*** Please provide your UML diagram (if you need to create one).

### Requirements

This is for keeping track of the requirements you fulfilled during the final project***.*** Please discuss each of the objectives/requirements listed in the final project assignment and how your final project meets/exceeds them.

My program successfully demonstrates user interaction with using WASD for movement.   
  
Processing is included with different If, elif, for, and while conditionals and functions. Uses game loop maintaining the game activity.  
  
Generates cleanly without any issues of performance. No processing time needed.   
  
Uses custom function \*\*\*  
  
Program uses pygame library to serve many different functions to draw border and player square easily.