

connect-four

0.1.0

Generated by Doxygen 1.9.1



<b>1 Connect-four</b>	<b>1</b>
1.1 Building and installing	1
1.2 TODO's	1
1.3 Licensing	1
<b>2 About</b>	<b>3</b>
2.1 Doxygen documentation	3
<b>3 Hierarchical Index</b>	<b>5</b>
3.1 Class Hierarchy	5
<b>4 Class Index</b>	<b>7</b>
4.1 Class List	7
<b>5 File Index</b>	<b>9</b>
5.1 File List	9
<b>6 Class Documentation</b>	<b>11</b>
6.1 Board Class Reference	11
6.1.1 Constructor & Destructor Documentation	11
6.1.1.1 Board() [1/3]	11
6.1.1.2 Board() [2/3]	11
6.1.1.3 Board() [3/3]	12
6.1.2 Member Function Documentation	12
6.1.2.1 CheckForWin()	12
6.1.2.2 IsFull()	12
6.1.2.3 operator=() [1/2]	12
6.1.2.4 operator=() [2/2]	13
6.1.2.5 PrintBoard()	13
6.1.2.6 SetPuck()	13
6.2 Computer Class Reference	13
6.2.1 Constructor & Destructor Documentation	14
6.2.1.1 Computer() [1/2]	14
6.2.1.2 Computer() [2/2]	14
6.2.2 Member Function Documentation	14
6.2.2.1 GetPlacement()	14
6.3 Game Class Reference	14
6.3.1 Detailed Description	15
6.3.2 Constructor & Destructor Documentation	15
6.3.2.1 Game() [1/3]	15
6.3.2.2 Game() [2/3]	15
6.3.2.3 Game() [3/3]	15
6.3.3 Member Function Documentation	15
6.3.3.1 Begin()	15

6.3.3.2 operator=() [1/2]	15
6.3.3.3 operator=() [2/2]	16
6.4 IUser Class Reference	16
6.4.1 Detailed Description	16
6.4.2 Constructor & Destructor Documentation	16
6.4.2.1 IUser() [1/2]	16
6.4.2.2 IUser() [2/2]	17
6.4.2.3 ~IUser()	17
6.4.3 Member Function Documentation	17
6.4.3.1 GetNumber()	17
6.4.3.2 GetPlacement()	17
6.4.4 Member Data Documentation	17
6.4.4.1 user_number	17
6.5 Player Class Reference	17
6.5.1 Constructor & Destructor Documentation	18
6.5.1.1 Player() [1/2]	18
6.5.1.2 Player() [2/2]	18
6.5.2 Member Function Documentation	18
6.5.2.1 GetPlacement()	18
<b>7 File Documentation</b>	<b>19</b>
7.1 docs/pages/about.dox File Reference	19
7.2 README.md File Reference	19
7.3 src/board.cpp File Reference	19
7.4 src/board.h File Reference	19
7.4.1 Variable Documentation	20
7.4.1.1 BOARD_HEIGHT	20
7.4.1.2 BOARD_WIDTH	20
7.5 src/computer.cpp File Reference	20
7.6 src/computer.h File Reference	20
7.7 src/game.cpp File Reference	20
7.8 src/game.h File Reference	21
7.8.1 Variable Documentation	21
7.8.1.1 PLAYERCOUNT	21
7.9 src/iuser.h File Reference	21
7.10 src/main.cpp File Reference	21
7.10.1 Function Documentation	22
7.10.1.1 main()	22
7.11 src/player.cpp File Reference	22
7.12 src/player.h File Reference	22

# Chapter 1

## Connect-four

This is the connect-four project for the requitment process of **S[&]T**.

### 1.1 Building and installing

See the BUILDING document.

### 1.2 TODO's

- Fix `CMAKE_TOOLCHAIN_FILE` in `CMakePrests.json` for linux so its not a hardcoded path
  - Visual Studio cross compiling cmake does not understand `$env{VCPKG_ROOT}` to get the environment variable from wsl
- Clear more warnings that clang-tidy suggests
- Maybe a sizeable board for more then 2 players as a extra feature
- Have doxygen docs generation work with the cmake target
  - Get comments from code in doxygen

### 1.3 Licensing

MIT



## Chapter 2

# About

### 2.1 Doxygen documentation

This page is auto generated using `Doxygen`, making use of some useful `special commands`.





## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Board . . . . .	11
Game . . . . .	14
IUser . . . . .	16
Computer . . . . .	13
Player . . . . .	17



## Chapter 4

# Class Index

### 4.1 Class List

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

<a href="#">Board</a>	11
<a href="#">Computer</a>	13
<a href="#">Game</a>	14
<a href="#">IUser</a>	16
<a href="#">Player</a>	17



## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

src/ <a href="#">board.cpp</a>	19
src/ <a href="#">board.h</a>	19
src/ <a href="#">computer.cpp</a>	20
src/ <a href="#">computer.h</a>	20
src/ <a href="#">game.cpp</a>	20
src/ <a href="#">game.h</a>	21
src/ <a href="#">iuser.h</a>	21
src/ <a href="#">main.cpp</a>	21
src/ <a href="#">player.cpp</a>	22
src/ <a href="#">player.h</a>	22



