Number Hangman

Jason Emerson

1st Iteration

CSC 5 - 48102

**Table of Contents**

1. Description of game
2. Core concepts used in the game learned in class
3. Layout of the game (screenshots included)
4. Psuedocode description of the game
5. Flowchart of the game
6. Continued Iteration (Project 2)

**1: Description**

Hangman has been around for a long time. Usually hangman is played with a series of words and as the player you would take a guess at a letter and attempt to complete the word. You are given a total of 6 lives (head, body, left arm, right arm, left leg and right leg). I wanted to make a new twist to the game with an attempt of making it more difficult. In this version of Hangman, you are given a series of number ranges. Your goal is to guess the number in that range within’ the 6 lives that you normally would get in a normal game of Hangman. You continue playing until all 6 lives have passed and your hangman has been hung so to speak.

**2: Core Concepts**

|  |  |
| --- | --- |
| **Lines of C++ code** | **230** |
| **Number of comment lines (function and program)** | **62 (beginning at System Libararies)** |
| **Blank Lines (separate functions)** | **27 (separating functions and processes)** |
| **Total Lines in Program (code, comments and blank lines)** | **319** |

|  |  |
| --- | --- |
| **Class Concepts** | |
| Do-While Loop | Main & showMenu() Function |
| While Loop | Main & Multiple Functions |
| If/Else Statements (including only if) | Multiple Functions |
| Switch Statements | Main & drawBoard() Function |
| Logical Operators |  |
| Arithmetic Operators |  |
| Relational Operators |  |
| Assignment Operators |  |
| Functions | Custom Functions Throughout |
| **Data Types** | |
| Int |  |
| Char |  |
| String |  |
| Boolean |  |
| **File I/O** | |
| Ifstream |  |
| ofstream |  |
| In.open(“filename”) |  |
| Out.open(“filename”) |  |
| In.close() |  |
| Out.close() |  |
| Out.clear() |  |
| **Random Number Function and Time** | |
| Srand() |  |
| Rand() |  |
| Time(0) |  |

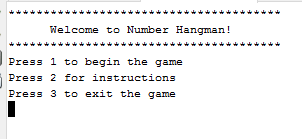
|  |  |
| --- | --- |
| **Libraries** | |
| Iostream | I/O Objects |
| Cstdlib | Random Number |
| Ctime | Random Number Seed |
| Fstream | File I/O |

**3: Game Layout**

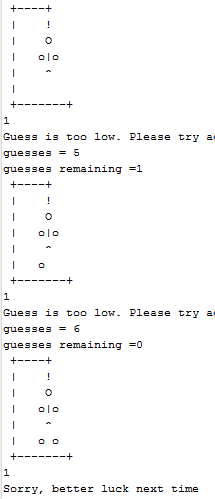
When you begin the game, you’re presented 3 options.

You can begin the game (starting from the range 1-10), read the instructions or

exit the game.



Once you begin the game you have 6 guess per range to guess the correct number. If you guess the correct number within’ the 6 guesses, the range will increase and you will have your 6 guesses reset. Each time you lose a life, your Hangman board will fill in another spot of the body. You lose the game by filling in the entire body.



Upon finishing the game you will be asked to enter your name and the game should write your name and score (score is equal to the total number of guesses you had left over) to the file for use for the next game. This is something I personally had issues getting to work properly and I hope to master this for Project 2.

