

# Report Assignment 5+6 SNAKE GAME

---

## Student Name:

Olesia Mykhailyshyn

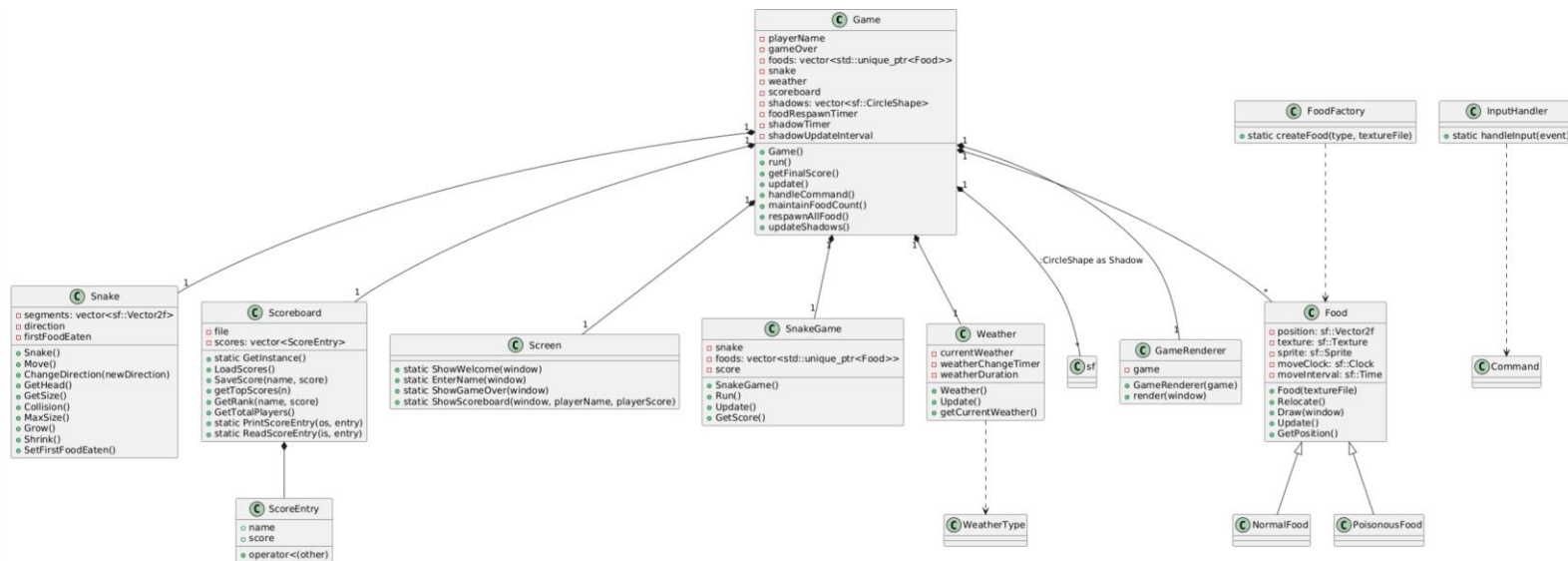
## Group:

опд 2

## Task:

The task involved implementing the classic Snake game in an object-oriented paradigm using the SFML library for visual rendering. The game was developed based on a previous design (Assignment 4) and includes unique features that differentiate it from a standard implementation. Specifically, two types of food were introduced: poisonous food (a red apple) that decreases the snake's size and normal food (a green apple) that increases it. Additionally, obstacles in the form of shaded circles were added, which periodically change their positions. A weather mode was also incorporated, where the snake moves faster during rain. Lastly, a leaderboard was implemented to save scores across games, allowing players to track their results over time.

