Project Title: M&I CandyPop!

Group Members:

1. Syeda Maha Fatima
2. Iqra Liaquat

1. Introduction:

Background:
Modern casual games like Candy Crush have taken the world by storm with their addictive puzzle mechanics and colourful graphics. Inspired by this genre, our project focuses on recreating a simplified version of such a game using Object-Oriented Programming (OOP) concepts in C++. This not only introduces the mechanics of puzzle gaming but also challenges us to implement real-world logic into code using proper design techniques.

Problem Statement:
Designing interactive games involves multiple challenges such as managing objects, user interaction, and visual representation. This project explores how OOP principles such as inheritance, polymorphism, and encapsulation can be applied to build a maintainable, scalable game structure.

Objectives:

- To build a functional and visually interactive puzzle game.
- To apply OOP concepts such as class design, inheritance, aggregation, composition etc.
- To test and showcase our programming skills by cloning a popular game.

2. Scope of the Project:

Inclusions:

- A playable Candy Crush-style game with integrated GUI.
- Grid-based game logic with matching, swapping, and crushing of candies.
- Automatic Crushing of candies included too.
- Levels Difficulty increasing by decreasing the moves.
- Scoring system based on match type (3, 4, 5 candies).

- Sound Included.
- Filing for user data included.

Exclusions:

- Online scoreboards and multiplayer features.
- Level progression.

3. Project Description:

Overview:

Our game implies the concept where the player swaps candies to match three or more identical ones in a row or column. Upon successful matching, the candies vanish, and new ones appear and scoring points in the process. Additionally, upon auto-crushing, score-points increase. This project applies OOP design through classes like Candy, Board, Player & Main-Menu following inheritance and encapsulation.

Technical Requirements:

- Language: C++
- IDE: Visual Studio Code
- SFML (GUI)

Project Phases:

1. Research: Understanding game logic by playing the original game and OOP design.
2. Design: Understanding the classes and all code structure.
3. Development: Simple Code for initial phase & fixing bugs.
4. Applying SFML(GUI)
5. Testing and Polishing: Final touch.

4. Methodology:

Approach:

We followed a simple approach towards our project by breaking the development into smaller modules, implementing and testing each feature iteratively. Key features were assigned and developed step-by-step with continuous feedback.

Team Responsibilities:

- Initially, we shall be dividing the console-based code.
- Collectively, we shall be debugging the code.
- After understanding GUI, we would apply it collectively.

5. Expected Outcomes:

Deliverables:

- A fully playable swapping game with basic functionalities & GUI.
- Project report.
- Source code with proper structure and classes.

Relevance:

This project will demonstrate understanding of ICT topics like interactive software development, object-oriented programming, file organization, and graphical user interface design — all core concepts in modern computing.

Software Needed:

- Installation of Graphics Library
- YouTube Tutorials for understanding the installation and setup.
- Tutorials for understanding GUI Basics.