Final Project

Design Document

By Annika Clark for CSC221 Final Project

[Introduction 2](#_Toc192066654)

[Project Functionality 2](#_Toc192066655)

[Design Process 2](#_Toc192066656)

[Project Development](#_Toc192066657) 3

[Pseudocode](#_Toc192066658) 3-4

[Flowchart](#_Toc192066659) 5

[Requirements](#_Toc192066661) 6

## Introduction

### Project Functionality

The player controls the direction the head of the snake moves in.

The snake moves at a set rate, and does not stop until the game ends.

If the snake collides with itself, the game ends.

If the game ends, the player can restart it.

If the head of the snake touches a fruit, the snake will grow. No other parts of the snake can "eat" a fruit.

The fruit will appear at a random location at the start of the game and after every time it is "eaten".

### Design Process

The main issue I ran into when developing this game was getting my IDE to recognize pygame. I eventually had to use a virtual environment and install pygame there.

Most of the design decisions I made were to fulfill requirements. Some, like the additional control schemes, were to make the game more comfortable to play for those who prefer alternatives to the arrow keys.

I really enjoyed figuring out the logic for how the snake would move. For a while, the game was just a program that displayed the default snake and a fruit, with no interactivity. Implementing movement was the most significant coding-specific hurdle, but it was an interesting exercise.

## Project Development

### Pseudocode

BEGIN

(initial setup)

INITIALIZE Pygame

SET grid dimensions (griddims) to 20

SET square size (gridsize) to 40

SET score to 0

CREATE game window with dimensions (griddims \* gridsize)

INITIALIZE font for score display

(rendering text)

DEFINE function draw\_text(text, font, color, x, y):

RENDER text on the window at position (x, y)

(making the snake and moving it)

DEFINE class Snake:

INITIALIZE body with starting positions

SET initial direction to right

FUNCTION draw\_snake():

FOR each segment in body:

DRAW segment on the window

FUNCTION slither(fruit\_position):

CALCULATE new head position based on direction

IF new head collides with wall OR body:

RETURN False (game over)

IF new head collides with fruit:

GROW snake and increase score

ELSE:

ADD new head to body

REMOVE last segment from body

RETURN True (snake is alive)

FUNCTION change\_direction(new\_direction):

IF new direction is not opposite to current direction:

SET direction to new\_direction

FUNCTION move\_snake(event\_list):

FOR each event in event\_list:

IF event is KEYDOWN:

CHANGE direction based on key pressed

IF event is JOYAXISMOTION:

CHANGE direction based on joystick movement

(making the fruit)

DEFINE class Fruit:

INITIALIZE fruit position randomly

FUNCTION grow\_fruit():

SET random position within grid dimensions

FUNCTION draw\_fruit():

DRAW fruit on the window

(begin setting up the actual game)

INITIALIZE clock for controlling game speed

CREATE Snake object

CREATE Fruit object

SET game\_over to False

(game loop)

WHILE True:

GET event\_list from Pygame

FOR each event in event\_list:

IF event is QUIT:

EXIT program

IF NOT game\_over:

CALL snake.move\_snake(event\_list)

IF snake.slither(fruit position) is False:

SET game\_over to True

CLEAR window

CALL snake.draw\_snake()

CALL fruit.draw\_fruit()

CALL draw\_text(score, font, color, position)

ELSE:

CALL draw\_text("Game Over", font, color, position)

CALL draw\_text("Press 'R' to Restart", font, color, position)

FOR each event in event\_list:

IF event is KEYDOWN AND key is 'R':

RESET score

CREATE new Snake object

CREATE new Fruit object

SET game\_over to False

UPDATE display

CONTROL game speed with clock

END

### Flowchart

### Requirements

The window is 20 by 20 tiles, exceeding the 10 by 10 tile requirement.

The snake moves continuously up, down, left, or right depending on input. By default, it moves right. This requirement has been exceeded by implementing a check to disallow reversing, optional WASD movement, and controller support.

The snake grows by one unit/segment every time it eats a fruit.

The fruit appears in a random, accessible position at the start of the game and after being eaten.

If the snake collides with itself or the walls, the game ends.

When the game ends, a "Game Over" message is displayed, and the player is prompted to restart by pressing R.

The player's score is displayed.