**Bug Report:**

The only bug that I have found is in the scenario of having 3 matching cards in the layout, if the player does not select the first card found in the layout like the help function says it would crash. I could not find a fix for it so I restricted the user from selecting the other 2 cards to make sure the program would not crash.

**Feature Report:**

**Missing Features:**

No missing features

**Extra Features:**

No extra features

**Description of the data structures:**

**Inheritance:**

Cpu and Human inherit from the player class

**Composition:**

1. Card: Deck
2. Cpu: None
3. Dealer: None
4. Deck: Card, Deck, Dealer
5. Game: None
6. Human: None
7. Player: None
8. Round: None
9. Source: Headers

**Classes:**

1. Card- Creates the cards for the game
   1. Data structures used: Vectors
2. Cpu – The Computer players action class, inherits from the player class
   1. Data structures used Vectors
3. Dealer – Distributes the cards from the deck
   1. Data structures used: Vectors
4. Deck – Holds the cards
   1. Data structures used: Vectors
5. Game – The main game which starts new or loaded rounds and keeps track of player scores.
   1. Data structures used: Vectors
6. Human - The human player action class, inherits from the player class
   1. Data structures used: Vectors
7. Player – Parent class of the human and cpu class
   1. Data structures used: Vectors
8. Round – Runs individual rounds within the game
   1. Data structures used: Vectors
9. Source – client file
   1. Data structures used: None

**Logs:**

1/21/20: 2 hours

Just did planning and made shell of card and deck classes.

1/22/20: 5 hours

Finished card class and deck class, deck class can take the cards created by the card class and make a 104-card deck.

1/23/20: 5.5 hours

Added more functions to the deck class such as drawCard(), created shell of round class, dealer class, and player class. Now able to draw a card and pass it to the dealer class. Also tested card and deck classes to find any bugs.

1/24/20: 3 hours

Finished dealer class and played the game with real cards to understand the game rules more.

1/26/20: 4 hours

Finished the basic round class, and successfully got the game to create the round and player objects, send the player objects to the round class, create the dealer class then made the card and deck class inside of it. Now the dealer class can pass the cards out to the player objects hands and the table cards.

1/27/20: 4 hours

Created a sortHand() in the player class which attempts to reorder the hand to make it easier to decide who goes first. I was having trouble with the sortHand() so I am going to try another route by making faceCards() in the player class which just looks through each players unsorted hands.

1/28/20: 4 hours

Got faceCards() to work but doesn’t handle ties yet.

1/29/20: 5.5 hours

Trimmed down some unnecessary for loops and made one big loop to deal the cards out to players and table in goFirst() function. (0.5 hours)

Finished goFirst() function, it can handle ties and determines who goes first. (2 hours)

Created the shell of Human and Cpu classes to extend the player class’ virtual playTurn() function. (0.5 hour)

Made the virtual function playTurn() which is inherited by the Human and Cpu class from the player class. (2 hours)

Started making UI for the Human class playTurn() function. (0.5 hours)

1/30/20: 2 hours

Continued working on human class playTurn() function(2 hours)

1/31/20: 0 hours

2/1/20: 1 hour

Human class playTurn() now returns the hand card index and table index they would like to play.(1 hour)

2/2/20: 1 hour

Fleshed out the human playTurn() more by including more input validation. (1 hour)

2/3/20: 4.5 hours

Added the menu before each turn (0.5 hours)

Cleaned up output (0.5 hours)

Made isLeagal() function to check if the player’s card can be played on the card they have chosen on the layout. (2.5 hour)

Added the removeCard() function to the player class when the player chooses it from their hand (1 hour).

2/4/20: 6.5 hours

Fixed a mistake where 10s were not displaying as X’s. (0.5 hour)

Added stacksOperations() function to the round class to perform the actions to represent stacking the cards the players play and other various behaviors related to stacking actions. (3 hours)

Added the addCapture() function to the player class to collect the cards the human player captures and puts it in their capture pool. (1 hour)

Added more checks in the stacksOperation() function to catch more stacking scenarios. (2 hours)

2/5/20: 7 hours

Half stacks captured are correctly collected and stored in the players capture pile and 3 stacks created by the stock card draw are correctly handled. (2.5 hours)

If stock card doesn’t make a 3 stack it is now added to the layout. (1.5 Hours)

Added the ability to combine 2 2-stacks of the same card to create a 4 stack and gives the player a point. (1.5 hours)

Added the showCapturePile() and showScore() functions to show both players captures and scores. (1.5 hours)

2/6/20: 1.5 hours

Added functionality to add any pairs the stock card makes when not matching with the played card. (1.5 hours)

2/7/20: 5 hours

Found bug when capturing two pairs from the table and fixed it (2.5 hours)

Added noPlay() function which checks if the player has no cards to play and asks what card they would like to discard to the layout. (2 hours)

Added askToPlay() function to ask player if they want to play another round. (0.5 hours)

2/8/20: 2 hours

Refactored many of the functions in the round and player/human/cpu classes (2 hours)

2/9/20: 3.5 hours

Fixed a bug that caused the game to crash when both the player card creates a pair and the stock card creates a pair to be captured (1.5 hours).

