NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES ISLAMABAD CAMPUS

Programming Fundamentals Spring 2025

Final Project

"Greatness isn't a gift—it's a grind. Chase your goals with fire, break through barriers, and grow through every challenge. The top is reserved for those who never back down."

Due Date: May 8, 2025

Time: 11:59 pm

Please follow the submission instructions. Failure to submit according to the format will result in a mark deduction. Submissions other than Google Classroom (e.g., email, etc.) will not be accepted, and no late submissions will be accepted. The group leader is responsible for the correct and timely submission of the Project; hence, no relaxation will be given to anyone.

Instructions and Submission Guidelines: **Total Marks: 120**

Plagiarism is strongly forbidden and will be very strongly punished. If we find that you have copied from someone else or someone else has copied from you (with or without your knowledge), both of you will be punished. You will be awarded a straight zero in the project (which can eventually result in your failure), and appropriate action as recommended by the Disciplinary Committee (DC can even award a straight F in the subject) will be taken.

This is a group (2 people) activity.

- Any use of AI or plagiarism in the report will result in a grade of zero for the project or an F in the course.
- *Using LLMs for rephrasing or paraphrasing will be considered an AI-generated document.*

- You can use anything to code this (preferably the contents you have covered in the course). If you use any other concept, be prepared to present it during your demonstration. You can get an idea of the implementation from any internet resource, but don’t try to copy any chunk of code exactly from any resource. Zero tolerance policy for plagiarism.
- **Try to understand and complete the project yourself, even if you cannot. Note that you will be mainly rewarded for your effort, not for whether you have completed the project.**
- Divide and conquer: Since you have many days, it is recommended that you divide the complete task into manageable subtasks.
- Imagination Powers: Use your imaginative powers to make this as interesting and appealing as possible. An excellent solution can get you bonus marks.
- Combine all your work in one .zip file.
- Name the .zip file as ROLL-NUM.DEPARTNMENT.SECTION.zip (e.g. 23i-0001.DS.B.zip).
- Submit the .zip file on Google Classroom.
- Use good programming practices (well-commented and indented code, meaningful variable names, readable code, etc.).
- You must follow the submission instructions to the letter, as failing to do so will result in a zero for the Project.
- Read and understand each instruction.

For any unspecified details, use your creativity to design solutions that align with the project's goals. Document your rationale in the report

****** Good Luck ******

Xonix Game

You are working as a game developer when a client approaches you with an idea for a game. Some features have already been implemented, while others are yet to be developed.

Basic Game Rules (these are already implemented):

- A player can move in any of the four directions: right, left, up, and down.
- While moving, if the player is building tiles, they cannot reverse their movement. For example, if the player is moving upwards and building tiles, they cannot move directly downward, which will cause the player's death.
- If an enemy touches a tile that is under construction, the player will die.

Your baseline starter code is available at:

https://github.com/embeddedmz/16-Games/tree/master/09%20Xonix

How to Build a Game on Linux:

Download source code, open terminal in the same folder, and execute the following commands

- sudo apt update
- sudo apt install cmake
- sudo apt install build-essential
- sudo apt install libsfml-dev
- mkdir build
- cd build
- cmake ..
- make (use “sudo apt install make” if not already installed on your system)

Now open build folder and there is an executable file with name xonix. After First compilation only make command need to compile code.

Now following are the Features you need to implement.

1. Basic Features 10 Marks

- **Single and Two Player Modes**
- **Start Menu:**
  - Start Game
  - Select Level
  - Scoreboard
- **End Menu:**
  - End the game properly
  - Show final score (highlight if it's a new high score)
  - Options: Restart, Main Menu, Exit Game

2. Difficulty & Enemy Count 20 Marks

Allow the player to select from the following modes:

- **Easy** – 2 enemies
- **Medium** – 4 enemies
- **Hard** – 6 enemies
- **Continuous Mode** – Starts with 2 enemies. Every 20 seconds, 2 more enemies are added. This continues throughout the game.

