

*Xonix Game*

*Report*

*By*

*Sara Kamran Saeed (24i-3018)*

*Wasif Mehmood (24i-0699)*

*Introduction:*

We developed a 2-player game inspired by the classic Xonix arcade game using C++ and SFML . The main idea is that both players move around a grid to capture area while avoiding enemies. Players can earn points, collect power-ups, and their scores are saved to a file so they can see who won. The game includes a start screen, gameplay loop, and a game over menu with a scoreboard.

Development of the project:

1. Planning the Game:

We initially discussed how the game would operate — such as what the players will do, how the enemies will move, how scoring would function, and what would be at the end.

2. Designing the Logic:

We divided the code into segments such as player movement, management of the grid, enemy AI, handling scores, and menus.

3. Coding the Game:

We employed SFML to render everything — the grid, tiles, enemies, and text such as score and timer.

4. Testing and Refining:

We tested the game repeatedly to correct bugs and ensure features such as scoring and power-ups functioned correctly.

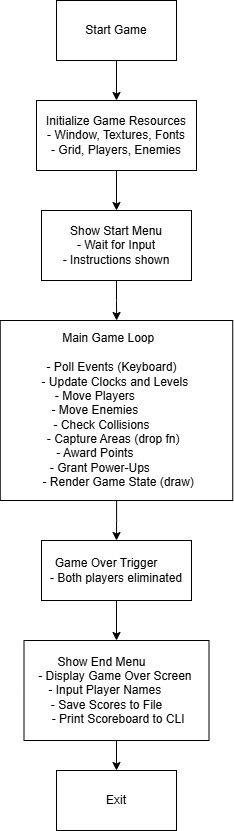
5. Final Touches:

We implemented a scoreboard system and game reset logic, so players can play several rounds conveniently.

Work flow diagram:

Below is the diagram for the workflow

Of the coding of the game

**

*Task distribution:*

. Single and multiplayer mode: Both student 1 and 2

Start menu: Student 1

End menu: student 2

Grid Setup and Logic: Source code

Multiplayer mode: both player 1 and 2

Levels: Both player 1 and 2

Enemy Movement and Bouncing: player 2

Area Capture Logic: Source code

Score & Multiplier System: player 1

Power-Up System: plater 1

UI: Score, Timer Display: player 2

Scoreboard: Save & Load Game Over Screen: Player 2

Asking for Player Names : player 1

Game Reset (Escape Key): Source code

Testing and Debugging: player 1 and 2

Workflow Diagram:player 1

Progress of features:

This project was developed step by step, and in each step a feature was implemented to add enjoyment and usability to the game. We first constructed a grid formed by tiles that created the stage where the game is being played. Then controls were added so the player is able to navigate using the arrow keys or WASD keys so anyone can easily play.

Then, we created the enemies. They were done through a special class that allows them to move automatically, bounce against walls, and increase the difficulty of the game. We then implemented the key feature of the game—area capturing. This section utilizes a technique called recursion to check and fill areas surrounded by the player so that they can occupy space on the grid.

To make the game more exciting, we included a score system. The players earn points depending on how much territory they take over. We also introduced power-ups, such as one that freezes the enemies temporarily, providing the player with a temporary advantage. There is a timer added to indicate how long the game has been ongoing.

To ensure the game is competitive, we developed a scoreboard that stores the best five scores to a file. When the game is over, a Game Over screen displays who won and provides an option to play again. We also implemented a reset option with the ESC key, so players can restart at any time.

Lastly, we added a feature where players are required to input their names before they save their scores. This tracks who got to the top places. All of these features were added incrementally to create a full and fun game.