



LOVELY
PROFESSIONAL
UNIVERSITY

Transforming Education Transforming India

Interim Python Project Report on **Snake Game**

Date of submission: 18th November 2024

Submitted by: Varshith

Roll No.: K23CHB43

Reg.no.:12301407

Course Code: CSM216

Section: K23CH

ACKNOWLEDMENT

I would like to express my sincere gratitude to my mentor, Aman Kumar Sir, for their invaluable support and guidance throughout this project. Their expertise in data structures and algorithms was instrumental in help in game understand the principles needed to build an effective scheduling and reminder system.

I am also thankful to my friends for their constructive feedback, which helped refine the functionality and design of the program, ensuring a user-friendly experience.

Lastly, I am grateful to **Lovely Professional University** and **Upgrad** for providing a conducive environment and access to learning resources, which made this project possible.

INDEX

s.no.	content	Pg.no.
1.	Introduction	
2.	Objective and Scope	
3.	Application tools	
4.	Flow Chart	
5.	preview	
6.	conclusion	
7.		

INTRODUCTION

The Snake Game is a timeless classic that has been a staple of video gaming since its introduction in the late 1970s. Originally gaining widespread popularity with its inclusion on early mobile phones, the game continues to captivate players of all ages with its simple yet addictive gameplay. The objective of the Snake Game is straightforward: guide the snake to eat as many fruits as possible while avoiding collisions with the walls or the snake's own body.

This project aims to bring the nostalgic charm of the Snake Game into the modern era using Python and the Turtle graphics library. Python is a versatile and powerful programming language known for its simplicity and readability, making it an excellent choice for developing a game that is both fun to play and educational for budding programmers. The Turtle graphics library, included in Python's standard library, provides an easy-to-use interface for creating graphical applications, allowing us to focus on the game's logic and design.

Features of the Snake Game

1. User-Friendly Interface:

- The game starts with a main menu created using Tkinter, a Python library for creating graphical user interfaces. The main menu provides options to start the game or exit the application, ensuring a smooth and intuitive user experience.

2. Difficulty Selection:

- Before the game begins, players can choose from three difficulty levels: Easy, Medium, and Hard. This allows players of all skill levels to enjoy the game, with each difficulty level adjusting the speed of the snake to provide an appropriate challenge.

3. Dynamic Gameplay:

- The core gameplay involves controlling the snake to collect fruits that randomly appear on the screen. Each time the snake eats a fruit, it grows longer, and the player's score increases. The game's speed gradually increases as the snake grows, adding to the challenge.

4. Score Tracking:

- The game features a scoring system that displays the player's current score on the screen. The score is updated in real-time as the snake collects fruits, giving players immediate feedback on their performance.

5. Pause Functionality:

- Players can pause the game at any time by pressing the spacebar. A "Paused" message is displayed on the screen, and the game resumes from where it left off when the spacebar is pressed again.

6. Game Over Screen:

- The game ends when the snake collides with the walls or its own body. A game-over screen displays the player's final score and offers an option to start a new game or exit.

Educational Value

This project is not only a fun game but also an excellent learning opportunity for anyone

interested in programming and game development. It covers a range of fundamental programming concepts and techniques, including:

- **Event Handling:** The game responds to user inputs (keyboard events) to control the snake's movement and manage the game's pause state.
- **Graphics and Animation:** Using the Turtle graphics library, the project demonstrates how to create and manipulate graphical elements on the screen.
- **Collision Detection:** The game includes logic to detect collisions between the snake and the walls, the snake and the fruits, and the snake with itself.
- **Game Logic:** The project's main game loop demonstrates how to manage game state, update game elements, and implement game rules in a structured manner.

By working on this project, developers can gain hands-on experience with these concepts, enhancing their programming skills and understanding of game development.

OBJECTIVES & SCOPE OF PROJECT

The primary objectives of this Snake Game project are as follows:

1. Develop an Engaging Gameplay Experience:

- Create a modern version of the classic Snake Game that is both fun and challenging for players of all ages.
- Incorporate different difficulty levels (Easy, Medium, Hard) to cater to various skill levels and preferences.

2. Enhance User Interaction:

- Design a user-friendly interface using Tkinter for game menus, difficulty selection, and game controls.
- Implement intuitive keyboard controls for smooth gameplay and a pause function for added convenience.

3. Implement Core Game Mechanics:

- Develop the core mechanics of the Snake Game, including snake movement, fruit collection, and collision detection.
- Ensure the game runs smoothly with responsive controls and clear graphical representation using the Turtle graphics library.