3. Movement Counter 5 Marks

- Track and display the number of moves made by the player. (A move is counted each time the player starts building tiles.)

4. Enemy Speed & Movement 20 Marks

- Track and display elapsed game time.
- Every 20 seconds, increase the enemies' movement speed by a fixed amount.
- After 30 seconds, half of the enemies switch from linear movement to specific geometric patterns.
  - Patterns must not be random; use at least **two defined geometric movement patterns** (e.g., **zig-zag**, **circular**).
  - Implement each pattern in separate functions and clearly label them in comments.

5. Scoring & Reward System 10 Marks

- Capturing a tile grants **1 point**.
- Capturing **more than 10 tiles in a single move** awards **double (×2) points**.
- A reward counter tracks how many times this bonus has been earned:
  - After 3 occurrences, the threshold reduces to **5 tiles** for the double points bonus.
  - After 5 total occurrences, capturing **more than 5 tiles** yields **×4 points**.
- Upon reaching a **score of 50**, the player earns a **power-up** that stops all enemies for **3 seconds**.
  - The power-up can be used any time until the end of the game.
  - Additional power-ups are granted at scores of **70, 100, 130**, and for every additional 30 points.
  - Unused power-ups stack and are stored in the player's inventory.

6. Scoreboard 10 Marks

- Maintain a file-based scoreboard (.txt file) that stores the **top 5 highest scores**, sorted in descending order.
- Include the score and time taken.
- On game over, if the player's score is in the top 5, update the file accordingly.

7. Two Player Mode Rules 25 Marks

- All features apply to both players.
- Both players play on the **same game board**:
  - **Player 1 Controls** – Arrow keys
  - **Player 2 Controls** – "W", "A", "S", "D"
- Shared game timer
- Individual scores and power-ups are displayed on their respective sides of the screen.
- Since both players share the same game board, they can interact with each other. If one player touches the other player or their in-progress (constructing) tiles, the following rules apply:
  - If both players collide while they are moving (constructing tiles), both players will die.
  - If Player 1 touches the constructing tile of Player 2, Player 1 will die — and vice versa.
  - If Player 1 collides with Player 2 while Player 1 is constructing tiles and Player 2 is not, Player 1 will die — and vice versa.
- Power-up usage in two-player mode:
  - Stops both the enemies **and the other player** for 3 seconds.
- When both players are eliminated, the game ends and the winner is determined based on the highest score.