## System Model:



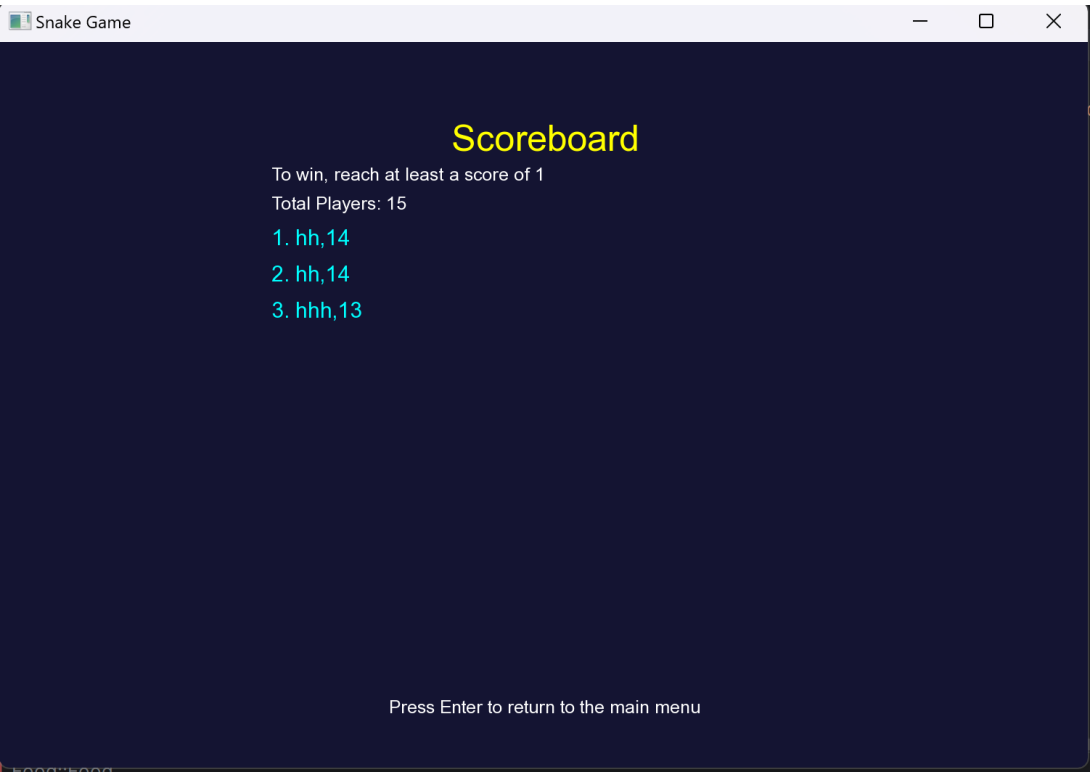
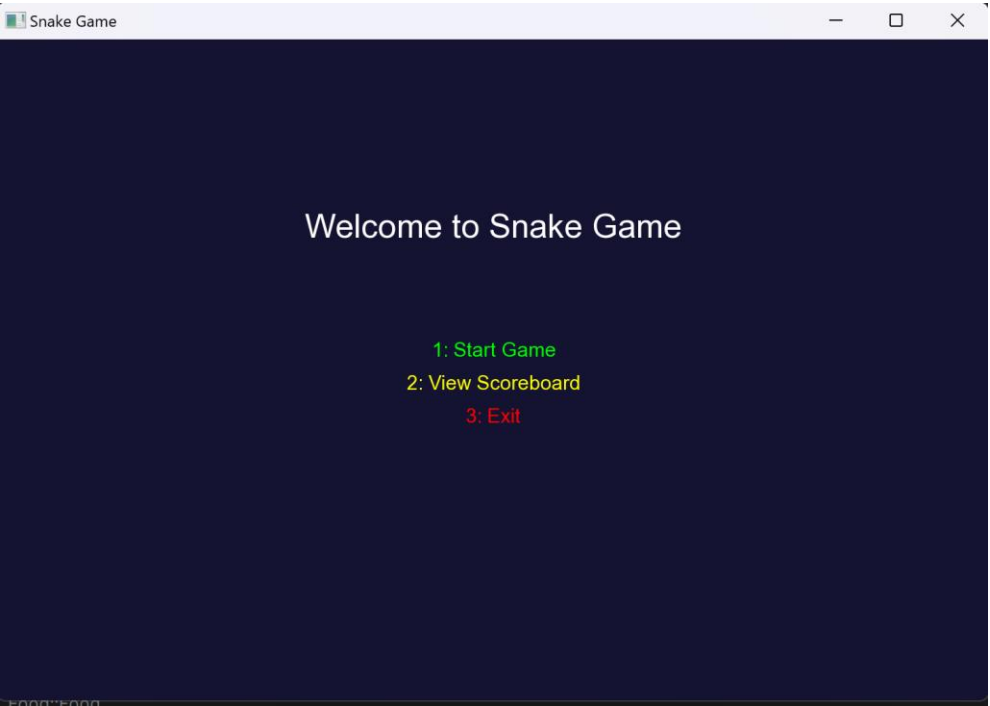
## GitHub Link:

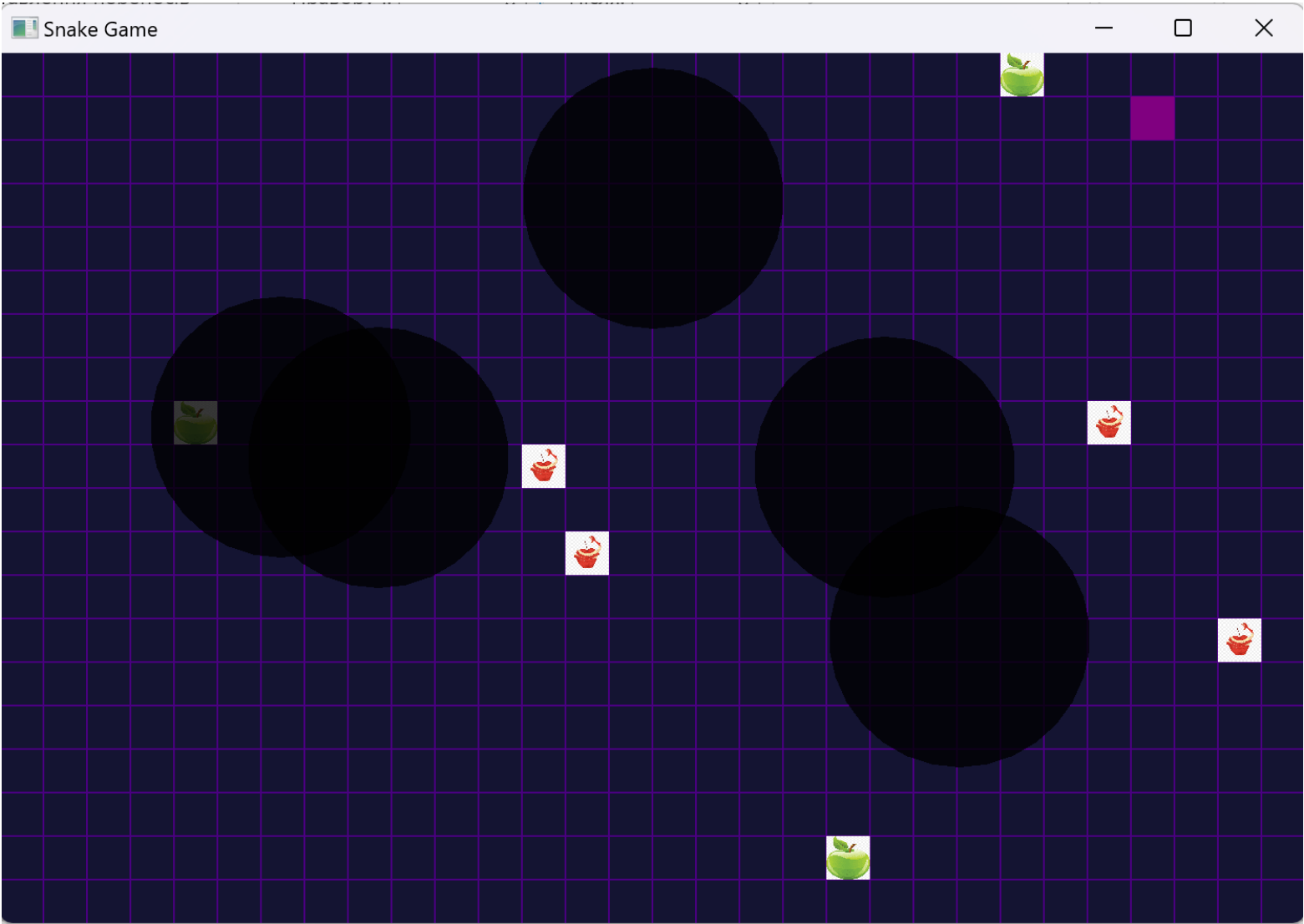
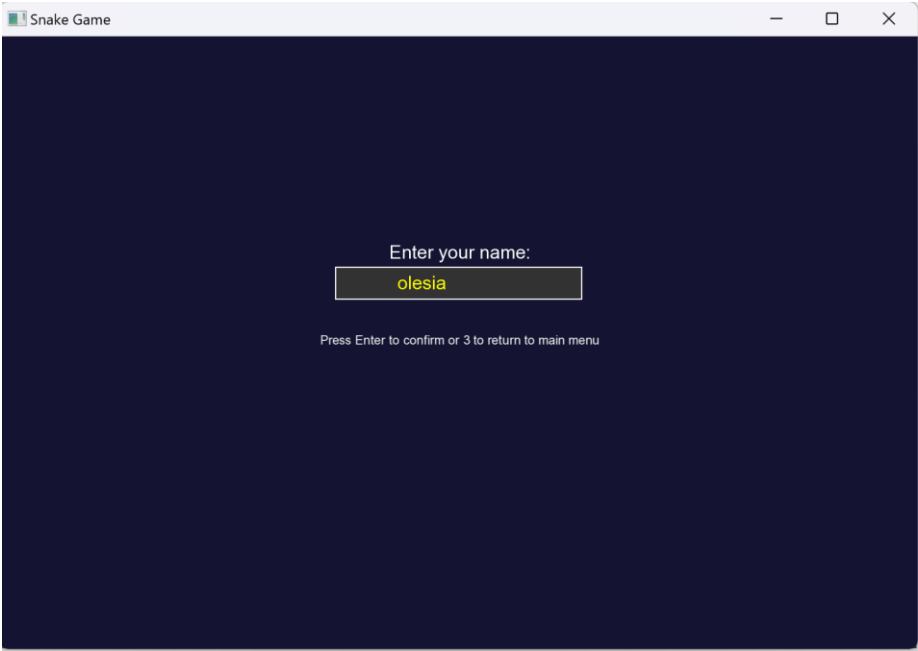
<https://github.com/VY-Assignments/asmt-5-game-engine-olesia-mykhailyshyn.git>

