**Project Phantom Pursuit**

**Nishant Shah (AM.SC.U4AIE23166)**

**Ansh Bajpai (AM.SC.U4AIE23108)**

**Abstract**

Our project leverages the **A**\* algorithm and graph theory to design an exciting game where a **ghost AI hunts players** in a dynamic and interactive environment. The game features a visually engaging user interface, sound effects, and multiplayer functionality enabled through **socket programming**. Players navigate through graph nodes while attempting to outwit the ghost AI, which adapts its movements in real time. The objective is clear: **survive as long as possible** while competing against other players in a multiplayer setting.

**Problem Statement**

Today’s games often feature predictable, pre-planned enemy movements that reduce excitement for players seeking challenges. This predictability diminishes the thrill for enthusiasts who enjoy problem-solving and adaptive gameplay. Additionally, creating a multiplayer version of such games poses a significant challenge, as it requires seamless real-time interaction among players while maintaining engaging gameplay.

**Solution**

We tackle these challenges by introducing a **powerful AI-driven ghost** that uses the *A algorithm*\* to dynamically hunt players. Unlike traditional games, the ghost adapts its movements, creating a challenging and unpredictable gameplay experience. To further enhance engagement, we implement **socket programming** to enable multiplayer functionality, allowing up to three players to join and compete in a shared game session

**Features and Gameplay**

1. **Graph-based Map**: The game world is structured as a graph with nodes (locations) and edges (possible movements).
2. **Dynamic AI**: The ghost AI uses the A\* algorithm to pursue players by minimizing the distance to them, making survival a real challenge.
3. **Multiplayer Mode**: Using **socket programming**, up to three players can join the game server, each assigned a unique symbol and tracked in a live lobby system.
4. **Interactive Lobby**: Displays details about connected players, including their **MAC addresses**, names, and in-game representations.
5. **Player Controls**: Players use keyboard inputs (**W**, **A**, **S**, **D**) to navigate the map or **EXIT** to leave the game.
6. **Difficulty Levels**: Players can select one of three difficulty modes for a customized experience.
7. **Real-time Updates**: The server continuously updates and broadcasts the game state, including ghost movements, player positions, and the remaining time.
8. **Sound and Visual Effects**: Built using pygame and playsound, enhancing the gaming experience.
9. **Leaderboard**: At the end of the game, a leaderboard displays the survival times of all players.

**Implementation Details**

1. **Server-Side Functionality**:
   * A **socket-based server** handles player connections, manages game states, and ensures real-time updates.
   * The server maintains a lobby that lists connected players with their **MAC addresses** and unique in-game identifiers.
   * The **ghost AI** tracks player movements and adjusts its strategy based on proximity, speed, and difficulty settings.
2. **Player Interaction**:
   * Players send movement commands to the server using predefined keys (**W**, **A**, **S**, **D**).
   * The server processes these commands, updates the game grid, and synchronizes it across all clients.
   * Players receive real-time feedback on their actions, including the ghost’s movements and survival timer.
3. **Game Mechanics**:
   * **Game Grid**: The map is a 16x16 grid with the ghost and players starting at predefined positions.
   * **Ghost Movement**: The ghost’s movement is calculated using the A\* algorithm to minimize distance to players.
   * **Caught Players**: When a player is caught by the ghost, their survival time is recorded, and the ghost’s speed increases, heightening the challenge for remaining players.
   * **Game Timer**: Players compete to survive for a total of 60 seconds while avoiding the ghost.
4. **Networking**:
   * **Socket Programming**: Players connect to the server using TCP sockets, enabling real-time communication and game synchronization.
   * **Multiplayer Synchronization**: The server manages all player actions and broadcasts updates to ensure a smooth and consistent multiplayer experience.

