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:

* 1. 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
  2. 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)
  3. A cool new splash screen.

1. **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.

1. Requirements:
   1. Simple Output
      1. main.cpp – line 365
         1. std::cout << "Welcome to the MazeGame!" << std::endl; //######## Requirement #1 ########
   2. Simple Input
      1. main.cpp – line 368 (variable defined on line 34)
         1. std::cin >> name; //######## Requirement #2 ########
   3. Conditionals
      1. main.cpp – line 366
         1. if(argc <= 1) { //######## Requirement #4 ########
   4. Logical Operators
      1. main.cpp – line 250 (defined on line 249)
         1. while(validChoice == false) { //######## Requirement #5 ########
   5. Loop
      1. main.cpp – line 241 (defined on line 29)
         1. while(finish == false) { //######## Requiremeent #6 ########
   6. Random Number
      1. main.cpp – line 273
         1. int randomNumberGenerator(void) { //######## Requirement #7 ########

int randomNum;

srand(time(NULL));

randomNum = rand() % 100 + 1;

return randomNum;

}

* 1. Error Categories
  2. Function
     1. Main.cpp – line 207
        1. void makeMove(node \*x) { //######## Requirement #10 ########
  3. Functional Decomposition
     1. Main.cpp – all lines
  4. Variable Scope
     1. Main.cpp – line 249 and line 115
        1. bool validChoice = false; //######## Requirement #12 ########
  5. Passing Mechanisms
     1. Main.cpp – line 87 and line 169 and line 184
        1. char getMove(node x) { //######## Requirement #13 ########
  6. Function Overloading
     1. Node.cpp – line 4 and line 7
  7. String
     1. Main.cpp – line 34
        1. std::string name; //######## Requirement #15 ########
  8. Recursion
     1. Main.cpp – line 235
        1. makeMove(goThere(x, getMove(\*x))); //######## Requirement #16 ########
  9. Multi-Dimensional Array
     1. Main.cpp – line 286
        1. boardArray[i] = new char[boardArrayY]; //######## Requirement #17 ########
  10. Dynamically Declared Array
      1. Main.cpp – line 284
         1. boardArray = new char\*[boardArrayX]; //######## Requirement #18 ######## Requirement #22 ########
  11. Command Line Argument
      1. Main.cpp – line 371
      2. Usage, run the program with your name after
         1. aziza::getName(argv[1]); //######## Requirement #19 ########
  12. Struct
      1. Main.cpp – line 81
         1. struct game { //######## Requirement #20 ########
  13. Class
      1. Main.cpp – line 65
         1. class InputSource { //######## Requirement #21 ########
  14. Pointer to an array
      1. Main.cpp – line 284
         1. boardArray = new char\*[boardArrayX]; //######## Requirement #18 ######## Requirement #22 ########
  15. Pointer to an object
      1. Main.cpp – line 43
         1. x->setNorthBool(north); //######## Requirement #24 ########
  16. Namespace
      1. Main.cpp – line 41
         1. namespace aziza { //######## Requirement #25 ########
  17. Header file
      1. node.h or main.cpp – line 26
         1. #include "node.h" //######## Requirement #26 ########
  18. Makefile
      1. Makefile
  19. Vector
      1. Main.cpp – line 37
         1. std::vector<node> roomList; //######## Requirement #28 ########
  20. Default Constructor
      1. Node.cpp – line 4 and line 7
  21. File IO
      1. Main.cpp – line 184
         1. void writeHighScore(game& x) { //######## Requirement #31 ########
  22. STL effects
      1. Vector, see section y above
  23. Inheritance
      1. Main.cpp – line 70
         1. class CinInputSource : public InputSource { //######## Requiement #33 ########
  24. Polymorphism
      1. Main.cpp – line 72
         1. virtual char getCommand() { //######## Requiement #34 ########