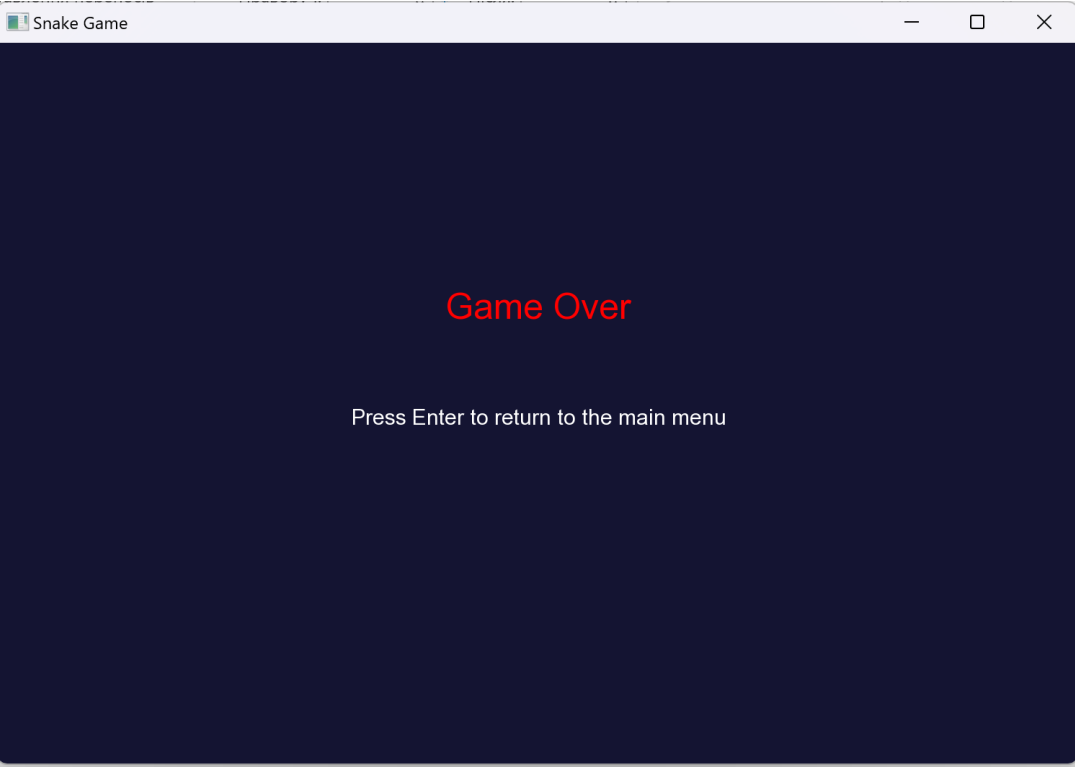
