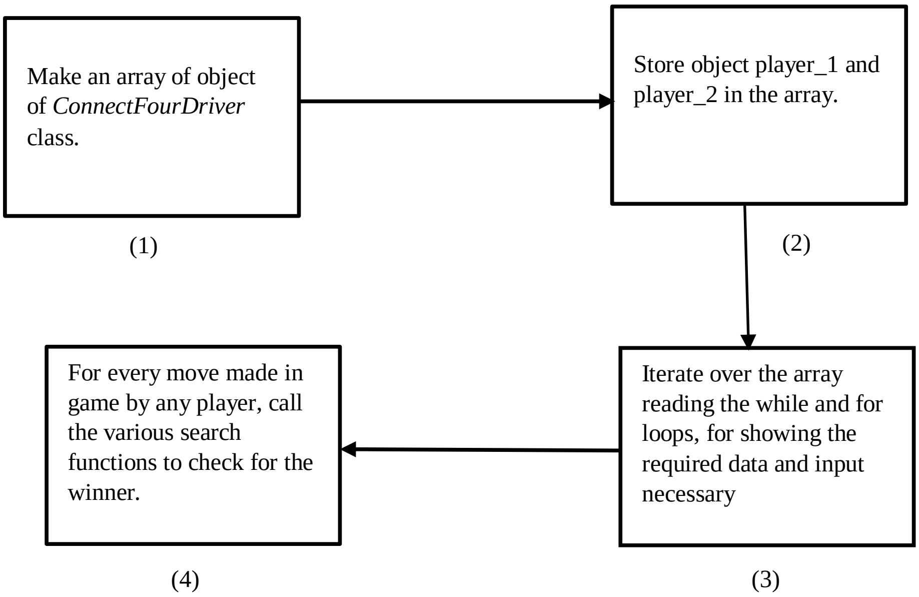
Report: Week 4

Assignment 4: Connect Four Game


Make an array of object of *ConnectFourDriver* class.
(1)
Store object player_1 and player_2 in the array.
(2)
Iterate over the array reading the while and for loops, for showing the required data and input necessary
(3)
For every move made in game by any player, call the various search functions to check for the winner.
(4)

The problem defined is for the creation a connect four game. This implementation is required to be done in two different mechanisms, the game is first played between two human players and the next is implemented where one of the players is a computer AI component.

The first implementation is done by creating an array object of class *ConnectFourDriver*. The main component of the program is the *ConnectFourDriver* class, the *HumanPlayer* class defines parameters for both the player instance – the name, player number. The *ConnectFourDriver* class has two interfaces *ConnectFourPlayerInterface* which contains the methods specific to the player class, the methods included are - ***takeTurn****()- which takes in the input of the player who chance it to play;* ***getName****() – it acquires the name of the player set in the HumanPlayer class;* ***getNumberOfWins****() – this method gets the total number of wins of an individual player throughout the game;* ***addWin****() – every game any of the player wins it is added and returned by the addWin method;* ***getGamePiece****() – gets the column number where the player ;****setGamePiece****() – sets the character that represents that player on the place received from the getGamePiece method;* ***setPlayerNumber****() - the character which represents a player is given by this method;*
The next interface *ConnectFourGameInterface* contains two methods ***getStats***() – this methods gets the scores of the place value when calculating where the game piece of the computer player should be placed. **playGame**() – this method initializes the game; this interface is implemented by the *ConnectFourGame* class. The player 1 and player 2 objects are created and stored in the array. Iterate over the array and print out the required data, asking to the desired input for the game to start. For every move made by the player update the game stats – the wins, for example. Eventually print out the results if the players discontinue to play the game.

For the second mechanism we tried approaching the problem using actual AI algorithm called the Minimax algorithm. Although the procedure followed for implementing the program is fairly the same as the previous problem. While implementing the minimax algorithm, we ran into few problems, which were a bit difficult to debug with it being a fairly long program. Due to the approaching deadline we had to go with the option of randomising the input which we had to provide for the computer player.

Although the second program was a bit challenging, we learnt a lot about how AI algorithms work and had even implemented most of the algorithm sans a few bugs. Apart from that the knowledge of how interfaces work in java, casting and scoping was useful while creating the programs.