PigDice:

|  |  |  |
| --- | --- | --- |
| Behaviour  Our program should handle | Input Example  When it receives | Output Example  It should return |
| 2 players taking turns | hold | Switch player |
| A player should know whether it's an X or an O and be able to report that (e.g. player.mark() could return "X" or "O") | Blank square selected by player to insert their token | X or O appears in grid |
| A space should know its coordinates and be able to be marked by a player (e.g., space.mark(player\_X)) | Space 3 is at Col 3 row 1 | Space 3 is a space, X or O |
| A space should be able to report who it's marked by (e.g. space.markedby() could return "X" or "O", or it could return a player object - \_player1 or player2). | Space 3 is marked by player 1 | Player number and x and O are independent |
| A board should create 9 spaces with the proper coordinates, and tell if there are three in a row marked by the same player. A board should be able to return a space by its coordinates (e.g., board.find(1, 2)). | Three in a row | Winner |
| A game should create 2 players and a board, be able to move to the next turn, know which player's turn it is, and be able to tell if the game is over or not. | Winner | Game over. No more entries. Reset board |
|  |  |  |

Constructor called Turn

Properties: player1CurrentTurn[array], player2CurrentTurn[array], player1Score, player2Score

Two objects Player1Turn and Player 2 Turn

Every time the Roll button is pressed a random integer from 1 to 6 is added to the xxxxTurn array unless it is a 1, in which case the xxxxTurn array is set to 0 and player switches

If it doesn’t switch because of a 0, it adds the total of the Turnarray to the score.

If player 2 is the computer, on easy mode it rolls twice then holds. On hard mode it rolls until array total => 20 then holds