PROJECT 2

<MONKEY GAME v2>

CSC17A - 42636
Cay, Shienne Patricia
Date: 06/02/2017

Introduction

Title: Monkey Game v2

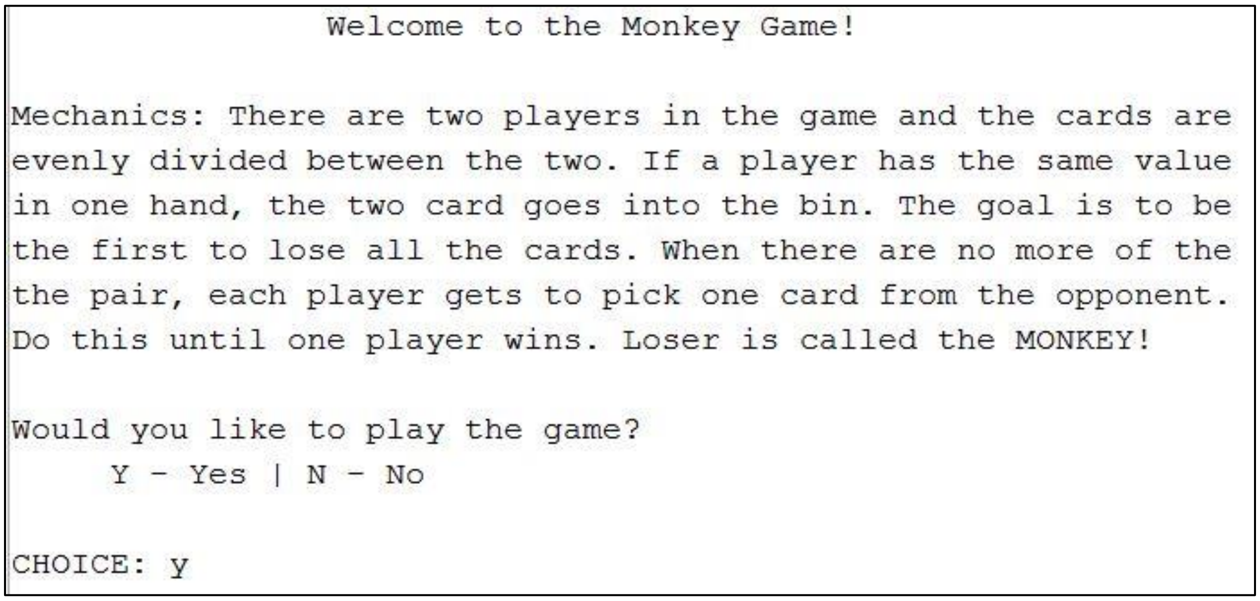
Just the same as in Version 1, Monkey Game is played using a deck of cards. It can have players up to 4 people but in this case, only 2 players are accommodated. Because I didn't have more time before, I tweaked the game a bit in determining the winner – which is the one with the highest point in the end. However, in this version, the one who eliminates all of his/her cards in hand is the winner, and whoever is left with the JOKER (a card I added in the game) is the monkey. In some cases, if both players wanted to quit the game with the option I gave without eliminating all cards, the winner is determined by the points they acquired in the game.

Just like the earlier version, game starts prompting the user if he or she wants to play the game after reading a bit of the synopsis of the game. If the user decides to play, the user will be asked which player he or she wants to be – and in that order, they will input their names. Shuffled cards will then be distributed to both players. The Player 1 will always have 27 cards, but it doesn't mean that the player will get the Joker as the cards are shuffled. There will be two rounds. In Round 1, players will discard all 'pairs' in their hands to gather points just like the first version. Pairs are the cards with the same value without regards for the suit. For example, if the player has a '7C' and a '7D', it is considered a pair, and it must be discarded. Two points are added each time a player discards a pair; however, when they get it wrong, and it is not a pair, there is a point deduction. After the end of this round, if no one got deduction, players must have even points even when the joker is around.

If a player has not seen a pair in hand and proceeded to end his/her turn, he/she will have the option to discard later, although it will only be one point addidtion if he/she ever does. The score board will be shown, and the game will prompt for next round. But before that, to be fair, the game will simulate a tossing of the coin to see who goes first in the next round. Heads will be Player 1, and Tails will be Player 2. After determining the one who goes first, the game will then proceed to Round 2.

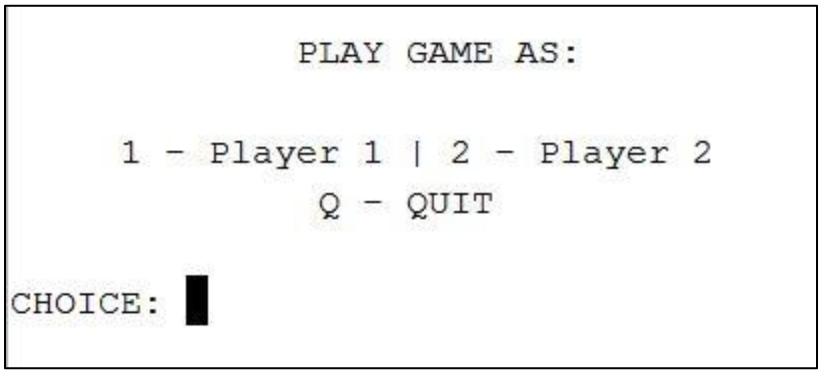
In this round, unlike the first version which is to find a pair of your card to the opponent's card, the objective now is to eliminate all your card in hand. The first one to eliminate all is the winner. The Joker among the cards will keep the game from having a tie. Although it can happen if a player decides to end the game along the way, and they have even scores on the board. Eliminating cards will go the same way as the version 1, you will get to pick a card from your opponent, but this time, you get to see what card you picked. The machine will ask if there is a pair in your hand in relation to the card you picked from the opponent. If you do so, you'll be able to eliminate that card in your hand and from opponent's hand. It will then change turns after every picking of cards. You don't get a point anymore, but if you choose a wrong card, you get a point deduction. If one player eliminates all of his/her cards, announcement of winner will be made and recorded to the "Records" text file and "Records" binary file.

How It Works



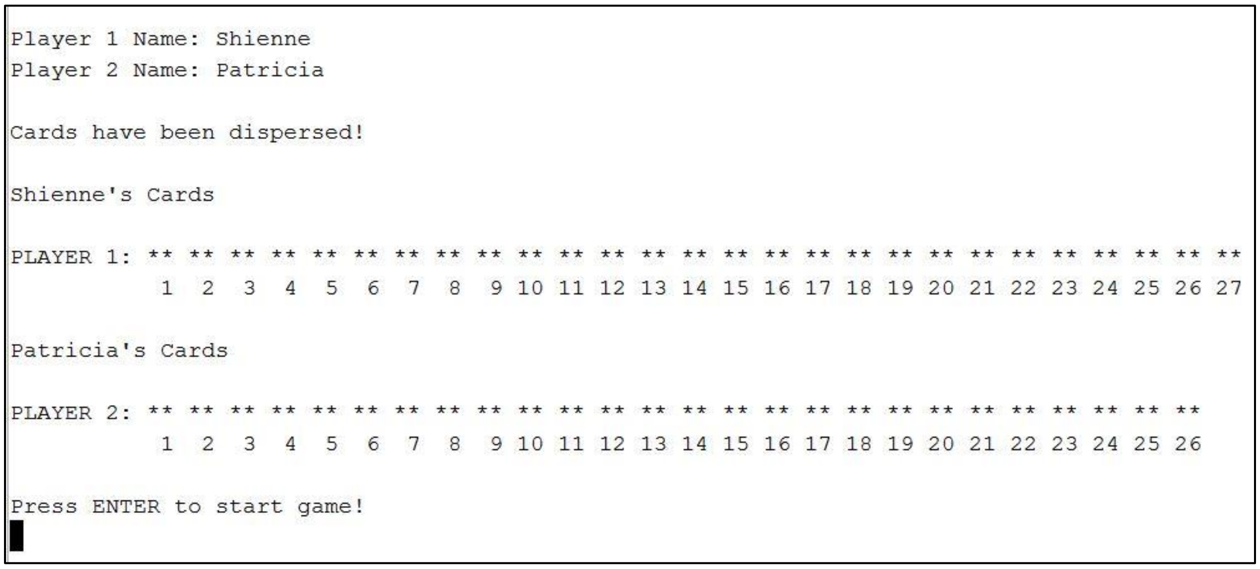
```
                Welcome to the Monkey Game!

Mechanics: There are two players in the game and the cards are
evenly divided between the two. If a player has the same value
in one hand, the two card goes into the bin. The goal is to be
the first to lose all the cards. When there are no more of the
the pair, each player gets to pick one card from the opponent.
Do this until one player wins. Loser is called the MONKEY!

Would you like to play the game?
     Y - Yes | N - No

CHOICE: y
```

Once the user hits play, the program will explain the simple mechanics of the game. The user will then be prompted to play or not. Should the user choose not to play, game ends and exits. If the user decides to play, player will be asked if he/she choose to be Player 1 or Player 2 of the game. Whoever the user chooses to be, he/she will have the first turn.



