Aryan Aziz

CS165 Final Project

1. **Explanation:** So, in the final project that I am actually submitting; it's a build off of the Week 10 assignment. This time, it includes all of the items that A15E2 had; with a few extra features.

   To be honest, I'm really disappointed about this. When I initially found out about the final project I had a lot of really cool ideas that I started building out; including a Reddit Reader and rudimentary advertising server (I work in the advertising industry); but I couldn't find usage for a lot of the requirements that you were asking for. So instead I had to settle for re-using something from before and just converting stuff to add the requirements. I did get a working version of the reddit reader running that pulls the Reddit JSON and shows you the post, upvotes, and downvotes however I don't have enough knowledge in making it cross-OS compatible.

   I also had a working version of the adserver running but I hit the same issue of not having enough of the items cross off the list.

   As for the design itself, I used my previous designs and then modified it to break it out correctly and added the new functions and features.

   New features include:
   a. A High scores system. After every game, the user's name is entered into a High scores document and stored for all eternity. A user can view the high scores by pressing 2 at the main menu
   b. A "layout view"; for people having a hard time visualizing the layout of the nodes it displays how the board is laid out (not with the connections because that would make it too easy)
   c. A cool new splash screen.
2. **Reflection:** It's very easy, and just as equally hard using all of these items in a single program. I realized there are so many different ways to do the same thing that everyone could end up writing the same program completely differently.

   Over the course of this class, I think what opened my eyes the most is the ability of not being able to just jump right in. I am a very hands on, no planning type of guy and when I get an assignment/project/ikea furniture I like to just jump right in and put it together without a plan or map. I learned the hard way in this class that the most important thing is to start with a good outline with basic ideas; and as you go along you'll see little pieces that you need to add.

I really went into this course, and the program in general, thinking I knew everything and everything would be fine; but I realized how little I know and how much I need to learn.

3. Requirements:
    a. Simple Output
        i. main.cpp – line 365
            1.
            ```
            std::cout << "Welcome to the MazeGame!" << std::endl; //######## Requirement #1 ########
            ```
    b. Simple Input
        i. main.cpp – line 368 (variable defined on line 34)
            1.
            ```
            std::cin >> name; //######## Requirement #2 ########
            ```
    c. Conditionals
        i. main.cpp – line 366
            1.
            ```
            if(argc <= 1) { //######## Requirement #4 ########
            ```
    d. Logical Operators
        i. main.cpp – line 250 (defined on line 249)
            1.
            ```
            while(validChoice == false) { //######## Requirement #5 ########
            ```
    e. Loop
        i. main.cpp – line 241 (defined on line 29)
            1.
            ```
            while(finish == false) { //######## Requiremeent #6 ########
            ```
    f. Random Number
        i. main.cpp – line 273
            1.
            ```
            int randomNumberGenerator(void) { //######## Requirement #7 ########
                 int randomNum;
                 srand(time(NULL));
                 randomNum = rand() % 100 + 1;
                 return randomNum;
            }
            ```
    g. Error Categories
    h. Function
        i. Main.cpp – line 207
            1.
            ```
            void makeMove(node *x) { //######## Requirement #10 ########
            ```
    i. Functional Decomposition
        i. Main.cpp – all lines
    j. Variable Scope
        i. Main.cpp – line 249 and line 115
            1.
            ```
            bool validChoice = false; //######## Requirement #12 ########
            ```
    k. Passing Mechanisms

- i. Main.cpp – line 87 and line 169 and line 184
  - 1. `char getMove(node x) { //######## Requirement #13 ########`

l. Function Overloading

- i. Node.cpp – line 4 and line 7

m. String

- i. Main.cpp – line 34
  - 1. `std::string name; //######## Requirement #15 ########`

n. Recursion

- i. Main.cpp – line 235
  - 1. `makeMove(goThere(x, getMove(*x))); //######## Requirement #16 ########`

o. Multi-Dimensional Array

- i. Main.cpp – line 286
  - 1. `boardArray[i] = new char[boardArrayY]; //######## Requirement #17 ########`

p. Dynamically Declared Array

- i. Main.cpp – line 284
  - 1. `boardArray = new char*[boardArrayX]; //######## Requirement #18 ######## Requirement #22 ########`

q. Command Line Argument

- i. Main.cpp – line 371
- ii. Usage, run the program with your name after
  - 1. `aziza::getName(argv[1]); //######## Requirement #19 ########`

r. Struct

- i. Main.cpp – line 81
  - 1. `struct game { //######## Requirement #20 ########`

s. Class

- i. Main.cpp – line 65
  - 1. `class InputSource { //######## Requirement #21 ########`

t. Pointer to an array

- i. Main.cpp – line 284
  - 1. `boardArray = new char*[boardArrayX]; //######## Requirement #18 ######## Requirement #22 ########`

u. Pointer to an object

- i. Main.cpp – line 43
  - 1. `x->setNorthBool(north); //######## Requirement #24 ########`

v. Namespace

- i. Main.cpp – line 41

1. `namespace aziza { //######## Requirement #25 ########`

w. Header file

  i. node.h or main.cpp – line 26

    1. `#include "node.h" //######## Requirement #26 ########`

x. Makefile

  i. Makefile

y. Vector

  i. Main.cpp – line 37

    1. `std::vector<node> roomList; //######## Requirement #28 ########`

z. Default Constructor

  i. Node.cpp – line 4 and line 7

aa. File IO

  i. Main.cpp – line 184

    1. `void writeHighScore(game& x) { //######## Requirement #31 ########`

bb. STL effects

  i. Vector, see section y above

cc. Inheritance

  i. Main.cpp – line 70

    1. `class CinInputSource : public InputSource { //######## Requiement #33 ########`

dd. Polymorphism

  i. Main.cpp – line 72

    1. `virtual char getCommand() { //######## Requiement #34 ########`