4. Incorporate a Scoring System:

- Track the player's score in real-time as the snake collects fruits.
- Display the current score on the screen and provide a game-over screen with the final score.

5. Provide an Educational Resource:

- Offer a practical example of game development in Python, demonstrating fundamental concepts such as event handling, graphics, animation, and game logic.
- Create an engaging project that serves as a learning tool for aspiring programmers and game developers.

Scope

The scope of this project includes:

1. Game Initialization and Setup:

- Design a main menu with options to start the game or exit the application.
- Implement a difficulty selection screen with options for Easy, Medium, and Hard levels.

2. Core Gameplay Development:

- Initialize the game screen with a specified layout and design.
- Create and control the snake's movement using keyboard inputs (arrow keys).
- Implement fruit placement logic, ensuring fruits appear randomly within the game boundaries.

3. Collision Detection and Game Logic:

- Detect collisions between the snake and fruits to increase the score and grow the snake.
- Detect collisions between the snake and game boundaries or itself to trigger the game-over state.

4. Scoring and Game State Management:

- Develop a scoring system to display the current score during gameplay.

- Implement a pause functionality that allows players to pause and resume the game with a key press (spacebar).
- Display a game-over screen with the player's final score and options to restart or exit the game.

5. User Interface Design:

- Use Tkinter to create a visually appealing and user-friendly interface for menus and game controls.
- Ensure the interface is intuitive and accessible for players of all skill levels.

6. Optimization and Performance:

- Optimize the game to run smoothly without significant lag or delays.
- Ensure the game logic and graphics rendering are efficient and responsive.

3.APPLICATION TOOLS

TOOLS USED INCLUDES:

The Snake Game project uses a variety of tools and libraries to create an engaging and interactive gaming experience. Here's a detailed look at the main tools utilized in this project:

1. Python:

- **Programming Language:** Python is the primary programming language used for developing the Snake Game. Its simplicity and readability make it an excellent choice for both beginners and experienced developers.

2. Turtle Graphics Library:

- **Graphics and Animation:** The Turtle graphics library is used to create and manipulate graphical elements on the screen. It provides an easy-to-use interface for drawing shapes, moving objects, and handling animations, making it ideal for game development.

3. Tkinter:

- **Graphical User Interface (GUI):** Tkinter is a standard Python library for creating graphical user interfaces. In this project, Tkinter is used to design the main menu, difficulty selection screen, and other interactive elements of the game. It helps in creating buttons, labels, and handling user input effectively.

4. Random Library:

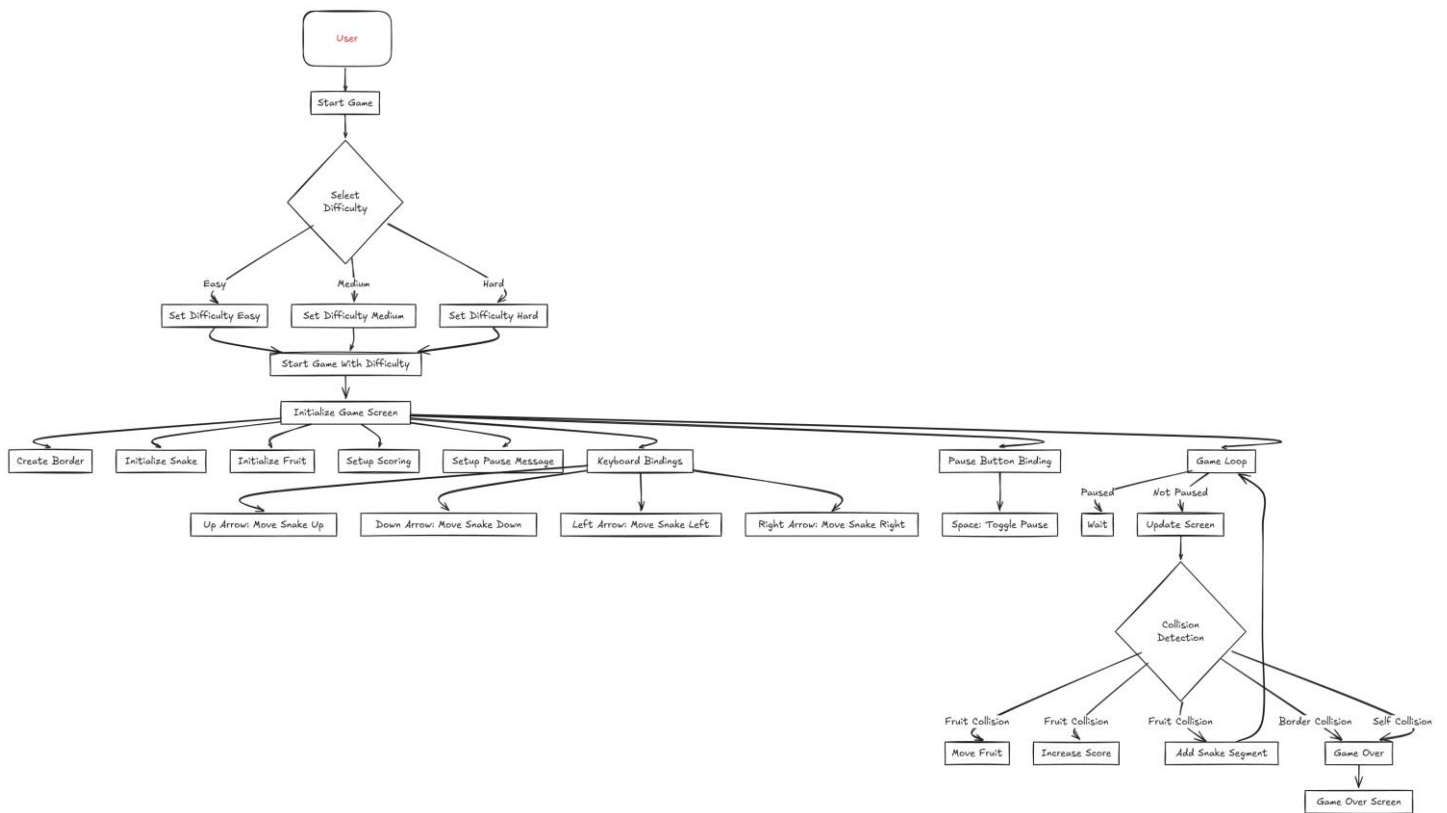
- **Random Placement:** The Random library is used to generate random positions for the fruits on the game screen. This ensures that the fruits appear at different locations each time, adding variability and challenge to the gameplay.

