C A S I N O
G A M E

CIS 5 41595
Mehak Lohchan
Winter 2023

TABLE OF CONTENTS

INTRODUCTION

A Casino is a facility that is comprised of many games where people can gamble. Some popular games consist of Roulette, Number Guessing, Poker, BlackJack, and so on.

GAME RULES/OBJECTIVES

In my Casino Game, I incorporated Number Guessing and BlackJack.

In Number Guessing, a Player place a bidding bet on a random number from 1-10, and if their guess stands correct, they receive a balance ten times of how much they bid. Otherwise, they lose the money they had bid on the number.

In BlackJack, the Player go against the Dealer. Both the Player and the Dealer start off with a deck of two cards. The Player is given the option to "Hit" or "Stay." If the player decides to "Hit" the receive another card. Whereas, if the player decides to "Stay", their turn is over and the dealer wins the game. The objective of this game is to have a deck of cards that add up to strictly 21, no more, no less. Whoever reaches this value first, wins the game.

At a Casino, people lose money and people win money. But, everyone deserves to have a good time, which is why I decided to incorporate another game called 2048. 2048 is a puzzle sliding game where the player must slide the numbered tiles on a grid and combine them to create the number 2048.

MY APPROACH

Finding an Algorithm to Create A Casino Game

To create an algorithm for my project, I researched on different types of Casino games and what the most popular games were. There were many factors I had to take into consideration such as:

1. How do I was going to construct the game
2. Which games I wanted to include in my project
3. How I would go about programming each game
4. How long will the User will be able to play the games

and so on.

My Thought Process

To help myself come up with answers to these questions, I created my code in small versions, such as Dr. Lehr has taught me in lecture, which really helped me understand what and what not to include in my project. As well as including many comments so that I am able to make sense of my code.

Summary

Project Size: 770 lines

This project includes many concepts that I have learned in Dr. Lehr's CIS 5 course, as well as the class textbook.
For example, I programmed two games in a Casino, where I utilized the switch case function to allow the User to play whichever game they would like.

PSEUDOCODE

Initialize

The enter the Casino, the User must place a deposit

If the User's Deposit is > 15
    They are allowed in the Casino

Else
    They are required to enter a minimum entry fee of $15

If User wants to play a game in the Casino
    They will be given an option to choose which game

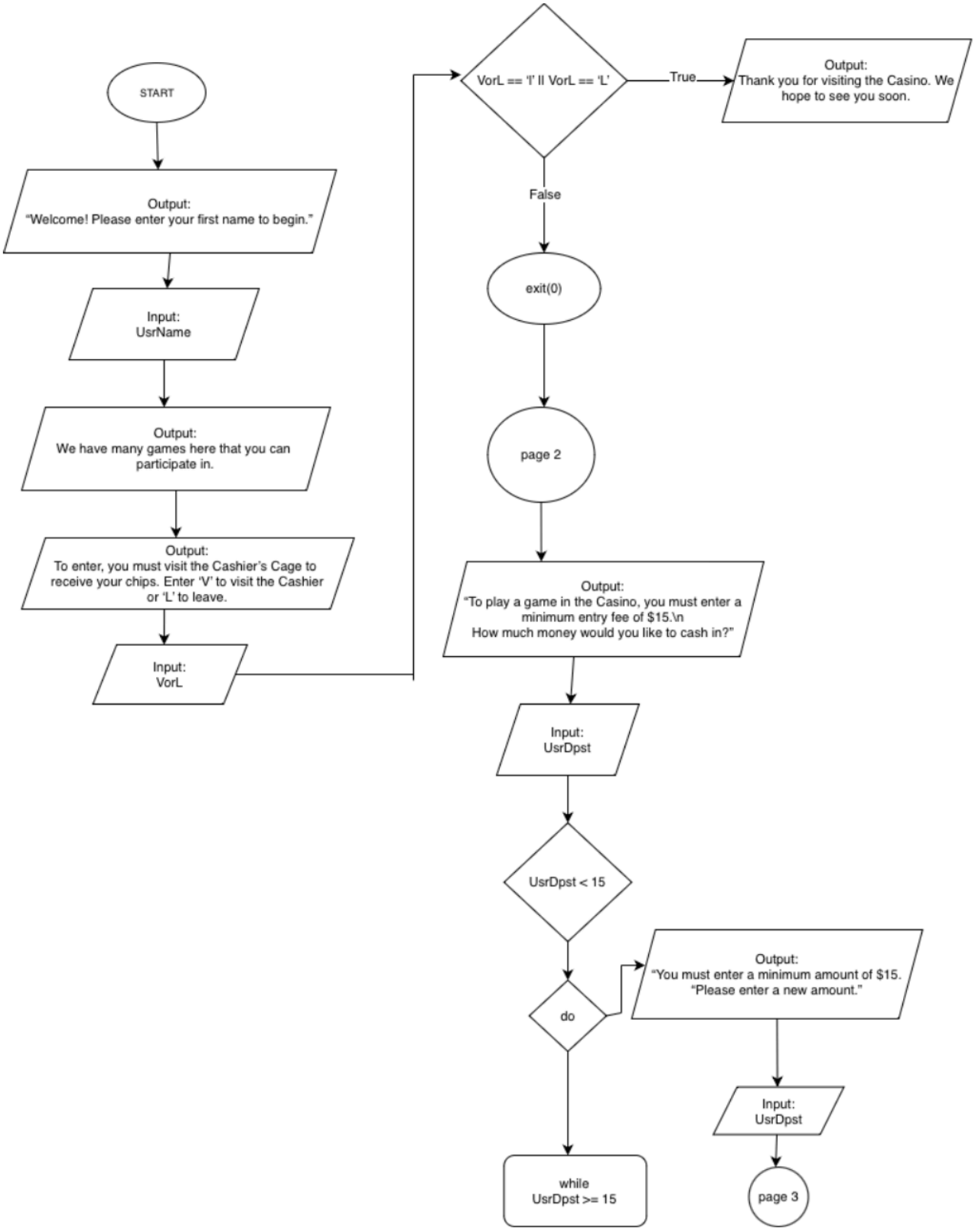
If the User picks game 'n' they are allowed in and are able to play the game