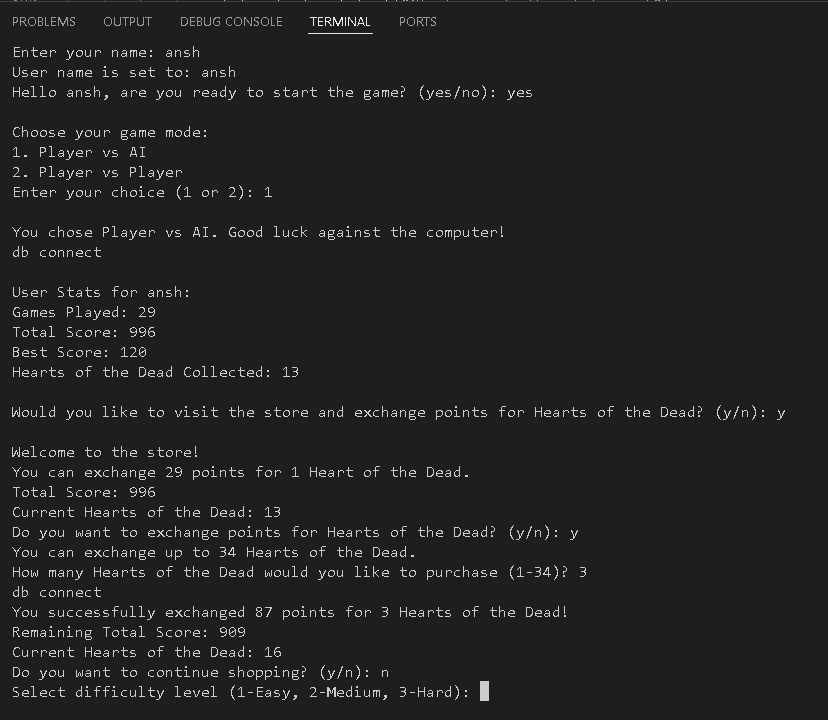
**Significance**

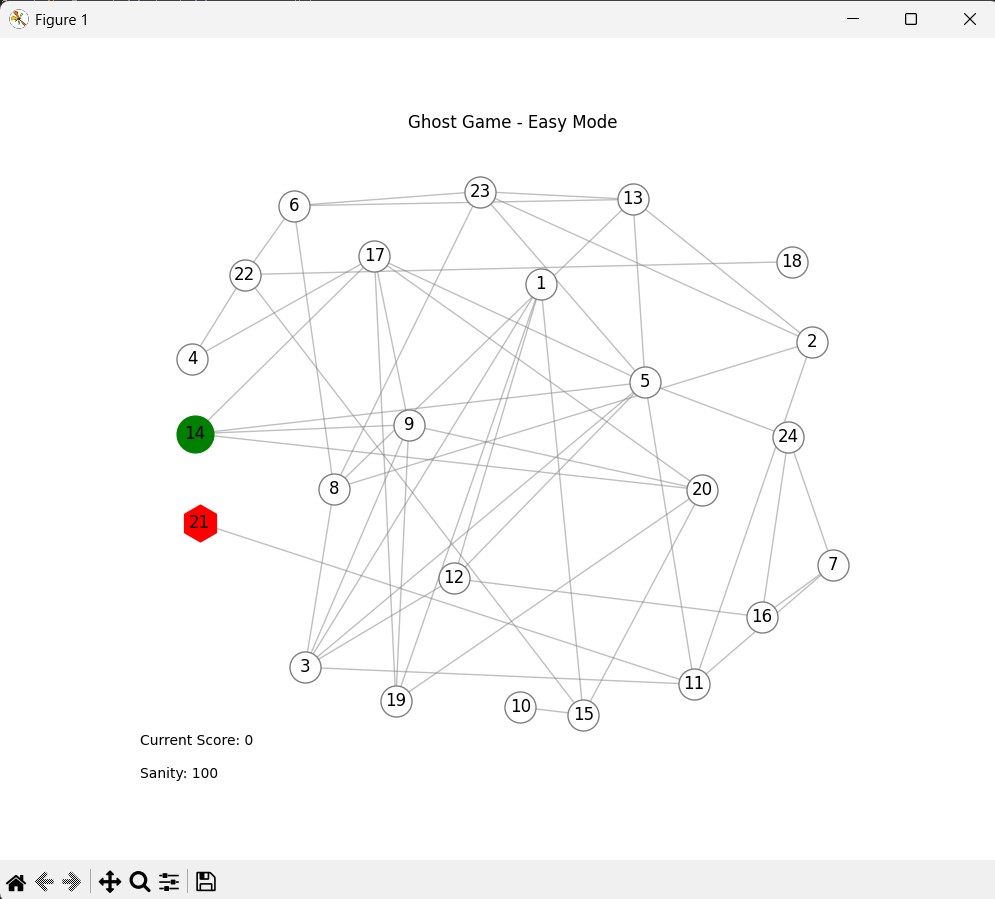
This game combines entertainment with skill-building by enhancing the player’s **decision-making abilities** and **strategic thinking**. The dynamic AI ensures every session is unique, while the multiplayer feature fosters competition and collaboration. It challenges players to adapt quickly, think critically, and refine their strategies, making it an ideal blend of fun and cognitive development.

