Swinburne University of Technology

COS20007 Object Oriented Programming

D Level Custom Program Initial Plan

PDF generated at 17:50 on Thursday 28th September, 2023

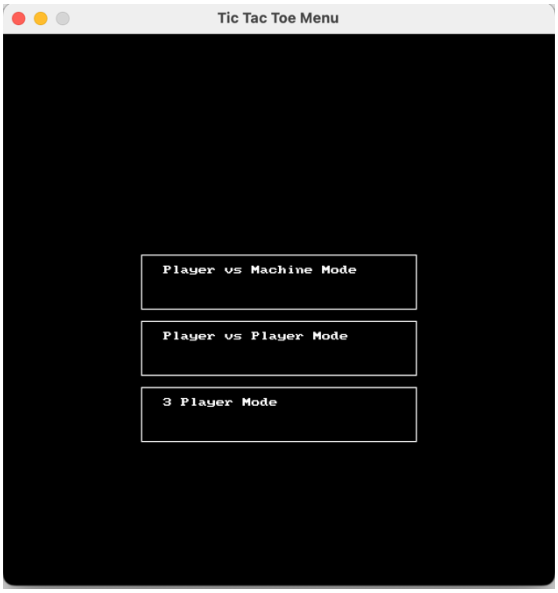
Design Overview for <<Tic Tac Toe Custom Project>>

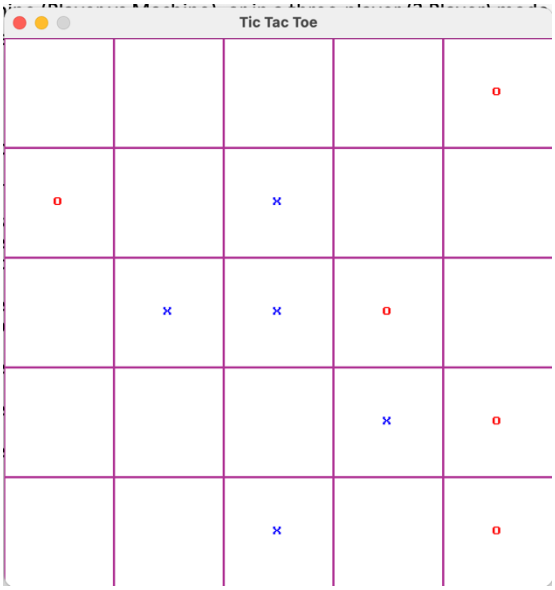
Name: Dang Khoa Le
Student ID: 103844421

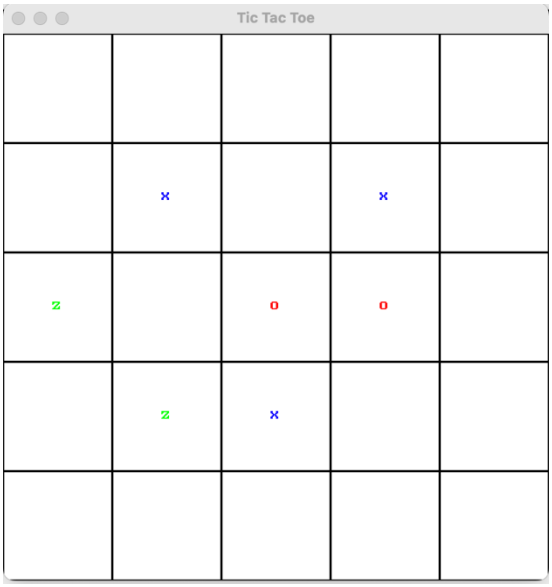
Summary of Program

The program will implement the classic Tic Tac Toe game in a 5x5 grid with three different game modes. Players can choose to play against another player (Player vs Player), against a machine (Player vs Machine), or in a three-player (3 Player) mode where they take turns. The game will track and display scores for each player and allow players to switch between modes from a menu.