If the User does not have a minimum of $5
    They are required to leave

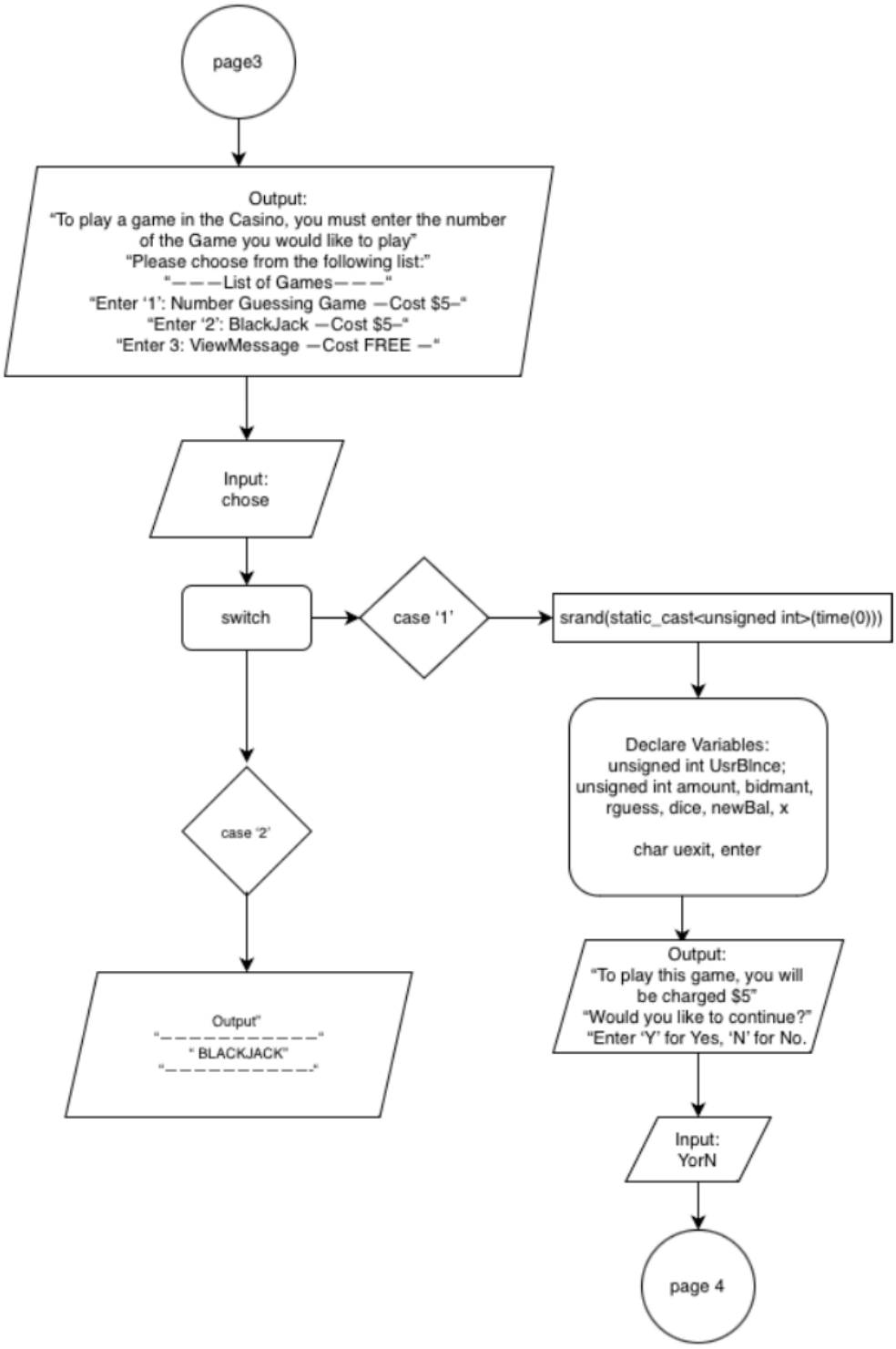
Else
    They must enter a Deposit greater than or equal to $5

If the Deposit is valid
    The User is allowed to play any game of their choice

FLOWCHART


START
Output:
"Welcome! Please enter your first name to begin."
Input:
UsrName
Output:
We have many games here that you can participate in.
Output:
To enter, you must visit the Cashier's Cage to receive your chips. Enter 'V' to visit the Cashier or 'L' to leave.
Input:
VorL
VorL == 'l' II VorL == 'L'
True
Output:
Thank you for visiting the Casino. We hope to see you soon.
False
exit(0)
page 2
Output:
"To play a game in the Casino, you must enter a minimum entry fee of $15.\n
How much money would you like to cash in?"
Input:
UsrDpst
UsrDpst < 15
do
Output:
"You must enter a minimum amount of $15.
"Please enter a new amount."
Input:
UsrDpst
page 3
while
UsrDpst >= 15

FLOWCHART CONT.