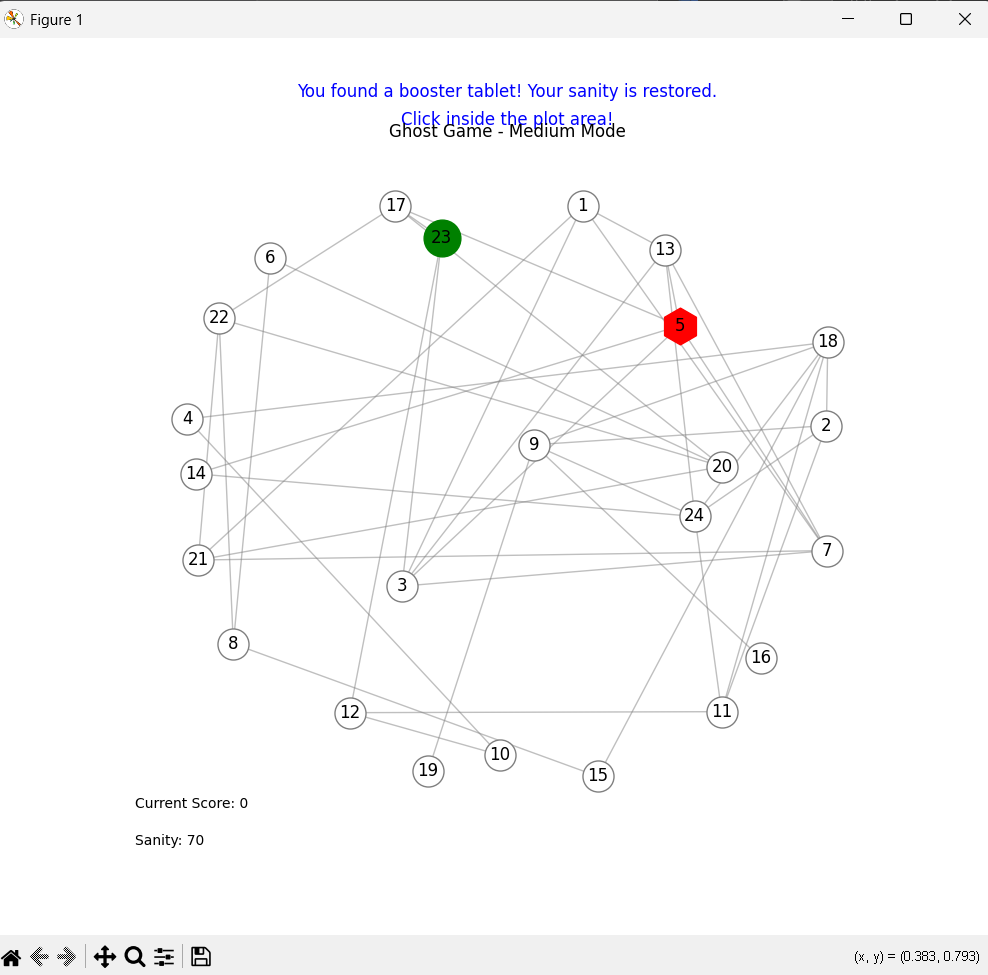
**Keywords**

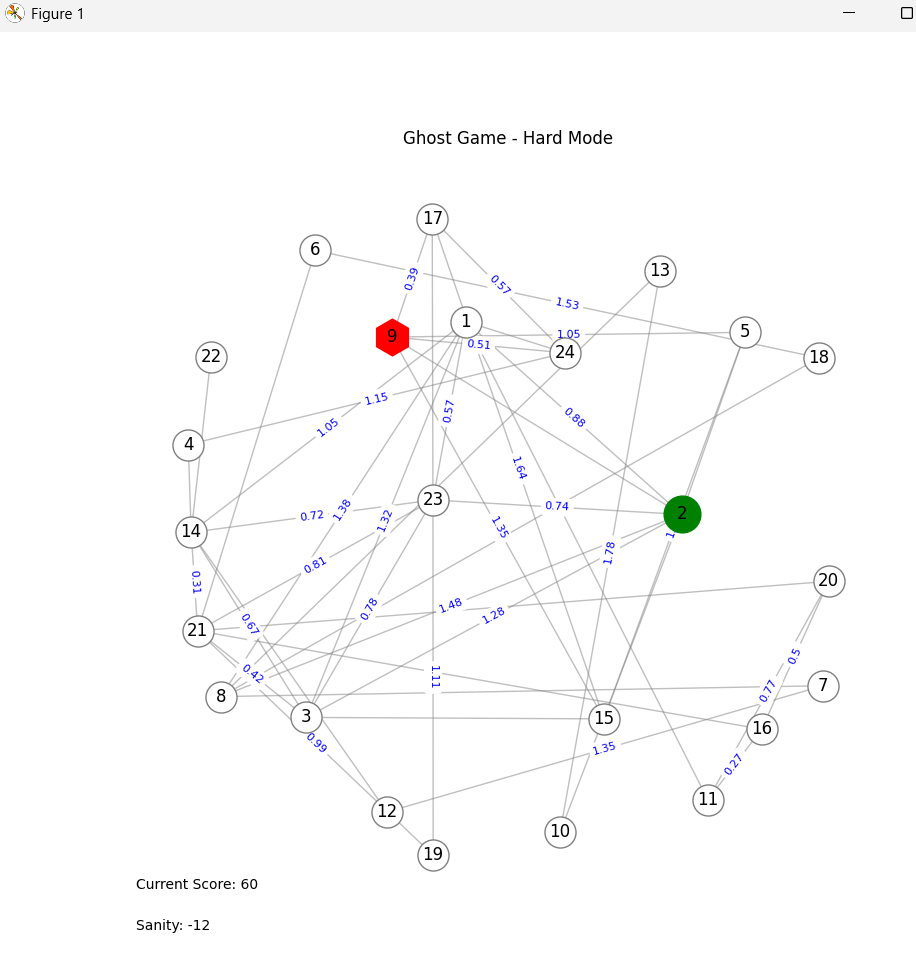
Graph Theory, A\* Algorithm, Socket Programming, Multiplayer Game, Dynamic AI, Survival Game.