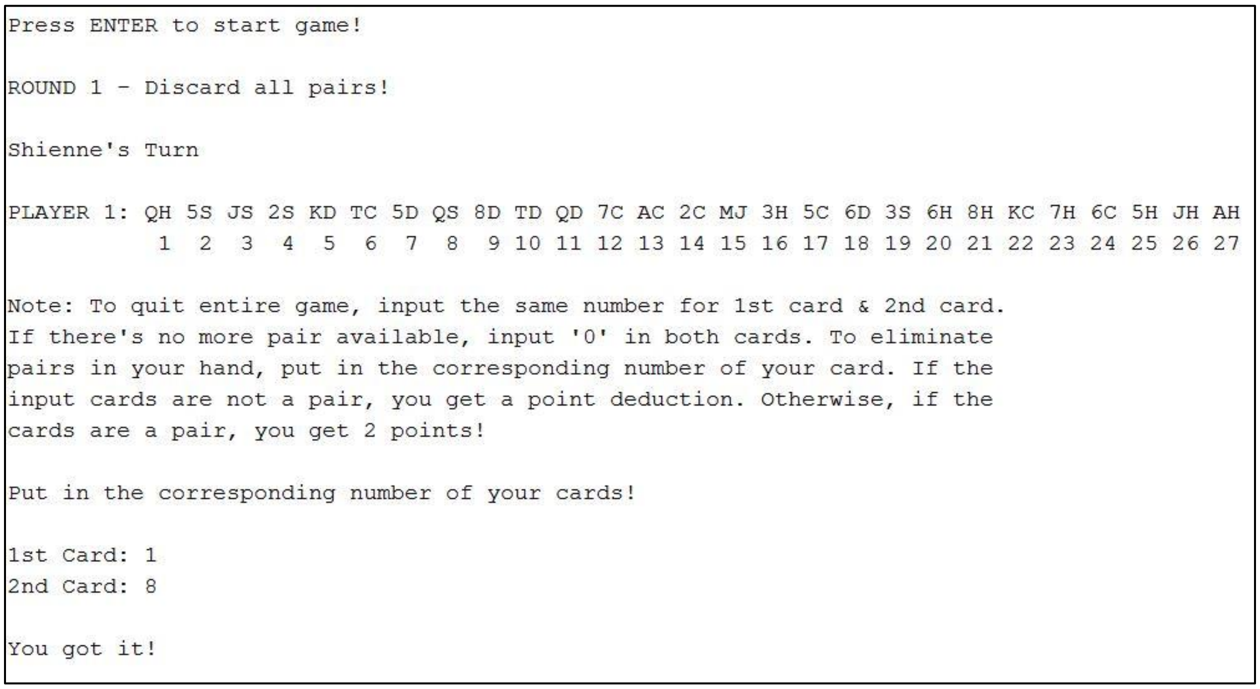
```
              PLAY GAME AS:

      1 - Player 1 | 2 - Player 2
               Q - QUIT

CHOICE:
```

There is still option to quit game in choosing players should the player wants to end game. Once user selects a player, the machine will prompt user for the names of two players for recording purposes. The names are used to determine whose turn it is especially 2nd round. After inputting the names for both players, machine will notify that cards have been dispersed (27, 26), and the machine will show both cards in hidden mode. One of them is Joker.



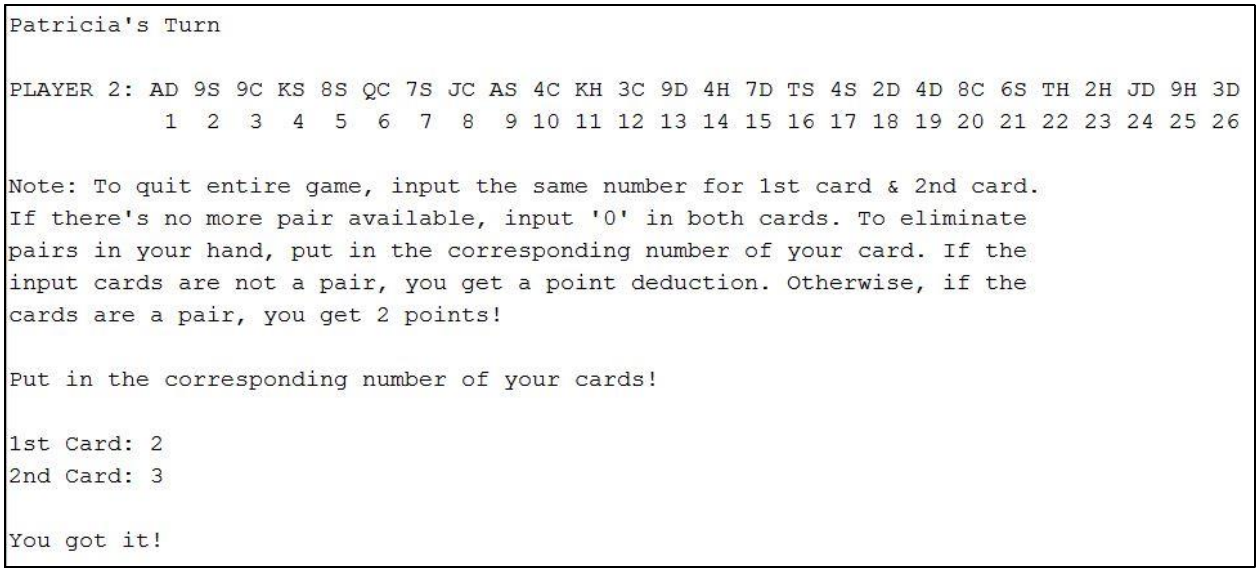
```
Player 1 Name: Shienne
Player 2 Name: Patricia

Cards have been dispersed!

Shienne's Cards

PLAYER 1: ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **
           1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Patricia's Cards

PLAYER 2: ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **
           1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Press ENTER to start game!
```

The game will immediately start once user presses ‘Enter’ key.

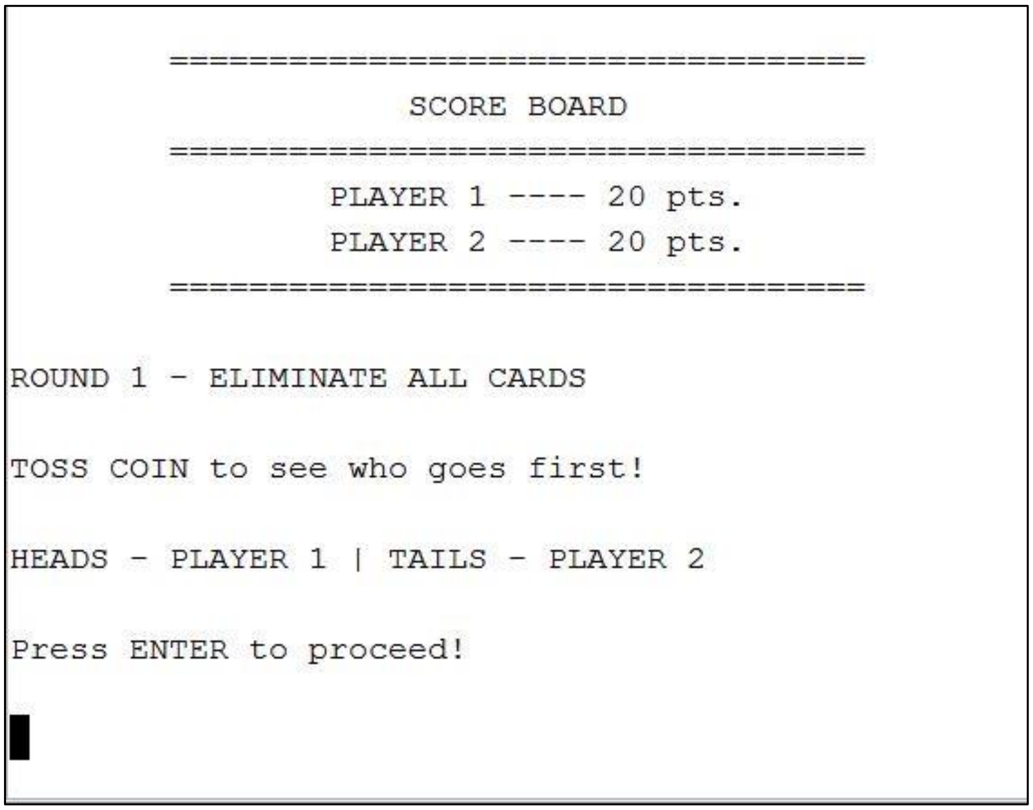


```
Press ENTER to start game!

ROUND 1 - Discard all pairs!

Shienne's Turn

PLAYER 1: QH 5S JS 2S KD TC 5D QS 8D TD QD 7C AC 2C MJ 3H 5C 6D 3S 6H 8H KC 7H 6C 5H JH AH
           1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

Note: To quit entire game, input the same number for 1st card & 2nd card.
If there's no more pair available, input '0' in both cards. To eliminate
pairs in your hand, put in the corresponding number of your card. If the
input cards are not a pair, you get a point deduction. Otherwise, if the
cards are a pair, you get 2 points!

Put in the corresponding number of your cards!

1st Card: 1
2nd Card: 8

You got it!
```