CONCEPTS UTILIZED

Libraries

<iostream> library

TYPE	DESCRIPTION	LOCATION
cout	Output	Line 192
cin	Input	Line 83
static_cast	Statically cast as a different variable	Line 329

<cstdlib> library

TYPE	FREQUENCY	DESCRIPTION	LOCATION
srand()	3	Random # seed	Line 122
rand()	8	Generates rand #	Line 498

<ctime> library

TYPE	FREQUENCY	DESCRIPTION	LOCATION
time	1	Set current time	Line 122

<string> library

TYPE	DESCRIPTION	LOCATION
string	Declare Variable	Line 45

<iomanip> library

TYPE	DESCRIPTION	LOCATION
set	Format spaces on game board	Line 531

<fstream> library

TYPE	DESCRIPTION	LOCATION
file.is_open()	Open file	Line 762
file.close()	Close file	Line 767

Data Types

TYPE	LOCATION
int	Line 762
unsigned int	Line 767
char	Line 48
string	Line 45
float	Line 46

Conditional Statements

TYPE	LOCATION
if	Line 390
if/else/if	Line 554, 557, 561
switch	Line 574
if/else	Line 528, 530

Loops

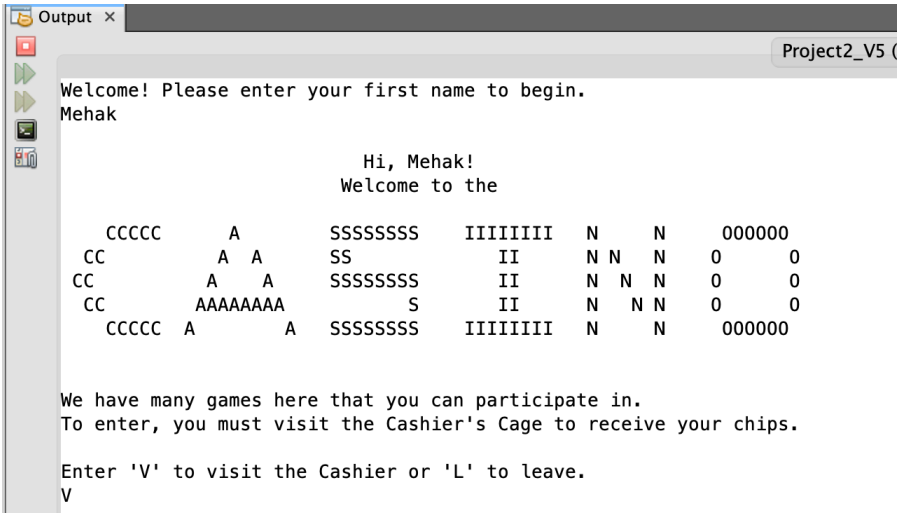
TYPE	LOCATION
do-while	Line 248, 260
for	Line 541
while	Line 343

REFERENCES

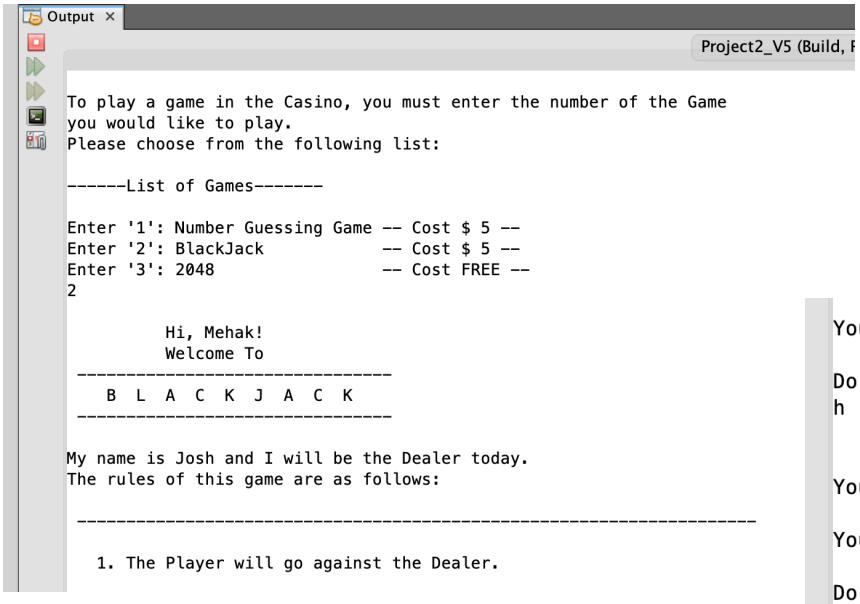
1. Dr. Lehr's Lectures/GitHub
2. "Starting Out with C++: From Control Structures through Objects" Gaddis, Tony 9th Edition (Textbook)

