

C Programmierkurs

TicTacToe (Games)

Institut: Beuth Hochschule für Technik Berlin
Dozent: Prof. Dr. Christian Forler
Url: <https://prof.beuth-hochschule.de/fileadmin/prof/cforler/index.html>
Email: [cforler\(at\)beuth-hochschule.de](mailto:cforler@beuth-hochschule.de)

Bei diesem Aufgabenblatt soll das Spiel TicTacToe implementiert werden.

Aufgabe 1 (4 Punkte) Die TicTacToe Bibliothek

Implementieren Sie unter anderem die folgenden Header Datei `tictactoe.h` und ggf. noch zusätzliche Helfermethoden die Ihnen das Leben erleichtern.

```
#pragma once

#include <stdbool.h>

#define BOARD_SIZE 3

enum ttt_entry { PLAYER_ONE = 'X', PLAYER_TWO = 'O', EMPTY_CELL = '_' };

struct ttt_ctx {
    char board[BOARD_SIZE][BOARD_SIZE];
    enum ttt_entry currentplayer;
    enum ttt_entry bot; // EMPTY_CELL == PvP Mode
};

void ttt_initialize(struct ttt_ctx *ctx);

void ttt_print_board(struct ttt_ctx *ctx);

bool ttt_full_board(struct ttt_ctx *ctx);

bool ttt_check_for_win(struct ttt_ctx *ctx);

void ttt_switch_current_player(struct ttt_ctx *ctx);

/* Diese Funktion gibt true zurück falls der derzeitige Spieler
 * einen validen Zug macht und wechselt den Spieler, ansonsten
 * gibt die Funktion nur false zurück.
 */
bool ttt_place_mark(struct ttt_ctx *ctx, int col, int row);
```

Aufgabe 2 (4 Punkte) TicTacToe das Spiel

Schreiben Sie ein Programm `ttt.c` die es zwei Spielern ermöglicht gegeneinander Tic-Tac-Toe zu Spielen. Verwenden Sie ihre Bibliothek aus Aufgabe 1.

```
-----  
| | | |  
-----  
| | | |  
-----  
| | | |  
-----
```

Player X. Please enter your coordinates (x y): 2 1

```
-----  
| | | |  
-----  
| | | X |  
-----  
| | | |  
-----
```

Player O. Please enter your coordinates (x y): 0 0

```
.  
. .  
-----  
| O | | X |  
-----  
| | O | X |  
-----  
| | | |  
-----
```

Player X. Please enter your coordinates (x y): 2 2

```
-----  
| O | | X |  
-----  
| | O | X |  
-----  
| | | X |  
-----
```

Sorry! Player O. You lost. :(
Player X. Congratulations! You won. :)

Aufgabe 3 (4 Punkte) TicTacToe Bot

Heute wollen wir einen Bot für unser TicTacToe Spiel implementieren.

Implementieren Sie unter anderem die folgenden Header Datei `tictactoe_bot.h` und ggf. noch zusätzliche Helfermethoden die Ihnen das Leben erleichtern.

```
#pragma once

#include <stdbool.h>

/* Der Bot soll erkennen, ob er in diesem Zug gewinnen kann. Falls ja,
 * soll er den entsprechenden Feld makieren und true
 * zurückgeben, ansonsten false. Vielleicht klappt es ja das nächste mal. :)
 */
bool ttt_bot_win(struct ttt_ctx *ctx);

/* Der Bot soll erkennen, ob er in diesem Zug eine Niederlage abwenden
 * kann. Falls ja, soll er den entsprechenden Feld makieren und true
 * zurückgeben, ansonsten false
 */
bool ttt_bot_rescue(struct ttt_ctx *ctx);

/* Der Bot makiert ein Feld. Überlegen Sie sich eine Gewinnstrategie.
 * Verwenden Sie dafür die beiden Funktionen ttt_bot_win() und
 * ttt_bot_rescue()
 */
void nextTurn(struct ttt_ctx *ctx);
```

Aufgabe 4 (4 Punkte) TicTacToe - Ultimate Edition (PvE und PvP)

Schreiben Sie ein TicTacToe-Spiel `ttt_ue.c`, welches den Zweispielermodus (PvP) und Einspielermodus (PvE) unterstützt.

Implementieren Sie unter anderem die folgenden Header Datei `tictactoe_game.h` und ggf. noch zusätzliche Helfermethoden die Ihnen das Leben erleichtern. Vergessen Sie nicht Ihren Code aus der Aufgabe 2 zu übernehmen.

```
#pragma once

#include <stdbool.h>

#include "tictactoe.h"

// starts the single player mode
void ttt_pve(struct ttt_ctx *ctx, enum ttt_entry bot_player);

// starts the player vs player mode
void ttt_pvp(struct ttt_ctx *ctx);

bool ttt_is_over(struct ttt_ctx *ctx);

// Prints "You Lost!", "Congratulations! You won.", or "Draw"
void ttt_print_win(struct ttt_ctx *ctx);

void ttt_reset_game(struct ttt_ctx *ctx);

void ttt_next_turn(struct ttt_ctx *ctx);
```