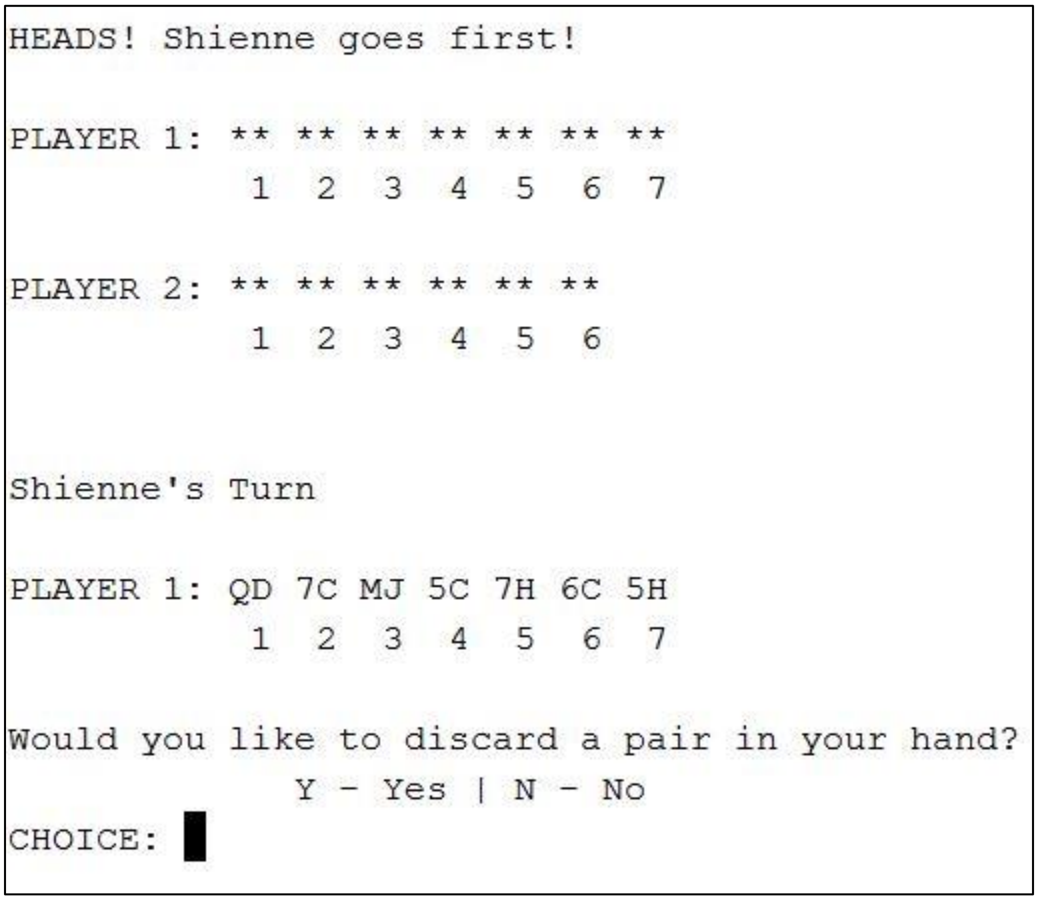
This loop will continue until current player 1 sees no more pair and inputs 0 in both cards. There it becomes the other player’s turn.


```
Patricia's Turn

PLAYER 2: AD 9S 9C KS 8S QC 7S JC AS 4C KH 3C 9D 4H 7D TS 4S 2D 4D 8C 6S TH 2H JD 9H 3D
           1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Note: To quit entire game, input the same number for 1st card & 2nd card.
If there's no more pair available, input '0' in both cards. To eliminate
pairs in your hand, put in the corresponding number of your card. If the
input cards are not a pair, you get a point deduction. Otherwise, if the
cards are a pair, you get 2 points!

Put in the corresponding number of your cards!

1st Card: 2
2nd Card: 3

You got it!
```

When players are done in eliminating pairs in their hands, score board will be shown, and as a feature I added, to determine who goes first next round, there will be tossing of coins. HEADS – Player 1 & TAILS – Player 2

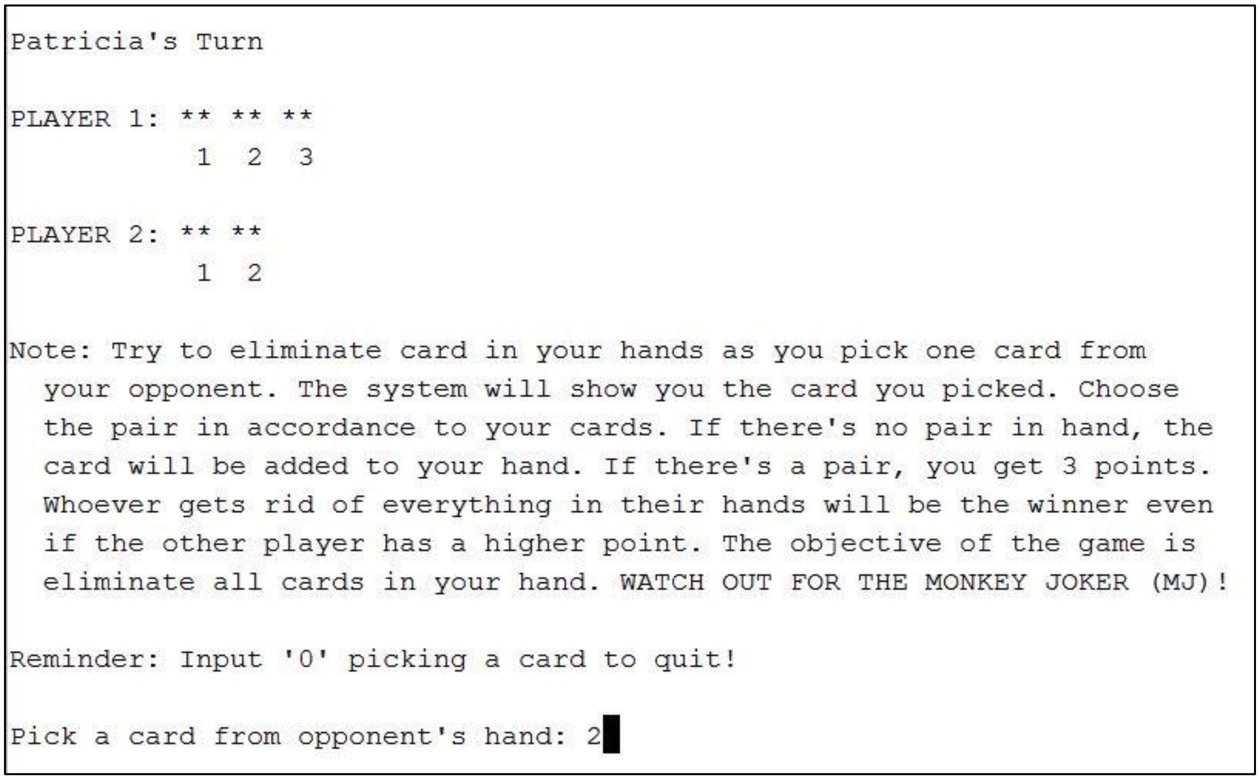

```
          ====================================
                       SCORE BOARD
          ====================================
                   PLAYER 1 ---- 20 pts.
                   PLAYER 2 ---- 20 pts.
          ====================================

ROUND 1 - ELIMINATE ALL CARDS

TOSS COIN to see who goes first!

HEADS - PLAYER 1 | TAILS - PLAYER 2

Press ENTER to proceed!
```

Eliminated pair after first round only gives the player 1 point instead of two. Once the tossing of coin is done, the game will ask for any additional elimination in case either player missed a pair in their hand. It will go on until both decide to not discard anything at all.



