

XONIX Game Management

Group: 10

Name: Muhammad Tayyab, Abdul Mateen

Roll Number: 23i-0738, 23i-0752

Course Name: Data Structures

Section: C

Instructor Name: Zahoor Hassan

Submission Date: 11 May,2025

Introduction

Overview:

We created a management system for game name Xonix. We enhanced the game by adding additional features. We used suitable data structures to simulate real world gaming mechanics.

Key Goals:

The key goals are:

- Implementing real world game mechanics.
- Adding secure authentication system.
- Adding both single and multiplayer mode
- Designing a creative and easy to use menu system.
- Implementing save and load game features
- Implemented a leaderboard to show the top scorers
- Desinging and updating player profile
- Using an AVL tree for theme management
- Developing a social system for by including accepting and sending friend requests feature
- Introduce match making and game room
- Using min heap to manage top scorers

Tools/Languages Used:

The tools used are:

- C++ for logic and data structures implementation.
- SFML for graphics and game interface
- Allowed C++ libraries like cstring, cstdlib and ctime etc
- We built our own data structures live AVL tree, priority queues and linklists

Work distribution

Work division:

Feature Module	Tayyab (23i-0738)	Abdul Mateen (23i-0752)
Login & Authentication	Done	
Main menu		Done
Single Player Scoring	Done	
Multiplayer		Done
Leaderboard	Done	
Match Making	Done	
Game Saving/Loading		Done
Friend System		Done
Inventory		Done
File Handling	Done	Done
Report Writing	Done	

Workflow and Distribution Timeline

The timeline of the project is follow:

Day 1-2:

Studied the base code provided and setup SFML on our devices and drawnUML

Day 3-5:

Implemeted the basic structure of the project and main menu

Day 6-7:

Did player profiling and implemented user authentication system

Day 8:

Implemented scoring of singleplayer .

Day 9-10:

Implemented multiplayer mode of game

Day 11:

Implemented leaderboard

Day 12:

Implemented theme and inventory system.

Day 13:

Implemented load/save game and match making.

Day 14:

Implemented friend system

Day 15:

Wrote the report of the project.

Data Structure Used

1.Array:

Used for creating priority queue. It was used for grid implementation. It provides direct access thus searching happens in $O(1)$ time

2.Linklist:

It was used in friend systems because it provides efficient insertion and deletion and it is dynamic in nature.

3.Priority Queues:

It was used for match making so that the top players are prioritized over other players.

4.AVL Trees:

It was used for storing themes in inventory system because AVL provides efficient sorted retrieval. It ensures that the search time is $O(\log n)$ irrespective of the tree.

5.Min heap:

It was used in implementing leaderboard. It is an efficient way of maintaining top 10 players. It allows quick access to the lowest score for replacement.

6.Hashing:

It is used in friend system for searching friends because it provides fast lookup.

Challenges & Solutions

Save/Load game:

It was very difficult to save and load game in text file.