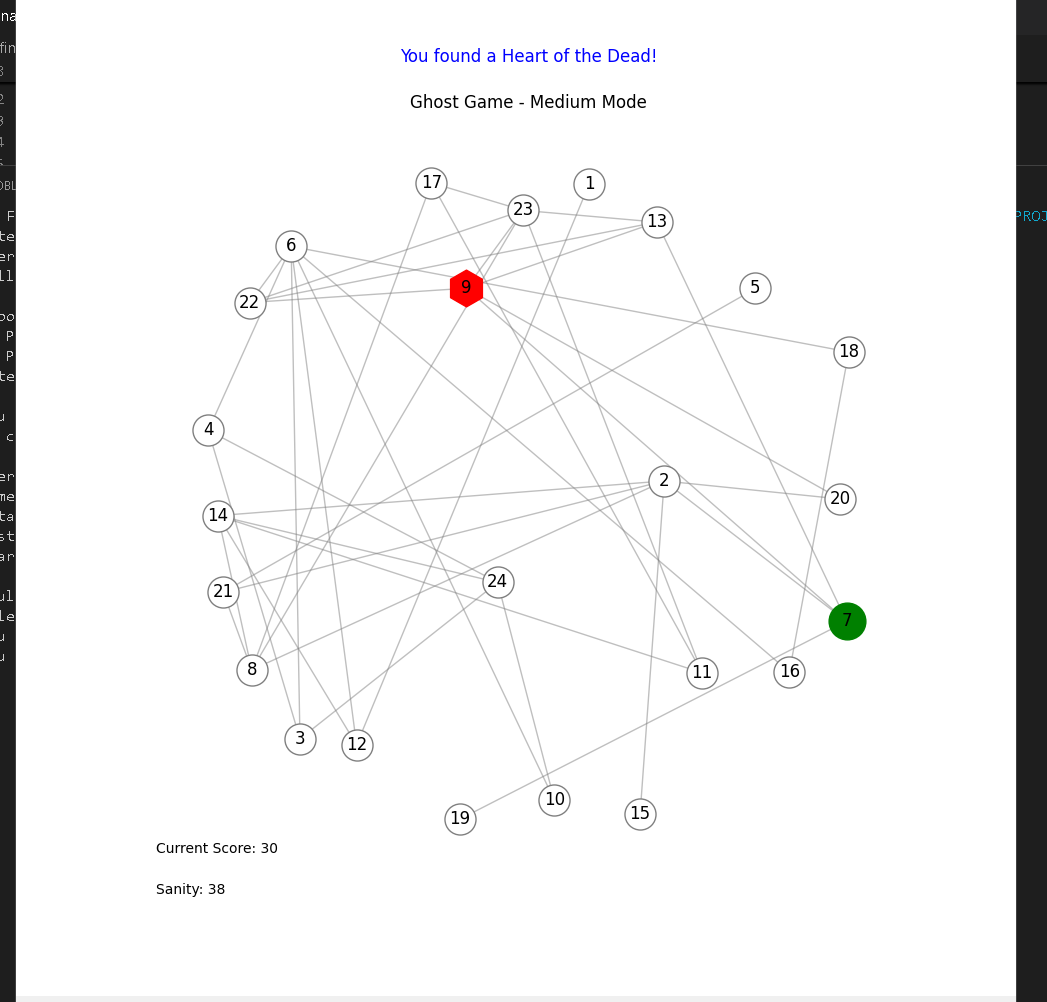


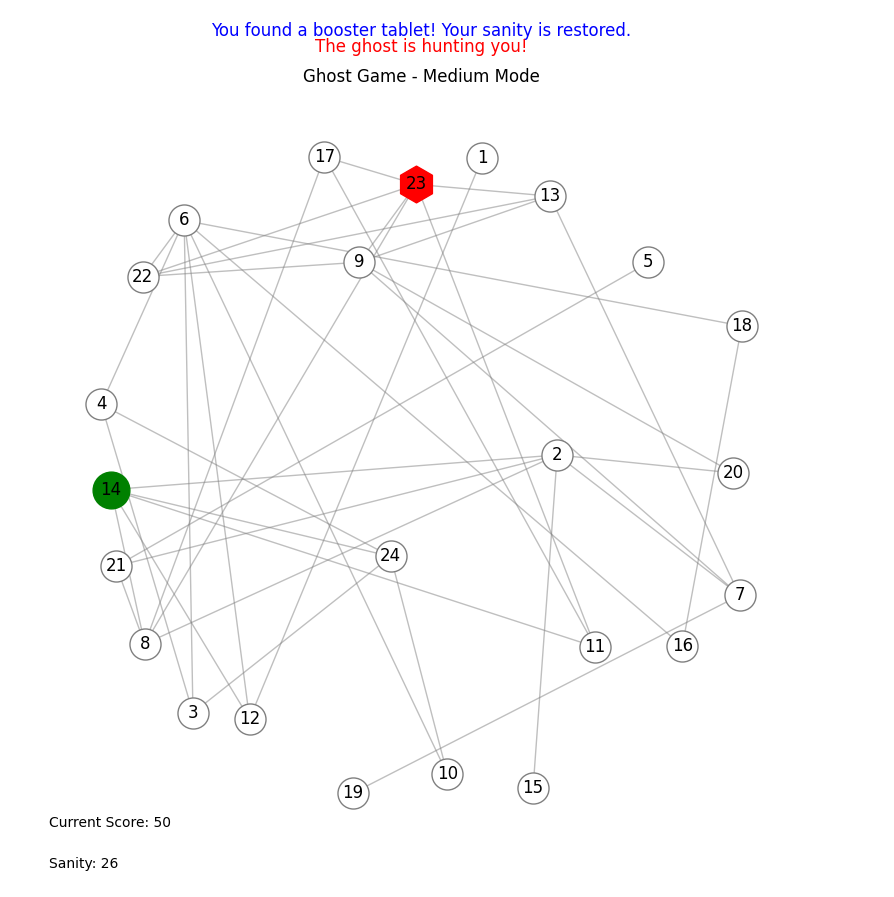
Outputs: 

