

Changes done in the UML Diagrams

Class Player

In this class we removed the method *getWin()* : *Boolean* because the grid class should handle the game state and logic. The abstract player class should only manage individual moves.

Instead of the method *isValidMove* we use the method *checkMove* because the Grid class already ensures a move is valid so checking it again in Player is redundant.

Class Human

The method *getWin()*: *Boolean* inherited from the abstract class player should also be deleted.

The object *scanner* has been added to scan the input.

Class Computer

The method *getWin()*: *Boolean* inherited from the abstract class player should also be deleted.

The object *random* has been added because the computer needs to do a random move.

Class Grid

In this class we added a method *display()*, this method is used to display the board. *makeMove(char, int, char): boolean* method has been added, so the player will be able to choose the position wanted using letters for rows, integers for columns and 'X' or 'O' for symbol.

The method *checkWin(char): boolean* has been added in this class instead of Player because it is the grid's role to manage the flow of the movements.

The method *isFull():boolean* has been added to check if the grid is full which will end the game.

Class Square

Made *value* private which help with encapsulating and preventing the direct modification of value.

Class Game

In this class we added the attributes *grid : Grid, player1 : Player, player2 : Player*.

The constructor *Game()* has been added and we added the methods *choosePlayers()* to determine who the players are.

The method *play()* has been added to control the actual gameplay loop, it ensures that players take turns, checks for a winner and determines when the game ends.