Program Design & Testing Document for Program 6
Lyell C Read

Problem Statement

The problem asks me to create a c++ version of the Connect4 game. This game must:

- Include a readme.txt file in the directory with the cpp
- CLI args will be used to determine [players] [x len] [y len]
- Will print the grid with numbers across the top
- 1 or 2 player functionality
-

More requirements (credit: Justin Goins, Canvas):

- Your program must display the updated game grid after each move.
- If a winner exists, the program must immediately declare the winner and ask if the player(s) want to play again.
- If no more moves are possible and no winner exists (i.e. the entire grid is full of tokens), the program must declare the game a tie and then prompt the player(s) to start a new game.
- Print an error message and recover when the player supplies an invalid column. This could be a column that doesn't exist ("Cat", -4, 142, etc) or it could be a column that is already full of tokens.
- For this program you must validate the incoming command line arguments and ensure that each of the three values is a non-negative number. If an invalid value is provided (negative number, floating point number, text string, etc) then the program must halt execution and display a message to indicate the problem.
- Play the game correctly based on rules and number of players.
- Continue to play until the user selects no.
- You must not have any global variables
- You must use a dynamic 2-dimensional array to represent the grid of tokens.
- The computer player must follow the rules of the game and can only drop tokens in columns that have at least one open space.
- Your functions need to focus on performing a particular task. In other words, you need to use good modular design. If your function uses more than about 20 lines of code, this may be an indication that the code should be split into multiple functions. If the TA notices that your code is not sufficiently modular, you will lose points.
- You must not have memory leaks.
- Segmentation faults are not allowed (e.g. your program crashes).

Understanding the Problem

As described in the requirements, the problem asks me to create a game of connect with the above listed constraints. This game will start with the user calling the executable with a parameter of players, and grid size (x,y).

Pseudo Code (Simplified)

input_Check (low, high)
    Returns input if on range and isint,
    Calls input_Check if not

check_Vertical (ptr --> grid)
    Iterates over all col's
    If there is a string of 4, then return the player that owns that string
    Otherwise, return -1

check_Horizonal (prt --> grid)
    Iterate over all horizontals
    If someone as a string of 4 in a row, return their number
    Else return -1

check_Diagonal (prt --> grid)
    Iterate over grid and check for diagonals
    If there is one, return the player's number
    Else return -1

check_All (ptr --> grid)
    Defines new array of values [3] from three checks
    If sum of all elements is -3, return -1
    Else return the first instance != -1 in the array[3] which would be the winner
    Delete array

draw_Grid (ptr --> grid)
    Draw grid
    Return

play_OK (ptr --> grid, col)
    Returns if a move in col is playable, or if that column is full

play_Value (ptr --> grid, value)
    Play the value in the column, dropping it to the right place.

```
player_Turn(prt --> grid, player)
        Announce whose turn it is
        Print the grid ?
        Ask where to play
        Check if play_OK (...), otherwise ask again.
        Play that value
        Print the grid

computer_Play (ptr --> grid)
        Play randomly in any non empty column
        Print grid

main(arguments)
        Check arguments
        Create a grid dynamically
        While cont = 1
                Set win to false
                While win is false,
                        If 1 player
                                Player plays
                                Win check, update?
                                Computer plays
                                Win check, update?
                        Else
                                For x in 0..1
                                        Player play
                                        Win check, update?

                Set cont to user input on “Play Again”

        Delete array
        Exit success
```

Predicted Results for Only Integer Input on range 0..1

Value	**What Should Happen**	**Does This Happen**
“”	Error - nothing entered	yes
“2”	Error - out of range	yes

“oasdjfasone”	Error - improper formatting	yes
“;”	Error - improper formatting	yes
“ “	Error - improper formatting	yes

Predicted Results for Execution input

Value	**What Should Happen**	**Does This Happen**
./a.out 2 x x | x on 1..infinity	2 player game	yes
./a.out 89k 2 3	Error - improper formatting	yes
./a.out 2 j 9	Error - improper formatting	yes