Oregon State University
College of Engineering

CS Capstone Progress Report

February 16, 2018

Ancestry Data Viewer

Prepared for

Ashley McGrath _____ Signature _____ Date

Prepared by

Group 22

Team Ancestry Data Viewer(ADVR)

YongPing Li _____ Signature _____ Date

Monica Sek _____ Signature _____ Date

Le-Chuan Chang _____ Signature _____ Date

Abstract

The purpose of our project is to read in a GEDCOM file and display its information onto an ancestry tree. The purpose of this progress report is to illustrate what we have done in the past 10 weeks and identify any problems that have impeded us and explain our solutions to these problems.

1 Introduction

The purpose of our project is to read in a GEDCOM file and display its information onto an ancestry tree. The GEDCOM file needs to be parsed through to make it easier for the software to read and to gather the required data. The ancestry tree will be displayed in a clear graph that contains series of parent and children nodes. The software has two applications: desktop and VR. A user should be able to toggle between desktop and VR mode if a VR headset is presented. In addition, the following special features should be included in the software: choosing a subsection of the tree, finding a common ancestor between two nodes (members), and toggling between a full ancestry tree and a direct lineage tree.

2 Yong Ping Li

3 Monica Sek

4 Le-Chuan Justin Chang

4.1 Introduction

In general, the current status of the components listed are that they are designed, but not implemented. The largest roadblock is primarily time management, as the capstone tended to be prioritized last, since there were no set goals. This is an issue that hasn't yet been resolved, though a proposed solution is just to set personal, hard deadlines. The status, problems and design of each component assigned is listed below.

4.2 Algorithm

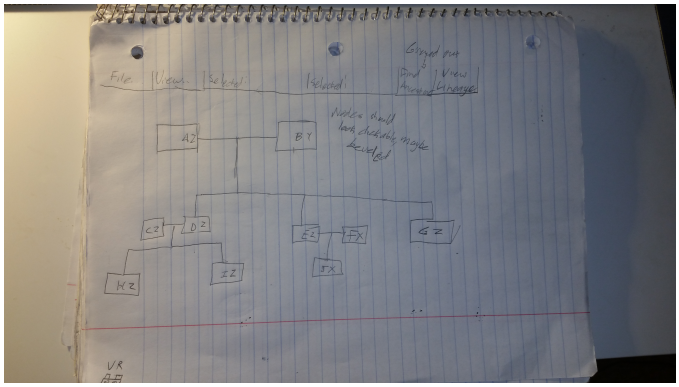
The algorithm has been designed, but it has not been implemented, nor has it undergone extensive testing to prove its correctness. The objective for the algorithm is that it generates nodes with enough space between them so the amount of overlap for lines is minimal, and that there are no nodes that overlap.

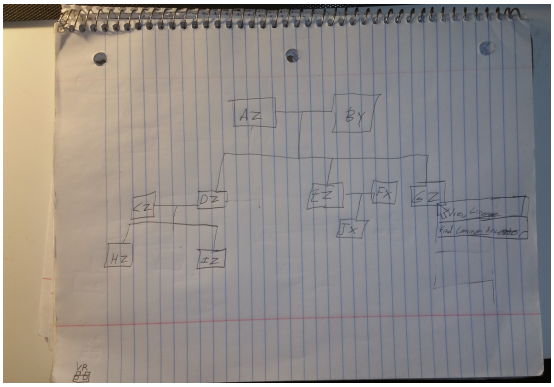
As stated before, the algorithm needs to be coded and tested. This will involve reading the data from the parser, then creating the algorithm from the design, then sending the design to the Unreal Engine visualization component for the data to be shown. There have not been any large roadblocks, aside from setting time aside to create and implement it. Overall, the algorithm is not as developed as it should be at this point in the term, though it is designed.