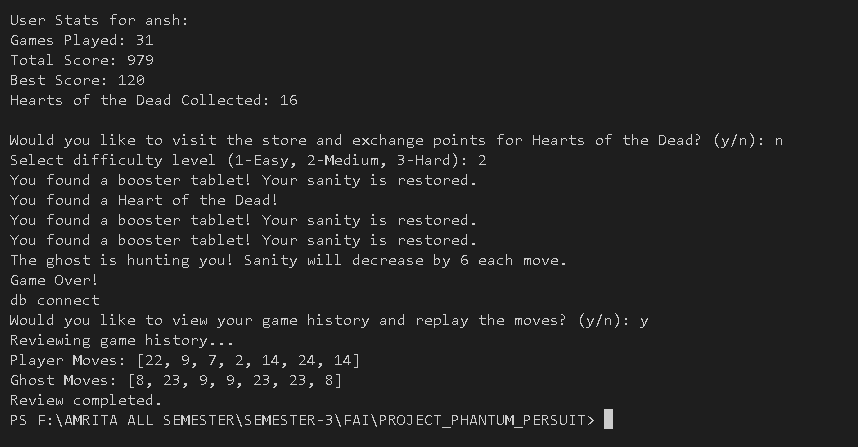












A screenshot of a computer screen

Description automatically generated

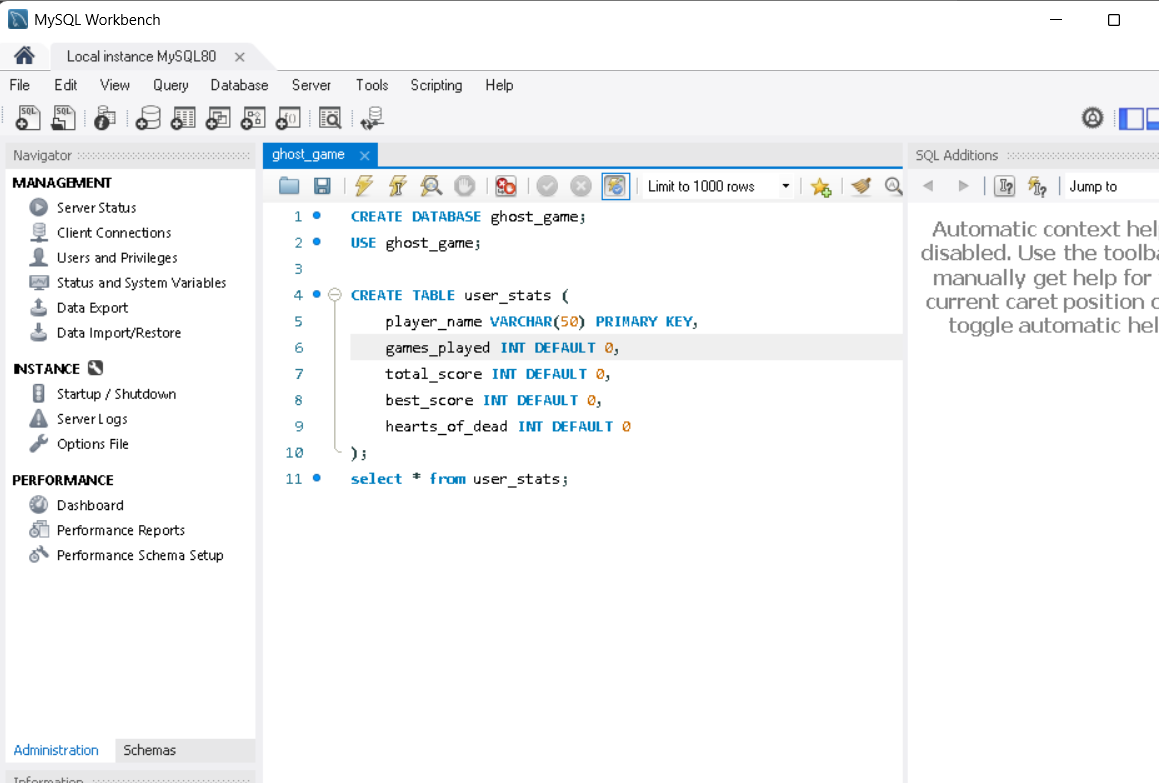
A screen shot of a computer

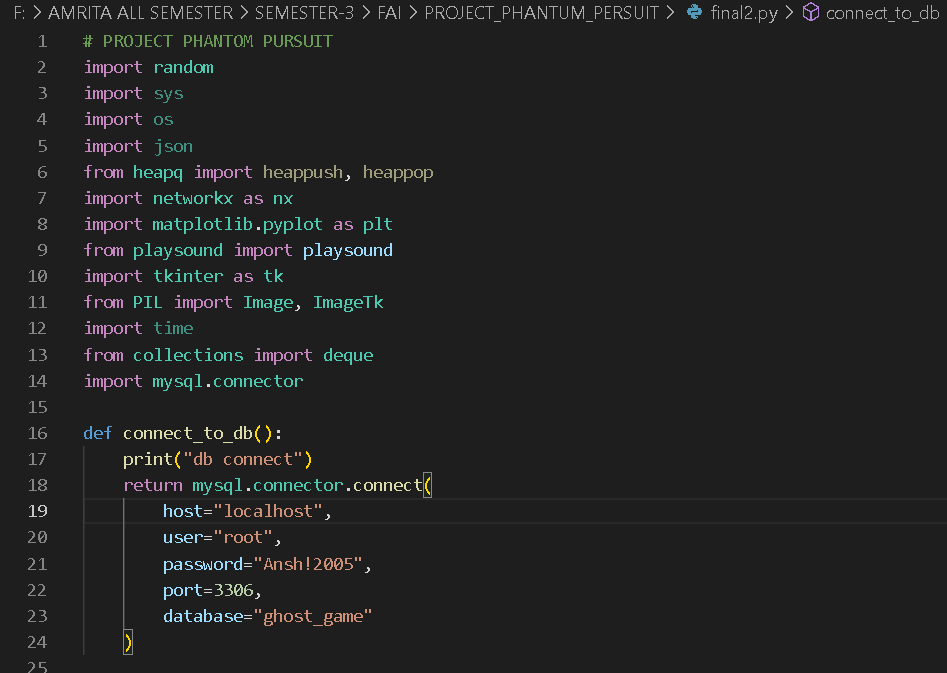
Description automatically generated

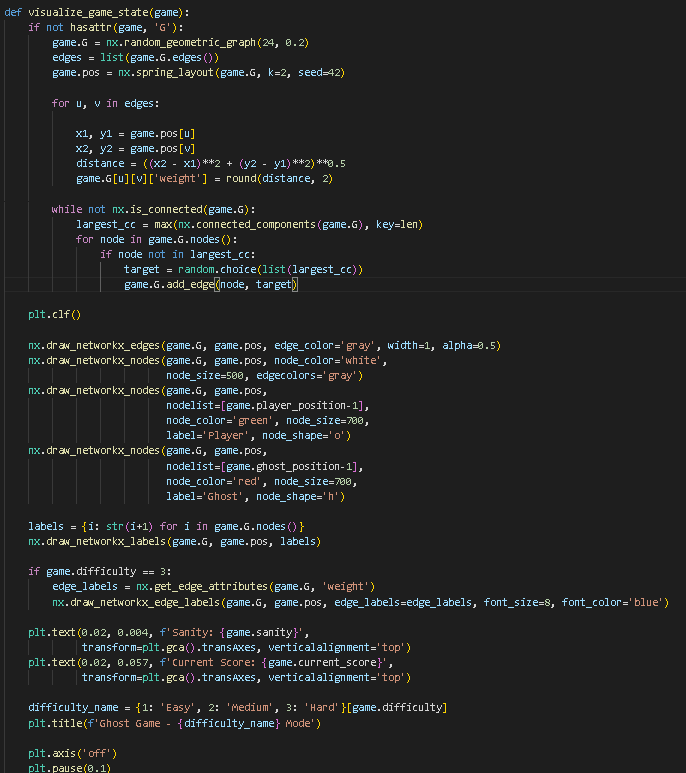
A screen shot of a black screen

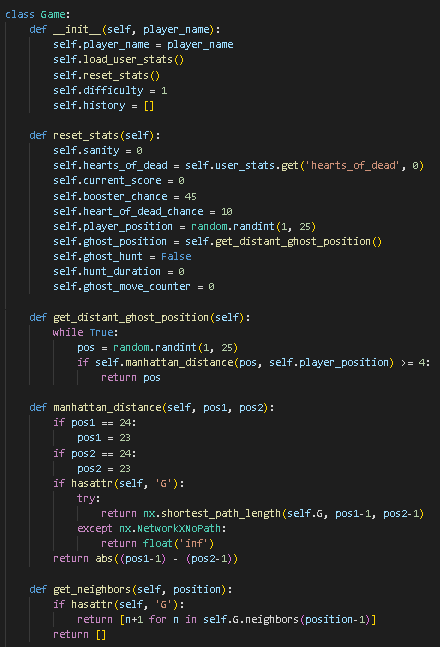
Description automatically generated

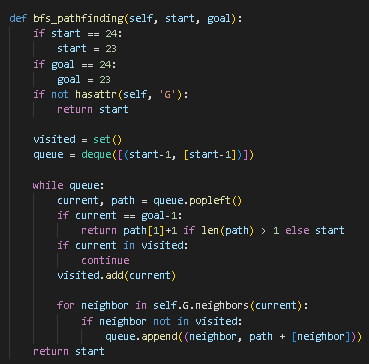
CODES AHEAD

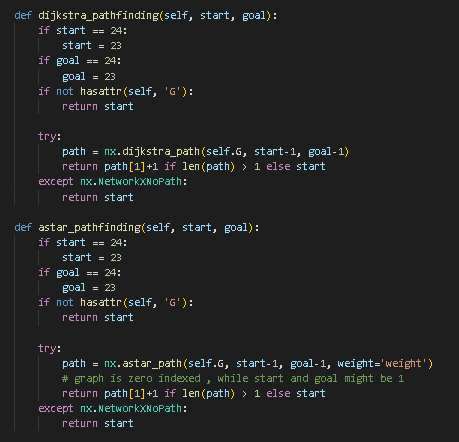
Codes: 

