|  |  |
| --- | --- |
| Author 1: | Jonathan Mabery |
| Email 1: | jmabery@uco.edu |
| Author 2: | Andrew Aprile |
| Email 2: | [aaprile@uco.edu](mailto:aaprile@uco.edu) |
| Course: | ADS. CMSC3613. CRN10040 |
| Assignment | Project 2. p02 |

**Give a brief discussion of your implementation: just like an explanation of your idea in an interview.**

**Answer:**

When we run our program, we input command line arguments in the form of the input and output files.

We made helper functions to output the target word and grid to the terminal and also the output file.

For our **read\_input() function**, we create a stream to hold the input called inputFile. We get the first line of the file and store the work into our target variable which is passed by reference to the function.

The second line of the file is the row and column number. We collect and store those into variables.

To setup the grid, we do two for loops; one to loop through the number of rows and one to look through the number of columns. At each grid box, we get the character from the file and store into the grid using the predefined set() function.

For the **search() function**, we check to see if the target character returns the \0 character. This indicated the end of a word of char type.

We next compare the character in the word with the character in the grid. If it doesn’t equal, we exit the search function. If it does equal, we continue.

The next part is the recursive calls. Before doing that, we need to mark the grid square with a temporary symbol so that the function ignores it when doing the 4 way search. Which symbol should we use you may ask? Well we used the \0 symbol; the same for indicating that we got to the end of the target word.

Once the function finds all the target characters, it will backtrack through each function call. When it does so, we can replace the \0 with the character that was in each grid box originally.

**Please also brief explain:**

1. **Why there is a nested loop in the main function to initiate the search procedure, compared to other examples of backtracking in the lectures?**

**Answer:**

This is to save time. It only calls the search function if you can find the first letter of the word in the

1. **What’s the purpose of using pointers and references in the function headlines, e.g., string& target, const char\* target, Grid<char>\* grid?**

**Answer:**

1. **How would you like to handle const char\* target part in the function of bool search(Grid<char>\* grid, int r, int c, const char\* target)?**

**Answer:**

A screenshot of a test run.