## Chapter 6

# Class Documentation

### 6.1 Board Class Reference

```
#include <board.h>
```

#### Public Member Functions

- [Board](#) ()=default
- [Board](#) (const [Board](#) &other)=delete
- [Board](#) ([Board](#) &&other)=delete
- [Board](#) & [operator=](#) (const [Board](#) &other)=delete
- [Board](#) & [operator=](#) ([Board](#) &&other)=delete
- bool [SetPuck](#) (std::pair< uint8\_t, uint8\_t > &point, uint8\_t player)
- bool [CheckForWin](#) (const std::pair< uint8\_t, uint8\_t > point)
- bool [IsFull](#) ()
- void [PrintBoard](#) ()

#### 6.1.1 Constructor & Destructor Documentation

##### 6.1.1.1 [Board\(\)](#) [1/3]

```
Board::Board ( ) [default]
```

##### 6.1.1.2 [Board\(\)](#) [2/3]

```
Board::Board (
    const Board & other ) [delete]
```

### 6.1.1.3 Board() [3/3]

```
Board::Board (
    Board && other ) [delete]
```

## 6.1.2 Member Function Documentation

### 6.1.2.1 CheckForWin()

```
auto Board::CheckForWin (
    const std::pair< uint8_t, uint8_t > point )
```

Check if last set puck has a winning position

#### Parameters

in	<i>point</i>	containing coordinates of a placed puck
----	--------------	---

#### Returns

true if last point was winnig

### 6.1.2.2 IsFull()

```
auto Board::IsFull ( )
```

Check if the board is full

#### Returns

true if full

### 6.1.2.3 operator=() [1/2]

```
Board& Board::operator= (
    Board && other ) [delete]
```



**6.1.2.4 operator=()** [2/2]

```
Board& Board::operator= (
    const Board & other ) [delete]
```

**6.1.2.5 PrintBoard()**

```
void Board::PrintBoard ( )
```

Print out a stylized connect four board on the command line

**6.1.2.6 SetPuck()**

```
auto Board::SetPuck (
    std::pair< uint8_t, uint8_t > & point,
    uint8_t player )
```

Set the puck on the board

**Parameters**

<i>in, out</i>	<i>point</i>	will be used for placing a puck on the board
<i>in</i>	<i>player</i>	value to be put on the board

**Returns**

true if value was succesfull in placing the puck

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

- [src/board.h](#)
- [src/board.cpp](#)

**6.2 Computer Class Reference**

```
#include <computer.h>
```

Inheritance diagram for Computer:

Collaboration diagram for Computer:

**Public Member Functions**

- [Computer](#) ()=default
- [Computer](#) (uint8\_t)
- const uint32\_t [GetPlacement](#) () override

## Additional Inherited Members

### 6.2.1 Constructor & Destructor Documentation

#### 6.2.1.1 Computer() [1/2]

```
Computer::Computer ( ) [default]
```

#### 6.2.1.2 Computer() [2/2]

```
Computer::Computer (
    uint8_t number )
```

### 6.2.2 Member Function Documentation

#### 6.2.2.1 GetPlacement()

```
const uint32_t Computer::GetPlacement ( ) [override], [virtual]
```

Return a randomised number between 0 and 6

Implements [IUser](#).

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

- [src/computer.h](#)
- [src/computer.cpp](#)

## 6.3 Game Class Reference

```
#include <game.h>
```

### Public Member Functions

- [Game](#) ()=default
- [Game](#) (const [Game](#) &other)=delete
- [Game](#) ([Game](#) &&other)=delete
- [Game](#) & operator= (const [Game](#) &other)=delete
- [Game](#) & operator= ([Game](#) &&other)=delete
- void [Begin](#) ()

### 6.3.1 Detailed Description

[Game](#) class this contains the startup of connect four by asking how many players should play After this the game will loop until the game reaches a end

### 6.3.2 Constructor & Destructor Documentation

#### 6.3.2.1 Game() [1/3]

```
Game::Game ( ) [default]
```

#### 6.3.2.2 Game() [2/3]

```
Game::Game (
    const Game & other ) [delete]
```

#### 6.3.2.3 Game() [3/3]

```
Game::Game (
    Game && other ) [delete]
```

### 6.3.3 Member Function Documentation

#### 6.3.3.1 Begin()

```
void Game::Begin ( )
```

Begin connect four game Ask user how many player will be playing And initialized said players

#### 6.3.3.2 operator=() [1/2]

```
Game& Game::operator= (
    const Game & other ) [delete]
```

### 6.3.3.3 operator=() [2/2]

```
Game& Game::operator= (
    Game && other ) [delete]
```

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

- [src/game.h](#)
- [src/game.cpp](#)

## 6.4 IUser Class Reference

```
#include <iuser.h>
```

Inheritance diagram for IUser:

### Public Member Functions

- [IUser](#) ()=default
- [IUser](#) (uint8\_t number)
- virtual [~IUser](#) ()=default
- virtual const uint32\_t [GetPlacement](#) ()=0
- const uint8\_t [GetNumber](#) ()

