CSE 213 – Final Project Rubric

Student Name: Marisa Loraas

Grader Name: Rachel Powers

NOTE: For the final project, the student will choose to either implement Go Fish or Uno. Only grade the portion for the project the user chose. Including the points from the general submission requirements, and extra credit (if they choose to implement networking), the total points a student can earn are 130 points for Go Fish, and 150 points for Uno.

Problem	Score	Total
Submission Requirements		**20**
Submission is named `cse213_<firstname>_<lastname>_final.tar.gz`	1	1
Code follows Google's style guide reasonably well, and uses four-space indentation	4	4
Every public class and public method has a Javadoc comment	2	5
The user included a UML diagram for all their classes that is readable and correct to the project	10	10
Go Fish		**100**
The project compiles and runs with no errors		5
The program accepts 4 players, one human and 3 AI		8
Cards.java contains an enumeration for all the 52 different card types		4
Deck.java contains an ArrayList of Card objects with all the cards in the deck, and has a method getCards(int) which returns a given number of cards from the ArrayList chosen at random.		6
The game begins when the player indicates they are ready (e.g. with a "play now" button), and when it begins each player is given unique name to differentiate them (e.g. player1, player2, etc.) and is dealt a hand of 7 randomly chosen cards.		10
The game order goes clockwise starting with the human player.		5
Upon a players turn, they can choose any other play to ask a card from. If they receive a card, their turn repeats. If the player they asked does not have the requested card, the game logic automatically indicates "go fish" has occurred and the turn is moved to the next player		10
The client GUI is clear and laid out in a readable manner		8
The client GUI has the following features: - A component that allows the player to indicate when they are ready to start		28

- A component to show all the cards the player has - A component that shows all the players currently in the game, and allows the player to select one to ask for a card - A component to allow the player to choose a card to ask for - A component to track the number of books each player has - A component to indicate when the game is over, and who the winning player is - A component to allow the player to opt into playing again when the game ends		
The game logic proceeds by the rules of Go Fish as described here: https://bicyclecards.com/how-to-play/go-fish/		12
The game ends when the combined total of books on the board is 13		4
EXTRA CREDIT: The player has implemented network functionality. This means: • The application has the files Client.java, ClientThread.java, Server.java, and ServerThread.java • The application accepts 4 human players over the network (instead of 3 AI and 1 local human) • The game does not begin until at least 2 clients have connected and both have indicated they are ready • The server indicates all game states to the clients (e.g. when “go fish” has occurred, how many books each player has, etc.)		10
Uno		**120**
The project compiles and runs with no errors	5	5
Cards.java contains two enumerations, one for card color and one for card symbol that can represent all 108 cards in an Uno deck.	10	10
Deck.java contains an ArrayList of Card objects with all the cards in the deck, and has a method getCards(int) which returns a given number of cards from the ArrayList chosen at random.	6	8
The program accepts 4 players, one human and 3 AI	12	12
The client GUI is clear and laid out in a readable manner	10	10
The game begins when the player indicates they are ready (e.g. with a “play now” button), and when it begins each player is given unique name to differentiate them (e.g. player1, player2, etc.) and is dealt a hand of 7 randomly chosen cards.	10	10
The game order goes clockwise starting with the human player, but changes if a card is played that changes the order of play (e.g. reverse)	10	10
The client GUI has the following features: - A component that allows the player to indicate when they are ready to start - A component to show all the cards the player has - A component that shows all the players currently in the game - A component to show the card at the top of the discard pile - A component to show how many cards each player has - A component to allow the player to choose a card to play - A component to show how many points each player has	34	40

- A component to indicate when the game is over and to show who won - A component to allow the player to opt into playing again if they wish		
The game logic proceeds by the rules of Uno as described here: https://en.wikipedia.org/wiki/Uno_(card_game)#Official_rules (NOTE: The game does NOT need to implement the rules that require vocal or otherwise nuanced interaction from players, such as calling “Uno” or challenging the Wild Draw Four card	15	15
EXTRA CREDIT: The player has implemented network functionality. This means: • The application has the files Client.java, ClientThread.java, Server.java, and ServerThread.java • The application accepts 4 human players over the network (instead of 3 AI and 1 local human) • The game does not begin until at least 2 clients have connected and both have indicated they are ready • The server indicates all game states to the clients	0	10
Total Score	129	**90/110/130**
Comments:		