Charlie Wyman

Gavin Tate

Parker Shore

Thomas Domser

CPSC 223 Shaded Trees Project - Design Document

Design Overview

This project runs with three code files (main.cpp, linked_list.hpp, and linked_list.cpp) and one data file. Our main.cpp file creates a StreetList object which calls all the necessary functions to run the program. The primary function of main.cpp is to begin the program; all of our user input processing and file data processing is taken care of in linked_list.cpp, which allows us to have access to any data we need wherever we need it.

Program Start Design

When the program begins it will run the setup() function inside linked_list.cpp. The setup function gives us a comprehensive way to call all the necessary functions required to begin the program. This includes collecting all data from the data file – street names, number of trees, and distance between trees – and storing them in a way that can be easily navigated by the backend. Street names are divided into two lists: one list of north/south streets, and one list of east/west streets. Even though this project only requires us to include east/west streets, we decided to show the user which north/south streets they are between to keep track of what block they are on.

Once the setup function has opened the data file and created the street lists, it then calls the functions “createAllNodes()” and “initializeAllNodes()” which create the framework for the

linked list and populate the linked list, respectively. The format of the linked list uses pointers that are stored as private data types in the linked_list class. The linked list framework is then populated in the initializeAllNodes() function using a nested for loop that iterates through the 2d matrix of tree amounts, storing each corresponding value from the data file to the “tree pointer” variable in the linked list.

Before setup is complete, it calls the “initializeAllDistances()” function, which reads all the tree distance information from the data file. This works logically very similar to the previous section, as it uses a pointer variable to store each of the tree distance information into the corresponding location of the linked list. Resultantly, each node on the linked list is populated with four pointers: a “previous” pointer, a “treeAmount” pointer, a “treeDistance” pointer, and a “next” pointer.

Program Running Design

Once the program is finished setting up, it will print a list of all the available street blocks to the user and prompt the user on which one they would like to select. The user will choose a number that corresponds with the street they would like to select. Once again, this is handled in the linked_list.cpp file. After the user inputs where to go, the program will show them what block they have chosen, as well as how many trees are on that block, and how far apart the trees are (in feet). The program will then prompt the user to move forwards in the list, backwards in the list, or exit the list entirely – allowing the user to choose a new street from anywhere on the list. This format will run indefinitely as a while loop, until the user inputs that they would no longer like to select a street (n). When that happens, the program ends.