Testing

Informal Testing

Stuff in testing notes go here.

Formal Testing

Test plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test # | Unit | What is getting tested | Input | Expected Result |
| 1 | UBoard. InitDraughts | This function should setup the draughts board at the start of each game. | Empty TArray | Functions returns an array with all the cells in the correct state, determined by the initial draughts board. |
| 4 | UBoard. ClearBoard | This function should be able to clear the board. | Populated TArray | No counters in all cells within the array. |
| 5 | UBoard. WhatPlayer | This function should be able to identify if a counter belongs to the player or the AI. | TArray populated with all counter types | Function returning FALSE for player counters and TRUE for AI counters. |
| 6 | UMove. MakeMove | This function should to use the user input to move a counter from one cell to another. | 4 integers: old row, old column, new row and new column | A counter getting removed from one cell and added to another cell, while maintaining its original value. |
| 7 | UMove. CheckLegalMove | This function should check the move, against certain criteria, in order to check its legality. | TArray and 4 integers: old row, old column, new row and new column | Function outputs TRUE if it is a legal move and FALSE if it is an illegal move. |
| 8 | UMove. PossibleLegalMoves | This function should use UMove.CheckLegalMove on a specified counter to find what moves it can make. | TArray and 2 integers: row and column | The function returns a list of the coordinates for all the legal moves for each counter. |
| 9 | UMove. AllPossibleLegalMoves | This function should use UMove.PossibleLegalMoves to find all the possible moves for the player or the AI. | TArray and Boolean | The function returns a list of all the arrays, which each contain the result of a legal move. |
| 10 | UAI. BoardVal | This function should evaluate the board that it is given. This function used in UAI.Minimax, min and max functions. | TArray with varying amounts of all counters. | The function returns an integer value for the board, which is dependent on the quantity of each counter on the board. |
| 11 | UAI. Min | This function should return the board with the lesser value, for use in USI.Minimax. | Two TArrays | The function returns the board with the lowest value, evaluated by UAI.BoardVal. |
| 12 | UAI. Max | This function should return the board with the greater value, for use in UAI.Minimax. | Two TArrays | The function returns the board with the highest value, evaluated by UAI.BoardVal. |
| 13 | UAI. Minmax | This function should output the next board, according to the move the AI has made, for use in the user interface. | TArray, Boolean and an integer (MaxDepth) | The function returns the array for the next turn, based on the parameters that it was given. |
| 14 | USaveLoad. Save | This function should be able to translate the board to text, so that it can save it to a text file. | TArray, difficulty (integer) and file name (string). | The function returns a save file with the value of each cell in the array and a value for the difficulty of the game. |
| 15 | USaveLoad. Load | This function should be able to translate a text file to variables for use in the board, then return that board for use in the user interface. | File name (string) and difficulty (integer). | The function loads each counter from the save file to their respective cell in the array. It must also correctly set the difficulty specified in the save file. |

++ MM test at the end (probably optional).