The current algorithm is designed as follows: The algorithm will ignore sibling relations, except with respect to placement for simplicity's sake. Before beginning, each parent pair is treated as a single node for placement purposes. The algorithm will start by iterating through the map that the data is stored in, and finding every pair of parents, or single parents if the spouse is unknown, that do not have parents. They will be labeled as root nodes, because they are the top of their tree. Each root will form trees that do not make contact with any of the other trees, though their nodes may appear in other trees. The algorithm will then go through each child attached to the root, attach their spouse, and then separate siblings by a fixed distance. Every subsequent child will separate their parent nodes as well, in order to maintain spacing. Finally, when the visualization occurs, there should be a line that connects nodes that are the same person if they are in a different location.

4.3 UI

The UI has been designed, though there are two designs, and some consideration on which design should be implemented.





The first design is more bulky, by virtue of having a permanently available menu always visible, as opposed to the second design having menus only appear when the user clicks on a node. The first design is currently favored, however, since while the first design has more always visible components, it is more intuitive precisely because of the menus being present.

There is one additional concern with both of the designs. The nodes should appear as though they can be clicked on, because they all should double as buttons that allow the user to select the person that the node represents. This will likely require research on how to influence users to click objects that they aren't certain are interactable.

The current status the UI is at is implementation. The UI needs to be created using the UMG UI creator, then implemented. The primary concern to implementing the UI is attaching functionality to the buttons, for instance, changing the view style. Another potential concern would be implementing user selection. The intended functionality of user selection is that if they click a node, it should show the person as who the user has selected, write it on the first available selected bar, and highlight that node, and if they click the person again, all of the modifications should be removed. Selecting people would also automatically make them candidates for Lineage View and Nearest Common Ancestors. To remedy this, the UMG UI documentation should be consulted, and if that fails, the Unreal forums.

4.4 VR UI

The VR UI has been designed, but the usability of the design has not been extensively tested. Fortunately, since the application's VR mode assumes that a VR controller is also in use, the UI is designed to be a menu attached to the fingers of a hand, which moves by the positional input from the controllers. Both hands serve the same purpose in terms of menus, but when selecting nodes, the controller that presses the button is also the one that selects the node. For instance, the user can activate find the nearest common ancestor with either hand, which will prompt both hands to switch to select a node mode. If the user points to two different nodes with both hands, but only wants to select the one that their right hand is pointing to, they simply press the select button on their right hand, and only the node that right hand is pointing at is selected. The user can then repeat the process to specify the other node with either hand.

The current status for this portion of the application is implementation, though it is further out than the other two portions, since the VR portions of the application has yet to be created. A possible solution to this particular roadblock would be to implement the portions that don't strictly need the rest of the system to be available, for instance, implementing the searches and views, or acquiring an asset for both of the user's arms.

The implementation of the VR UI should blend aspects of the normal UI with new aspects. In particular, the VR UI should keep the function implementation of the original UI, but instead of binding them to physical buttons on the menu, the functions should instead be bound to the buttons on the controller. Another addition to that menu is that the button menus should switch to a selection menu when necessary, for instance, when they user selects a menu option that requires node selection. Additionally, selection needs to be redone to accommodate for when the user points at a node. Finally, the hands that the user sees in VR mode must be created and designed to be of sufficient length, so that the user can adjust how far away the menu is from their face.

5 Conclusion

The purpose of our project is to read in a GEDCOM file and display its information onto an ancestry tree. The GEDCOM file needs to be parsed through to make it easier for the software to read and to gather the required data. The ancestry tree will be displayed in a clear graph that contains series of parent and children nodes. The software has two applications: desktop and VR. A user should be able to toggle between desktop and VR mode if a VR headset is presented. In addition, the following special features should be included in the software: choosing a subsection of the tree, finding a common ancestor between two nodes (members), and toggling between a full ancestry tree and a direct lineage tree.

6 Retrospective

Positives	Deltas	Actions
Established Technologies	Workflow is slow	Establish a tighter schedule and work on time management.