Network Implementation of *UNO*!

Network Programing - COMP - 2100 - 09

Professor Salem Othman

Chom Sreng, Fabio Tran, Peter Garrity

Due Date: October 19, 2021

Implementation Plan

We will use the Python Programming Language and Wireshark to recreate the card game *UNO*. Players will represent clients and will connect to the main server to initiate a game. Our application will be able to facilitate a full game of Uno until there is a winner or until the game is terminated.

Why This Project Was Chosen

This project was chosen because we wanted to make a fun and interactive game that involves multiple clients connecting to a host server. Our program will incorporate socket programming and networking knowledge we learned this semester which makes it an appropriate selection for this final project assignment. There are many card games we could have programmed such as Blackjack or Poker. *UNO* is different from these other card games because the rounds are fairly long compared to traditional casino card games. This means our program will have clients connected to the host for an extended period until there is a winner. In addition, while traditional casino card games are mostly formulaic and routined based, *UNO* has a characteristic that makes it unpredictable which is that a player/client must say “Uno” when they have one card left, if not they would have to draw more cards. This characteristic and others add to the complexity of the program which makes it interesting for us to implement.

Stages of Implementation

To start our project we will first work on implementing the essentials of the game such as a class for the cards and subtypes for the special cards. After that we will work on implementing the multiplayer functionality, starting with just the ability to play number cards and then adding in the more complex cards. If we are able to complete this with

plenty of time to spare, we hope to explore the possibility of adding a GUI. This is an ambitious goal as our team has very little experience with Python let alone experience in Python's GUI libraries.

Timeline

Tentative Project Timeline	
10/21	Plan out project outline and details
10/28	Begin coding client and server
11/04	Continue implementing code
11/18	Testing and debugging
12/02	Final Presentation Report and Presentation