Richard Moot
CS 162-400
December 8, 2015

Final Project

Design

The design stage of creating the game was by far the most challenging portion of this assignment. Since it was quite open-ended, it was hard to come up with a suitable idea that would sufficiently express all of the project requirements. I had decided at this stage to select something that could somewhat believably take place within a terminal, so that there was a greater sense of realism (although not much) when playing the game. The idea was that the player has hacked into a network so they could recover their identity from a server. They would navigate through the different machines on the network to try and find the passcodes for accessing the server.

I had decided that I would have a main controller class that would negotiate the gameplay, so that I could keep the main function fairly clean. The Network class would be most responsible for this, since logistically this would be facilitating moving amongst machines. The Machines would be the main abstract class for where all the spaces would be for the player to move amongst. There would then be three types of machines that we would have: Server, Computers, and Mobile. The server would be the goal location for the user to get into, with Computers and mobile housing the information needed to access the server.

Since this is designed like a network, each Machine on the network will point to every other machine on the network. This allows the player to go wherever they would like at the cost of their chances of being detected increasing. Also taking actions within a machine increases detection, so the player is encouraged to find their key information as quickly as possible. I plan to have everything simplified and compartmentalized, so that functions would be very clear through the process (and make testing easier). Player would be responsible for knowing where they are and what information they have, Machines would be responsible for housing information and granting it to the user.

Testing

Testing the application that involves this many components required taking an incremental approach. Each piece would be tested before being committed to the entire project, and interactivity would be tested before it would allowed to go in. This started with ensuring that all of the rooms were properly initialized by checking menus for each one. This coupled with the detection going up ensure that we had navigation working properly prior to having any state changes with the player or within any of the rooms.

Then I had implemented the actions that would be taken by the user piece by piece, with just returns in their place prior to implementation to gracefully handle them throughout testing. My testing process ultimately sparked my desire to allow the user to skip the intro dialogue, since it had seriously take time away from me being able to test further. Once all pieces were implemented, I had planned on smoke testing the entire game by navigating to make sure each menu option went to their proper place or displayed the correct message.

Smoke testing included first going through the victory scenario of obtaining the server password from the email of the work machine, triggering the 2-factor code, and getting the 2-factor code from the text messages. Then attempting to access the server. Once this basic functionality is verified, then robust testing should take place. This requires utilizing each menu option for each room to verify the messages displayed. Disconnections due to being detected would ultimately be tested in this process as well.

Overall, the testing process led to most of the fine-tuning in development because it helped make sure that we had no dead-ends and all components of the system were functional.

Reflection

Although I had struggled with the initial concept portion of this assignment, it has thoroughly solidified my comfort in using far more aspects of C++ but also delegating responsibilities to appropriate classes. My comfort with pointers at the beginning of the quarter was quite shaky at best, but now has been solidified. Passing the player pointer to be modified by the Machines and passing the Machine pointers to the player was seamless, where before this would have been daunting (like it was with the Tournament assignment).

I had even had a few moments where I had initially created 4 functions for each of the Computer derived classes, but realized that I could consolidate all of those into one function inside of the Machine class that they could all inherit from (the playerSelection function). It even became easier to understand inside of a given class whether or not it would be able to access information in another class, which led me into making use of forward declarations for the first time to be able to properly pass pointers to classes.

Overall this has been an interesting quarter that had a rocky start with the Game of Life assignment, and leading into one of my most involved and well organized programming projects that I had done thus far.