Will Bouasisavath

007547473

CSCI 112

BlackJack Project

This project is based on a BlackJack system where users can enjoy a fun and exciting way to learn BlackJack. The purpose of this project will allow us to comprehend the fundamentals of BlackJack in an easier format. The goal of BlackJack is to get 21 without going over while beating the dealer hand. The application will give a realistic feel of the casino's most popular game where you can perform activities that are done in a real casino environment.

```c
//  BlackJack
//  Will Bouasisavath on 12/17/20.
//  007547473
//  CSCI 112

#include <stdio.h>
#include <time.h>
#include <stdlib.h>

int shuff(int cards[]) //function to shuffle deck of cards
{

    int t;
    int i;
    int desk[52];
    for (i = 0; i < 52; i++)
        desk[i] = (i / 13 + 3) * 100 + i % 13 + 1;

    srand(time(NULL));
    for (i = 0; i < 52; i++)
    {
        do
        {
            t = rand() % 52;
        } while (desk[t] == 0);
        cards[i] = desk[t];
        desk[t] = 0;
    }

    return 0;
}

int convert_jkq(int a) //face value of cards is 10
{
    if ((a%100 == 11) ||(a%100 == 12) ||(a%100 == 13)) return (a/100)*100 + 10;
    else return a;
}

```

Part1 and 2 uses <time.h> for srand time(NULL) and variable case statements.

```c
void pic(int num) //potential number of cards
{
    char fl;
    int po_num;

    fl = num / 100;
    po_num = num % 100;
    switch (po_num)
    {
        case 1:
        {
            printf("* %c *\n", fl);
            printf("* A *\n");
            break;
        }
        case 2:
        case 3:
        case 4:
        case 5:
        case 6:
        case 7:
        case 8:
        case 9:
        case 10:
        {
            printf("* %c *\n", fl);
            printf("* %2d *\n", po_num);
            break;
        }
        case 11:
        {
            printf("* %c *\n", fl);
            printf("* J *\n");
            break;
        }
        case 12:
        {
            printf("* %c *\n", fl);
            printf("* Q *\n");
            break;
        }
        case 13:
        {
            printf("* %c *\n", fl);
            printf("* K *\n");
            break;
```

```c
    //whether player receives another cards
    i = 0;
    for (i = 0; i < 3; i++)
    {
        char j = 'n';

        printf("Would you like to hit? Input y or n:\n");
        do{
            j = getchar();
        } while (j!='y' &&j!='n');

        if (j =='y')
        {
            printf("Play receives another card.\n");
            pcards[i + 2] = cards[i + 4];
            printf("and your card %d is:\n", i + 3);
            pic(pcards[i +2 ]);

            if (pcards[i + 2] % 100 == 1)
            {
                printf("A value of the card %d, input 'y' for 11 or 'n' for 1:\n", i + 3);
                do{
                    d = getchar();
                } while (d!='y' && d!='n');
                if (d == 'y')
                {
                    printf("chosen value 11 for card A.\n");
                    psum = psum + 11;
                }
                else if(d == 'n')
                {
                    printf("chosen value 1 for card A.\n");
                    psum = psum +1;
                }
            }
            else if (convert_jkq(pcards[i+2]) %100 ==10) psum = psum + 10;
            else psum = psum + pcards[i+2]%100;

            if (psum > 21)
            {
                printf("Sum of Player's cards now:%d\n\n",psum);
                printf("Dealer win!\n");
                return 1;
            }
            else if (psum == 21)
```

```c
int play(void) //function to start the game
{
    int i;
    int psum=0;
    int bsum=0;
    int pcards[5]={0};
    int bcards[5]={0};
    int cards[52];
    char go_on;
    char d;

    printf("Welcome to BlackJack!\n"
    "Good Luck and Have fun! Press Enter:\n");
    do{
        go_on = getchar();
    } while (go_on != '\n');
    printf("\n");

    //shuffles the cards
    shuff(cards);

    //gives the cards
    pcards[0] = cards[0];
    pcards[1] = cards[1];
    bcards[0] = cards[2];
    bcards[1] = cards[3];

    //the 2 cards player get
    printf("Top of Dealer's cards:\n");
    pic(bcards[0]);
    printf("\n");
    printf("Cards of Player:\n");
    pic(pcards[0]);
    //printf("\n");
    pic(pcards[1]);
    //printf("\n");

    i = 0;
    for (i = 0; i < 2; i++)
    {
        if (pcards[i] % 100 == 1)
        {
            printf("A value of the card %d, input 'y' for 11 or 'n' for 1 :\n", i + 1);
            do{
                d = getchar();
            } while (d!='y' && d!='n');
```

```c
    //the 2 cards of Dealer/Computer
    //i=0;
    printf("Dealer's cards:\n");
    pic(bcards[0]);
    pic(bcards[1]);

    if (bcards[0]%100 + bcards[1]%100 == 2)
    {
        bsum=12; //two A cards
        printf("Sum of Dealer's cards now:%d\n\n", bsum);
    }
    else if ((convert_jkq(bcards[0]))%100 + (convert_jkq(bcards[1]))%100 ==1)
    {
        bsum=21;
        printf("Sum of Dealer's cards now:%d\n\n", bsum);
        printf("Dealer win!\n");
        return 1;
    }
    else if (bcards[0]%100==1 || bcards[1]%100==1)
    {
        bsum=(bcards[0]+bcards[1])%100+(rand()%2)*10;
        printf("Sum of Dealer's cards now:%d\n\n", bsum);
    }
    else
    {
        bsum = (convert_jkq(bcards[0]))%100 + (convert_jkq(bcards[1]))%100;
        printf("Sum of Dealer's cards now:%d\n\n", bsum);
    }

    //whether computer get another cards until bsum>16
    //i=0;
    for (i=0; i<3 && bsum<17; i++)
    {
        bcards[i+2]=cards[i+7];
        printf("Dealer's card %d is:\n", i+3);
        pic(bcards[i+2]);

        if (bcards[i+2]%100 == 1)
        {
            if (bsum+11 <= 21)
            {
                printf("Dealer has chosen A as 11\n");
                bsum = bsum+11;
                printf("Sum of Dealer's cards now:%d\n\n", bsum);
            }
```

Part 3, 4, 5 uses global variables of value of cards with if else statements giving cycles of certain cards for both the player and the dealer.

Part 6 determines whether player wins or dealer wins!

```c
    //the last step
    if (bsum>21 || psum>bsum)
    {
        printf("Player wins!\n");
        return 0;
    }
    else if (psum == bsum)
    {
        printf("Tie!\n");
        return 3;
    }
    else if (psum < bsum)
    {
        printf("Dealer wins!\n");
        return 1;
    }

    return 3;
}

int main(void) // function for asking player to play again
{
    char again;

    play();

    printf("\nPlay again? Input 'y' or 'n':\n");
    do{
        again = getchar();
    } while (again!='y' && again!='n');

    if (again == 'y')
    {
        printf("\nGood Luck!\n\n");
        main();
    }

    return 0;
}
```

blackjackoutput.txt

```
Welcome to BlackJack!
Good Luck and Have fun! Press Enter:


Top of Dealer's cards:
*  *
*  7 *

Cards of Player:
*  *
* K *
*  *
*  3 *
Sum of Player's cards now:13

Would you like to hit? Input y or n:
y

Play recieves another card.
and your card 3 is:
*  *
*  8 *
Sum of Player's cards now:21

Player win!

Play again? Input 'y' or 'n':
y
```