

## TicTacToe Project

Generated by Doxygen 1.9.7



---

<b>1 Class Index</b>	<b>1</b>
1.1 Class List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Class Documentation</b>	<b>5</b>
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 LetAIMove()	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
<b>4 File Documentation</b>	<b>9</b>
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
<b>Index</b>	<b>11</b>



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">Board</a>	.....	<a href="#">5</a>
<a href="#">Enum</a>	.....	<a href="#">7</a>
<a href="#">Tile</a>	.....	<a href="#">7</a>



## Chapter 2

# File Index

### 2.1 File List

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

Project_CP/ <a href="#">Board.h</a> . . . . .	9
Project_CP/ <a href="#">Player.h</a> . . . . .	10
Project_CP/ <a href="#">Tile.h</a> . . . . .	10





## 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

related 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

<i>player</i>	
---------------	--

##### 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

<i>tile</i>	
-------------	--

### 3.1.2.6 OtherPlayer()

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

Switches between players.

#### Parameters

<i>player</i>	
---------------	--

#### 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.](#)

```
00001
00002 //Declarations (headers) of board related functions
00003
00004 #pragma once
00005 #include "Player.h"
00006 #include <vector>
00007 #include "Tile.h"
00008
00016 class Board
00017 {
00018     char** board;
00019     int boardSize;
00020     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();
```

```

00087
00091     ~Board();
00092
00096     void PrintBoard();
00097
00103     Tile EnterMove(Player player);
00104
00109     void MakeMove(Tile tile);
00110
00116     Player OtherPlayer(Player player);
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 };
00016

```

# Index

- Board, [5](#)
  - BoardFull, [6](#)
  - CheckForWin, [6](#)
  - EnterMove, [6](#)
  - LetAIMove, [6](#)
  - MakeMove, [6](#)
  - OtherPlayer, [7](#)
- BoardFull
  - Board, [6](#)
- CheckForWin
  - Board, [6](#)
- EnterMove
  - Board, [6](#)
- Enum, [7](#)
- LetAIMove
  - 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](#)