**4: Psuedocode**

Player selects a choice from the menu

If choice 1

Then play the game

If choice 2

Then display instructions

If choice 3

Then display high score

If choice 4

Then exit the game

If player selects choice 1

Player inputs a guess from a given range

If guess = too high or too low

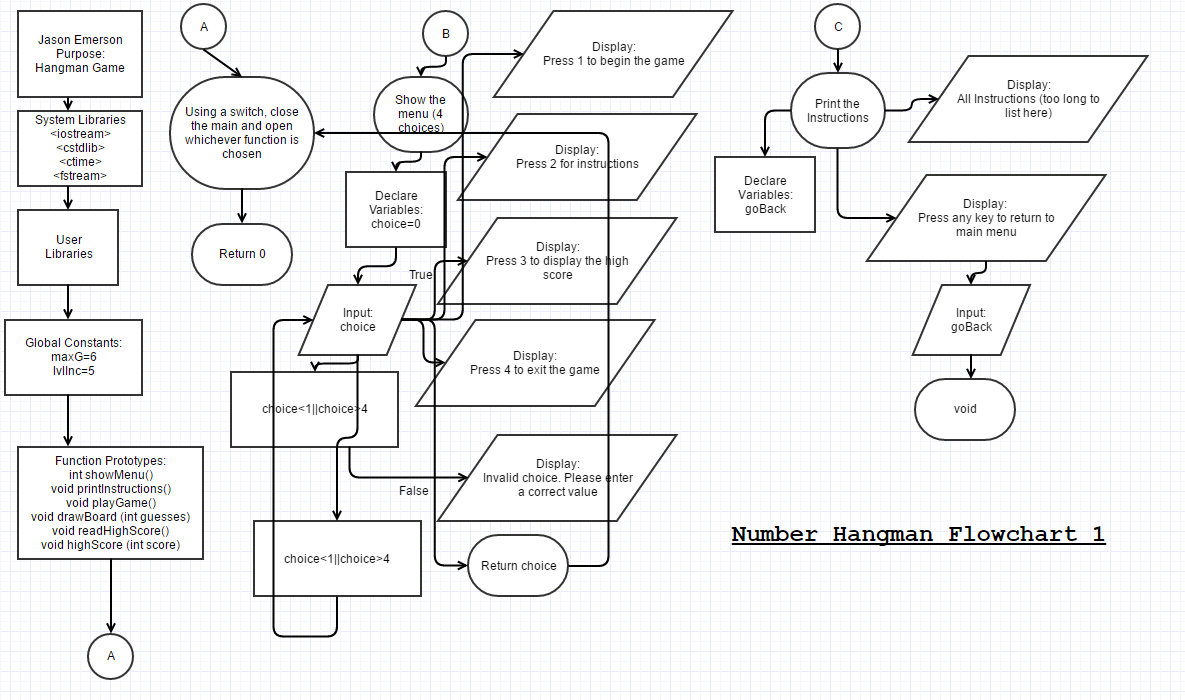
Then player inputs another guess and total guess -1