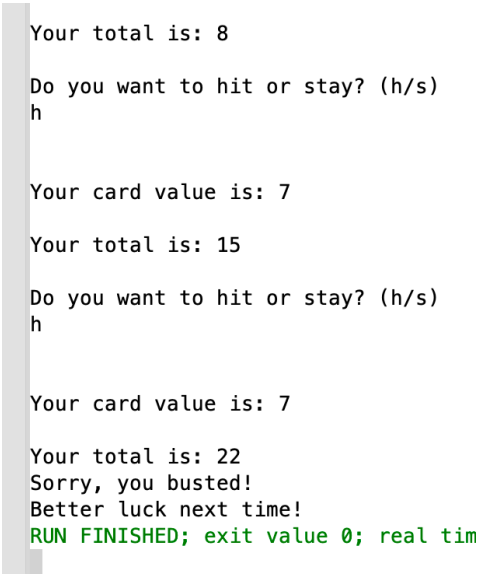
PROOF OF A WORKING PRODUCT

Welcome Screen

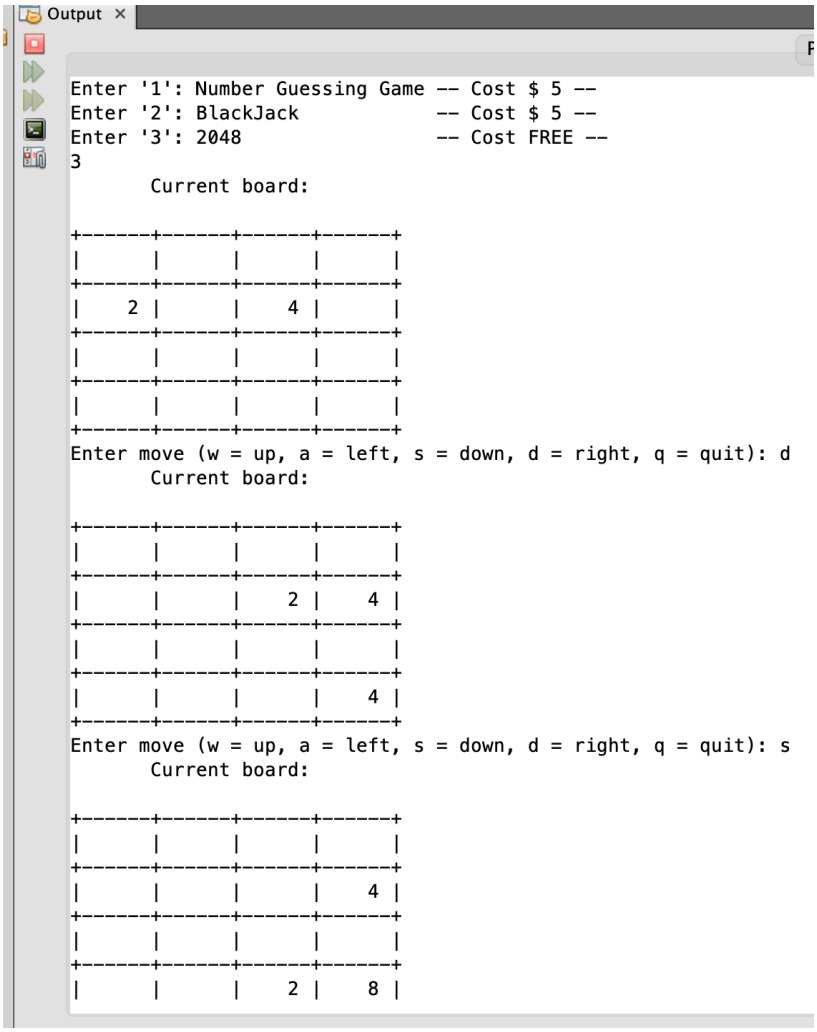


Game Selection(Utilizing Switch Case)





2048
2D Dimensional Array



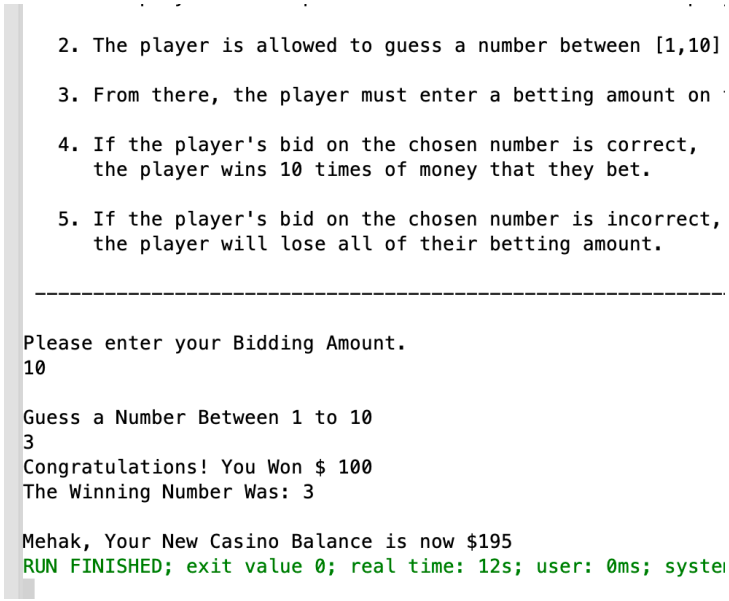
```
Output ×
Enter '1': Number Guessing Game -- Cost $ 5 --
Enter '2': BlackJack            -- Cost $ 5 --
Enter '3': 2048                 -- Cost FREE --
3
       Current board:

+------+------+------+------+
|      |      |      |      |
+------+------+------+------+
|    2 |      |    4 |      |
+------+------+------+------+
|      |      |      |      |
+------+------+------+------+
|      |      |      |      |
+------+------+------+------+
Enter move (w = up, a = left, s = down, d = right, q = quit): d
       Current board:

+------+------+------+------+
|      |      |      |      |
+------+------+------+------+
|      |      |    2 |    4 |
+------+------+------+------+
|      |      |      |      |
+------+------+------+------+
|      |      |      |    4 |
+------+------+------+------+
Enter move (w = up, a = left, s = down, d = right, q = quit): s
       Current board:

+------+------+------+------+
|      |      |      |      |
+------+------+------+------+
|      |      |      |    4 |
+------+------+------+------+
|      |      |      |      |
+------+------+------+------+
|      |      |    2 |    8 |
```

Number Guessing Game



```
  2. The player is allowed to guess a number between [1,10]

  3. From there, the player must enter a betting amount on

  4. If the player's bid on the chosen number is correct,
     the player wins 10 times of money that they bet.

  5. If the player's bid on the chosen number is incorrect,
     the player will lose all of their betting amount.

 ------------------------------------------------------------

Please enter your Bidding Amount.
10

Guess a Number Between 1 to 10
3
Congratulations! You Won $ 100
The Winning Number Was: 3

Mehak, Your New Casino Balance is now $195
RUN FINISHED; exit value 0; real time: 12s; user: 0ms; syste
```

