UIC Spring 2019

CS 342 Project 5

UNO

Team Number: 14

Section: 41553

Members: Michael Aiello, Ryan Anderson, Thomas Hein, Anthony Slas

---

Team

Name	***Email***
Michael Aiello	maiell5@uic.edu
Ryan Anderson	rander42@uic.edu
Thomas Hein	thein4@uic.edu
Anthony Slas	aslas2@uic.edu

GitHub Repository Link

Game Description

We are making a JavaFX multiplayer game. The game we are going to attempt to replicate is the classic card game, UNO.

Every player is dealt 7 cards. The remaining cards are placed into a draw pile. The top card of the draw pile is placed into the discard pile.

Now for the gameplay. Given their hand, every player attempts to *match* the top card on the discard pile. *Match* means to match the card in their hand with the card on the top of the discard pile by number, color, or symbol/action. Or the player can play a wild card. If the player chooses not to play a card, they must draw a card from the draw pile.

The game continues until any player has one card left. At this point, the player with one card must declare they have UNO. Then they must either play that card if applicable, leaving them with zero cards making them the winner, or draw a card and continue with normal gameplay.

If a winner is decided then the round is over, and the winner gets all the opponents cards in their hand. Then, the points are scored (what cards are in winners hand), and the game starts over. Players play until a designated score. Usually 500, but it can be anything.

There are special cards called action cards. Reverse, skip, draw two, wild, and wild draw four. These cards are worth more when it comes to scoring.

Scoring is also relative to the players hand in the end of the game. Below is more information from the UNO rules website.

Here is a link with details about the special actions cards as well as scoring procedures.

[UNO rules]

Languages and Frameworks

We will be developing this project in Java with JavaFX, using sockets and threads.

Client Server Relationship

We plan to have a similar client server relationship as our last team project.

The server will continually listen for players until 4 players are in a player pool at which point gameplay will begin.

The server will create new client threads accordingly as the players connect. The server will store these threads in some data structure. Most likely a HashMap.

The client will send information to the server (what cards in hand, what card played, did they pass on their turn, are they quitting, etc) in the form of strings. The server will then parse these messages accordingly and take action.

The server will implement all game logic, and handle the distribution/dealing of cards.

User Interface Description

Being a server client game, there are going to be two GUIs.

Server GUI

The server GUI will have basic server functionality. Data fields to enter server info (port number, etc), buttons to turn the server on and off. In addition to the server functionality it will have some aspects necessary for the game of UNO. The server GUI will have a player table (essentially a game lobby, displaying who is the the lobby at any given moment). It will also have a message box displaying game information (game in progress, waiting on players, if there is a winner, game restart, etc).

Client GUI

The client GUI will have basic client functionality. Data fields to enter connection info (port and IP info), buttons to connect to the server and quit the game. The client GUI will also have UNO related buttons and images. These include clickable buttons with images of the card, button to skip turn, button to draw card, button to declare UNO.