Chris Fortier – CSCIE-50b Term Project

Synopsis: Created a version of the classic board game Clue. Imagine that you a player trapped in a mysterious house with five other people when a tragic murder occurs. Nobody can leave until the murder is solved. You know that one of the players is the murder, but you don't know who it is. You also don't know what room it happened in or what weapon was used. It is your job to compete against the other guests trying to figure it all out!

Cool Tech Stuff/Challenges: I knew this would be a huge project once I really started working on it and wanted to embody the philosophies of object oriented programming along with the "unix" mindset of keeping things as separate as possible. On that note, there are nine total classes involved. The main Clue class is the controller and keeps everything running smoothly. There are several tiers of classes for the user interface as well as a couple tracker "engines" to keep track of things. There are around 600 spaces(buttons), six pieces, and twenty-four cards *per player* to keep track of. As such there is some pretty complex code involved. There are also numerous methods solely for communicating between the different layers of the program. The major challenge was keeping track of everything in my head and getting it all to work together. Also, in creating the logic for the computer players I used random numbers as much as possible to try and simulate a human player.

Loading Instructions: I created a linux .sh script and a Windows .bat script to compile all of the necessary files. If you're wondering - I used multiple computers during the development and ended up doing work in linux, Mac OSX, and Windows while creating this. Either of the scripts while compile the necessary files. Once everything is compiled simply run 'java Clue' to start.

Gameplay: As the only human player, the game begins with you. Simply press the 'Roll Dice' button and the game will begin. You will see spaces light up in blue if they are available for you. If a room is available you may go there and start a rumor/accusation. You will be prompted (through a custom rumorFrame) to select a Murderer, Room, and the Weapon. If you choose the correct one, you win! Otherwise you're rumor will be compared with the next player to see if they have any of those cards. If they do, that would prove your rumor wrong and end your turn. Though you will be allowed to see which cards matched. The game continues in this fashion until someone wins the game. On the technical side, each piece has a tracker to track which cards have already been seen. The computer players will randomly choose a rumor based on what they haven't seen.

Expansion: I've tried to write the whole program as modularly as possible and could expand it in multiple ways. It would be neat to allow multiple human players. Though that would require making some way to hide the data from the other users. It would also be neat to add some kind of actual artificial intelligence to the computer players. Oh, and a snazzier user interface would be a nice feature.

Reactions: I thought it was a great project and this was definitely the best programming class I've ever taken. I feel like I've advanced several years in my knowledge of programming with this course. Things I learned: as projects and programs get incrementally more features, the complexity increases exponentially…. Proper, solid planning is absolutely essential.