

Assignment #1 Design/Testing

Understanding the Problem

This problem is asking me to create a program that will operate like the game connect four, but it will be able to support an $n \times m$ board with connecting p pieces. The program will take in these values as command-line arguments with the options `-r`, `-c`, and `-p` to indicate rows, columns, and pieces, and these option/value pairs can be provided in any order. The program needs to handle inappropriate data, i.e. non-positive integers, and ask the user again for the correct input, but it doesn't have to recover from bad options. It must continue to play the game, until the user no longer wants to play, without creating a memory leak.

Devising a Plan/Design

Create a Makefile for running the multiple files of the overall program.

Construct a struct in the file `connect_four.h` that reads exactly like the following: `struct game { char **board; int r, c, p; char p1; char p2; };`

Create a program file named `connect_four.cpp` that will contain all the programs functions except the main function.

Create `bool is_valid_arguments(char *info[])` function that will the input for the size of the board are valid.

Create `void set_game_info(game *, char *info[])` function input the information into the pervious created structure, to its corresponding variables.

Create `char** create_table(int, int)` function that will take the dimensions inputted for the size of the game board, and actually create it according to that given information.

Create `bool check_winner(game)` function that overall will check for the winner of the game as the game is played.

Check if there's a winning connection in the horizontal, diagonal, or vertical directions.

Continue game/loop if winner is not yet determined.

Once a winner is determined, print out a message of the winner, and ask the user if they would like to play again.

Create `void play_game(game *)` function that will mostly facilitate the playing of the game.

Prompt the user what color they would like to be for player one, which will also set the remaining color for player two.

Prompt the user which column they would like to place their piece, initializing the game board as game is played.

Print an error message and recover when user inputs a non-existing column or an invalid input.

After each play, call the `check_winner` function, which will determine the activity of the loop that continues the game.

Create void delete_table(game *) function that will delete the information for the creation of the game board that was placed into the initial structure, in order to prevent a memory leak as if the user wants to play the game multiple times.

There will be a loop that will delete the

Create a program file named play_cf.cpp that will hold the overall program's main function.

Create main(int argc, char *argv[]) function.

Testing

Program complies, but will not run/print anything.

Input Values	Expected Output	Did Actual Meet Expected?
-	-	-