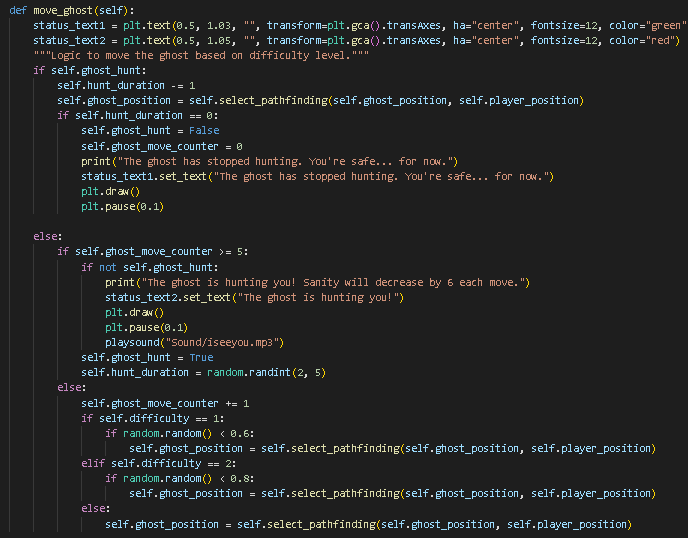


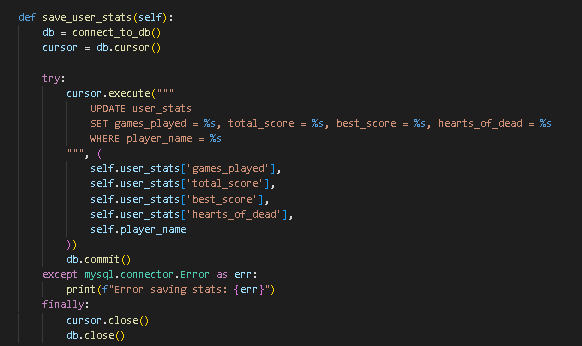
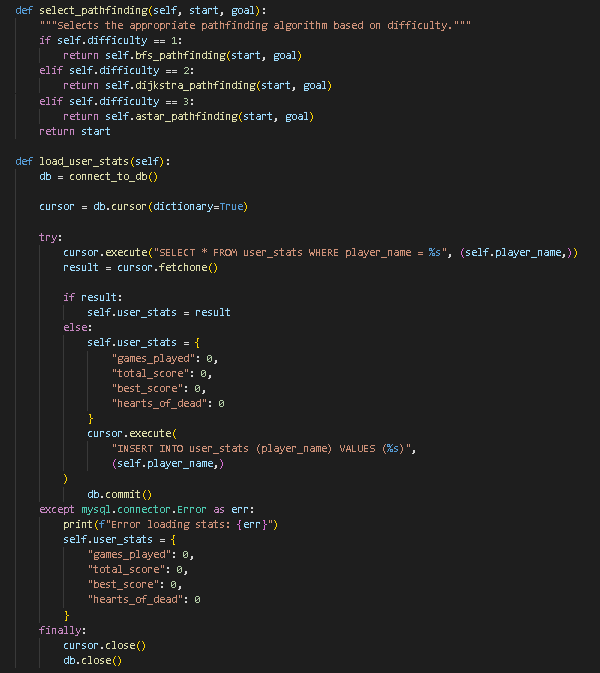


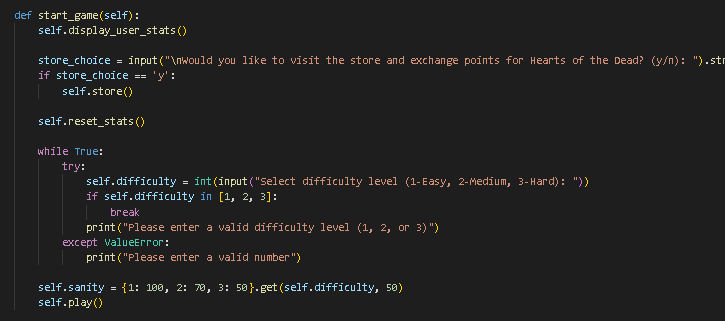
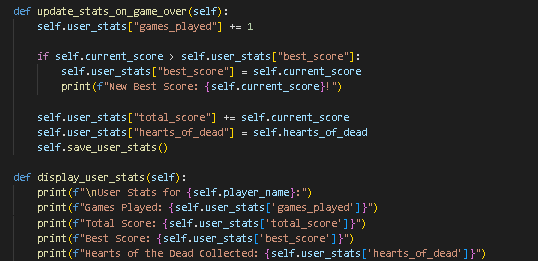
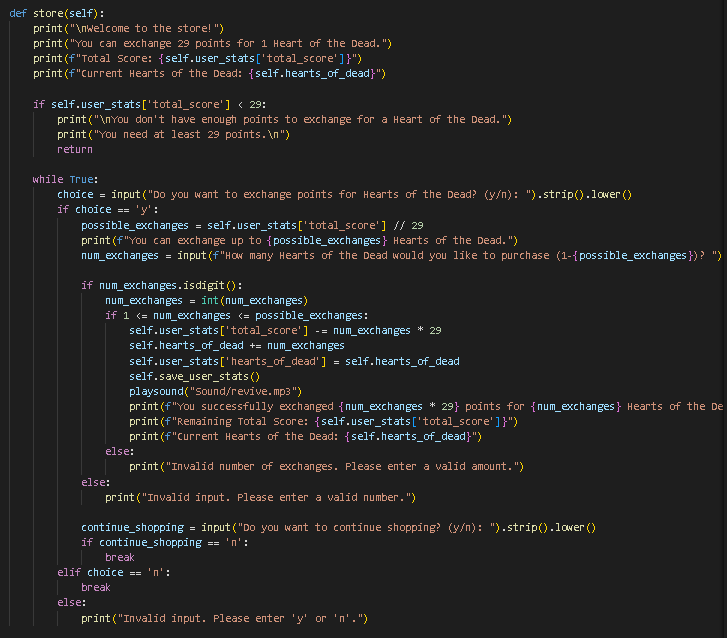


