

Home exam for PG3400

Kjetil Raaen

November 2016. Deadline Thursday 15th of December

Instructions

This assignment is graded A-E with F for fail and counts 40% of your final grade. Competent and idiomatic use of the programming language C is more important than fancy features while grading. A short description of how the program and your thought processes have worked will help grading. Try to avoid memory leaks and buffer overflows. The product should compile and run under Ubuntu 16.10.

The delivery should contain the code and a `readme.txt` which explains any decisions made in the assignment.

The game “Othello”

The purpose of this assignment is programming a functional game of “Othello” also known as “Reversi”. While the rules of this game are simple, playing the game well requires both practice and theory. You can find all rules of the game on Wikipedia.

Because we have not worked with any graphics library in this course, this assignment uses simple console input and output. I have attached code to make and print the board.

Requirements

- The game should ask for the names of the players.
- Then it should print the board and prompt for input, mentioning the player by name.
- The input should be a field on the board, denoted by letter and number, such as “b3”.
 - If the player enters invalid input, they should be allowed to try again.
 - This input should ideally accept some variants, such as “b 3” and “B,3”. That is, upper and lower case as well as space or comma between coordinates.
- The system should then check if the move is legal, and prompt again if not. (Hint: This should not be 8 individual checks for the directions, rather one check repeated 8 times.)
- If the move is legal, the system should update the board by the rules of the game, and print the new board.
- The game should check if the game is finished, and if not ask the next player to play.

- Note that if one player cannot move, the other player can take multiple turns.
- If the game is finished the winner should be announced and the program should exit.
- During the game, each move should be recorded and saved to a file. Describe the format in attached documentation.
- Include a makefile that compiles the project correctly, compiling only the files changed since last time.

In our grading criteria, “independence” is considered critical factor for top grades. The following is not covered in detail in the lectures, and thus will only be relevant for the highest grades. To show a high degree of independence, replace the text output with a graphical user interface. You are free to use whichever library you like. Suggestions include Cairo, GTK+ , SDL or ncurses. Other independent improvements are also appreciated.