**Tic-Tac-Toe Game Project:**

**Submitted By:**

Hassan Farooq

**Submitted To:**

Mr. Irfan

**Introduction:**

The Tic-Tac-Toe Game Project is a simple yet engaging implementation of the classic game of Tic-Tac-Toe, designed to demonstrate the principles of Object-Oriented Programming (OOP). This project aims to provide a comprehensive understanding of OOP concepts through the design and implementation of a functional game that adheres to the principles of encapsulation, inheritance, and polymorphism.

**Vision and Scope:**

The vision behind this project is to create a clean, intuitive, and interactive Tic-Tac-Toe game that is easy to understand and play. The scope encompasses the development of a single-player mode against the computer, featuring a graphical user interface (GUI) for enhanced user experience. Future enhancements may include multiplayer modes and advanced AI opponents.

**Designing of the Project (Class Diagram):**

The project is structured around several key classes, each representing a distinct entity within the game:

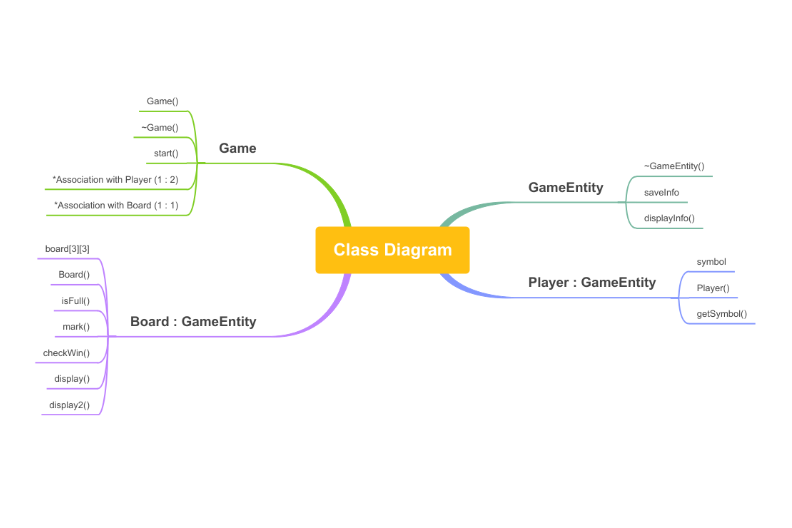
GameEntity: A base class defining common properties and behaviors for game entities.

Player: Inherits from GameEntity, representing a player with a unique symbol ('X' or 'O').

Board: Also inherits from GameEntity, modeling the game board and containing methods to manage the game state.

Game: Orchestrates the game flow, handling player turns, win conditions, and game termination.

**Class Diagram:**

****

**Implementation:**

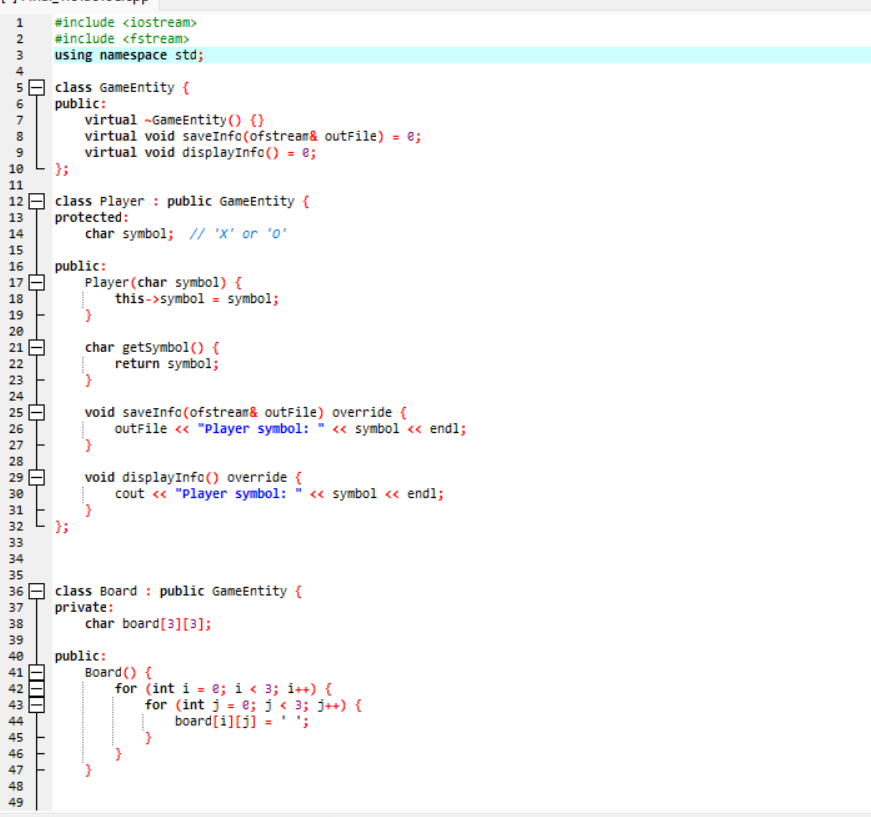
The implementation phase involved translating the design into functional code, focusing on the interaction between classes and the overall game mechanics. Key functionalities include:

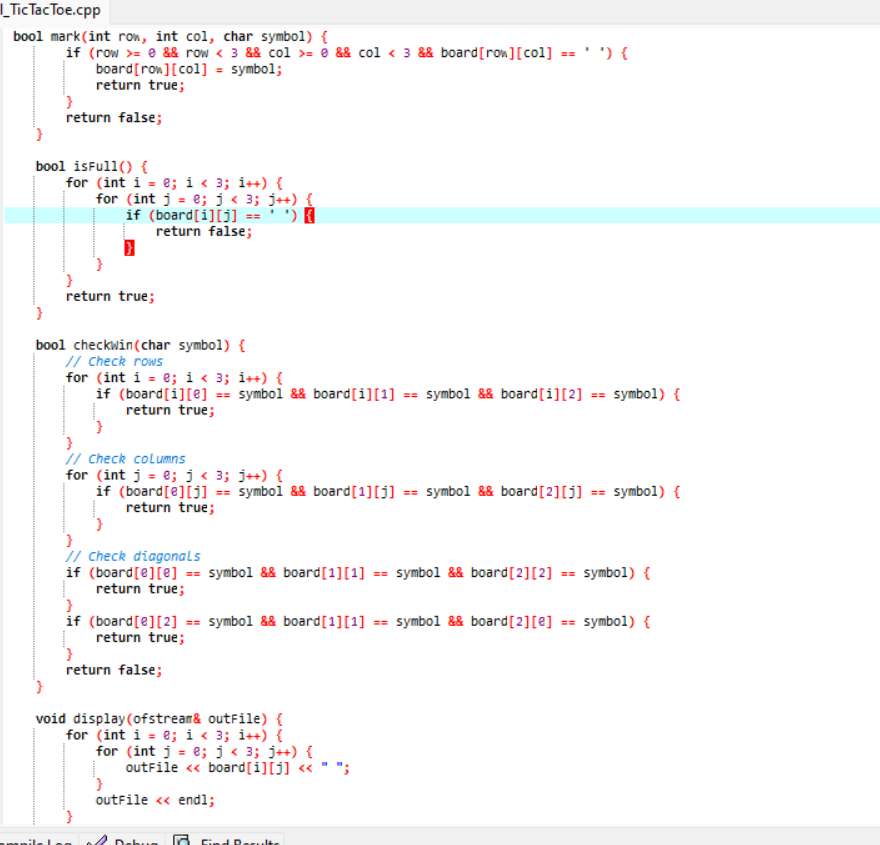
Initialization: Setting up the game board and initializing player objects.

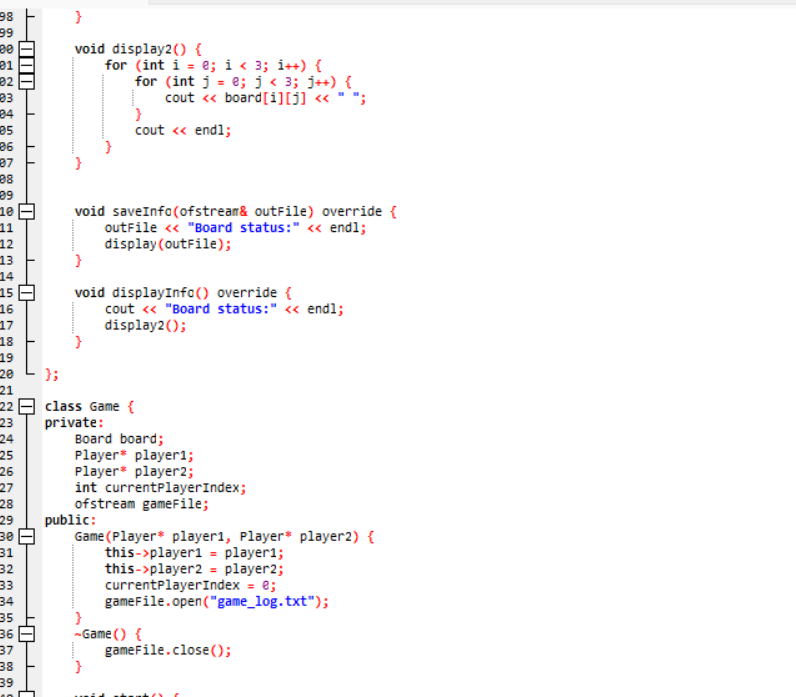
Gameplay: Implementing the game loop, player moves, win condition checks, and displaying the game state.

