

Xonix Game

Group#25

Imama Sarwar (23i-3048)
Hiyam Rehan (23i-0039)

Data Structures (CL-2001)
CS-4C

Submitted To:
Zahoor ul Hassan & Arslan Aslam

May 13, 2025

Introduction

This document details the development of a Xonix-inspired game, serving as a practical demonstration of game development methodologies. The project features a compelling single-player experience centered around strategic territory acquisition, a dynamic scoring framework with escalating rewards and tactical power-ups, persistent storage for player achievements, and an engaging competitive multiplayer mode.

The primary objectives of this project were:

- Implementing fundamental Xonix mechanics with responsive player input and intelligent enemy behavior.
- Developing a comprehensive scoring system that incentivizes strategic gameplay and incorporates dynamic bonus structures.
- Integrating power-up functionalities to introduce strategic depth and player advantages.
- Establishing a persistent leaderboard system to promote competition and track player rankings.
- Designing an intuitive user interface with clear navigation and visually appealing, selectable themes.
- Implementing secure user authentication and comprehensive player profile management.
- Enabling seamless multiplayer interaction, including friend request management and efficient matchmaking.
- Providing options for saving and loading single-player game states and allowing user customization through themes.
- Incorporating a friend request system to facilitate social interaction among players.
- Implementing a theme inventory allowing players to personalize their visual game experience.

Technical Specifications

Development Environment:

- Language: C++
- Graphics Library: SFML (Simple and Fast Multimedia Library)
- Additional Libraries: cstring, iostream, fstream, cstdlib, time.h

This document will proceed to elaborate on the work distribution and implementation timeline, along with the data structures used, challenges encountered, current images, and future features/conclusion.

Work Distribution

Features	Work Division
Login and Authentication	Imama Sarwar
Menus (start, levels, mode selection, game over, and pause game)	Hiyam Rehan
Levels (Level 1, 2, and 3)	Imama Sarwar
Points System	Hiyam Rehan and Imama Sarwar
Player Profile	Imama Sarwar
Multiplayer Mode	Hiyam Rehan
LeaderBoard	Hiyam Rehan
Game Room/Matchmaking	Imama Sarwar
Send/Accept Friend Requests	Imama Sarwar
Save Game	Hiyam Rehan
Inventory	Imama Sarwar

Workflow & Implementation Timeline

The development of the Xonix game followed a structured yet agile approach, with daily meetings and goal setting. Below is a breakdown of the planning and implementation phases:

Planning Phase

- **April 26 (Saturday):**
 - The project officially began. A group was formed, and initial discussions were held to understand the game requirements.
- **April 28 (Monday):**
 - A team meeting was conducted to thoroughly review the project documentation.

- The core game flow was finalized, which includes:
Login/Signup → Start Menu → Level Selection → Mode Selection (Single/Multiplayer)
→ Game → Game Over screen.

Development Phase

Date	Work Done
April 28, 2025 (Monday)	Log in/sign up with file handling for user data and main menus based on finalised user flow
April 29, 2025 (Tuesday)	Theme inventory system using AVL trees
April 30, 2025 (Wednesday)	Leaderboard using Min-Heap, How to Play screen, and level system with varied difficulty
May 1, 2025 (Thursday)	Points system with power-ups managed via stack structure, matchmaking/game room via priority queue, saving game state, and setting up the basic structure of the multiplayer mode
May 2, 2025 (Friday)	Continued development of multiplayer functionalities
May 3-4, 2025 (Saturday-Sunday)	Completed Friend Request System
May 5-12, 2025 (Monday-Saturday)	Multiplayer mode, Report, and final touches
May 13, 2025 (Sunday)	Submission



Figure1: Gantt Chart about workflow & implementation timeline

Key Development Challenges and Resolutions

During the development process, we encountered several technical and coordination-related challenges.

One major technical challenge was getting the multiplayer mode to work properly. When we implemented something for one player, it worked fine. But when we tried to apply the same logic to the second player,

things started to break down or behave unexpectedly. This made progress slow and frustrating at times. We handled it by testing a lot, going through trial and error, and researching better ways to handle input and logic for both players.