PROGRAM

```
/*
 * File:   main.cpp
 * Author: Mehak Lohchan
 * Created on February 10th, 2023, 2:01 PM
 * Purpose: Project 2: Casino Game
 *           VERSION 5
 */

// System Libraries
#include <iostream>  // Input Output Library
#include <iomanip>   //Format Library
#include <cstring>   //String Object
#include <cstdlib>   //Random number generator
#include <ctime>
#include <fstream>
using namespace std;

//User Libraries

//Global Constants not Variables
//Science, Math, Conversions, Dimensions

//Function Prototypes
//       2048
int board[4][4];        //4x4 GameBoard
void print_board(int board[4][4]);
bool game_over();
void shift_left();
void add_duplicate_left();
void shift_right();
void add_duplicate_right();
void shift_up();
void add_duplicate_up();
void shift_down();
void add_duplicate_down();
void display_board();
void bblSort(vector<int> arr[], int);
int binSrch(int arr[], int, int, int);
string readFile(string);
```

```
//Execution begins here at main
int main(int argc, char** argv) {

    //Declare Variables
    string UsrName;            //Allow User to enter their
name
    float UsrDpst;             //The min deposit User has
to make to enter Casino
    char VorL, YorN;           //Visit or Leave   /   Yes
or No
    char chose;
    int UsrBlnc;               //User Balance Through Each
Game

    //Initialize Variables
    //Ask User to input their Player (first) Name
    cout<<"Welcome! Please enter your first name to begin.
\n";
    cin>>UsrName;

    //Introduce Player to Casino and Rules
    cout<<"\n                          Hi, "<<UsrName<<
           "! \n                        Welcome to
the\n";

    cout<<"\n    CCCCC      A        SSSSSSSS    IIIIIIII
N     N     OOOOOO   ";
    cout<<"\n  CC          A  A      SS             II
N N   N    O      O  ";
    cout<<"\n CC          A    A     SSSSSSSS       II
N  N  N    O      O  ";
    cout<<"\n  CC        AAAAAAAA           S       II
N   N N    O      O  ";
    cout<<"\n    CCCCC  A        A   SSSSSSSS    IIIIIIII
N     N     OOOOOO   \n";

    cout<<"\n\nWe have many games here that you can
participate in.";

    //Allow User to make a deposit
    cout<<"\nTo enter, you must visit the Cashier's Cage to
receive your chips.\n"<<
```

```
            "\nEnter 'V' to visit the Cashier or 'L' to
leave.\n";
    cin>>VorL;

    //If User wants to leave the Casino, Display goodbye
message and exit
    if (VorL == 'l' || VorL == 'L'){
        cout<<"Thank you for visiting the Casino. We hope
to see you soon!";
        exit(0);
    }

    //Allow User to Pick which Game they want to play
    cout<<"\nTo play a game in the Casino, you must enter a
minimum "
            "entry fee of $15.\n";
    cout<<"\nHow much money would you like to cash in?\n";
    cin>>UsrDpst;

    //While User Deposit is < $15, allow User to re-enter
amount until min/higher than 15
    if (UsrDpst < 15){

        do {
            cout<<"\nYou must enter a minimum amount of
$15."<<endl;
            cout<<"\nPlease enter a new amount.\n";
            cin>>UsrDpst;
            }
        while (UsrDpst < 15);
    }

    //User must enter a minimum entry fee amount of $15
    if (UsrDpst >= 15){

        //If the User's Deposit is >= $15, they may enter
the Casino
        cout<<"\nGreat! You may now enter the Casino.\n\n";

        cout<<"\nTo play a game in the Casino, you must
enter the "
                "number of the Game \nyou would like to
play."<<endl;
```

```
        cout<<"Please choose from the following
list:"<<endl;
        cout<<"\n------List of Games-------\n";
        cout<<"\nEnter '1': Number Guessing Game -- Cost $
5 -- "<<endl;
        cout<<"Enter '2': BlackJack            -- Cost $ 5
-- "<<endl;
        cout<<"Enter '3': 2048                 -- Cost FREE
-- "<<endl;
        cin>>chose;


    // -- GAME 1 -- //
    switch (chose){
        case '1': {

            /* GAME 1: Number Guessing Game - Cost $5
             Allow User to guess a random number and if
this random number matches the
             winning number, then User wins x money.
             */

            //Set Random Number Seed
            srand(static_cast<unsigned int>(time(0)));

            unsigned int UsrBlnc;     //User Balance
Through Each Game


            //Declare Variables
            unsigned int amount, bidamnt,    //Amount, Bid
Amount
                         rguess,             //Random Guess
(Utilizing random num seed)
                         dice,
                         newBal,
                           x;               //Var used to
calculate equations

            char uexit,                       //User Exits
                 enter;                       //Allow user
to Enter to Continue Game
```

```
            //Initialize Variables
            cout<<"\nTo play this game, you will be charged
$ 5.\n";
            cout<<"\nWould you like to continue?";
            cout<<"\nEnter 'Y' for Yes, 'N' for No.\n";
            cin>>YorN;


            if (YorN == 'y' || YorN == 'Y'){
                //Note: User must have a Casino Balance of
at least $15
                if (UsrDpst >= 15){

                    //In this case, get the User Deposit

                    //Subtract cost of game from User's
Casino Balance, then output
                    x = UsrDpst - 5;

cout<<"\n----------------------------------------";
                    cout<<"\n    Your new Casino Balance is
$ "<<x;

cout<<"\n----------------------------------------\n";
                    }
            }

            //Map/Process the inputs -> Outputs

            //If User enters N, provide an option to exit
            if (YorN == 'n' || YorN == 'N' ){
                cout<<"Would you like to exit?";
                cout<<"\nEnter 'Y' for Yes, 'N' for No.\n";
                cin>>YorN;

                if (uexit == 'y' || uexit == 'Y'){
                        exit(0);

                }
            }

            if (UsrBlnc > 0){
```

```
                //Checks to see if the User wants to Visit
the Casino
                if (VorL == 'v' || VorL == 'V' || UsrDpst
>= 15){
                    cout<<"\n\nHi, "<<UsrName<<"! Welcome
to the "
                            "Number Guessing Game!";

                    cout<<"\nMy name is John.\n";
                }

                //Checks to see if the User wants to leave
the Casino
                    else if (VorL == 'l' || VorL == 'L'){
                        cout<<"\nWe're sorry to see you go.