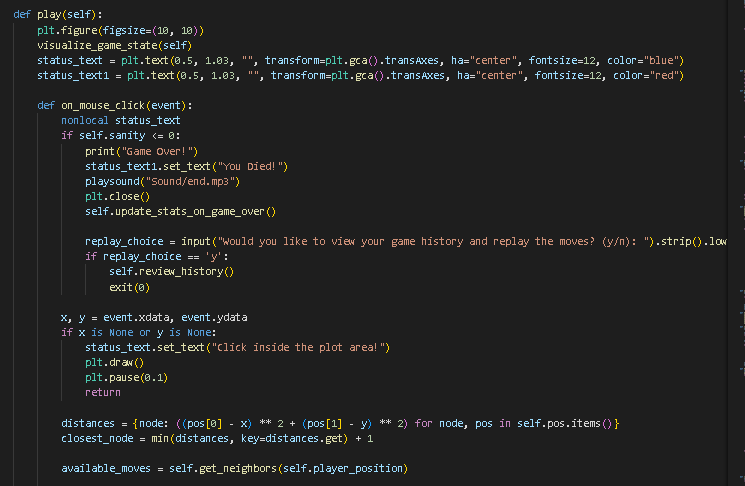
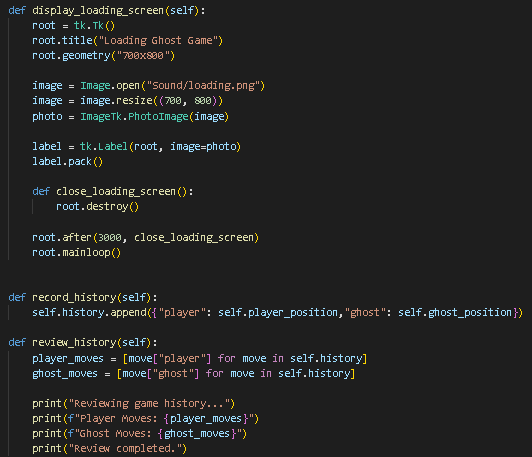
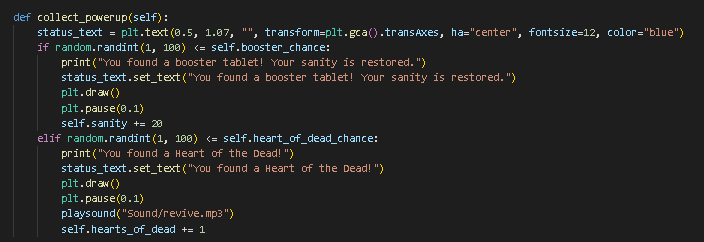


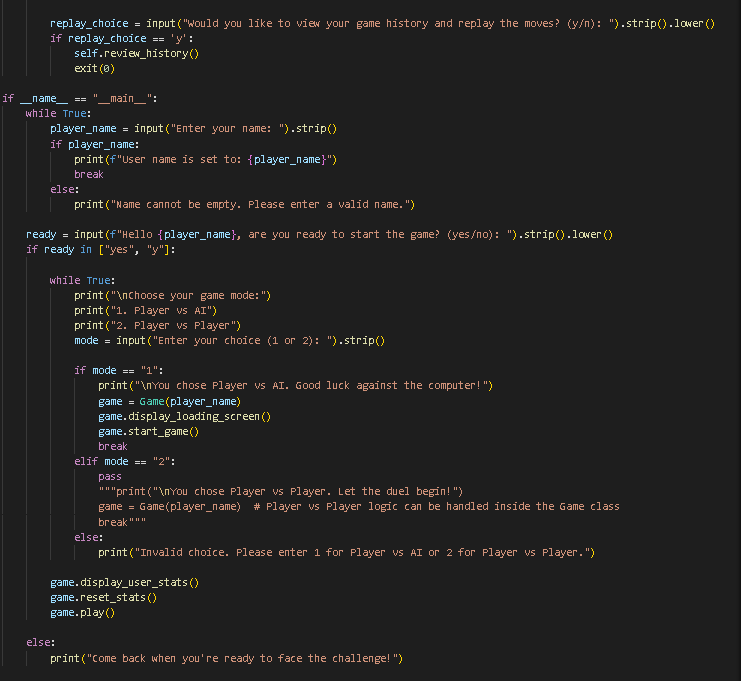
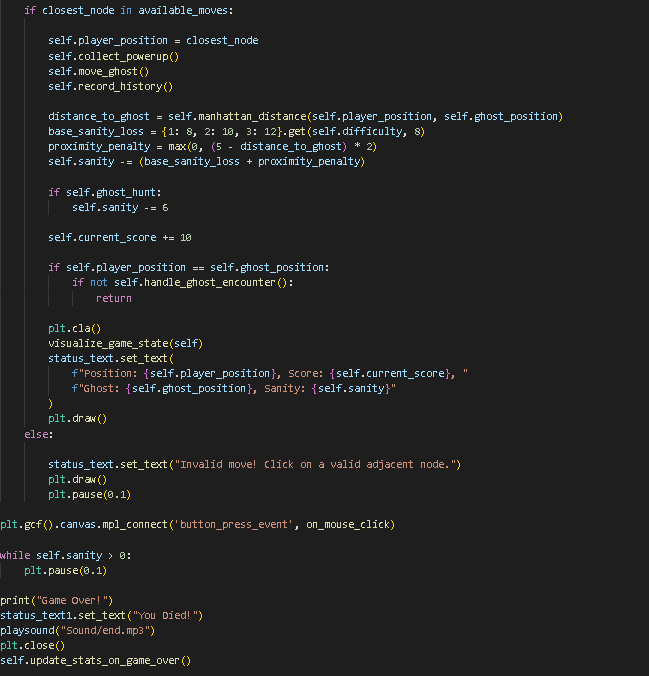