Another issue was team coordination. Sometimes, one of us would make changes to the code without informing the other. Later, when we tried to combine our work, it would lead to bugs or strange behavior. To fix this, we decided to meet regularly at university, discuss what we were working on, and give each other updates more frequently. This really helped us stay on the same page.

Lastly, time management was tough. We had this project alongside 3–4 other projects, plus presentations and assignment deadlines all within two weeks. To manage everything, we stayed back after classes to work together, divided up tasks clearly, set deadlines, and often worked late into the night to make sure we finished everything on time.

Data Structures Used

Throughout the development of this game, several key data structures were used to support core features efficiently.

- **Min Heap – Leaderboard:** A min heap was used to maintain the leaderboard, where the player with the highest score is always at the root. This structure allowed us to efficiently keep track of the top scorer and reorder ranks when new scores were added, ensuring constant-time access to the highest score.
- **Priority Queue – Matchmaking Queue/Game Room:** The matchmaking system was implemented using a priority queue, where players with the highest scores were given higher priority. Players were selected for game rooms based on both their availability and their position in the queue, allowing fair and skill-based matchmaking in real time.
- **AVL Tree – Themes Inventory:** The themes inventory was managed using an AVL tree to allow quick searching, insertion, and display of themes. AVL trees maintain balance, ensuring that the lookup operations (such as selecting or searching for a theme) remain consistently fast.
- **Linked List – Save Game System:** The saved game system was managed using a Linked List. They allow for efficient dynamic memory allocation, fast insertion and deletion, and minimal wasted storage. It is also easily serialized for writing onto a file, which is needed for save game management system.

- **Hash Table – Friend Requests:** Hash tables were used to manage the friend request system. They allowed for quick access to player data using usernames as keys, making it efficient to check existing friends, sent/received requests, and update friend relationships without unnecessary traversal or delay.

Conclusion

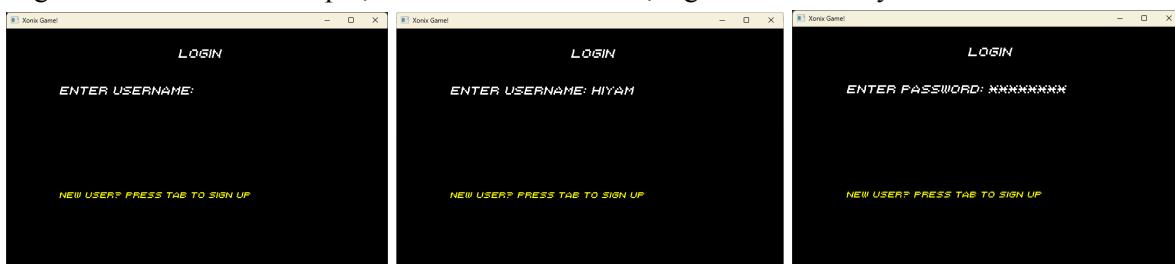
This project proved to be a valuable learning experience for our team on multiple levels. Beyond technical implementation, we developed essential soft skills such as teamwork, time management, effective communication, and creative problem-solving. A major takeaway was understanding how common data structures are used to power features we see every day in apps and games, like friend systems, leaderboards, and matchmaking.

Throughout the development process, we successfully implemented key features including login and authentication, a points system, player profiles, multiplayer mode, leaderboard, matchmaking queue, friend request handling, game saving, and a theme inventory system. These components came together to form a functional and engaging game.