\n"
                                "Please come again!\n";
                }


            //User must have $5 or more in their Balance to
play this game
            if (UsrDpst > 5){
                //Display rules of Game 1
                cout<<"\n
------------------------------------------------------------
-----------------\n";
                cout<<"  The rules of this game are as
follows:\n"
                        "\n   1. The player must deposit an
initial amount to start playing the game.\n"
                        "\n   2. The player is allowed to
guess a number between [1,10].\n"
                        "\n   3. From there, the player
must enter a betting amount on the number chosen.\n"
                        "\n   4. If the player's bid on the
chosen number is correct,\n"
                            "      the player wins 10
times of money that they bet.\n"
                        "\n   5. If the player's bid on the
chosen number is incorrect,\n"
                            "      the player will lose
all of their betting amount.\n";
```

```
                cout<<"\n
------------------------------------------------------------
-----------------\n";


                //Allow User to enter an Initial and Bid
Amount
                do {

                    //If User Balance is < 5, display
message of minimum entry fee
                    if (UsrDpst < 5){
                        cout<<"\nSorry, you must enter a
minimum of $5 to be "
                                "eligible to play this game
\n"
                                "or have an Account Balance
of $5.\n";

                        //Allow User to try again
                        cout<<"\nPlease enter the minimum
entry fee.\n";
                        cin>>UsrDpst;

                    if (UsrDpst == 0 || UsrDpst < 5){
                        cout<<"\nSorry, you are not
eligible to play this game.\n";
                    }
                }

                }
                while (UsrDpst < 5);


            //Map/Process the inputs -> Outputs

                /* If User enters a bid amount > than
amount deposited
                   Have them re-enter amount until valid*/
                do {
                    cout<<"\nPlease enter your Bidding
Amount.\n";
                    cin>>bidamnt;
```

```
                    cout<<"\n";

                    /* If User's bidding amount is greater
than what they deposit
                     Display error message*/

                    if (bidamnt > UsrDpst){
                        cout<<"\nSorry, you are not allowed
to bid more than "
                                "your Casino Balance.\n"
                                "\nPlease re-enter amount.
\n";
                        cin>>bidamnt;
                    }
                }
                while (bidamnt > UsrDpst);

               //Allow the User to Pick a number [1, 10]
                do {
                    cout<<"Guess a Number Between 1 to
10\n";
                    cin>>rguess;

                    /* If the User's Random Guess if <= 0
or > 10,
                       then display an error message*/
                    if (rguess <= 0 || rguess > 10){
                        cout<<"Oops! The Number should be
Between 1, 10\n"
                                "Please Try Again.\n";
                    }
                }
                //User's Guess has to be <= 0 but less than
10 for it to be valid
                while (rguess <= 0 || rguess > 10);


                //If the dice number IS equal to the User's
random guess..
                if (dice == rguess){
                    amount = bidamnt * 10;
                    UsrBlnc = x + amount;
```

```
                    cout<<"Congratulations! You Won "<<"$
"<<amount;
                }

                //If the dice number IS NOT equal to User's
random guess..
                else if (dice != rguess) {
                    UsrBlnc = x - bidamnt;
                    cout<<"\nSorry, You Lost "<<"$
"<<bidamnt<<
                            "\nBetter Luck Next Time!\n";
                }

                //Display the winning number from dice
                cout<<"\nThe Winning Number Was: "<<"3
"<<"\n"; //test

                //Calculate new User Balance by adding the
newBal + UserBlnce

                //Output User's New Casino Balance
                cout<<"\n"<<UsrName<<", Your New Casino
Balance is now $"
                        <<UsrBlnc;

                    }
                }
            }

            break;
        }
    }
        switch (chose){
            case '2': {

            /* GAME 2: BlackJack - Cost $5
            Allow User to guess a random number and if this
random number matches the
            winning number, then User wins x money.
            */

                //User Introduction
```

```
                cout<<"\n          Hi, "<<UsrName<<"! \n
Welcome To \n";
                    cout<<"
--------------------------------";
                    cout<<"             \n    B  L  A  C  K
J  A  C  K\n";
                    cout<<"
--------------------------------";

                    //Introduce Dealer and Rules
                    cout<<"\n\nMy name is Josh and I will"
                           " be the Dealer today.\n";
                    cout<<"The rules of this game are as
follows:\n";

                //Display rules of Game 2
                cout<<"\n
------------------------------------------------------------
----------\n"

                        "\n   1. The Player will go against
the Dealer.\n"
                        "\n   2. You will be giving a
randomized card, as well as the Dealer.\n"
                        "\n   3. From there, you will have
the option to Hit or Stay.\n"
                        "\n   4. If you choose Hit, you
will receive another card,\n"
                               "      or if you choose Stay,
you will end your turn.\n"
                        "\n   5. The goal of this game is
to have a deck of cards that,\n"
                               "      add up to 21.\n"
                        "\n   6. Whoever reaches 21 first,
wins. If you go above 21,\n "
                               "      the Player/Dealer
Busts.\n";

                cout<<"\n
------------------------------------------------------------
----------\n";

                //Set Random Number Seed
```

```
                srand(static_cast<unsigned int>(time(0)));

                //Declare Variables
