Final Project

Design Document

By Julian Black 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

A 600 by 600 window is created with the intimal snake head, the food block, and the score. When the player enters an arrow key the snake begins moving in that direction until another arrow key is entered. When the snake head enters a food block, the player gains one block in length, and the score increases by one. The game continues until the player loses by either running into themselves or the wall, in which case the game stops and two texts appear declaring game over, showing the players score and giving two options to either restart or quit.

### Design Process

I tried to write it myself but when I really had no idea what to do, I looked over other’s tutorials and tried to work out what they were doing and why. Then I tried to quiz myself on the workings of the code in order to understand it. I also wanted the snake to not start moving until a movement key was pressed to give the player a chance to get their bearings. There was a problem where the snake’s head could sometimes appear within the body when you held down multiple keys at once. It seemed to be that the variables could be changed more than once within a single set of events. I found that if I separated the setting of the direction and the change of direction then the problem wouldn’t happen.

## Project Development

### Pseudocode

Width, Height = 600, 600

White = (255,255,255)

Block\_size = 38

Snake\_pos = ((width of screen)/2), ((height of screen)/2)

Score\_text = sysFont (timesnewroman)

While running

Score = 0

Snake\_speed = [0,0]

If the window is Xed

Quit game

If a key is pressed and the player is not moving in that direction

Set direction to ‘direction’

If direction equals a ‘direction’

Set snake\_speed to change in x y coordinates we want

If (coordinates for the snake head exist within the snake body coordinates, or coordinates for the snake head exist outside of the game window) conditions is met

Go to the game over screen

Display two messages, one saying game over and displaying the score and the other describing the options they can take during the game over screen

If the q key is pressed

Game closed

If the r key is pressed

Run the game loop

Udpdate snake to move in a particular direction, and eat food

Create a new head for the snake by adding the snake\_speed to old snake head coordinates

If the new head coordinates are the same as the food coordinates

Generate a new food object

Add one to the score

Else

Remove the last element in the snake body list

Insert the new head within the beginning of the snake body

Display all objects

fill screen with black

for each block within the snake body

draw rectangle (surface = screen, color = (0,255,0), (rectangle dimensions = x and y of each block within the snake body, block\_size))

draw rectangle (surface = screen, color = (255,0,0), (rectangle dimensions = x and y of the current food object, block\_size)

score\_text = score\_font.render(score, antialiasing = true, color = white)

screen.blit(score\_text, x+10, y+10)

### Flowchart

A diagram of a flowchart

AI-generated content may be incorrect.

### Requirements

* The game should be played on a window containing at least a 10-by-10 tile set.

The game is currently a 16 by 16 tile set

* The snake should move continuously in one of four directions (up, down, left, right).

The snake starts with no movement but as soon as the user enters an input, it moves continuously

* The player should control the direction using the arrow keys.

The player can control the snake by entering the arrow keys

* Every time the snake eats a food item, it should grow longer by one unit.

The snake can eat a food item and grow longer by one unit

* Food should appear randomly on the game screen after being eaten by the snake.

Food does randomly appear after being eaten by the snake

* The game should end if the snake collides with:
  + The walls of the screen.
  + Its own body.

The game does end when the player hits the walls of the screen and when it hits it’s own body

* Display a Game Over message when the snake dies.

A game over message is displayed when the snake dies

* Display the player's score, which is based on how many food items the  
  snake has eaten.

A score based on the amount of food items eaten is displayed

* There should be an option to replay the game after a game over.

There is an option to replay the game after a game over