**Platoon Implementation Manual**

**Introduction**

Platoon is my very first game, It’s my final project for my Object Oriented Computing class and is made using Java and Javafx. The object of the game is to make five piles using ten randomly drawn cards from a playing card deck (minus aces and jokers). You will then select one of your piles to go up against one of five computer generated piles. The goal is to have three of your piles or “hands” win.

This project has a bit of animation along with score tracking and card values. It also includes special properties for the face cards.

**Project Structure**

Src/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 |\_\_\_\_\_ /Game/ /resources/  
 |\_\_\_\_\_\_\_ Main.java |\_\_\_\_\_\_\_\_\_ card\_back.jpg  
 |\_\_\_\_\_\_\_ ComputerPlayer.java |\_\_\_\_\_\_\_\_\_ card\_Images.txt   
 |\_\_\_\_\_\_\_ HumanPlayer.java | (list of every card name)  
 |\_\_\_\_\_\_\_\_\_ ...5 spades  
 king diamonds  
 etc.

Main.java

* Includes main UI
* Most methods and their functions
* Holds most of the variables used

HumanPlayer.java

* Used to call methods for the human player
* Uses the code in Main.java exactly

ComputerPlayer.java

* Most methods are altered to cater to a non player
* New methods to compensate for the different functionality of existing methods

**Game Mechanics**

A player is given ten cards and is given five piles to sort them into. A pile cannot have more than five cards and the player can move up to five cards at a time. If they wish they may also move a card already in a pile to another pile or back to their hand by pressing the “Q” key. Once the player’s hand is empty and all the piles are filled, they will be given the option to start the game.

They will now see their piles and the computer generated piles, now the player can now select one pile each from themselves and the computer and press the “Play Hand” button. The game will check what pile contains the best score or right face card and award the winner a point, the first to three points wins. If both players end up with two it is a draw.

Win Conditions:

Each card is given the value of its number (1 = 1, 2 = 2, etc.) with a King having a value of 10 and Queens and Jacks with a value of 1. The face cards have special properties that can turn a hand around.

Jack: swaps the hands of the player and computer

Queen: loses no matter what unless it goes against a King

King: wins no matter what unless it goes against a Queen

If both hands were to have Kings or Queens, it will calculate the value of the pile, the pile with the highest score after that, wins.

**Assets**

Assets are stored in the src/resources/ directory. It contains images of playing cards including a back side, and a text document with the name of every card image so it can drawn from in the code.

**Animations**

There are some animations like when selecting or hovering over a card, little blue highlights will appear at its corners. There is also animations for moving the cards in between the hand and piles.

**Computer Player**

The computer player does not just make hands randomly, it has a priority order of hands to make, in order it looks to make

Jack and Queen > Queen > King > low score > high score

If it reaches the end of this cycle, it will check if every pile has at least one card, if not, it will move some number cards to those empty piles.

**Issues**

Currently the restartGame method doesn’t work properly, the game doesn’t load further than after the round start button appears.

I created methods one by one, adding features as I thought of them, because of this and the short deadline, a lot of the code is unorganized and likely has some useless or repeat code.