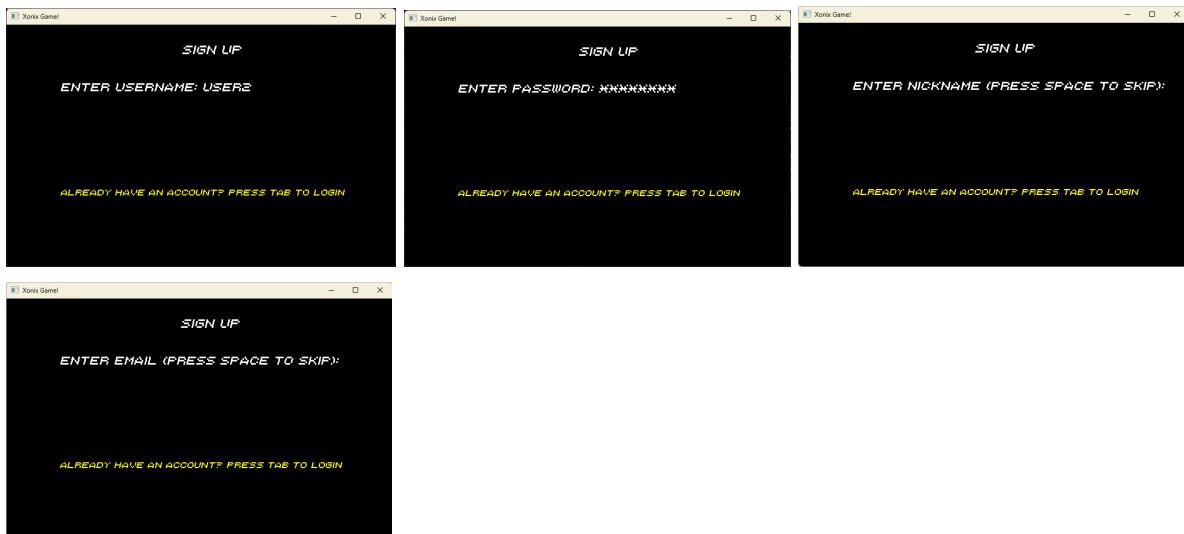
Looking ahead, several enhancements could further improve the gameplay experience. Features such as in-game chat, character customization, and special levels or exclusive themes for top scorers would add depth and personalization. These additions would make the game more interactive, personal, and rewarding for players.

Screenshots

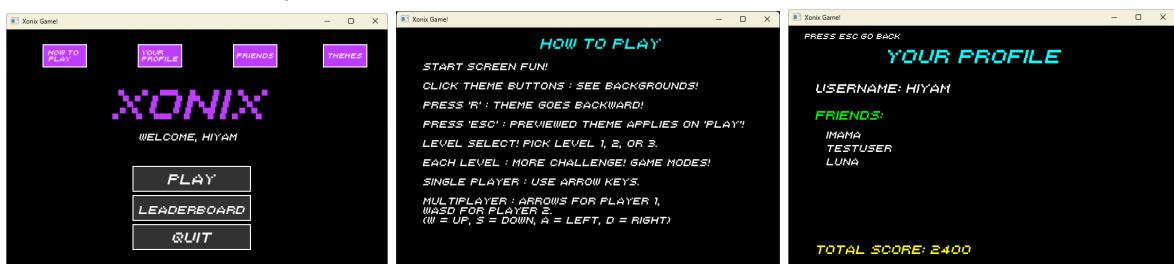
Login screen: Takes user Input, and if the user is found, logs in successfully



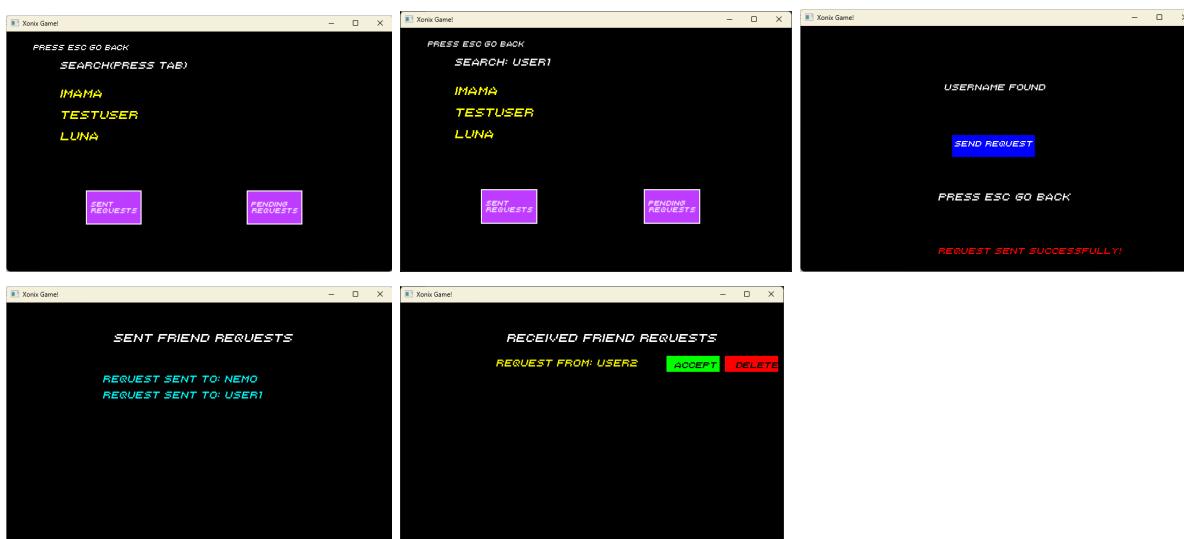
If TAB is pressed, the sign-up page is shown (input username, password, nickname, email)