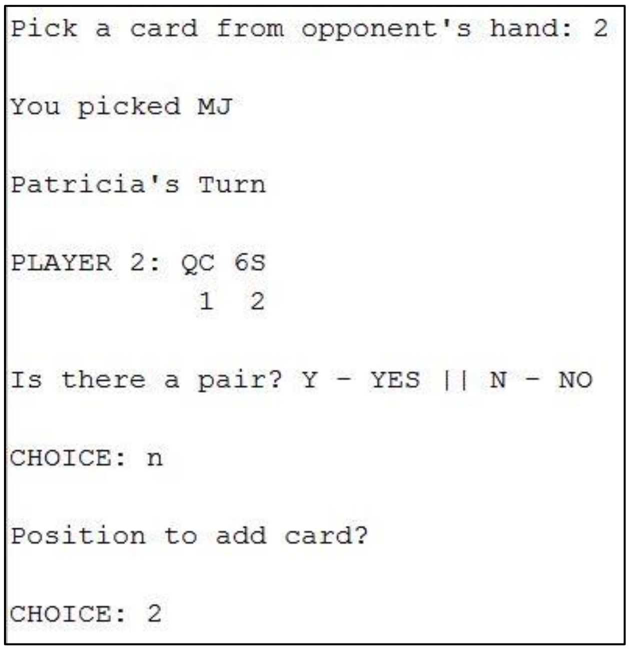
```
HEADS! Shienne goes first!

PLAYER 1: ** ** ** ** ** ** **
           1  2  3  4  5  6  7

PLAYER 2: ** ** ** ** ** **
           1  2  3  4  5  6


Shienne's Turn

PLAYER 1: QD 7C MJ 5C 7H 6C 5H
           1  2  3  4  5  6  7

Would you like to discard a pair in your hand?
             Y - Yes | N - No
CHOICE: █
```

If both player does not want to discard anything or does not have anything to discard, the game commence to the second part of the second round which is to pick a card from the opponent, and then, eliminate a pair of that card from player’s hand.



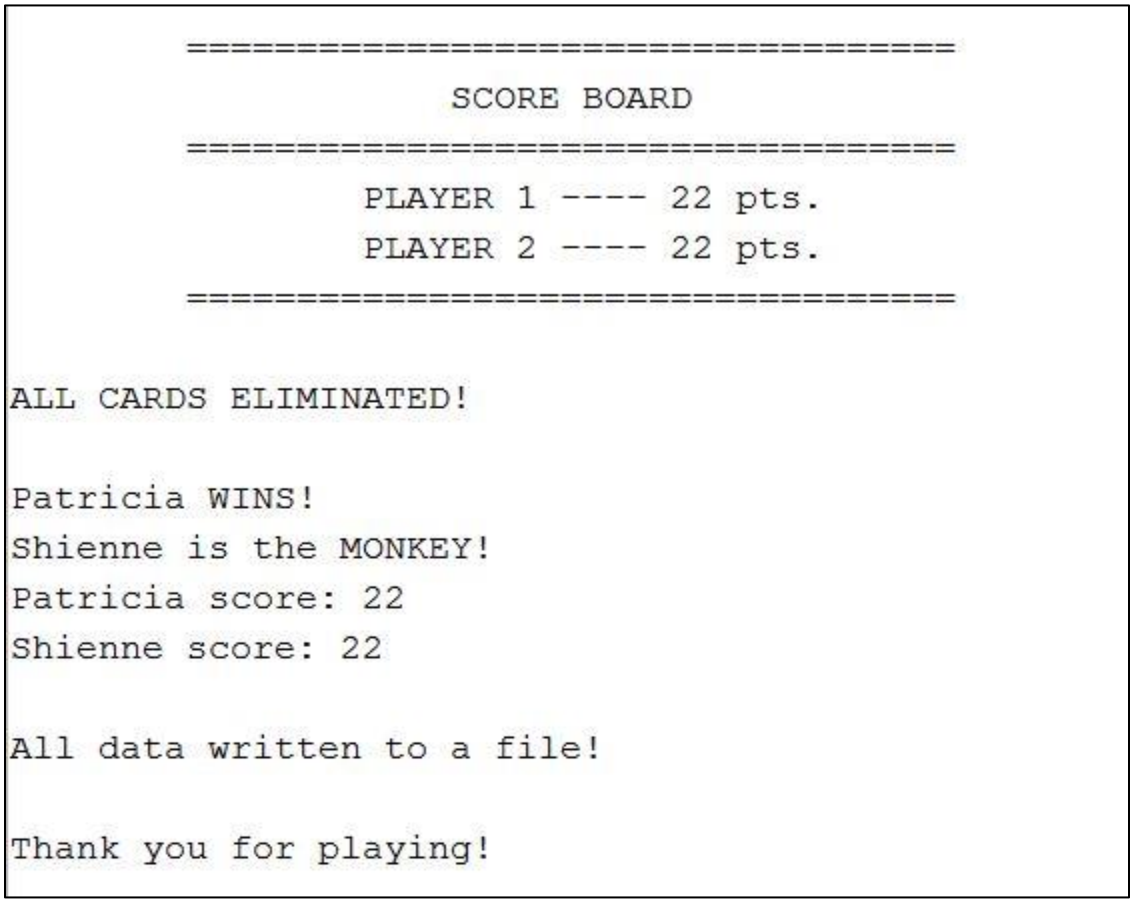
```
Patricia's Turn

PLAYER 1: ** ** **
           1  2  3

PLAYER 2: ** **
           1  2

Note: Try to eliminate card in your hands as you pick one card from
  your opponent. The system will show you the card you picked. Choose
  the pair in accordance to your cards. If there's no pair in hand, the
  card will be added to your hand. If there's a pair, you get 3 points.
  Whoever gets rid of everything in their hands will be the winner even
  if the other player has a higher point. The objective of the game is
  eliminate all cards in your hand. WATCH OUT FOR THE MONKEY JOKER (MJ)!

Reminder: Input '0' picking a card to quit!

Pick a card from opponent's hand: 2
```

If a player unluckily picks “MJ” which stands for Monkey’s Joker, he/she will have the option on where to place the card in hand, so the other opponent won’t have any idea where he/she put it. There’s a good chance that the other player will get it back.



```
Pick a card from opponent's hand: 2

You picked MJ

Patricia's Turn

PLAYER 2: QC 6S
           1  2

Is there a pair? Y - YES || N - NO

CHOICE: n

Position to add card?

CHOICE: 2
```

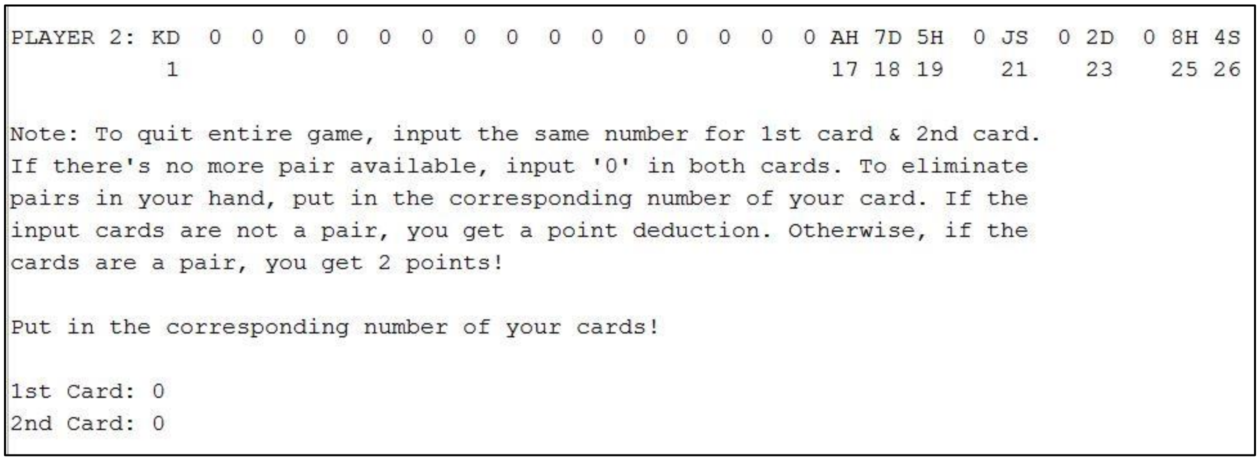
The game will continue until one player decides to end game by inputting ‘0’ in pick a card from opponent or if a player eliminates all of his/her cards in hand leaving only the Joker to the opponent.