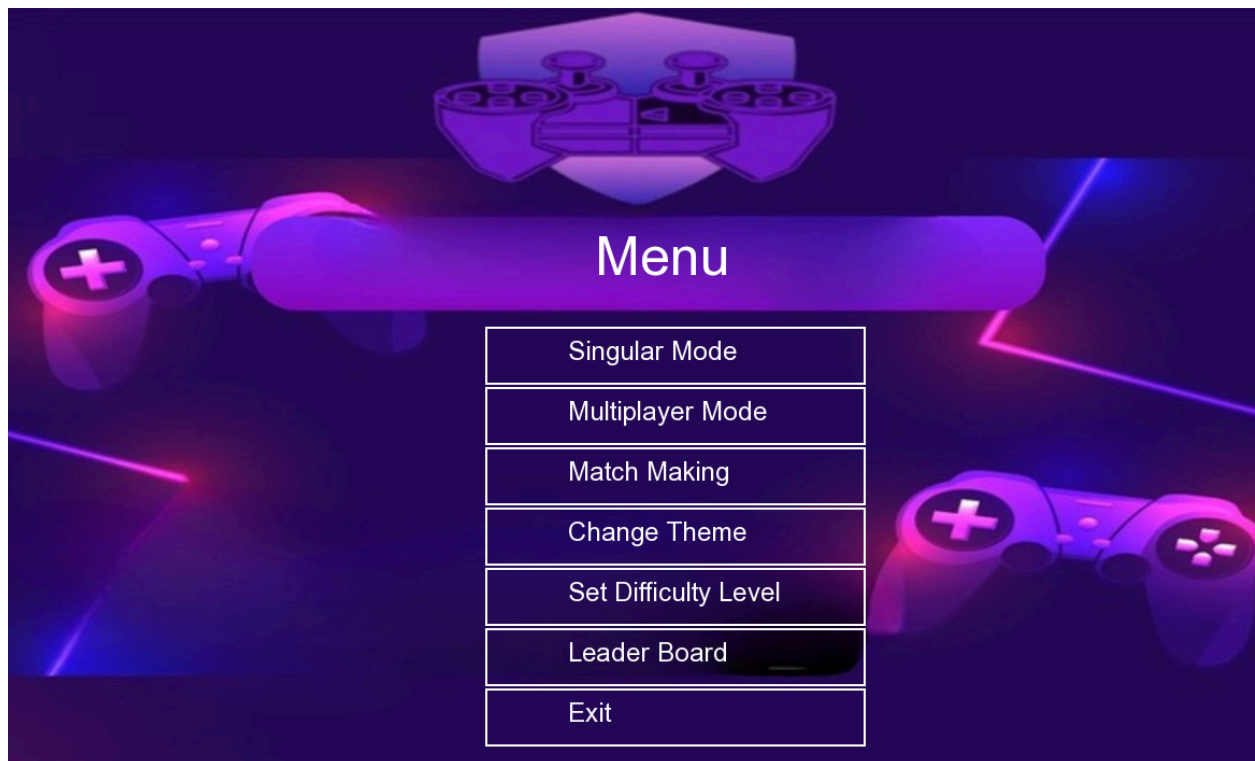
Sol: We created a new saved file for each save by the player with its timestamp

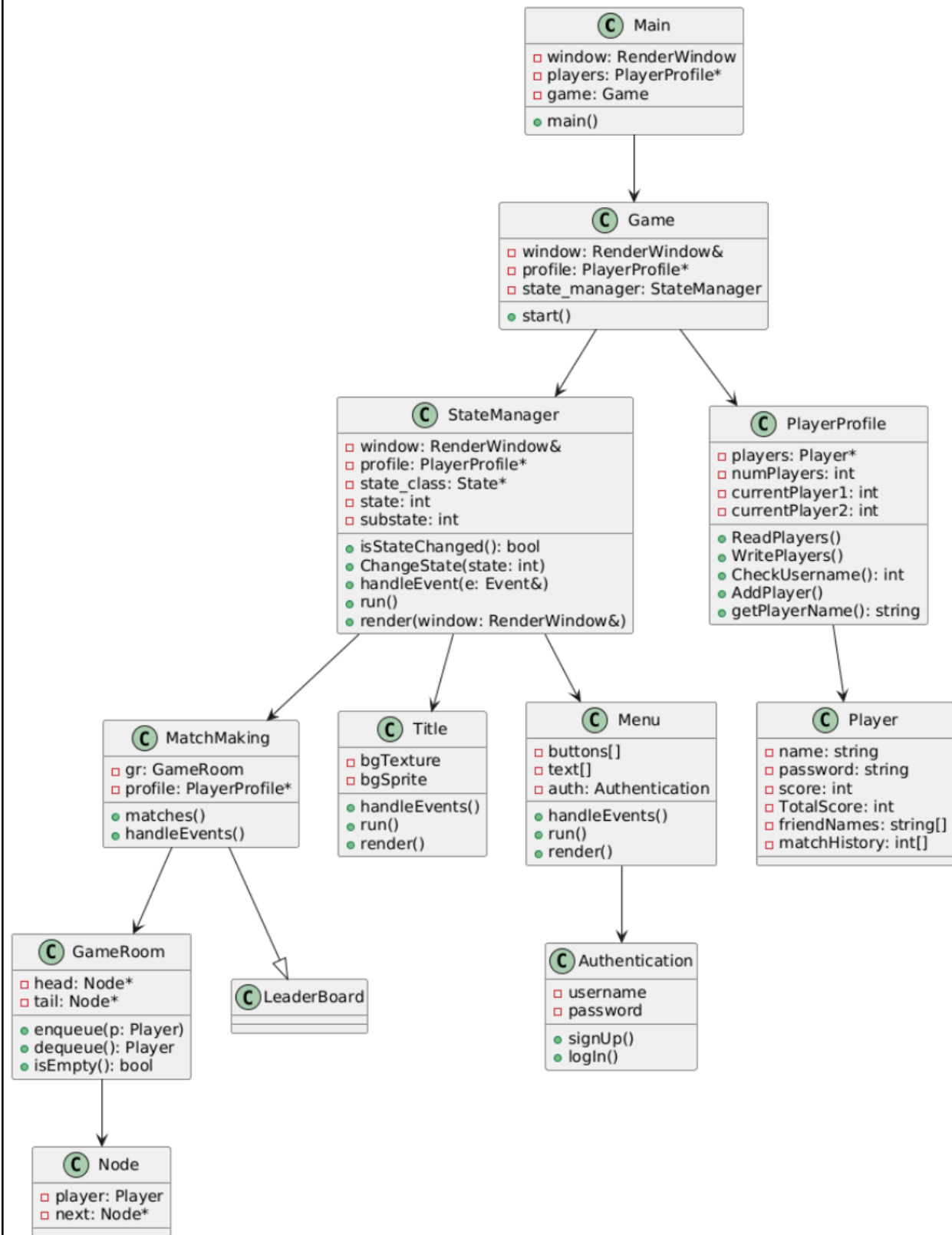
Multiplayer Collision Rules:

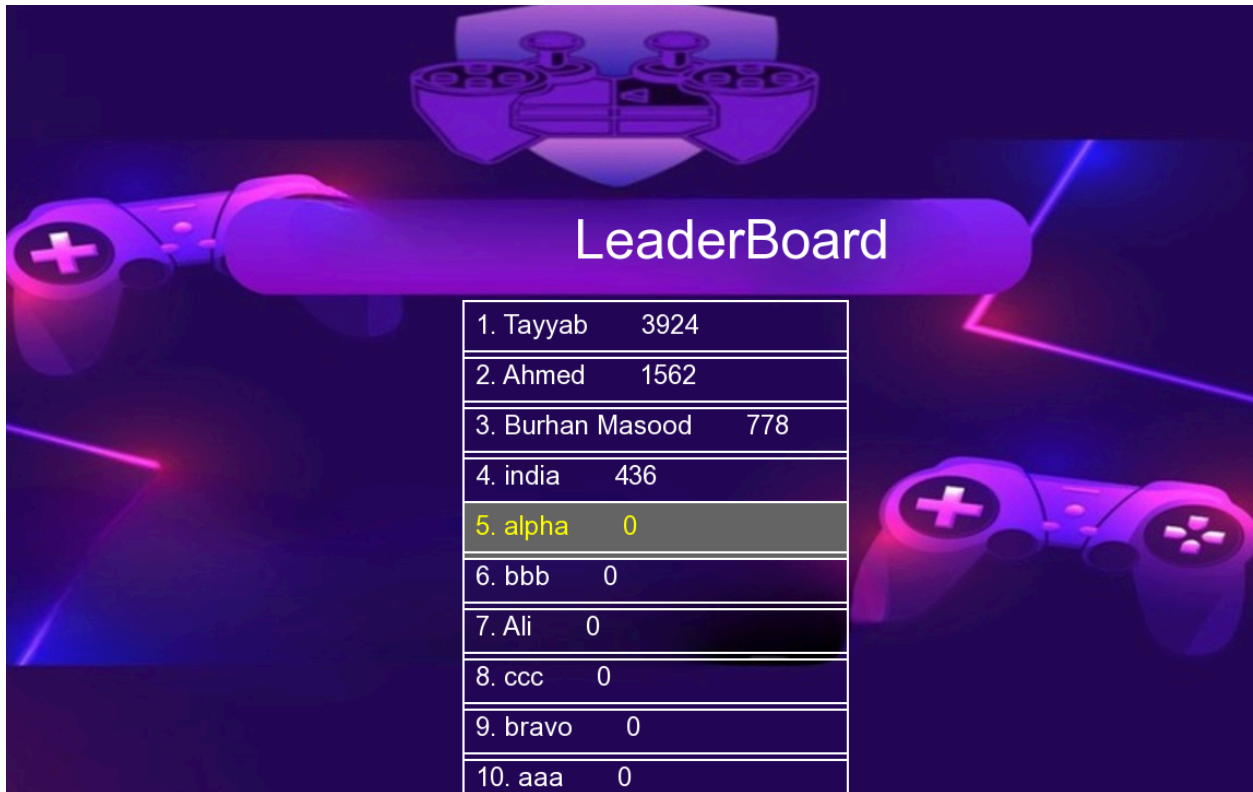
It was very difficult to simulate multiplayer because of its complex collision rules.