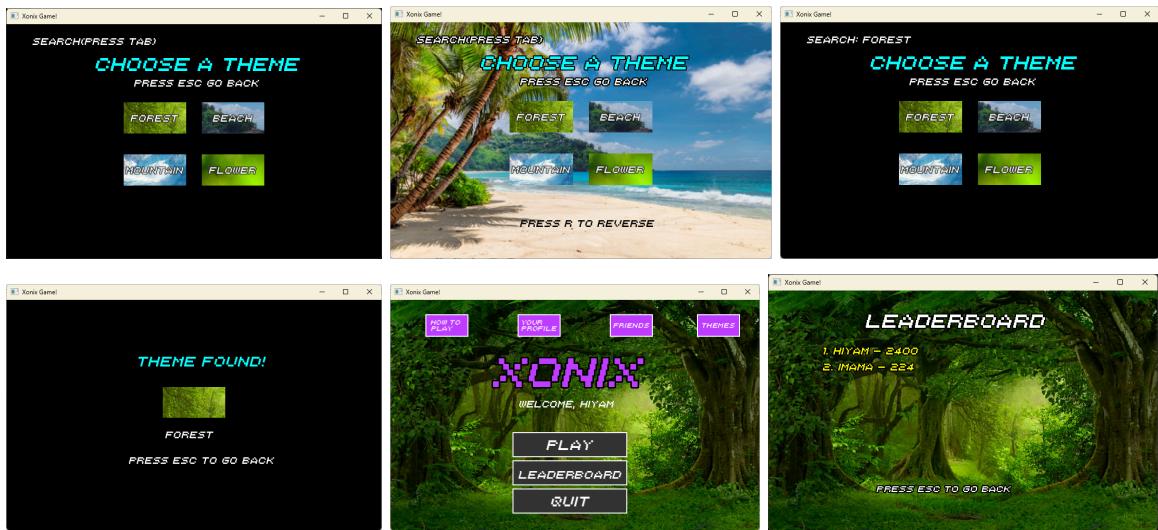
Start menu, How To Play Screen, Your Profile Screen:



Your Friends Screen:



Themes Menu, Theme application, Leaderboard Screen:



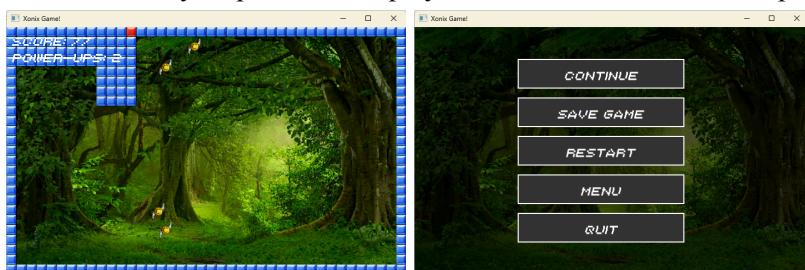
Upon clicking on the play button, the user can choose their level and single/multiplayer mode



Level 1, 2, 3 (left to right)



Takes arrow keys input from the player, and SPACE will enable a powerup. Press ESC to pause



Game over occurs when your trail collides with an enemy. The game is won when you've covered more than 65% of the playfield.



WorkFlow Diagram