Sample Output:








```
This is the Tic Tac Toe game by a 5x5 grid. Made by Khoa Le.

Instruction:
Press [SPACE] to change the grid color, [ESC] to restart the current match.
Once you make a streak in any direction, you win the game.

The game has 3 modes: Player vs Machine, Player vs Player, and 3 Player Mode.
X player will always go first.

The Winner is X side.
[X] 1 - 0 [O]
The Winner is X side.
[X] 2 - 0 [O]
The Winner is O side.
[X] 2 - 1 [O]
```

Required Roles

Table 1: <<Drawing>> (class)

Responsibility	Type Details	Notes
Manages the game board, player move, machine move strategies, and checking for winner method.	**Fields**: board (2D array of PlayerType), markers (List of Player), _gridColor (Color), _gridSize (int), _vsMachineMode (bool), _vsPlayerMode (bool), _threePlayerMode (bool), _currentPlayer (PlayerType), XScore (int), OScore (int), ZScore (int). **Methods**: ChangeGridColor(), Draw(), PlaceMarker(), FindStreak(), MakeMachineMove(), CheckForWinner(), ThreePlayerWinnerCheck(), HandleInput(), SwitchToPlayerVsMachineMode(), SwitchToPlayerVsPlayerMode(), SwitchToThreePlayerMode(), UpdateScores(), Reset()	This class stores most of the game's fundamental set-ups and functionalities, logics.

Table 2: << PlayerType >> (enumeration)

Value	Notes
X, O, Z, Empty.	This enumeration represents the types of players and/or markers on the board.

Table 3: << Player>> (class)

Responsibility	Type Details	Notes
Represents a player with a specific player type (X, O, or Z).	**Fields**: Type (PlayerType).	Represent the real player (human player)

Table 4: << Machine>> (class)

Responsibility	Type Details	Notes
Represents a machine player with a specific player type (marker O).	**Fields**: Type (PlayerType).	Represent the machine player (AI player)

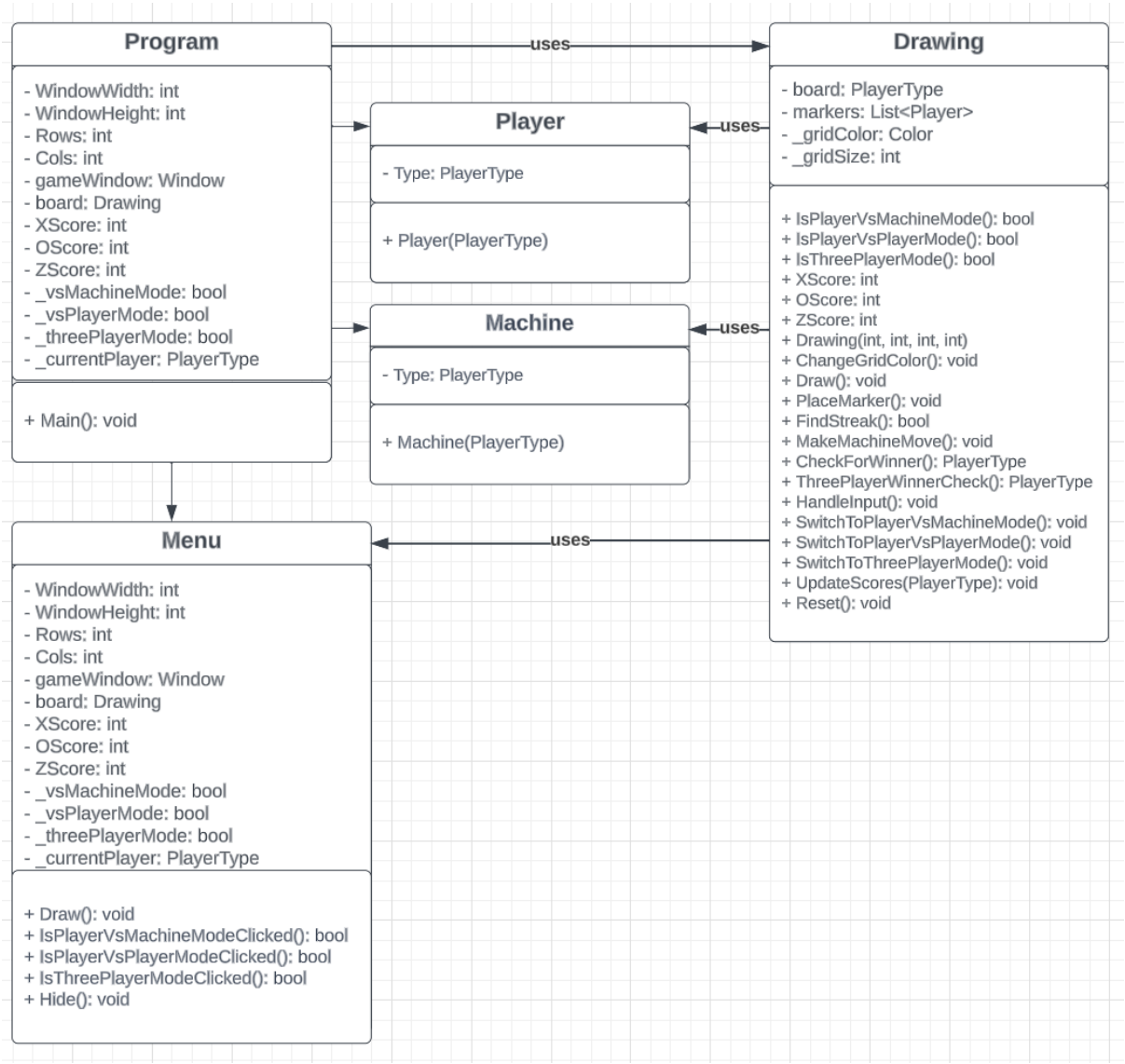
Table 5: <<Menu>> (class)

