Design Report for Custom Program

Assignment 1 of COS10009

Introduction to Programming

Semester 1,2018

Written by Darren Frederick Chan Chyun Yaw

Student ID: 100085926

Date: 11/5/2018

# User Manual

1. Execute the application program.
2. Enter your name.
3. Press any key to continue to the game screen.
4. Use ‘R’ or ‘r’ keys to go right and ‘D’ or ‘d’ to go downwards.
5. You may press ‘Q’ or ‘q’ to quit the game at any time.
6. The game ends when the player reaches the bottom right most number in the board.

# Program Design

When the player executes the program, in the main function, an integer variable named ‘a’ is initialized and a struct named init is made. Afterwards, the function startScreen is called with the address of struct name init. In the startScreen function, a char variable called input is initialized with a size of 1000. A welcome text is printed and the readFile function is called. The readFile function prints the name and score of previous players on to the scoreboard from a text file named score. Following that, the player is asked to input their name which is then stored in the struct variable called name in the init struct, which is passed into the startScreen function, this is done by using pointers and functions, specifically, strcpy. The input is then flushed. The game then greets the player by name and prompts the player to press any key to continue, this is done by using \_getch(). The screen is cleared. The startScreen function ends.

Back in the main function, the grid() function is executed. In the grid() function, many types variables are declared and initialized. However, with a myriad of variables declared and initialized, we look at the more important variables which are the 2d array and the movement and tracking mechanism. To delve into the topic, the 2d array was initialized to a size of [15][8]. Furthermore, the movement mechanism involved the declaration of two arrays, titled row\_sel and col\_sel, their function is to keep track of the position the player by storing the x-axis and y-axis data in the row\_sel and col\_sel arrays. Both arrays come initialized with the starting point and ending point, which is (0,0) and (14,7). After declaring the variables, the game is then seeded to ensure that at every execution of the program, different numbers are displayed and stored in the 2d array. This is done by using the code below

srand((unsigned)time(NULL))

This seed’s the game based on the time when executed. A for loop with a nested for loop is used to assign values of 0-9 randomly to the 2d array. Following this, a while loop is declared, it prints the 2d array to the screen followed by a input menu detailing the score and controls to the game. When a user inputs a character, the screen is cleared, the col\_box increments by 1 to “move” the cursor to the right, if the col\_box exceeds 7, it decremented by 1, this is to prevent bugs. Else, the position of the cursor is added to the row\_sel and col\_sel array and the score is increased based on the value in that position of the 2d array. The 2d array and input menu is printed again. This works vice versa for the other option, except for incrementing by 2 to the row\_box and if the row\_box exceeds 14, it is decremented by 2. If the player inputs ‘Q’ or ‘q’, the while loop is broken and the score is returned.

Back again in the main function, the integer variable ‘a’ now holds the score of the game. The next function to be executed is the endScreen function which is passed the ‘a’ variable containing the score and the address of the init struct. In the endScreen function, congratulatory text is printed and the score is displayed. The writeFunction is then executed to write the name and score to the score text file. The endScreen function ends and signals the end of the main function as well. The program has finished execution.

I’ve implemented several codes that are not covered in the course so far to my knowledge. I included the usage of string, Boolean, conio libraries for usage in my code. I used string copy or strcpy to copy the inputted string from the user into the struct variable. Furthermore, I used the Boolean variable foundMatch to track the position of the cursor. The for loop will run until foundMatch is true or “a ” is out of range. FoundMatch is set to true if there is any ‘a’ for which row==row[a] && col==col\_sel[a] is true. |%d| will print if foundMatch is true. Lastly, getche() was used to allow users to input without pressing Enter.

# Flowchart of main function