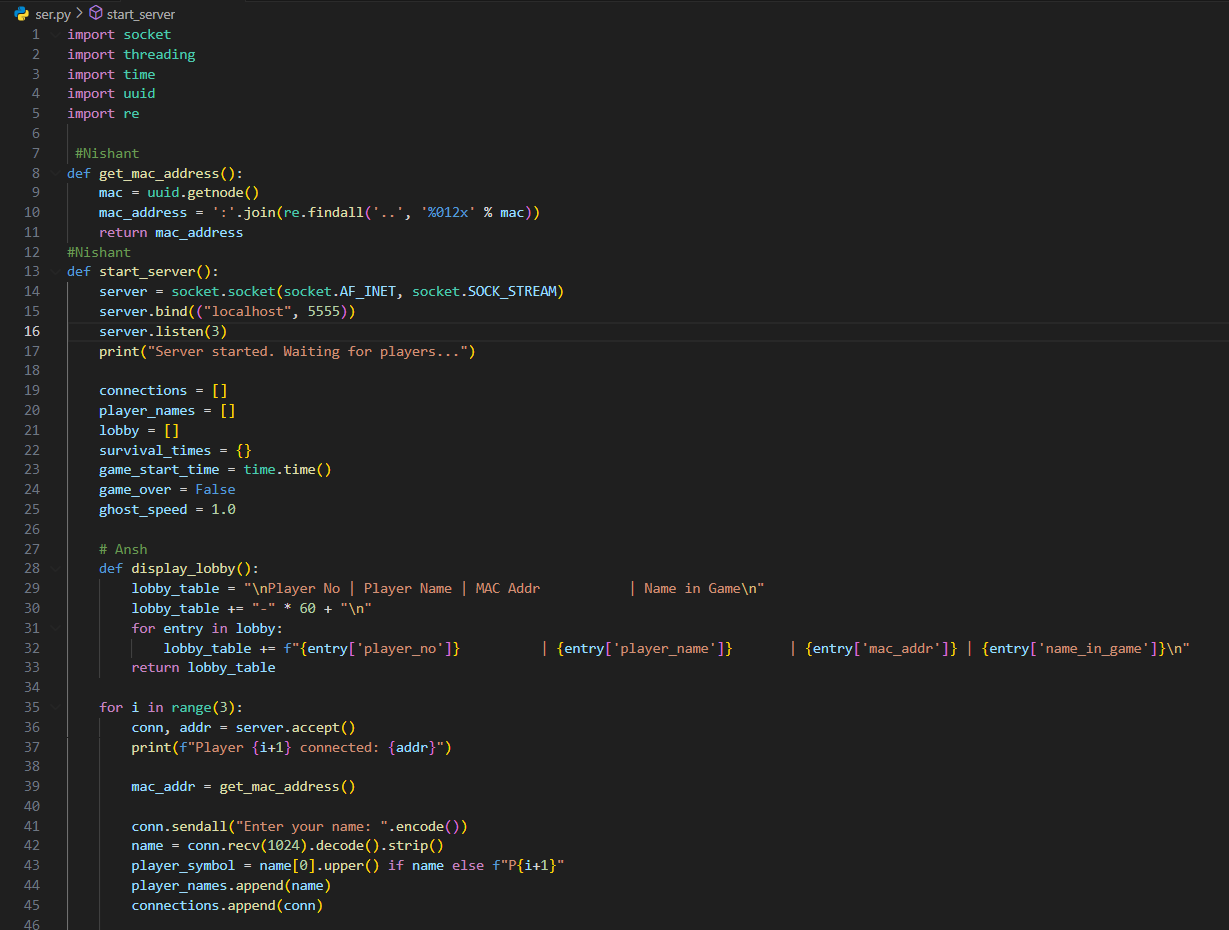










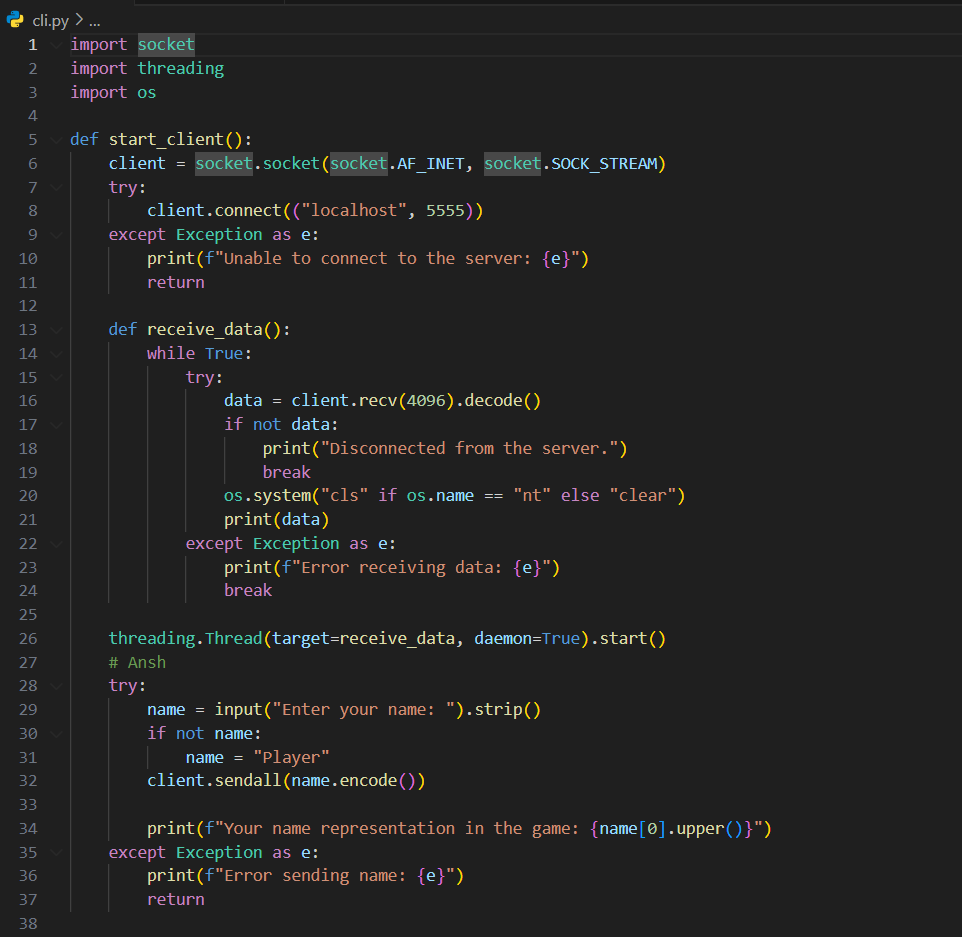


A screen shot of a computer program

Description automatically generated A screen shot of a computer program

Description automatically generated A screen shot of a computer program

Description automatically generated

Client: 

A computer screen shot of text

Description automatically generated