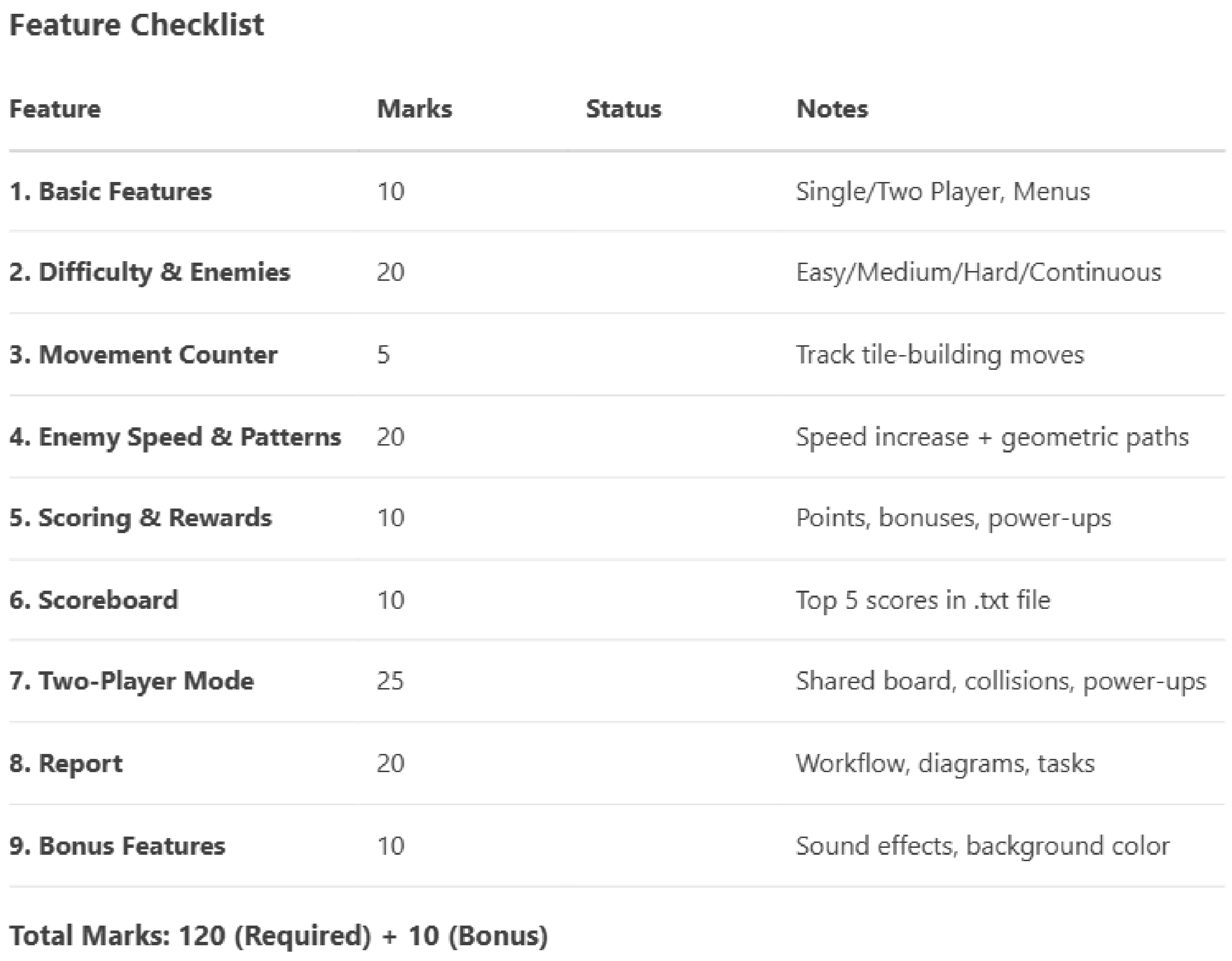
8. Report 20 Marks

A detailed report minimum of 4-6 pages will also be included, *covering the overall project workflow, the game development approach,* ***workflow diagram(e.g., using draw.io, Figma, Lucidchart),*** *task distribution between both team members, and the implementation progress of each feature according to the given requirements.*

9. Bonus Features 10 Marks

- Add sound effects for events such as capturing tiles, achieving bonuses, or activating power-ups.
- Change the background color from black to a suitable alternative when a power-up is used.

Note: Bonus marks will only be awarded after completing all required project features.



Feature Checklist

Feature	Marks	Status	Notes
1. Basic Features	10		Single/Two Player, Menus
2. Difficulty & Enemies	20		Easy/Medium/Hard/Continuous
3. Movement Counter	5		Track tile-building moves
4. Enemy Speed & Patterns	20		Speed increase + geometric paths
5. Scoring & Rewards	10		Points, bonuses, power-ups
6. Scoreboard	10		Top 5 scores in .txt file
7. Two-Player Mode	25		Shared board, collisions, power-ups
8. Report	20		Workflow, diagrams, tasks
9. Bonus Features	10		Sound effects, background color

Total Marks: 120 (Required) + 10 (Bonus)

Restricted Libraries and functions:

- <vector>
- <algorithm>
- Built–in lists
- Built-in queues
- Built–in stack

Allowed Libraries

- <SFML/Graphics.hpp>
- <time.h>
- <Cmath>
- <cstdlib>
- <string>
- <cstring>