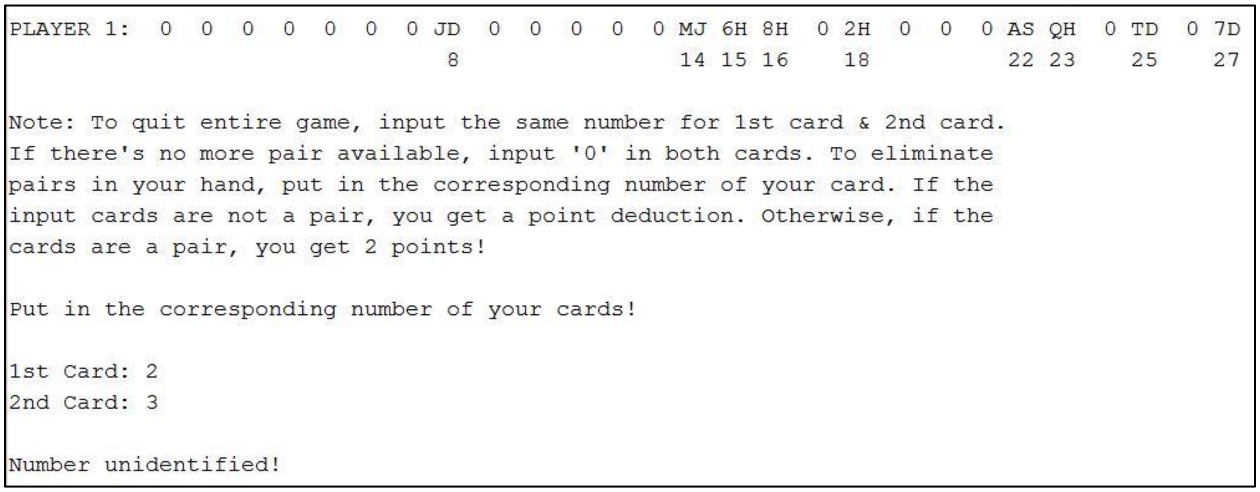
```
               ===================================
                          SCORE BOARD
               ===================================
                      PLAYER 1 ---- 22 pts.
                      PLAYER 2 ---- 22 pts.
               ===================================

ALL CARDS ELIMINATED!

Patricia WINS!
Shienne is the MONKEY!
Patricia score: 22
Shienne score: 22

All data written to a file!

Thank you for playing!
```

All data will then be recorded to a txt file and a bin file.

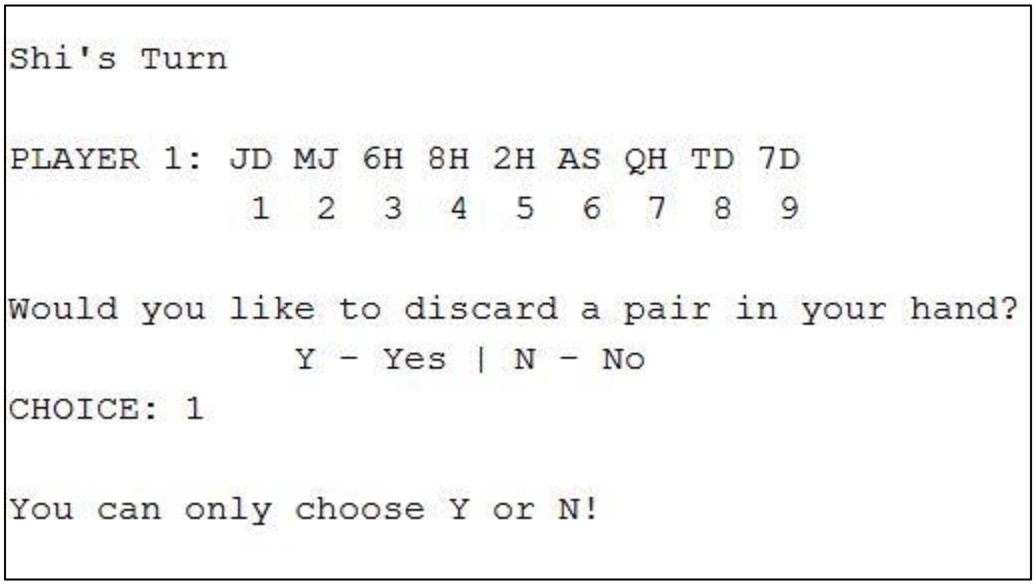
Additional Photos/Screenshots



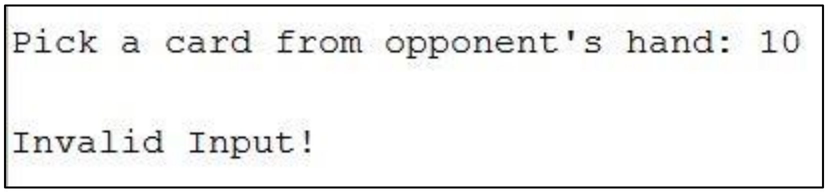
```
PLAYER 2: KD  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 AH 7D 5H  0 JS  0 2D  0 8H 4S
           1                                              17 18 19    21    23    25 26

Note: To quit entire game, input the same number for 1st card & 2nd card.
If there's no more pair available, input '0' in both cards. To eliminate
pairs in your hand, put in the corresponding number of your card. If the
input cards are not a pair, you get a point deduction. Otherwise, if the
cards are a pair, you get 2 points!

Put in the corresponding number of your cards!

1st Card: 0
2nd Card: 0
```



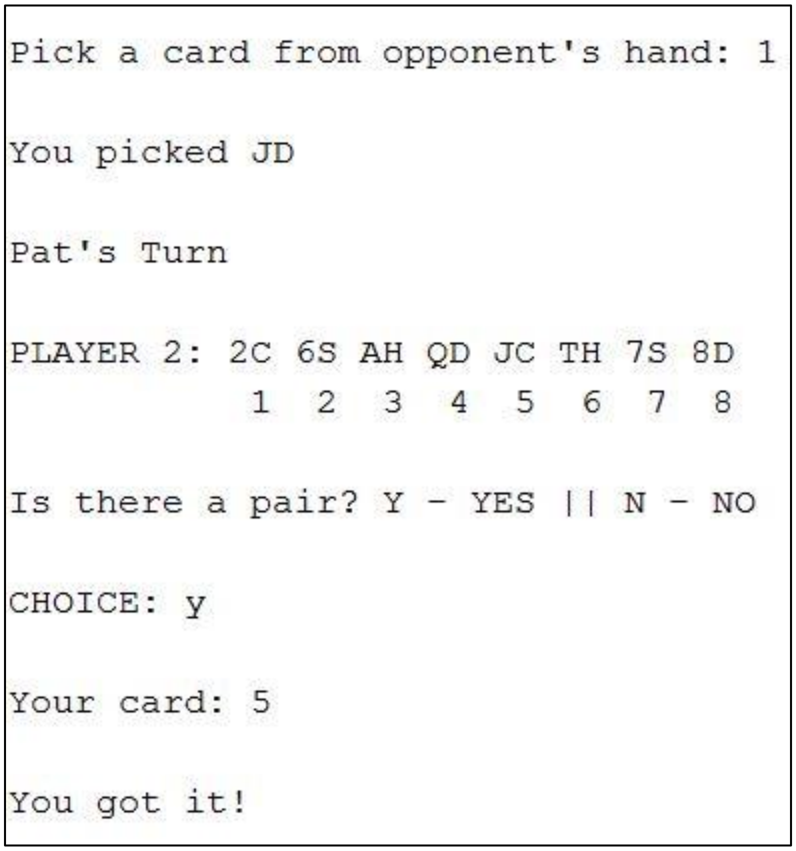
```
PLAYER 1:  0  0  0  0  0  0  0 JD  0  0  0  0  0 MJ 6H 8H  0 2H  0  0  0 AS QH  0 TD  0 7D
                               8                 14 15 16    18          22 23    25    27

Note: To quit entire game, input the same number for 1st card & 2nd card.
If there's no more pair available, input '0' in both cards. To eliminate
pairs in your hand, put in the corresponding number of your card. If the
input cards are not a pair, you get a point deduction. Otherwise, if the
cards are a pair, you get 2 points!

Put in the corresponding number of your cards!

1st Card: 2
2nd Card: 3

Number unidentified!
```



