**Textual Description**

This system is a virtual card game where the user plays against an opponent, which is the computer. The user has 3 decks to choose from, each deck consists of 20 different cards. Each player starts the game at 100 life, and whoever reaches 0 life first loses the game. Each round, the user and the opponent both select a card to fight each other and the loser of the round loses 20 life. Each player starts the game with three cards in hand and draws a card after each round. If a player would draw a card while there are no cards in the deck, the deck reshuffles all of the cards previously played by that player back into the deck. This system contains a Card class, a Deck class, a Participant class, an AutoOpponent class, a User class, a UI class, and a GameController class.

The Card class contains five attributes and the setter and getter functions for those attributes. A card has a power attribute that is an integer that ranges between 2 and 15, a rarity attribute which is common, rare, epic, or legendary, a type attribute which is red, green, or blue, an archetype attribute which is a knight, a wizard, or a dragon, and an image attribute which contains the image icon for the card in the game.

The Deck class contains two stacks, one for the player’s deck which is for cards that have not been played yet and one for the discard which is where cards go after they are played. The class also contains a function to load from a text file the deck chosen by a player, a function to shuffle the deck, a function to draw a card from the top of the deck, and a function to switch between the discard stack and the deck if the deck is empty.

The Participant class is a superclass of the user and autoOpponent classes. This class attributes are a Deck object, an arrayList of type Card of size 3 that contains the hand of the participant, an int which has the health of the user, and the card position of the card a player chose to play. This class also contains setters and getters for those attributes. The participant class is initialized to have 100 health and card position at -1 to show that the user has not chosen a card yet. In addition, the participant class has a function to draw a card from the deck and to reset all of its attributes.

The User class inherits all of the attributes and functions of the participant class and contains only one function that returns an object of the card type which is the card selected by the user.

The AutoOpponent class inherits all the attributes and functions of the participant class and contains a function that chooses a card to play based on the difficulty of the game. If the game is set to easy, the computer will randomly choose a card to play from its hand. If the difficulty is set to hard, the computer will select the card with the highest power in its hand to play.

The GameController class is the class that controls the gameplay. This class also has an attribute that is a collection of 3 decks which contains all the decks that the user can choose from, an attribute that is a UI object, and two participants, one which is the user and the other is the opponent. This class also contains all the event listeners for the buttons implemented in the UI class. The GameController has a nested class called PlayCardListener which contains the game play of the game. The PlayCardListener contains a function that is called when the user selects play card. The function first makes sure that a card has been selected before pressing that button, if there wasn’t a message is displayed to the user to select a card and nothing happens. Otherwise, the function gets the both cards and compares them by power, type, and archetype. After the winner of the round has been decided, the function updates the life total and the life bar of the losing player.

The UI class represents the user interface and contains all the elements that are displayed to the user and all the buttons that the user clicks on to perform actions. When the user clicks on a button in the UI it send a signal to the corresponding action listener in the game controller.