TicTacToe Project

Generated by Doxygen 1.9.7

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

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

Chapter 3

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.
- ∼**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 LetAIMove (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

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()

```
Tile Board::EnterMove (
            Player player )
```

Function responsible for getting the input from user.

Parameters

player	

Returns

Tile

3.1.2.4 LetAIMove()

```
Tile Board::LetAIMove (
            Player player )
```

Function that return the best move for AI to play.

Returns

Tile

3.1.2.5 MakeMove()

```
void Board::MakeMove (
             Tile tile )
```

Short function for altering the board state.

Parameters

tile	

3.1.2.6 OtherPlayer()

```
Player Board::OtherPlayer (
             Player player )
```

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**

3.3.1 Detailed Description

square on the board

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

- Project_CP/Tile.h

Chapter 4

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.

```

//Declarations (headers) of board related functions

#pragma once
#include "Player.h"
#include <vector>
#include "Tile.h"

class Board
{
    char** board;
    int boardSize;
    int winningCount;

    std::vector<Tile> FindNeighbours(Tile tile, char** board);

    Tile NextTileInSequence(Tile currentTile, Tile previousTile);

    std::vector<Tile> WinningSequence(Tile parentTile, Tile neighbour, int winningCount);

    bool MatchSequences(std::vector<Tile> winningSequence, char** board);

    bool CheckTheSequence(Tile tile, std::vector<Tile> neighbours, int winningCount, char** board);

    int EvaluateBoard();

    int MiniMax(char** board, int depth, bool isMax);

public:
    Player winner;

    Board();

    ~Board();

    void PrintBoard();

    Tile EnterMove(Player player);

    void MakeMove(Tile tile);

    Player OtherPlayer(Player player);

    Player CheckForWin();

    bool BoardFull();

    Tile LetAIMove(Player player);
};

```

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.

```

#pragma once
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.

```

#pragma once
class Tile
{
public:
    int row;
    int column;
    char symbol;
};

```

Index