Responsibility	Type Details	Notes
Manages the game menu and user interface to select game modes.	**Fields**: _window (Window), _PlayerVsMachineModeButton (Rectangle), _playerVsPlayerModeButton (Rectangle), _threePlayerModeButton (Rectangle). **Methods**: Draw(), IsPlayerVsMachineModeClicked(), IsPlayerVsPlayerModeClicked(), IsThreePlayerModeClicked(), Hide().	This class stores the game modes menu, explicitly differs each mode after entering any given modes.

Table 5: <<Program>> (class)

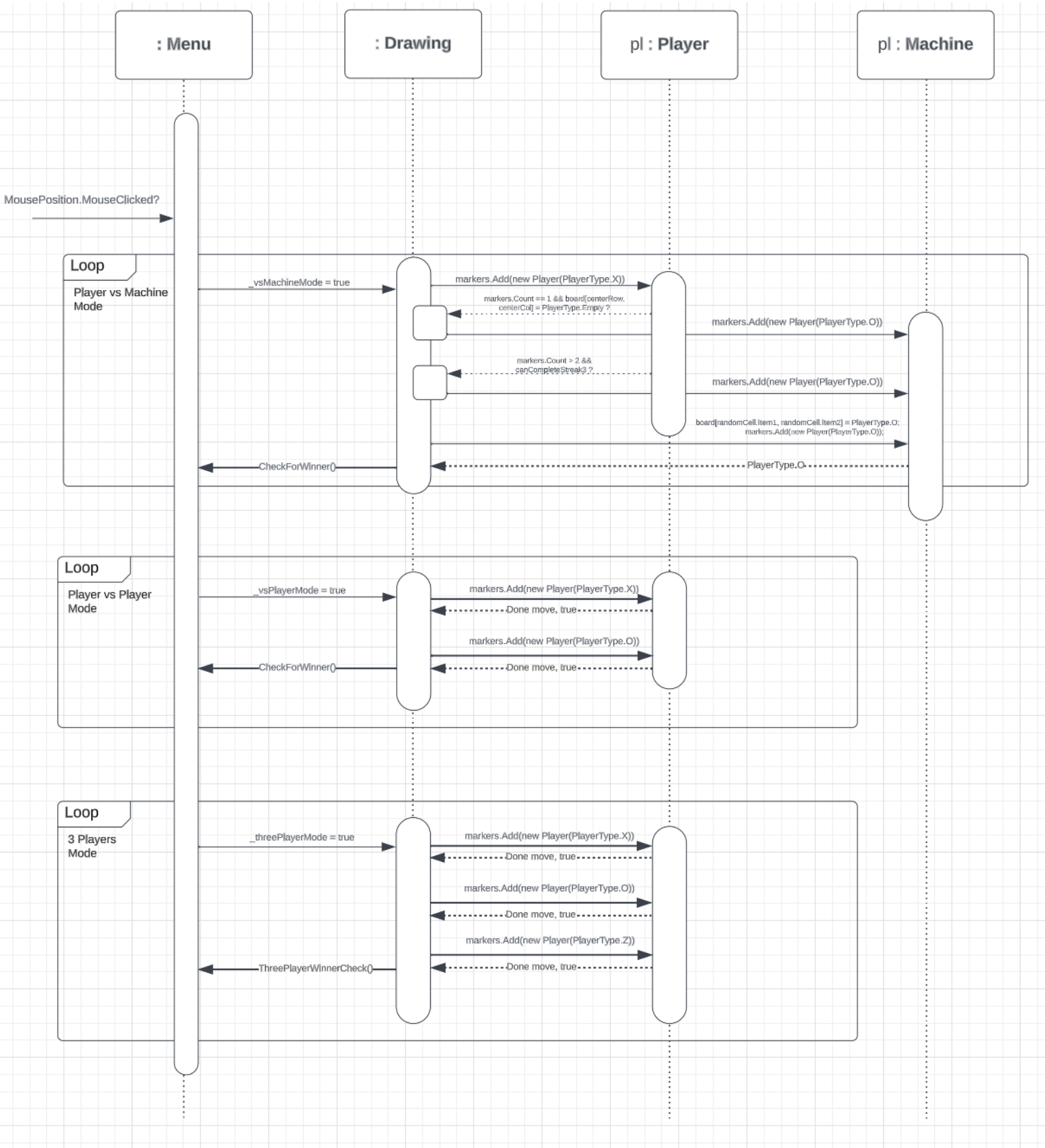
Responsibility	Type Details	Notes
Field type, parameter, and return types. A general set-up for game sketch, defines modes and print game instruction.	**Fields**: WindowWidth: int, WindowHeight: int, Rows: int, Cols: int, gameWindow: Window, board: Drawing, XScore: int, OScore: int, ZScore: int, _vsMachineMode: bool, _vsPlayerMode: bool, _threePlayerMode: bool, _currentPlayer: PlayerType. **Methods** Main(): void	A general/Main class for generic parameters, types.

Class Diagram



Sequence Diagram

Below is the general Sequence Diagram of the Concept Design for ‘Tic Tac Toe’ game.


: Menu
: Drawing
pl : Player
pl : Machine
MousePosition.MouseClicked?
Loop
Player vs Machine Mode
_vsMachineMode = true
markers.Add(new Player(PlayerType.X))
markers.Count == 1 && board[centerRow, centerCol] = PlayerType.Empty ?
markers.Add(new Player(PlayerType.O))
markers.Count > 2 && canCompleteStreak3 ?
markers.Add(new Player(PlayerType.O))
board[randomCell.Item1, randomCell.Item2] = PlayerType.O;
markers.Add(new Player(PlayerType.O));
CheckForWinner()
PlayerType.O
Loop
Player vs Player Mode
_vsPlayerMode = true
markers.Add(new Player(PlayerType.X))
Done move, true
markers.Add(new Player(PlayerType.O))
CheckForWinner()
Done move, true
Loop
3 Players Mode
_threePlayerMode = true
markers.Add(new Player(PlayerType.X))
Done move, true
markers.Add(new Player(PlayerType.O))
Done move, true
markers.Add(new Player(PlayerType.Z))
ThreePlayerWinnerCheck()
Done move, true