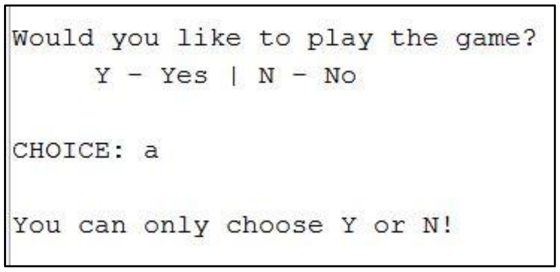
```
Shi's Turn

PLAYER 1: JD MJ 6H 8H 2H AS QH TD 7D
           1  2  3  4  5  6  7  8  9

Would you like to discard a pair in your hand?
             Y - Yes | N - No
CHOICE: 1

You can only choose Y or N!
```



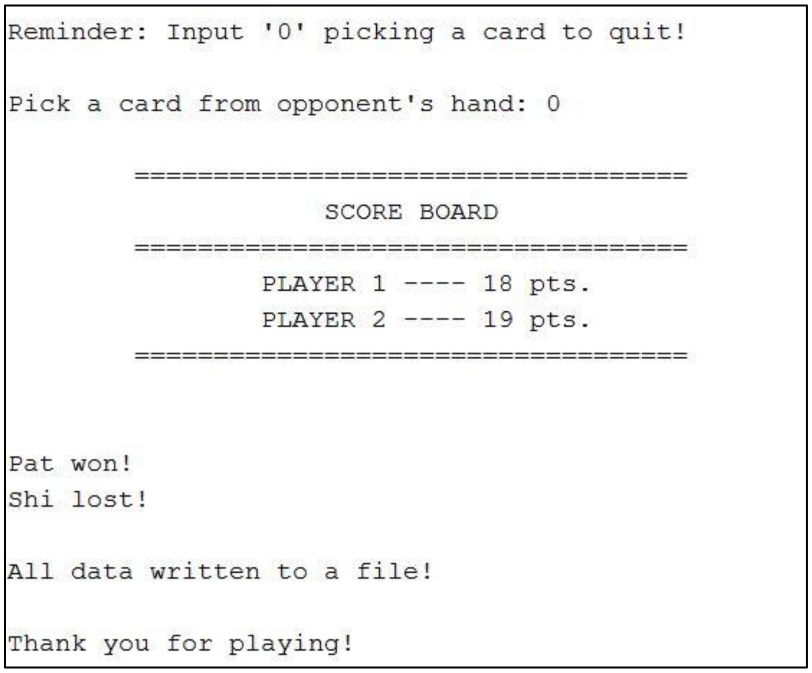
```
Pick a card from opponent's hand: 10

Invalid Input!
```



```
Pick a card from opponent's hand: 1

You picked JD

Pat's Turn

PLAYER 2: 2C 6S AH QD JC TH 7S 8D
           1  2  3  4  5  6  7  8

Is there a pair? Y - YES || N - NO

CHOICE: y

Your card: 5

You got it!
```



```
Would you like to play the game?
     Y - Yes | N - No

CHOICE: a

You can only choose Y or N!
```



```
Reminder: Input '0' picking a card to quit!

Pick a card from opponent's hand: 0

        ===================================
                   SCORE BOARD
        ===================================
                PLAYER 1 ---- 18 pts.
                PLAYER 2 ---- 19 pts.
        ===================================



Pat won!
Shi lost!

All data written to a file!

Thank you for playing!
```

Summary

Project Size: 461 lines
The Number of Variables: Main – about 30

Just like the first version, I used the same structure for applying memory allocation from file deck and player cards. Unlike before, I was now able to write not only the text files but also the binary file. Most menus and user input prompt now have input validation. I created an abstract class with nothing in it but a pure virtual function. It is to be inherited by one of my classes with the virtual function overridden in the child class. This took me a week again to remake as most of my functions were now move to classes. I used a lot of boolean operators just for the counter of so-while loops. I also used templates in my RytFile class for writing to text files and binary files. I've used operatator overloading for round counting

I had a hard time incorporating the templates, and I also had to think of a way to somehow inherit a class that is already templated. In the end, I couldn't do it so I just inherited an abstract class. I had to create one because I didn't know where it could fit my codes. Instead of using the trash array all the time, I incorporated vectors in the second round since it can remove and add elements on command unlike allocated memory or static arrays where you still need to make a copy. It was useful. I also had a hard time figuring out what to do with my functions when I added classes to the project, but I think it turned out well.

Description

The main point I programmed in this project is recreating the second round of the game where you have to be able to remove and add elements on command using vectors. It was tricky because I have to copy the element first before I delete it. Since I was able to solve most of the hard part in project 1, I only had to worry about the second round.

Cross Reference for Project 1

Chapter	Section	Topic	Where in code – Line Number
9	2	Pointer Variables	Players.h (27, 28),
	5	Initializing Pointers	Players.cpp (20, 21)
	7	Pointers as function argument	Players.cpp (23)
	8	Dynamic Memory Allocation	Players.cpp (239, 255-258)
	9	Returning Pointers from Functions	Players.cpp (238, 254)
10	3	C-strings stores as array	47
11	2	Combining data into structures	Cards.h
	3	Accessing structure members	Players.cpp (256-257…)
	5	Arrays of Structures	Players.h (28)
	8	Returning structure from function	Players.cpp (276)
	9	Pointers to structures	Players.h (28)
12	1	File Operations	FileRyt.h (49-116)
13		Classes	Players.h, ElimRnd.h, Points.h, RytFile.h
	3	Defining an Instance of a Class	117, 213, 446
	7	Constructors	Players.h, ElimRnd.h, RytFile.h
	9	Destructors	Players.h, RytFile.h
14	3	Memberwise Assignment	117, 213, 446
	5	Operator Overloading	Players.h (50)
15	4	Redefining Base Class Function	ElimRnd.h (36)
	6	Polymorphism and Virtual Member Functions	Points.h (19)
	7	Abstract Base Classes and Pure Virtual Functions	Points.h (19)
16		Templates	FileRyt.h

Pseudo-code

```
/*
 * File:   main.cpp
 * Author: Shienne Cay
 * Created on June 2, 2017, 9:20 PM
 * Purpose: Project 2 v2
*/

//Input/Output Stream
//Vector Library
//File Stream
//String Library
//Time Library
//C Standard Library

//User Libraries

//Global Constants
//Such as PI, Vc, -> Math/Science values
//as well as conversions from one system of measurements
//to another

//Function Prototypes
//Function to check pairing of a card

//Executable code begins here! Always begins in Main

//Set the random number seed
//Declare Variables
//Player Names
//Start game, choose player, discard pairing, eliminate card
//Game Play counter, Player Scores
//Count for array number
//Trash array - where discarded cards are put in
//LOOPS: x-play game, y-choose player,
 //a-card elimination, b-pair discard, c-elimination cards from opponents
//change-pair elimination, play-change player
//end-end all loop for score tally, elim is for eliminating leftover pair of cards in player's
hands
//1st Card, 2nd Card, Pick from opponent, Your hand
```

```
//Explain rules

//Prompt user for game play
    //If they chose to play, get to next loop
    //If not, end game
    //Invalidate if input isn't 'Y' or 'N'
//Continue loop if x remains true

//Prompt user to choose player
    //If they chose 1, become player 1
    //Exit loop
    //If they chose 2, become player 2
    //Exit loop
    //If they chose 'Q', end game
    //Invalidate if input '1', '2', or 'Q'
//Continue loop if y remains true

//Prompt user to enter name
//Create object of player using names as parameter
    //Show both cards anonymously
    //Use overload operator to indicate round


//Start first part of the game - loop hand card elimination
    //If each player had their turn, exit elimination loop
    //Show card of the user current playing whether Player or Player 2
        //Set a true for player change

//Start elimination loop
    //Prompt user for first card
    //Prompt user for second card
        //If any input is less than 0 or greater than 27, invalidate!
            //Show card again based on player playing
        //If user put '0' in both card, exit elimination loop
    //If player 1, change to player 2
        //Add game play counter
    //If player 2, change to player 1
        //Add game play counter
    //Set a to false
    //If input is the same except 0, quit game play
    //If input is none other than above, check player
        //If player 1
        //Check if input has already been used
        //If not, validate if cards are a pair
        //If cards are a pair, add to trash array
        //Add two points for player
        //Show Player Cards
    //If cards are not a pair
        //Deduct 1 point
    //If input has already been used
        //Invalidate and show player cards
    //Do the same if player is player 2
//Continue loop if a remains true
//Continue loop if change remains true

//Create Eliminate second round object
//Show current points
//Show cards left from player's hand
//Create Vector of strings for second round elimination
    //Increment round play
//Fill the created vectors with cards left in players hand
//Show play round
//Toss coin to determine who's first
    //Use random generator for coin tossing
        //If heads, player 1 goes first
        //If tails, player 2 goes first
//Show cards anonymously

//If a player discards all of his or her cards, end loop
    //Reset gameplay counter;
    //Set b for additional card as true
    //Show whoever players' turn is based on coin toss
    //Ask If they have remaining cards in hand to discard
        //Prompt user for answer
            //If yes, prompt entering the two cards
            //If player 1
                //Check if number input is on the limit
                //If yes, check if it is a pair
                //If it is, erase or eliminate card in vector
                //If it's not a pair, invalidate
                    //Deduct 1 point in score
                    //If went over the limit, show number unidentified
            //Do the same if player is player 2

//If player chose not to discard
```

```
//Increment Game counter and change player
//If both game play counter meets requirements, exit loop by setting b as false
    //Declare c as true
    //Change back player to the real chronology
//reset game play counter

//if there's no cards left for one player, end all loop

//If cards still exist, show cards
//Explain rules
//Prompt user to input card number from hand
//Input invalidation
//Prompt user to input card number from hand
    //If player chose to quit, tally score
//Show the picked card from the opponent
//Prompt user to input number from opponent's hand
    //Prompt user if there is a pair from the picked card to player's hand
        //If yes, input number corresponding that card
        //Validate if it's a pair
            //If it is, score is added and the cards are eliminated
                //Then change player
            //If it's not a pair, let player know
                //Deduct 1 point from their score
    //If there's no pair, ask user where to position in his card the card that is to be added in his hand
//Invalidate if user's input exceeded the limit
    //Add card to player's hand
    //Increment game play counter
        //If game play counter meets requirement, exit loop
         //Do the same if player 2

//Continue loop if c remains true
 //Continue loop until end turns false
//Set new score into class
//Show score board
//Create data write class to write all data to text file and binary file
//End of Game
```

