Snake Game Project Report

Project Summary

Our project focused on building a Python-based version of the classic **Snake Game**. The player controls a snake that grows each time it eats food, with the difficulty increasing as the game progresses. We implemented **score tracking**, a **local high-score board**, and **keyboard controls**.

This project gave us hands-on experience in **object-oriented programming (OOP)**, **GUI/game loop development**, and proper **Git-based version control**.

The project aligned with several core course objectives:

- **OOP Concepts:** We created Game, Snake, Food, and collision exception classes.
- **Game Loop and GUI:** Implemented using **pygame**, with initial exploration of Tkinter Canvas.
- **Version Control and Packaging:** Maintained a well-structured repository named **group 18-snake-game** with meaningful commit history to individual contributions.
- **Testing:** Incorporated multiple **pytest** cases to validate logic and ensure code reliability.

Functional and UI Requirements

We achieved the following functional requirements:

- Snake grows after consuming food.
- Proper collision detection with itself and walls.
- Real-time score updates.

The UI included:

- Grid-based game window.
- Live score display.
- Start/Pause buttons.
- A basic high-score list.

MVP and Stretch Goals

- **MVP Goals:** Fully completed a playable snake game with scoring, growth mechanics, and game-over detection.
- **Stretch Goals:** Planned but not implemented levels, power-ups, and save/replay functions. These remain opportunities for future iterations/additions.

Challenges Faced

- Balancing speed progression so that the game remained playable without being too easy or too hard.
- Fixing collision bugs in early versions where game-over states were triggered wrongly.
- Ensuring UI responsiveness, especially at higher snake speeds.

Reflection

- **Strength:** Consistent use of Git with meaningful, team-based commits.
- **Area to Improve:** UI polish and additional animations could make the game more engaging.
- **Potential Blocker:** Dependence on Python versions ≥ 3.10 may limit compatibility across some environments.

Future Work

Future improvements could include:

- Introducing **multiple levels** with unique map layouts.
- Adding **power-ups** such as score multipliers or temporary slow-downs.
- Implementing **save and replay** functionality for players to review or continue games.

Contributors

Coordinator/Lead: Uche Ebubechukwu Uche Uduma

Developers (Core Logic):

- Success Oluwafemi Ali
- Oyedokun Yusuf

QA/CI & Tests:

- Farid Abdurrahman
- Ali Garba Muazu

Docs/Presenters:

- Richmond Adoki
- Ike-Onuigbo Gideon Ikechi
- Muhammad Najibullah Abubakar
- Mohammed Mustapha Shehu

Conclusion

The Snake Game project was a valuable experience that helped us apply programming concepts to a practical problem. By combining OOP design, GUI/game loop programming, testing, and collaborative version control, we delivered a working game that met our MVP criteria. Although stretch goals remain for future development, this project provided strong exposure to both technical and teamwork skills.