Started the CPU playTurn() function (2 hours).

2/10/20: 5 hours

The CPU player can use the same functions as the human player. (2.5 hours)

Refactored code in the startRound() function. (0.5 hours)

Fixed bug where in H0 if the stock card matched the card the player added to the layout it wouldn’t capture. (2 hours).

2/11/20: 7.5 hours

Added more functions to the CPU players class to make the most optimal move e.g. look for 3 stacks to complete rather than just completing the first pair found. (2 hours)

Made the help function in the Human class. (2 hours)

Formatted the output of the game to match the specifications (3 hours).

Refactored code in the Human class. (0.5 hours)

2/12/20: 5 hours

Serialization function can save the state of the game to a text file now (3 hours)

Finished CPU player, now plays the way that matches the project specification (2 hours).

2/13/20: 7 hours

Fixed bug where choosing a card greater than or equal to 10 was not a valid choice. (1.5 hours)

Added the checkIfSameHand() function which checks if both players have the same hand, if they do it resets the deck and player hands then the dealer deals the cards out again. (2 hours)

Added function to ask player what layout card to play the stock card to if there is more than one match (2 hours)

Added function to let CPU choose what layout card to play the stock card to if there is more than one option (1.5 hours)

2/14/20: 5 hours

Created the loadGame() function which reads a text file and resumes a saved game.(2 hours)

Refactored code in the startRound() function of the Round class. (2 hours)

Started a function in the round class to start a game when given a load file. (1 hour)

2/15/20: 8 hours

Finished the loading functions in the round and game classes to make it possible to read from a specified file and resume the game. (3 hours)

Fixed bugs where the games loaded from a file were not displaying the correct data. (2 hours)

Refactored more code in the startRound() class to make it easier to load a game. (3 hours)

2/16/20: 4 hours

Refactored code in round class. (2 hours)

Improved UI such as informing the player who goes first in next round based on points scored. (1 hour).

Added in input validation for the handSelect(), layoutSelect(), and when inputing the name of the load file. (1 hour)

2/17/20: 5 hours

Tried fixing formatting and load file bugs such as not reading all the hand cards and round scores not being added (5 hours)

2/18/20: 0 hours

2/19/20: 0 hours

2/20/20: 0 hours

2/21/20: 0 hours

2/22/20: 0 hours

2/23/20: 0 hours

2/24/20: 4.5 hours

Fixed help function not coming up occasionally (0.5 hours)

Fixed cpu player not picking the ‘-‘denoted 3 stacks (0.5 hours)

Fixed next player not playing the correct player (0.5 hours)

Fixed AI not prioritizing triple stacks first (3 hours)

2/25/20: 7.5 hours

Fixed bug where the game ended when the cpu still had a card (1.5 hours)

Fixed bug where score was not correctly displayed when loading in the game (2 hours)

Improved the player help function by adding detections of not stacked 3 matching cards (1 hour)

Fixed bug where human player not able to collect 3 non stacked matching cards if they play the same card (3 hours).

2/26/20: 1 hour

Started working on making cpu’s moveSelector() function to look at the capture pile to see if any pairs can be completed if there are no 3 stacks to capture(1 hour)

2/27/20: 5.5 hours

Completed the part of cpu’s moveSelector() function that captures pairs that are also in capture pile (1.5 hours).

Human help now looks in capture pile to see if any pairs match and can create a 4 stack (2 hours)

Fixed bug where when collecting 3 separate cards on the layout the wrong cards were collected (2 hours)

2/28/20: 0 hours

2/29/20: 0 hours

3/1/20: 0 hours

3/2/20: 3 hours

Fixed bug where cpu wasn’t choosing the best option when given the choice of two layout cards (1 hour)

Documenting (2 hours)

3/3/20: 2 hours

Documenting (2 hours)

3/4/20: 0 hours

3/5/20: 0 hours

3/6/20: 0 hours

3/7/20: 0 hours

3/8/20: 4 hours

Fixed bug in the checkStacks() function that didn’t capture 4 cards when the player, stock, and 2 cards on the layout matched (3 hours)

Documenting (1 hour)

3/9/20: 4 hours

Added error catching when inputting file name in load field. (1 hour)

Documenting (3 hours)

**How to run the program:**

Start an empty project in visual studios, add the source files and test files to the projects folder. Now you can compile the code then run it to be able to play.

**Screen shots:**

**New Game being started and player playing its turn with help**

**A screenshot of a cell phone

Description automatically generated**

**Computer player playing its turn after the human**

**A screenshot of a cell phone

Description automatically generated**

**Computer capturing**

**A screenshot of a cell phone

Description automatically generated**

**Selecting the save option in the menu**

**A screenshot of a cell phone

Description automatically generated**

**Player selecting help in the menu**

**A screenshot of a cell phone

Description automatically generated**

**Player selecting make a move in the layout**

**A close up of a black background

Description automatically generated**

**Player selecting exit in the menu**

**A close up of a black background

Description automatically generated**

**Example of loading in a file**

**A screenshot of a cell phone

Description automatically generated**