Program

```
/*
 * File:   main.cpp
 * Author: Shienne Cay
 * Created on June 2, 2017, 9:20 PM
 * Purpose: Project 2 v2
 */

//System Libraries
#include <iostream>
#include <vector>
#include <fstream>
#include <string>
#include <ctime>
#include <cstdlib>
using namespace std;

//User Libraries
#include "Players.h"
#include "Points.h"
#include "ElimRnd.h"
#include "FileRyt.h"

//Global Constants
//Such as PI, Vc, -> Math/Science values
//as well as conversions from one system of measurements
//to another

//Function Prototypes
bool check(int [], short, short, int);

int main(int argc, char** argv) {
    srand(static_cast<unsigned int>(time(0)));

    //Declare Variables
    char name1[80], name2[80];
    char start, player, discard, yElim;
    int gmPly1=0, gmPly2=0, p1scor=0,
        p2scor=0, cntr1=1, cntr2=1;
    int trsh1[28]={}, trsh2[27]={};
    bool x=true, y=true,
         a=true, b=true, c=true
         change=true, play=true,
         end=true, elim=true;
    short frst, scnd, pick, pair;

    cout<<"\t\tWelcome to the Monkey Game!"<<endl<<endl;
    cout<<"Mechanics: There are two players in the game and the cards are\n"
        <<"evenly divided between the two. If a player has the same value\n"
        <<"in one hand, the two card goes into the bin. The goal is to be\n"
        <<"the first to lose all the cards. When there are no more of the\n"
        <<"the pair, each player gets to pick one card from the opponent.\n"
        <<"Do this until one player wins. Loser is called the MONKEY!"<<endl<<endl;

    do {
        cout<<"Would you like to play the game?"<<endl;
        cout<<"     Y - Yes | N - No"<<endl;
        cout<<"\nCHOICE: ";
        cin>>start;
        if (start=='y'||start=='Y') x=false;
        else if (start=='n'||start=='N') return 0;
        else {
            cout<<"\nYou can only choose Y or N!"<<endl<<endl;
            cin.clear();
        }
    } while(x);

    do {
        cout<<"\n             PLAY GAME AS:"<<endl<<endl;
        cout<<"     1 - Player 1 | 2 - Player 2"<<endl;
        cout<<"              Q - QUIT"<<endl;
        cout<<"\nCHOICE: ";
        cin>>player;
        if (player=='1') {
            play=true;
            y=false;
        }
        else if (player=='2') {
            play=false;
            y=false;
        }
        else if (player=='Q'||player=='q') return 0;
        else {
            cout<<"\nYou can only choose 1, 2 or Q!"<<endl;
            cin.clear();
        }
    } while(y);

    cin.ignore(256, '\n');
    if (play) {
        cout<<endl;
        cout<<"Player 1 Name: ";
        cin.getline(name1, 80);
        cout<<"Player 2 Name: ";
        cin.getline(name2, 80);
    }
    else {
        cout<<endl;
        cout<<"Player 2 Name: ";
        cin.getline(name2, 80);
        cout<<"Player 1 Name: ";
        cin.getline(name1, 80);
    }

    Player plyNum(name1, name2);
    cout<<"\nCards have been dispersed!"<<endl<<endl;
    plyNum.shwBoth();
    cout<<"Press ENTER to start game!"<<endl;
    cin.get();
    ++plyNum;
    cout<<"ROUND "<<plyNum.plyRndG()<<" - Discard all pairs!"<<endl;

    do {
        if (gmPly1==1&&gmPly2==1) change=false;
        else {
            plyNum.shwCrd(play);
            a=true;
            do {
                cout<<"Note: To quit entire game, input the same number for 1st card & 2nd card."<<endl;
                cout<<"If there's no more pair available, input '0' in both cards. To eliminate"<<endl;
                cout<<"pairs in your hand, put in the corresponding number of your card. If the"<<endl;
                cout<<"input cards are not a pair, you get a point deduction. Otherwise, if the"<<endl;
                cout<<"cards are a pair, you get 2 points!"<<endl<<endl;
                cout<<"Put in the corresponding number of your cards!"<<endl<<endl;
                cout<<"1st Card: ";
                cin>>frst;
                cout<<"2nd Card: ";
                cin>>scnd;
                if (frst<0||frst>27||scnd<0||scnd>27) {
                    cout<<"\nNumber unidentified!"<<endl<<endl;
                    if (play) plyNum.shwLeft(play, trsh1);
                    else plyNum.shwLeft(play, trsh2);
                }
                else {
                    if (frst==0&&scnd==0) {
                        if (play) {
                            gmPly1++;
                            play=false;
                        }
                        else {
                            gmPly2++;
                            play=true;
                        }
                        a=false;
                    }
                    else if (frst==scnd) return 0;
                    else {
                        if (play) {                    //If player 1
                            if (check(trsh1, frst, scnd, 28)) {
                                if (plyNum.valid8(frst, scnd, play)) {
                                    trsh1[cntr1]=frst; cntr1++;
                                    trsh1[cntr1]=scnd; cntr1++;
                                    cout<<"\nYou got it!"<<endl;
                                    p1scor+=2;
                                    plyNum.shwLeft(play, trsh1);
                                }
                                else {
                                    cout<<"\nIt's not a pair!"<<endl;
                                    cout<<"1 point is deducted to your score!"<<endl<<endl;
                                    p1scor--;
                                }
                            }
                            else {
                                cout<<"\nNumber unidentified!"<<endl<<endl;
                                plyNum.shwLeft(play, trsh1);
                            }
                        }
                        else {
                            if (frst==27||scnd==27) {
                                cout<<"\nNumber Unidentified!\n\n";
                                plyNum.shwLeft(play, trsh2);
                            }
                            else {
                                if (check(trsh2, frst, scnd, 27)) {
                                    if (plyNum.valid8(frst, scnd, play)) {
                                        trsh2[cntr2]=frst; cntr2++;
                                        trsh2[cntr2]=scnd; cntr2++;
                                        cout<<"\nYou got it!"<<endl;
                                        p2scor+=2;
                                        plyNum.shwLeft(play, trsh2);
                                    }
                                    else {
                                        cout<<"\nIt's not a pair!"<<endl;
                                        cout<<"1 point is deducted to your score!"<<endl<<endl;
                                        p2scor--;
                                    }
                                }
                                else {
                                    cout<<"\nNumber unidentified!"<<endl<<endl;
                                    plyNum.shwLeft(play, trsh2);
                                }
                            }
                        }
                    }
                }
            } while(a);
        }
    } while(change);

    Elimin8 secRnd(p1scor, p2scor);
    cin.ignore(256, '\n');
```

```
secRnd.points();
plyNum.setLeft(trsh1, trsh2);
vector<string> plyr1(plyNum.getLft1());
vector<string> plyr2(plyNum.getLft2());
++plyNum;       //Increment round play
plyNum.filVec(plyr1, plyr2, trsh1, trsh2);