                int plyrsum = 0, dealrsm = 0,       //
Player Sum, Dealer Sum
                        card;

                bool pturn = true, dturn = false,
                        pbust = false, dbust = false;

                char HorS, again;                   /*Hit
or Stay,
                                                    If
Player wants to play Again*/

                for (again = true; again == true;) {
                    //Player's Turn
                    while (pturn) {
                        //Randomize card number, then add
that to the Player's Sum
                        card = rand() % 11 + 1;
                        plyrsum += card;

                        cout<<"\n\nYour card value is:
"<<card<<endl;
                        cout<<"\nYour total is:
"<<plyrsum<<endl;

                        //If the Player's Sum is > 21, The
Player Busts
                        if (plyrsum > 21) {
                            pbust = true;
                            cout<<"Sorry, you busted!
\nBetter luck next time!";
                            exit(0);
                        }

                        //Ask Player if they want to Hit or
Stay
                        cout<<"\nDo you want to hit or
stay? (h/s)\n";;
                        cin>>HorS;
                        if (HorS == 's') {
```

```
                        pturn = false;
                        dturn = true;
                    }
                }

                //Dealer's Turn
                while (dturn) {
                    //Randomize card number, then add
that to the Dealer's Sum
                    card = rand() % 11 + 1;
                    dealrsm += card;

                    cout<<"\nThe Dealer's card value
is: \n"<<card<<endl;
                    cout<<"\nThe Dealer's total is:
\n"<<dealrsm<<endl;

                    //If the Dealer's Sum is > 21,
Dealer Busts, then exit game
                    if (dealrsm > 21) {
                        dbust = true;
                        cout<<"\nDealer busted!
\n"<<endl;
                        exit(0);
                    }
                    //If the Dealer Sum is >= 17, then
the Dealer takes another turn
                    if (dealrsm >= 17) {
                        dturn = false;
                    }
                }

                //Determine the Winner
                //If the Player Busts
                if (pbust) {
                    cout<<"\nThe Dealer wins!"<<endl;
                }

                //If the Dealer Busts, Player Wins
                else if (dbust) {
                    cout<<"-------------------------";
                    cout<<"     CONGRATULATIONS
";
```

```
                        cout<<"-------------------------";
                        cout<<UsrName<<"You win!"<<endl;
                    }

                    else {
                        if (plyrsum > dealrsm) {
                            cout<<"You win!"<<endl;
                        }
                        else if (dealrsm > plyrsum) {
                            cout<<"Dealer wins!"<<endl;
                        }
                        else {
                            cout<<"The Dealer and the
Player have\n"
                                    "reached a Tie."<<endl;
                        }
                    }

                    //If the Player wants to play again
                    cout<<"Would you like to play again?
('Y'/'N)\n";
                    cin>>again;

                    if (again == 'n' || again == 'N') {
                        again = false;
                    }
                    else {
                        //Reset the Game for Attempt 'n'
                        plyrsum = 0, dealrsm = 0;
                        pturn = true, dturn = false;
                        pbust = false, dbust = false;
                    }
                }
            }
            break;
        }


        switch(chose){
            case '3':
                //Set random number seed
                srand(static_cast<unsigned int>(time(0)));
```

```
                //Declare Variables and Functions
                int count = 0;
                char move;

                /*
                2048 main function

                w - shift up
                a - shift left
                s - shift down
                d - shift right
                q - ends the game
                :4x4 2D list of integers representing game
board
                */

                //Map/Process the inputs -> Outputs
                while (count < 2) {
                int i = rand() % 4;
                int j = rand() % 4;
                if (board[i][j] == 0) {
                    board[i][j] = (rand() % 2 + 1) * 2;
                    count++;
                }
            }
                //Main game loop
                while (!game_over()) {
                    // Display the game board
                    cout<<"       Current board:\n"<<endl;
                    print_board(board);

                    //Get player move
                    cout<<"Enter move (w = up, a = left, s
= down, d = right, q = quit): ";
                    cin>>move;

                    //Perform move
                    switch (move) {
                        case 'w':
                            shift_up();
                            add_duplicate_up();
                            break;
                        case 'a':
```

```
                        shift_left();
                        add_duplicate_left();
                        break;
                    case 's':
                        shift_down();
                        add_duplicate_down();
                        break;
                    case 'd':
                        shift_right();
                        add_duplicate_right();
                        break;
                    case 'q':
                        if (move == 'q'){
                            cout<<"\nThank you for
playing!";
                            exit(0);
                        }
                }
                //Add a new tile to the board
                int i = rand() % 4;
                int j = rand() % 4;
                while (board[i][j] != 0) {
                    i = rand() % 4;
                    j = rand() % 4;
                }
                board[i][j] = (rand() % 2 + 1) * 2;

                //Check if the player has reached 2048
                for (int i = 0; i < 4; i++){