### Protected Attributes

- uint8\_t [user\\_number](#) {0}

### 6.4.1 Detailed Description

Interface class for the user that play the game of connect four This is used so the derived classes [Player](#) and [Computer](#) can be called. from the game class

### 6.4.2 Constructor & Destructor Documentation

#### 6.4.2.1 IUser() [1/2]

```
IUser::IUser ( ) [default]
```

#### 6.4.2.2 IUser() [2/2]

```
IUser::IUser (
    uint8_t number ) [inline]
```

#### 6.4.2.3 ~IUser()

```
virtual IUser::~IUser ( ) [virtual], [default]
```

### 6.4.3 Member Function Documentation

#### 6.4.3.1 GetNumber()

```
const uint8_t IUser::GetNumber ( ) [inline]
```

#### 6.4.3.2 GetPlacement()

```
virtual const uint32_t IUser::GetPlacement ( ) [pure virtual]
```

Implemented in [Player](#), and [Computer](#).

### 6.4.4 Member Data Documentation

#### 6.4.4.1 user\_number

```
uint8_t IUser::user_number {0} [protected]
```

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

- [src/iuser.h](#)

## 6.5 Player Class Reference

```
#include <player.h>
```

Inheritance diagram for Player:

Collaboration diagram for Player:

## Public Member Functions

- [Player](#) ()=default
- [Player](#) (uint8\_t)
- const uint32\_t [GetPlacement](#) () override

## Additional Inherited Members

### 6.5.1 Constructor & Destructor Documentation

#### 6.5.1.1 [Player\(\)](#) [1/2]

```
Player::Player ( ) [default]
```

#### 6.5.1.2 [Player\(\)](#) [2/2]

```
Player::Player (
    uint8_t number )
```

### 6.5.2 Member Function Documentation

#### 6.5.2.1 [GetPlacement\(\)](#)

```
const uint32_t Player::GetPlacement ( ) [override], [virtual]
```

Get players input between 0 and 6

Implements [IUser](#).

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

- [src/player.h](#)
- [src/player.cpp](#)

## Chapter 7

# File Documentation

### 7.1 docs/pages/about.dox File Reference

### 7.2 README.md File Reference

### 7.3 src/board.cpp File Reference

```
#include <algorithm>
#include <string_view>
#include "board.h"
#include <fmt/color.h>
#include <fmt/core.h>
#include <fmt/format.h>
#include <fmt/ranges.h>
Include dependency graph for board.cpp:
```

### 7.4 src/board.h File Reference

```
#include <array>
#include <utility>
#include <cstdint>
Include dependency graph for board.h: This graph shows which files directly or indirectly include this file:
```

#### Classes

- class [Board](#)

#### Variables

- constexpr uint8\_t [BOARD\\_HEIGHT](#) {6}
- constexpr uint8\_t [BOARD\\_WIDTH](#) {7}

## 7.4.1 Variable Documentation

### 7.4.1.1 BOARD\_HEIGHT

```
constexpr uint8_t BOARD_HEIGHT {6} [constexpr]
```

### 7.4.1.2 BOARD\_WIDTH

```
constexpr uint8_t BOARD_WIDTH {7} [constexpr]
```

## 7.5 src/computer.cpp File Reference

```
#include <random>
#include "computer.h"
#include "board.h"
Include dependency graph for computer.cpp:
```

## 7.6 src/computer.h File Reference

```
#include "iuser.h"
Include dependency graph for computer.h: This graph shows which files directly or indirectly include this file:
```

### Classes

- class [Computer](#)

## 7.7 src/game.cpp File Reference

```
#include <chrono>
#include <iostream>
#include <thread>
#include "game.h"
#include <fmt/printf.h>
Include dependency graph for game.cpp:
```



## 7.8 src/game.h File Reference

```
#include <memory>
#include "board.h"
#include "computer.h"
#include "player.h"
```

Include dependency graph for game.h: This graph shows which files directly or indirectly include this file:

### Classes

- class [Game](#)

### Variables

- constexpr uint8\_t [PLAYERCOUNT](#) {2}

#### 7.8.1 Variable Documentation

##### 7.8.1.1 PLAYERCOUNT

```
constexpr uint8_t PLAYERCOUNT {2} [constexpr]
```

## 7.9 src/iuser.h File Reference

```
#include "board.h"
```

Include dependency graph for iuser.h: This graph shows which files directly or indirectly include this file:

### Classes

- class [IUser](#)

## 7.10 src/main.cpp File Reference

```
#include <iostream>
#include <string>
#include "game.h"
```

Include dependency graph for main.cpp:

### Functions

- auto [main](#) () -> int

### 7.10.1 Function Documentation

#### 7.10.1.1 main()

```
auto main ( ) -> int
```

## 7.11 src/player.cpp File Reference

```
#include "player.h"  
#include <iostream>  
#include <fmt/printf.h>  
Include dependency graph for player.cpp:
```

## 7.12 src/player.h File Reference

```
#include "iuser.h"  
Include dependency graph for player.h: This graph shows which files directly or indirectly include this file:
```

### Classes

- class [Player](#)