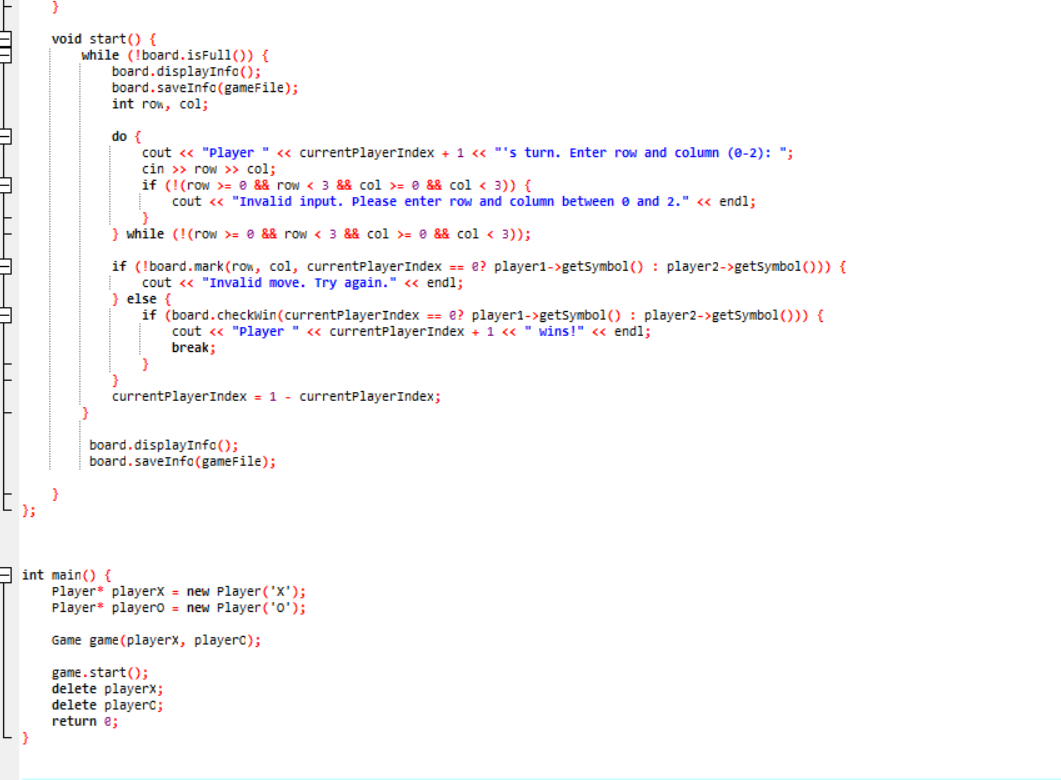
Input Handling: Managing player inputs and validating them against the game rules.

**Screen Shots of Code:**

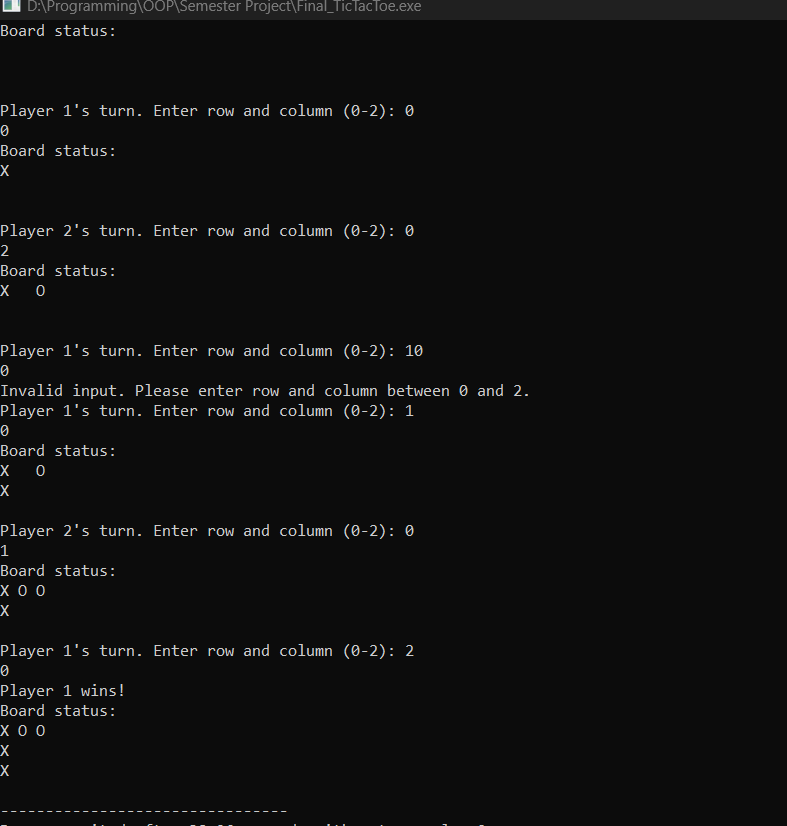
****







**Output Screen Shots**



**Concepts of OOP Used in Projects**

Encapsulation: Hiding internal details and exposing only necessary functions through class interfaces.

Inheritance: Leveraging inheritance to define common behaviors across different types of game entities.

Polymorphism: Allowing game entities to behave differently based on their type, demonstrated through overriding virtual functions.