Final Project Design Document

Irfanullah Ahmadi

Course: CSC221

Date. 4/19/2025

Introduction

* A simple Snake game implemented using Python and the pygame library.
* Users control a snake, eat food, and grow longer, avoiding collisions with walls and itself.

Background/Problem Statement

* Classic arcade game that introduces fundamental programming concepts (loops,

Conditional, collision detection, etc.).

Scope

* Basic Snake gameplay with a score counter.
* In-scope Snake movement, food generation, collision detection, score tracking.
* Out-of-scope: Advanced AI opponents, complex game modes, effects.

Project Functionality and functional requirements:

* Snake movement: The Snake should move continuously in a specified direction (up, down, right, left) based on user input.
* Food Generation: Food should appear at random locations on the game grid.
* Snake growth: When the snake eats food, its length should increase.
* Collision Detection: Detect collisions between the snake and the wall, and the snake and itself. Trigger game over if a collision occurs.
* Score Tracking: Maintain and display the player’s score (e.g., increase score when food is eaten).

Non-functional requirements

* Performance: The Game should run smoothly at a reasonable frame rate.
* Usability: Controls should be intuitive and responsive

Design Process

* Game Grid: Use a grid-based system to simplify snake and food positioning.
* Snake Representation: The snake is represented as a list of connected blocks/ rectangles.

Project Development

* phase 1: initialize pygame, set up game window, define colors, snake variable.
* Phase 2: implement snake movement and food generation.
* Phase 3: Implement collision detection and snake growth.
* Phase 4: implement score tracking and game over logic.

Development Tools

* Python
* Pygame library
* Text editor or IDE (VS Code, PyCharm etc.)

Pseudocode

* Key movement:
* Snake movement:

Flowchart

Overall Project Flow:

* You’ll need to draw the flowchart here. Start with ‘initialize Game’, then show the main game loop: Snake Movement ->Collision Check ->Score Update -> Render, and Game Over condition.]

Snake Movement

Is it eating food ?

Rectangle game loop(while not game over)

Score update

Rectangle Handle events

YES NO YES NO

Generate New food, increase snake length

[Diamond] Restart/ Quit? [Diamond] Eat food?

Yes NO Yes NO

Generate New food, increase snake length

[Rectangle] initialize Game

[Rectangle] Draw Game

[Rectangle] Control Game Speed

[Back to Game loop]

1] End Game (Quit)

UML Diagram

* For a simple UML Diagram Class, define the classes you use (e.g., snake, Food) with their attributes and methods, and relationships.]