cout<<"ROUND "<<plyNum.plyRndG()<<" - ELIMINATE ALL CARDS"<<endl<<endl;
cout<<"TOSS COIN to see who goes first!"<<endl<<endl
   <<"HEADS - PLAYER 1 | TAILS - PLAYER 2"<<endl<<endl;
cout<<"Press ENTER to proceed!"<<endl<<endl;
cin.get();
unsigned short coin=rand()%2+1;
if (coin==1) {
   cout<<"HEADS! "<<plyNum.getNme1()<<" goes first!"<<endl<<endl;
   play=true;
}
else {
   cout<<"TAILS! "<<plyNum.getNme2()<<" goes first!"<<endl<<endl;
   play=false;
}

secRnd.shwCrds(plyr1, plyr2);
cout<<endl;

do {
   if (plyr1.size()==1||plyr2.size()==1) end=false;
   else {
      gmPly1=0; gmPly2=0;
      b=true;
      do {
         if (play) cout<<endl<<plyNum.getNme1()<<"'s Turn"<<endl;
         else cout<<endl<<plyNum.getNme2()<<"'s Turn"<<endl;
         secRnd.elimCrd(plyr1, plyr2, play);
         cout<<"Would you like to discard a pair in your hand?"<<endl;
         cout<<"            Y - Yes | N - No                 "<<endl;
         cout<<"CHOICE: ";       //Prompt user for answer
         cin>>discard;
         if (discard=='Y'||discard=='y') {
            cout<<"\nInput corresponding number."<<endl<<endl;
            cout<<"1st Card: ";
            cin>>frst;
            cout<<"2nd Card: ";
            cin>>scnd;
            if (frst==0&&scnd==0) return 0;
            if (play) {                    //If player 1
               if (frst<plyr1.size()&&frst>0&&scnd>0&&scnd<plyr1.size()) {
                  if (secRnd.valid8(plyr1, plyr2, frst, scnd, play)) {
                     if (frst>scnd) {
                        plyr1.erase(plyr1.begin()+frst);
                        plyr1.erase(plyr1.begin()+scnd);
```

```
                }
                else {
                    plyr1.erase(plyr1.begin()+scnd);
                    plyr1.erase(plyr1.begin()+frst);
                }
            }
            else {
                cout<<"\nIt's not a pair!\n\n";
                p1scor--;
            }
        }
        else {
            cout<<"\nNumber unidentified!\n\n";
        }
    }
    else {
        if (frst<plyr2.size()&&frst>0&&scnd>0&&scnd<plyr2.size()) {
            if (secRnd.valid8(plyr1, plyr2, frst, scnd, play)) {
                if (frst>scnd) {
                    plyr2.erase(plyr2.begin()+frst);
                    plyr2.erase(plyr2.begin()+scnd);
                }
                else {
                    plyr2.erase(plyr2.begin()+scnd);
                    plyr2.erase(plyr2.begin()+frst);
                }
            }
            else {
                cout<<"\nIt's not a pair!\n\n";
                p2scor--;
            }
        }
        else {
            cout<<"\nNumber unidentified!\n\n";
        }
    }
}
else if (discard=='N'||discard=='n') {
    if (play) {
        gmPly1++;
        play=false;
    }
    else {
        gmPly2++;
        play=true;
    }
    if (gmPly1==1&&gmPly2==1) b=false;
}
else {
    cout<<"\nYou can only choose Y or N!"<<endl<<endl;
    cin.clear();
```

```
            b=true;
        }
    }while (b);
    c=true;
    if (play) play=false;
    else play=true;
    gmPly1=0; gmPly2=0;
    do {
        if (plyr1.size()==1||plyr2.size()==1) {
            end=false;
            c=false;
        }
        else {
            if (play) cout<<endl<<plyNum.getNme1()<<"'s Turn"<<endl;
            else cout<<endl<<plyNum.getNme2()<<"'s Turn"<<endl;
            cout<<endl;
            secRnd.shwCrds(plyr1, plyr2);
            cout<<"\nNote: Try to eliminate card in your hands as you pick one card from"<<endl;
            cout<<"  your opponent. The system will show you the card you picked. Choose"<<endl
                <<"  the pair in accordance to your cards. If there's no pair in hand, the"<<endl
                <<"  card will be added to your hand. If there's a pair, you get 3 points."<<endl
                <<"  Whoever gets rid of everything in their hands will be the winner even"<<endl
                <<"  if the other player has a higher point. The objective of the game is "<<endl
                <<"  eliminate all cards in your hand. WATCH OUT FOR THE MONKEY JOKER
(MJ)!"<<endl<<endl;
            cout<<"Reminder: Input '0' picking a card to quit!"<<endl<<endl;
            cout<<"Pick a card from opponent's hand: ";
            cin>>pick;
            while (secRnd.limit(plyr1, plyr2, play, pick)||cin.fail()) {
                cout<<"\nInvalid Input!\n\n";
                cin.clear();
                cin.ignore(256, '\n');
                cout<<"Pick a card from opponent's hand: ";
                cin>>pick;
            }
            if (pick==0) {
                c=false;
                end=false;
            }
            else {
                secRnd.picked(plyr1, plyr2, play, pick);
                elim=true;
                do {
                    if (play) cout<<endl<<plyNum.getNme1()<<"'s Turn"<<endl;
                    else cout<<endl<<plyNum.getNme2()<<"'s Turn"<<endl;
                    secRnd.elimCrd(plyr1, plyr2, play);
                    cout<<"Is there a pair? Y - YES || N - NO "<<endl<<endl<<"CHOICE: ";
                    cin>>yElim;
                    cout<<endl;
                    if (yElim=='Y'||yElim=='y') {
                        cout<<"Your card: ";
    cin>>pair;
    if (secRnd.valid82(plyr1, plyr2, pick, pair, play)) {
        if (play) {
            plyr2.erase(plyr2.begin()+pick);
            plyr1.erase(plyr1.begin()+pair);
            p1scor++;
            play=false;
        }
        else {
            plyr1.erase(plyr1.begin()+pick);
            plyr2.erase(plyr2.begin()+pair);
            p2scor++;
            play=true;
        }
        cout<<"\nYou got it!"<<endl<<endl;
        elim=false;
    }
    else {
        cout<<"\nIt's not a pair!\n\n";
        if (play) p1scor--;
        else p2scor--;
    }
 }
else if (yElim=='n'||yElim=='N') {
    bool rmze=true;
    short pos;
    if (play) {
        do {
            cout<<"Position to add/place card?\n\nCHOICE: ";
            cin>>pos;
            if (pos<plyr1.size()&&pos>0) rmze=false;
            else {
                cout<<"\nPosition must be between 0 and "<<plyr1.size()<<"!"<<endl<<endl;
                cin.clear();
            }
        }while(rmze);
        auto it=plyr1.insert(plyr1.begin()+pos, plyr2[pick]);
        plyr2.erase(plyr2.begin()+pick);
        play=false;
        elim=false;
        if (play) gmPly1++;
        else gmPly2++;
        if (gmPly1==1&&gmPly2==1) c=false;
    }
    else {
        do {
            cout<<"Position to add card?\n\nCHOICE: ";
            cin>>pos;
            if (pos<plyr2.size()&&pos>0) rmze=false;
            else {
                cout<<"\nPosition must be between 0 and "<<plyr2.size()<<"!"<<endl<<endl;
```

```
                        cin.clear();
                    }
                }while(rmze);
                auto it=plyr2.insert(plyr2.begin()+pos, plyr1[pick]);
                plyr1.erase(plyr1.begin()+pick);
                play=true;
                elim=false;
                if (play) gmPly1++;
                else gmPly2++;
                if (gmPly1==1&&gmPly2==1) c=false;
              }
            }
            else {
              cout<<"\nYou can only choose 'Y' or 'N'\n\n";
              cin.clear();
              elim=true;
            }
          }while (elim);
        }
      }
    } while(c);
  }
}while (end);

secRnd.setScr(p1scor, p2scor);
secRnd.points();

RytFile<short> data(p1scor, p2scor);
data.dataRyt(name1, name2, plyr1, plyr2);

cout<<"\nAll data written to a file!"<<endl<<endl;
cout<<"Thank you for playing!"<<endl;

return 0;
}
```