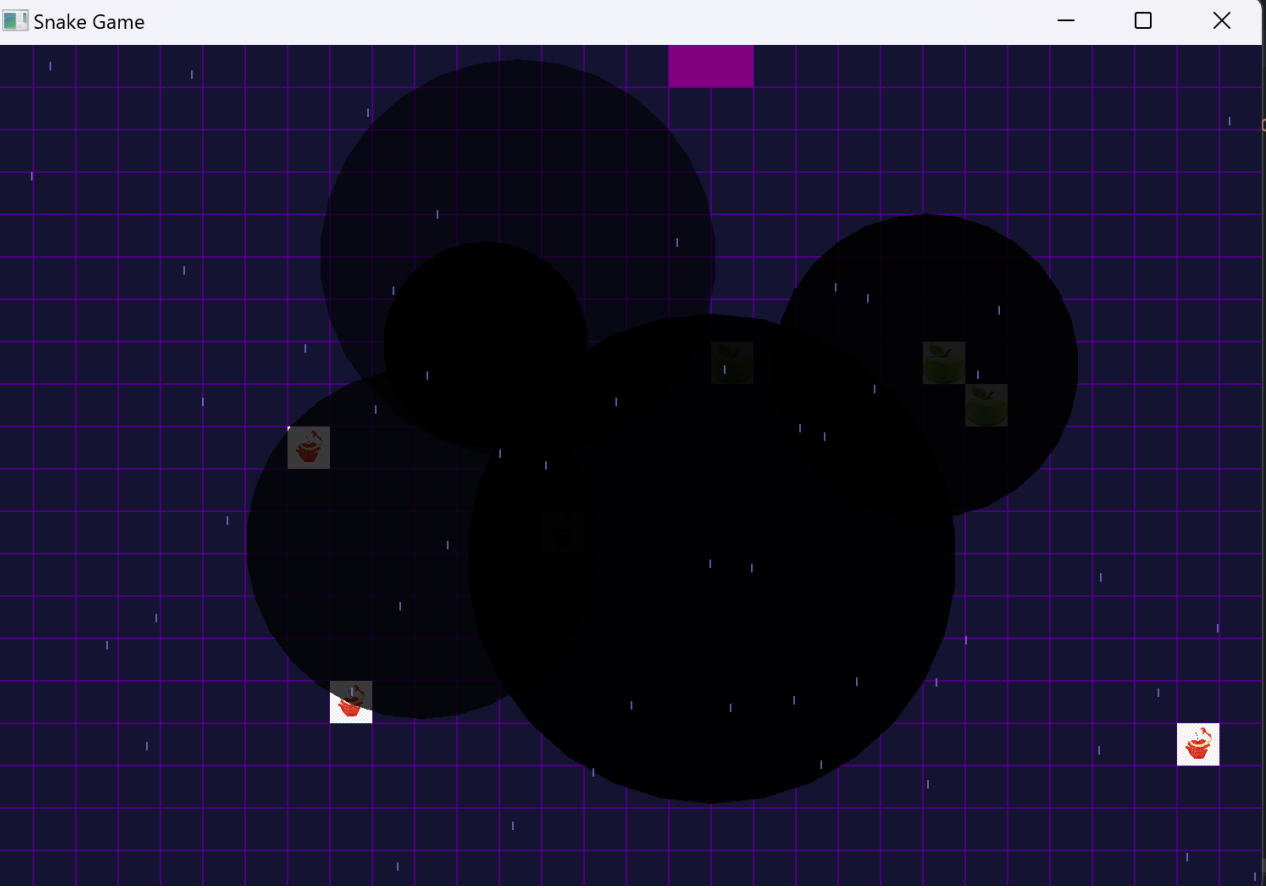
## Solution Description:

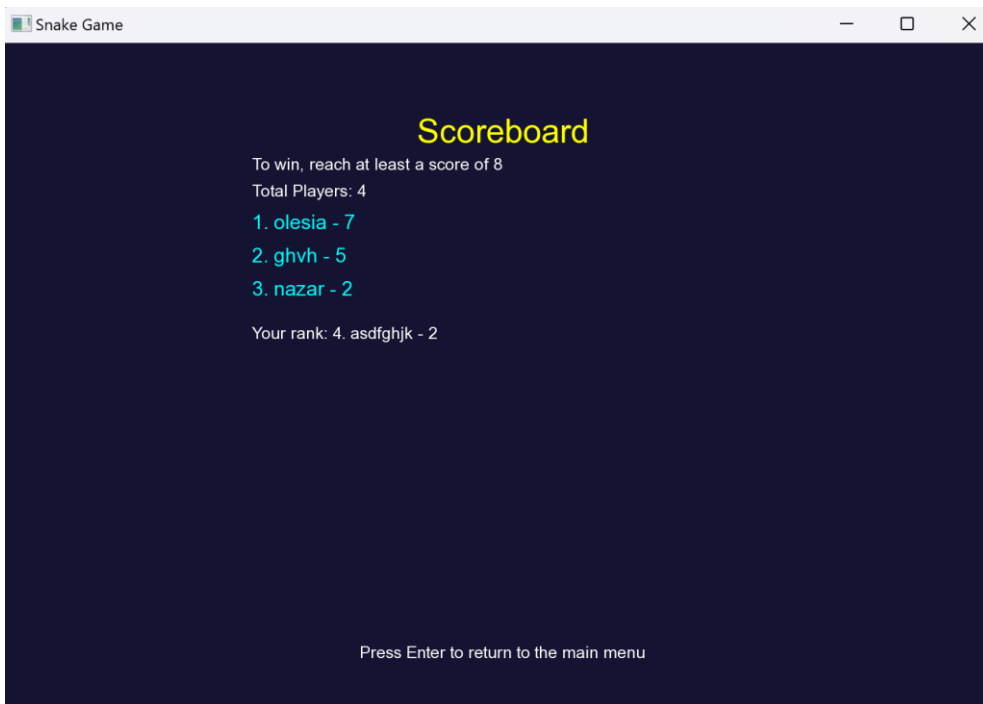
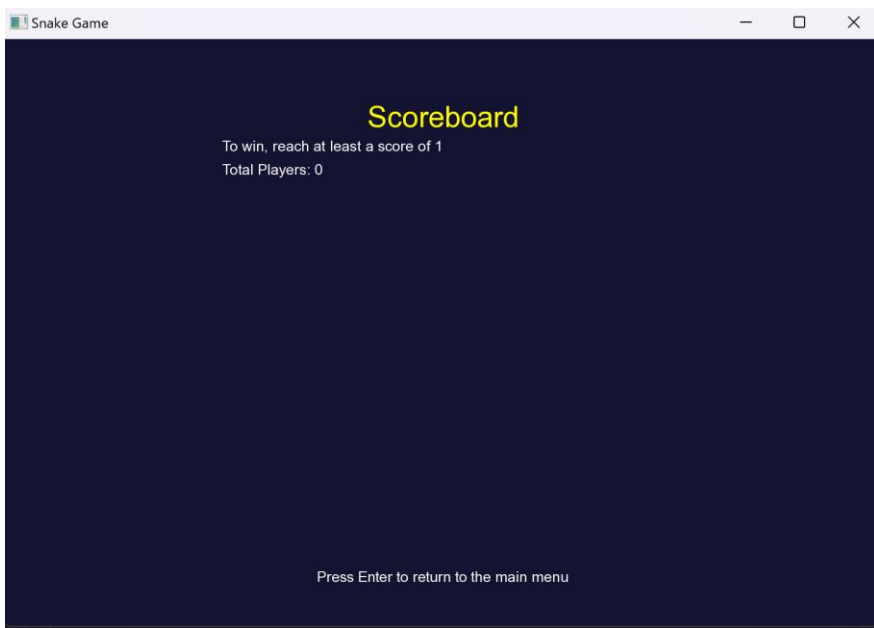
The system is structured with a **Game** class to manage the primary game loop, updates, and interactions. The **Snake** class handles movement, growth, and collisions, while **Food** and its subclasses (**NormalFood** and **PoisonousFood**) represent food items that affect the snake's size differently. Obstacles are implemented as shaded circles that periodically change positions, creating dynamic challenges for the player. The **Weather** class introduces a rain effect that temporarily increases the snake's movement speed, adding variability to gameplay. Input handling is managed through the **InputHandler** class, which processes player controls. A **Scoreboard** class, using the Singleton pattern, tracks and displays top scores, allowing players to save and view their progress across games. The codebase is modular, with each feature encapsulated in its own class, supporting readability and ease of future expansion. This structured design aligns with the object-oriented approach, ensuring organized and maintainable code.