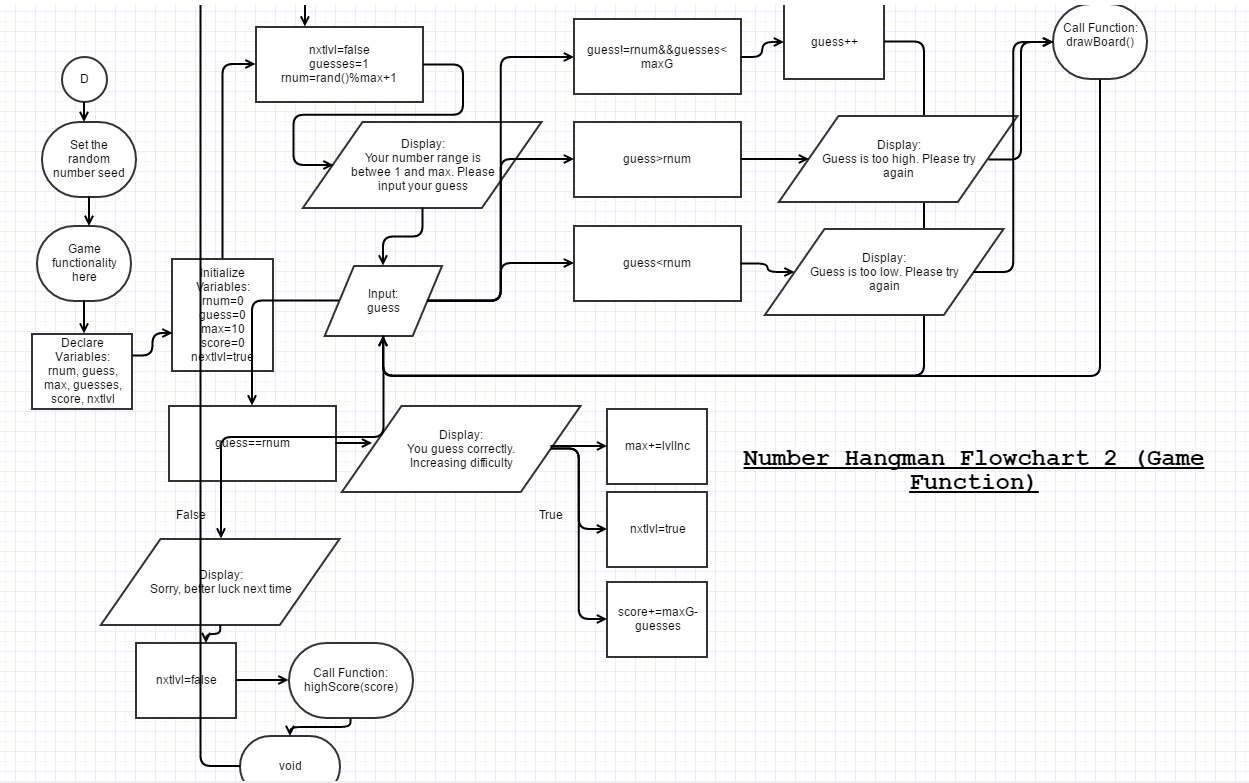
If guess = correct

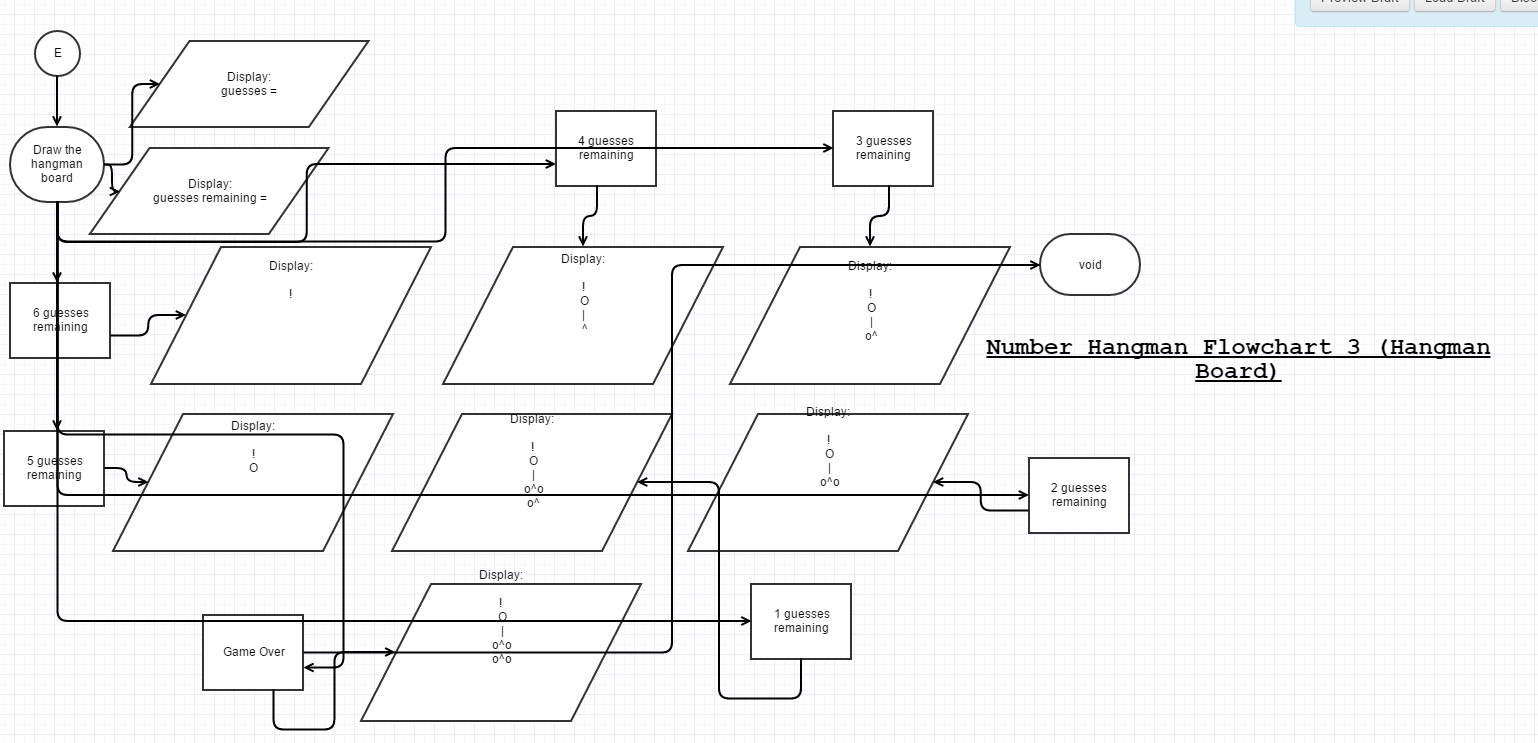
Then player beats that level and range increments +5