5. Time Library:

- **Time Management:** The Time library is used for handling time delays and pauses in the game. It helps control the speed of the snake and implement the pause functionality, ensuring smooth and responsive gameplay.

Flow Chart

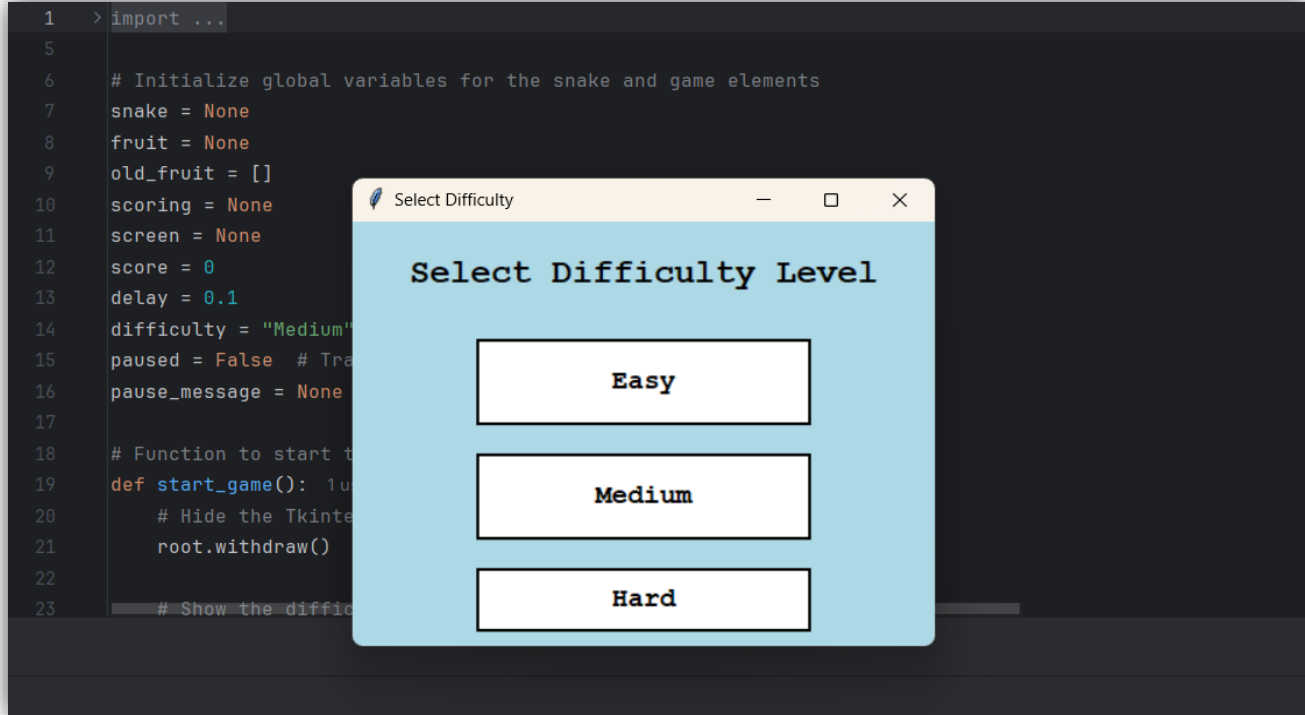
Data Flow Daigram:



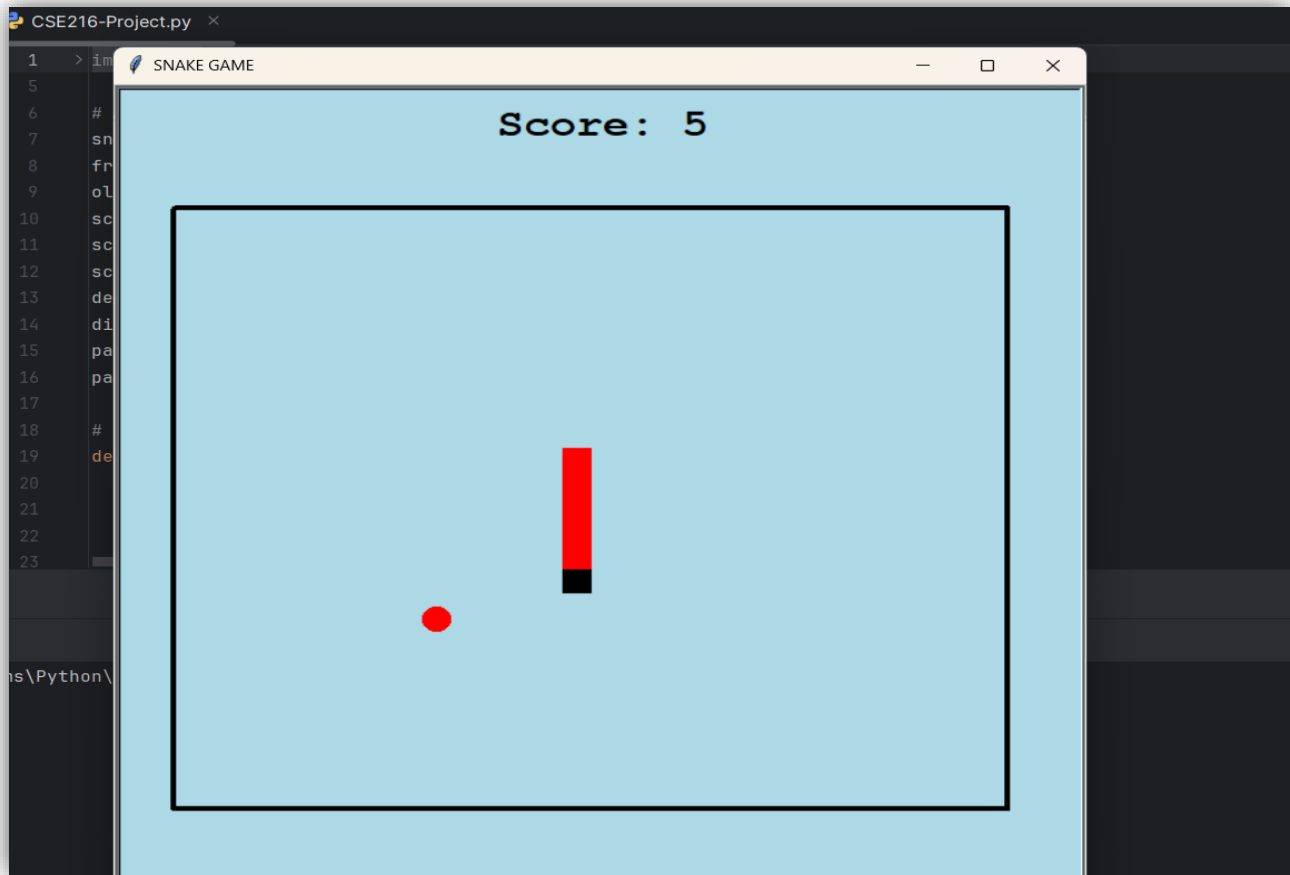
Starting page:



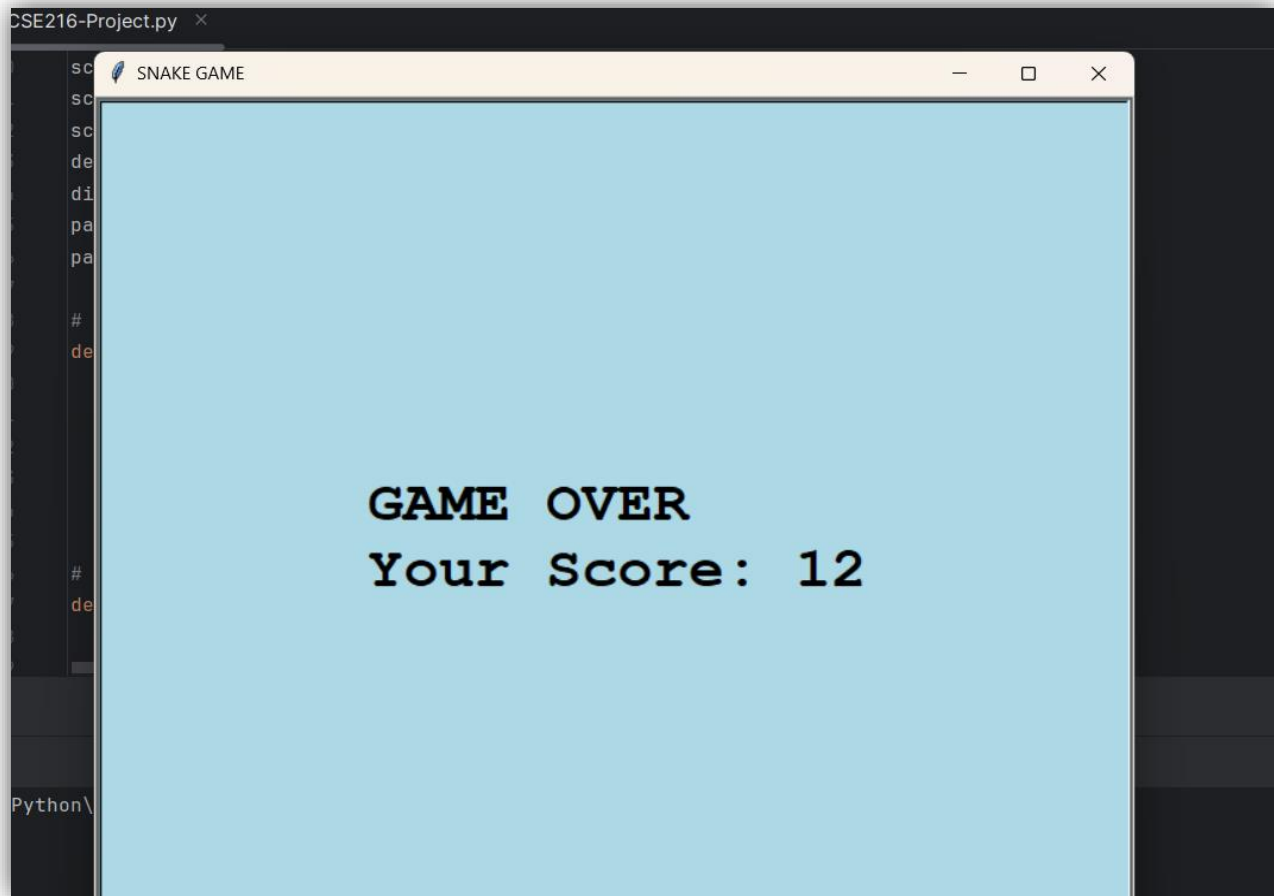
Choose Difficulty:



Snake Game :



Game Ended Page:



Conclusion

The Snake Game project successfully brings the classic and nostalgic experience of the snake game into the modern era with a variety of enhanced features and functionalities. This project not only serves as an engaging and interactive game but also as an educational tool for those interested in learning about programming and game development.

Key Achievements:

1. User-Friendly Interface:

- The game provides a smooth and intuitive interface using Tkinter, making it accessible to players of all ages and skill levels.

2. Engaging Gameplay:

- With different difficulty levels, real-time scoring, and responsive controls, the game keeps players engaged and challenged.

3. Educational Value:

- This project demonstrates fundamental programming concepts such as event handling, graphics rendering, collision detection, and game state management using Python and its libraries.
- By working on this project, developers can gain hands-on experience with these concepts, enhancing their programming skills and understanding of game development.

4. Customization and Enhancement:

- The project offers ample opportunities for further customization and enhancement. Developers can add new features, improve graphics, or integrate additional functionalities to extend the game's capabilities.