Testing:









## Conclusions:

*What was implemented:* The project successfully implemented a unique version of the classic Snake game using SFML for graphical rendering, following an object-oriented approach. Key features include two types of food items: normal food (green apple) that increases the snake's size, and poisonous food (red apple) that decreases it. Dynamic obstacles, represented as shaded circles, periodically change their positions to add complexity to the gameplay. Additionally, a weather feature was introduced, where rain temporarily increases the snake's speed. A leaderboard was also implemented to track and save scores across multiple sessions, providing a competitive aspect to the game.

*Differences from the proposed model:* The actual implementation deviated slightly from the initial design by incorporating enhanced gameplay elements, such as moving obstacles and weather effects, which were not part of the original model. These additions provided greater depth and replayability to the game. Additionally, the Scoreboard was designed using the Singleton pattern to ensure consistency in score tracking across sessions, which was not specified in the proposed model.

*Additional Notes:* The development process involved addressing several challenges, such as synchronizing obstacle movement with gameplay updates and managing performance when rendering dynamic visuals. Implementing the weather effect required careful adjustment of the snake's speed to balance gameplay. Using SFML for graphical rendering posed a learning curve, but it allowed for smooth visual effects and enhanced interaction. The modular design of the code supports maintainability and provides a foundation for potential future extensions or improvements to the game.

## **Appendices:**

<https://www.youtube.com/watch?v=xtZHJxYA6q8&list=PLiZZKL9HLmWMF8PlzvZu2WOC9kjs1zzhm>

<https://www.youtube.com/watch?v=jyDlxqVOZMA>

*and there were many other videos and repositories I have seen to implement this game*