Sol: We implemented this by using flags for each scenario.

ScreenShots

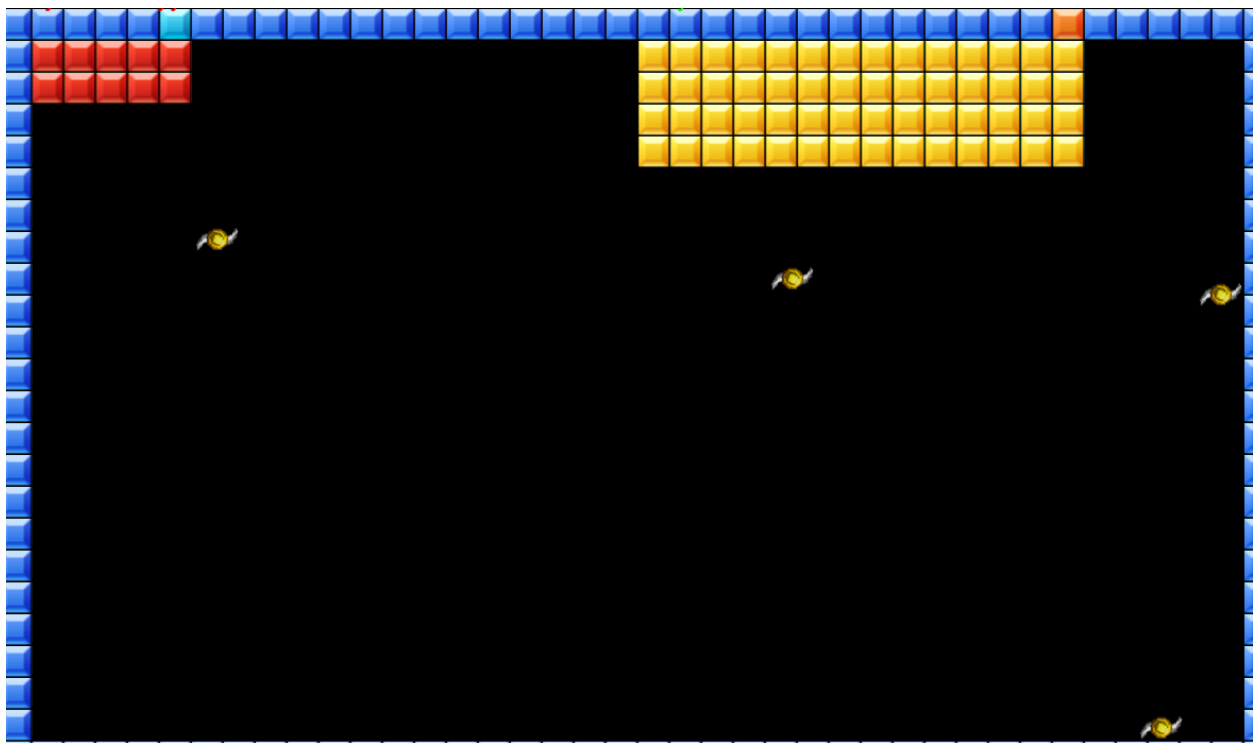






LeaderBoard

1. Tayyab	3924
2. Ahmed	1562
3. Burhan Masood	778
4. india	436
5. alpha	0
6. bbb	0
7. Ali	0
8. ccc	0
9. bravo	0
10. aaa	0



Conclusion

This project taught us how to apply advanced data structures in real-time applications. We developed core programming logic for game mechanics, authentication, matchmaking, leaderboard management, and UI interactions. All components were implemented using our own built data structures

Suggestions for improvements

- Add a computer opponent in singleplayer mode.
- Add online real time multiplayer mode