TicTacToe Project

Generated by Doxygen 1.9.7

11

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 Board Class Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Function Documentation	6
3.1.2.1 BoardFull()	6
3.1.2.2 CheckForWin()	6
3.1.2.3 EnterMove()	6
3.1.2.4 LetAlMove()	6
3.1.2.5 MakeMove()	7
3.1.2.6 OtherPlayer()	7
3.2 Enum Class Reference	7
3.2.1 Detailed Description	7
3.3 Tile Class Reference	7
3.3.1 Detailed Description	8
4 File Documentation	9
4.1 Project_CP/Board.h File Reference	9
4.2 Board.h	9
4.3 Project_CP/Player.h File Reference	10
4.4 Player.h	10
4.5 Project_CP/Tile.h File Reference	10
4.6 Tile.h	10

Index

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Board																							5
Enum																							7
Tilo																							-

2 Class Index

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Project_CP/Board.	h.									 							 					ç
Project_CP/Player.	h									 							 					10
Project CP/Tile.h										 							 					10

File Index

Class Documentation

3.1 Board Class Reference

Public Member Functions

• Board ()

Constructor of the class Board, it creates a pointer that points to an array of pointers filled with whitespaces.

• \sim Board ()

Destructor of the class Board.

· void PrintBoard ()

Prints current state of the board.

• Tile EnterMove (Player player)

Function responsible for getting the input from user.

• void MakeMove (Tile tile)

Short function for altering the board state.

• Player OtherPlayer (Player player)

Switches between players.

• Player CheckForWin ()

Function that returns which player won.

• bool BoardFull ()

Simple function which tells if the board is full.

• Tile LetAlMove (Player player)

Function that return the best move for AI to play.

Public Attributes

· Player winner

3.1.1 Detailed Description

realted to operations on the game board

6 Class Documentation

3.1.2 Member Function Documentation

3.1.2.1 BoardFull()

```
bool Board::BoardFull ( )
```

Simple function which tells if the board is full.

Returns

bool

3.1.2.2 CheckForWin()

```
Player Board::CheckForWin ( )
```

Function that returns which player won.

Returns

Player

3.1.2.3 EnterMove()

Function responsible for getting the input from user.

Parameters

player

Returns

Tile

3.1.2.4 LetAlMove()

Function that return the best move for AI to play.

Returns

Tile

3.2 Enum Class Reference 7

3.1.2.5 MakeMove()

Short function for altering the board state.

Parameters

tile

3.1.2.6 OtherPlayer()

Switches between players.

Parameters

player

Returns

Player

The documentation for this class was generated from the following files:

- Project_CP/Board.h
- Project_CP/Board.cpp

3.2 Enum Class Reference

3.2.1 Detailed Description

represents a player as a char

The documentation for this class was generated from the following file:

• Project_CP/Player.h

3.3 Tile Class Reference

Public Attributes

- int row
- int column
- char symbol

8 Class Documentation

3.3.1 Detailed Description

square on the board

The documentation for this class was generated from the following file:

• Project_CP/Tile.h

File Documentation

4.1 Project_CP/Board.h File Reference

```
#include "Player.h"
#include <vector>
#include "Tile.h"
```

Classes

· class Board

4.2 Board.h

Go to the documentation of this file.

```
00002 //Declarations (headers) of board related functions
00003
00004 #pragma once
00005 #include "Player.h"
00006 #include <vector>
00007 #include "Tile.h"
80000
00016 class Board
00017 {
00018
          char** board:
00019
          int boardSize;
          int winningCount;
00021
00028
          std::vector<Tile> FindNeighbours(Tile tile, char** board);
00029
00036
          Tile NextTileInSequence(Tile currentTile, Tile previousTile);
00037
00045
          std::vector<Tile> WinningSequence(Tile parentTile, Tile neighbour, int winningCount);
00046
00053
          bool MatchSequences(std::vector<Tile> winningSequence, char** board);
00054
00063
          bool CheckTheSequence(Tile tile, std::vector<Tile> neighbours, int winningCount, char** board);
00064
00069
          int EvaluateBoard();
00070
00078
          int MiniMax(char** board, int depth, bool isMax);
00079
00080 public:
00081
          Player winner;
00082
00086
          Board();
```

10 File Documentation

```
00087
00091
          ~Board();
00092
00096
          void PrintBoard();
00097
00103
          Tile EnterMove(Player player);
00104
00109
          void MakeMove(Tile tile);
00110
          Player OtherPlayer (Player player);
00116
00117
00122
          Player CheckForWin();
00123
00128
          bool BoardFull();
00129
00134
          Tile LetAIMove(Player player);
00135 };
00136
```

4.3 Project_CP/Player.h File Reference

Enumerations

```
• enum class Player { X = 'X', O = 'O', None = ''}
```

4.4 Player.h

Go to the documentation of this file.

```
00001
00003 #pragma once
00007 enum class Player { X = 'X', O = 'O', None = ' ' };
```

4.5 Project_CP/Tile.h File Reference

Classes

• class Tile

4.6 Tile.h

Go to the documentation of this file.

```
00001

00002 #pragma once

00009 class Tile

00010 {

00011 public:

00012 int row;

00013 int column;

00014 char symbol;

00015 };
```

Index

```
Board, 5
    BoardFull, 6
    CheckForWin, 6
    EnterMove, 6
    LetAlMove, 6
    MakeMove, 6
    OtherPlayer, 7
BoardFull
    Board, 6
CheckForWin
    Board, 6
EnterMove
    Board, 6
Enum, 7
LetAlMove
    Board, 6
MakeMove
    Board, 6
OtherPlayer
    Board, 7
Project_CP/Board.h, 9
Project_CP/Player.h, 10
Project_CP/Tile.h, 10
```

Tile, 7