

Charlie Wyman

Gavin Tate

Parker Shore

Thomas Domser

CPSC 223 Shaded Trees Project - Specifications Document

Front-End Features

- *Selecting a Street*
 - Purpose: The user is asked to select a street at the start of the program. This gives them the ability to choose a jumping-on point into the linked list, after which they can navigate through the list forward and backwards.
 - Assumptions: The user is expected to have a way to interact with the program.
 - Inputs: The program will print the street blocks out to the user in a numbered list. The user will then be prompted to input the number that corresponds with the street block that they want to select.
 - Outputs: Once the desired street is selected, the program will output information on that street to the user – number of trees and distance between trees.
 - State Changes: A state change will occur when the user inputs a number.
 - Expected Behavior: If the user inputs a number that isn't on the list, or if they input any characters other than a number on the list, they will be notified that the input is invalid, and will be prompted to re-enter a valid number.
- *Navigating Forward*

- Purpose: The user will be given the option to navigate forward on the linked list, allowing them to iterate through each node on the list.
- Assumptions: The user is expected to have a way to interact with the program.
- Inputs: When the user is on a street (node), they will be prompted to input 'f' for forward. If they input 'f', the program will take them to the next node.
- Outputs: The program will output the information of the new node they traveled to – tree amount and tree distances.
- State Changes: A state change occurs when the user inputs 'f'.
- Expected Behavior: When the user inputs 'f', the program is expected to iterate to the next node unless the user is already at the end of a street, or is at the end of the entire list. If this is the case, the program will notify the user that they have reached the end of the street and will not be able to go forward.
- *Navigating Backward*
 - Purpose: The purpose of this feature is to allow the user to navigate backwards through the street list, one node at a time.
 - Assumptions: The user is expected to have a way to interact with the program.
 - Inputs: The user is prompted to input 'b' to navigate backwards.
 - Output: After the user navigates backwards, the program will output the information of the previous node to the user – tree amounts and tree distances.
 - State Changes: A state change occurs when the user inputs 'b'.
 - Expected Behavior: After input, the program will take the user to the previous node in the list. However, if the user is already at the starting end of a street, they

will be notified that they cannot move back on that street any further, and the program will ask them for a different input.

- *Printing All Lists in Geographic Representation*

- **TBD**

- *Graceful Exit*

- Purpose: The purpose of this feature is to allow the user to easily exit the program when they are finished.
 - Assumptions: The user is expected to have a way to interact with the program.
 - Inputs: The user will input the letter 'n' after prompted.
 - Outputs: The program will output "Exiting traversal." to let the user know that they have exited the program.
 - State changes: A state change will occur when the user inputs 'n'.
 - Expected behavior. Every time the user navigates to a new node on the list, they will be asked if they would like to leave the list (n). After this, the user has exited the list, but is asked if they would like to choose a new street from the list (y/n). If the user enters 'n', the program will print "Exiting traversal" and will end.

Back-End Features

- *Insert*

- Purpose: The purpose of this feature is to insert variables into the linked list structure.
 - Assumptions: This feature assumes that the format of the linked list and its associated pointer variables has already been constructed.

- Inputs: The data file will input all the tree amounts and tree distances into the linked list.
- Outputs: This will allow the program to output tree amount and distance data to the user from the corresponding node that the user has chosen.
- State Changes: A state change occurs every time new tree data is inserted from the file, as this extends the linked list by one node.
- Expected Behavior: The linked list is expected to have access to the information that has been inserted into it from the file.
- *Printing*
 - Purpose: The purpose of printing from the backend is to have a way to convey information with the user and request inputs from the user.
 - Assumptions: The user is expected to have a way to read the information printed from the program.
 - Inputs: The printing aspect of this program requires input from the data file.
 - Output: The information from the data file will be output for the user in a way that is comprehensible and readable.
 - State Changes: A state change occurs every time the program navigates to a new node on the linked list, as each node has different information that must be printed for the user to read.
 - Expected Behavior: The program is expected to print the list of street names at the start, and is also expected to print the tree amount and distance information of each node as the user navigates between different nodes on the list.
- *Search*

- Purpose: The purpose of this feature is to allow the user to select any node on the list as a starting point to enter the list.
- Assumptions: The user is expected to have a way to interact with the program.
- Inputs: The user will input a number that corresponds with the number of the street (node) that they would like to navigate to on the list.
- Outputs: When the user selects the desired node on the list, the program will output the information of the node they selected – tree amounts and tree distances.
- State Changes: A state change will occur when the user selects a node. A state change will also occur if the user exits the list and decides to choose a new street from the list.
- Expected Behavior: The program is expected to navigate to any specific node on the list after the corresponding input.