                    for (int j = 0; j < 4; j++){
                        if (board[i][j] == 2048){
                            cout<<"Congratulations, you
win!"<<endl;
                        }
                    }
                }
            }
                //Game over
                cout<<"Game over!"<<endl;
                break;
        }

    }
```

```
//Function Calls
void print_board(int board[4][4]){
    for (int i = 0; i < 4; i++) {
        cout<<"+------+------+------+------+"<<endl;
        for (int j = 0; j < 4; j++) {
            cout << "| ";
            if (board[i][j] == 0) {
                cout<<"    ";
            } else {
                cout<<setw(4)<<board[i][j];
            }
            cout<<" ";
        }
        cout<<"|"<<endl;
    }
    cout<<"+------+------+------+------+"<<endl;
}

void display_board(){
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
            cout<<board[i][j]<<"\t";
        }
        cout<<endl;
    }
}

void shift_left(){
    for (int i = 0; i < 4; i++) {
    int k = 0;
    for (int j = 1; j < 4; j++) {
        if (board[i][j] != 0) {
            if (board[i][k] == 0) {
                board[i][k] = board[i][j];
                board[i][j] = 0;
            } else if (board[i][k] == board[i][j]) {
                board[i][k] *= 2;
                board[i][j] = 0;
                k++;
            } else {
                k++;
                board[i][k] = board[i][j];
```

```
                if (j != k) {
                    board[i][j] = 0;
                }
                }
            }
        }
    }
}

void add_duplicate_left(){
    for (int i = 0; i < 4; i++) {
        for (int j = 1; j < 4; j++) {
            if (board[i][j] != 0 && board[i][j] == board[i]
[j-1]) {
                board[i][j-1] *= 2;
                board[i][j] = 0;
            }
        }
    }
}

void shift_right() {
    for (int i = 0; i < 4; i++) {
    int k = 4 - 1;
    for (int j = 4 - 2; j >= 0; j--) {
        if (board[i][j] != 0) {
            if (board[i][k] == 0) {
                board[i][k] = board[i][j];
                board[i][j] = 0;
            } else if (board[i][k] == board[i][j]) {
                board[i][k] *= 2;
                board[i][j] = 0;
                k--;
            } else {
                k--;
                board[i][k] = board[i][j];
                if (j != k) {
                    board[i][j] = 0;
                }
                }
            }
        }
    }
```

```
}

void add_duplicate_right(){
    for (int i = 0; i < 4; i++) {
        for (int j = 4 - 2; j >= 0; j--) {
            if (board[i][j] != 0 && board[i][j] == board[i]
[j+1]) {
                board[i][j+1] *= 2;
                board[i][j] = 0;
            }
        }
    }
}

void shift_up(){
    for (int j = 0; j < 4; j++) {
    int k = 0;
    for (int i = 1; i < 4; i++) {
        if (board[i][j] != 0) {
            if (board[k][j] == 0) {
                board[k][j] = board[i][j];
                board[i][j] = 0;
            } else if (board[k][j] == board[i][j]) {
                board[k][j] *= 2;
                board[i][j] = 0;
                k++;
            } else {
                k++;
                board[k][j] = board[i][j];
                if (i != k) {
                    board[i][j] = 0;
                }
                }
            }
        }
    }
}

void add_duplicate_up(){
    for (int j = 0; j < 4; j++) {
        for (int i = 1; i < 4; i++) {
            if (board[i][j] != 0 && board[i][j] ==
board[i-1][j]) {
```

```
                board[i-1][j] *= 2;
                board[i][j] = 0;
            }
        }
    }
}

void shift_down(){
    for (int j = 0; j < 4; j++) {
    int k = 4 - 1;
    for (int i = 4 - 2; i >= 0; i--) {
        if (board[i][j] != 0) {
            if (board[k][j] == 0) {
                board[k][j] = board[i][j];
                board[i][j] = 0;
            } else if (board[k][j] == board[i][j]) {
                board[k][j] *= 2;
                board[i][j] = 0;
                k--;
            } else {
                k--;
                board[k][j] = board[i][j];
                if (i != k) {
                    board[i][j] = 0;
                }
                }
            }
        }
    }
}

void add_duplicate_down(){
    for (int j = 0; j < 4; j++) {
        for (int i = 4 - 2; i >= 0; i--) {
            if (board[i][j] != 0 && board[i][j] ==
board[i+1][j]) {
                board[i+1][j] *= 2;
                board[i][j] = 0;
            }
        }
    }
}
```

```
bool game_over(){
    // Check if any move is possible
    bool moves_possible = false;
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
            if (board[i][j] == 0) {
                // An empty cell is available
                moves_possible = true;
                break;
            }
            // Try moving the tile to the left
            if (j > 0 && board[i][j-1] == board[i][j]) {
                moves_possible = true;
                break;
            }
            // Try moving the tile to the right
            if (j < 4-1 && board[i][j+1] == board[i][j]) {
                moves_possible = true;
                break;
            }
            // Try moving the tile up
            if (i > 0 && board[i-1][j] == board[i][j]) {
                moves_possible = true;
                break;
            }
            // Try moving the tile down
            if (i < 4-1 && board[i+1][j] == board[i][j]) {
                moves_possible = true;
                break;
            }
        }
        if (moves_possible) {
            break;
        }
    }
    if (!moves_possible) {
        // No moves are possible
        print_board(board);
        cout << "Game over!\n";
    }
}

void bblSort(int arr[], int n) {
```

```
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

int binSrch(int arr[], int low, int high, int target) {
    while (low <= high) {
        int mid = (low + high) / 2;
        if (arr[mid] == target) {
            return mid;
        }
        else if (arr[mid] < target) {
            low = mid + 1;
        }
        else {
            high = mid - 1;
        }
    }
    return -1;
}

string readFile(string filename){
    string result = "";
    ifstream file(filename);
    if (file.is_open()) {
        string line;
        while (getline(file, line)) {
            result += line + "\n";
        }
        file.close();
    }
    return result;
}
```