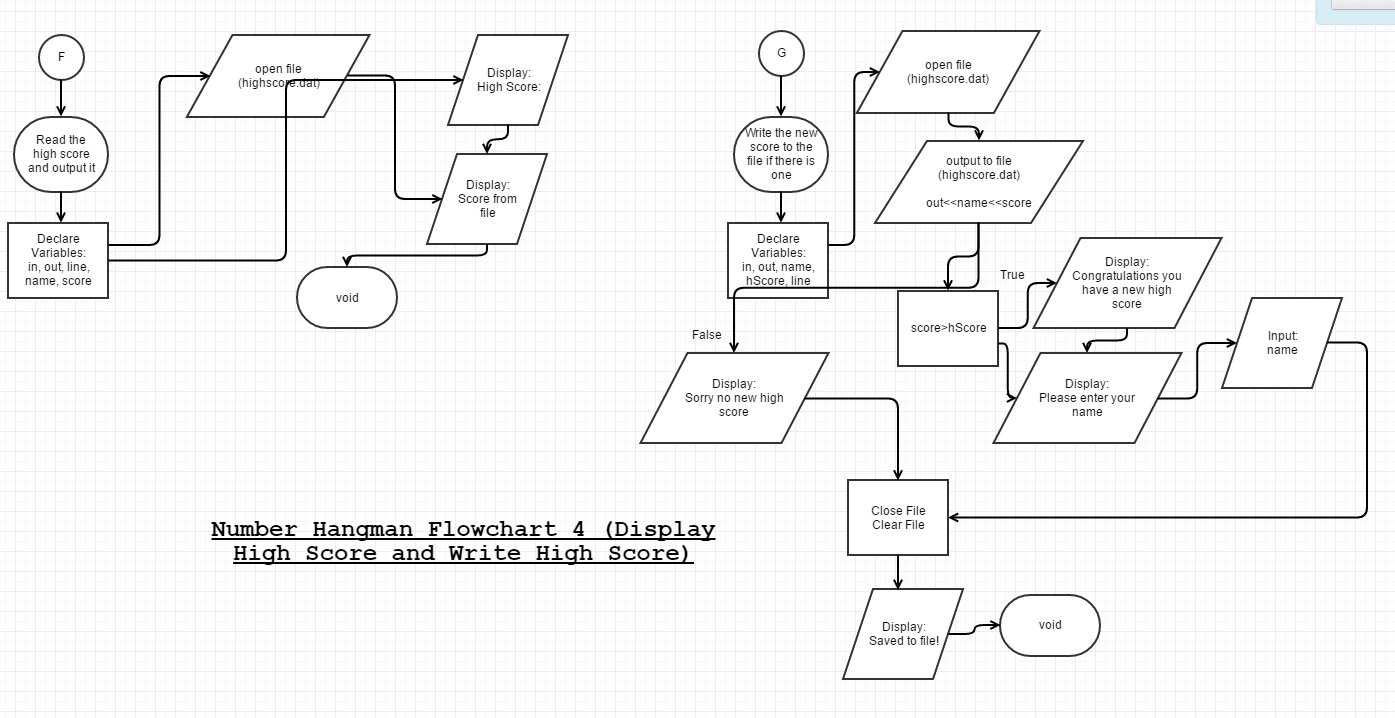
Repeat

**4: Flowcharts**

****

****





**5: Continued Iteration (Project 2)**

This project definitely gave me some trouble. I began this project creating one of my favorite casino games Blackjack. I was unable to get somethings to work properly and I am not as familiar with arrays so I was unable to complete the game how I had expected. With 4 days left until the project was due I decided to create Hangman but with a twist and I ended up with this. With this project most of it was pretty self explanatory except I had much difficulty using the File I/O system. I was unable to get it to write the name and score every time to the file. Usually only the score. For Project 2 I am looking to finish my Blackjack code, perfect it and polish it up with some visual adjustments I have in mind. If I were to finish this game for Project 2, I would like to fix the File I/O for the name as well as add support for multiple players and multiple high scores. I also would like to make the display of the game a bit more polished and presentable in terms of clearing lines and only showing what is going on at that moment.

All in all I had a lot of fun doing this project. I have some familiarity in Java as well as C++ so I had expected to make a game with ease and instead I had a very nice challenge creating this game and it required me to actually take a step back and think over just brushing everything aside.