Project 1

<Blackjack Game>

**CIS-5-42480**

**Name: Jesus Contreras**

**Date: 4/22/2018**

**Introduction:**

Title: Blackjack Game

Blackjack is a card game with an objective of reaching a total score of 21. Each card has a value set for it as such:

Ace: 1 or 11

Cards numbered 2 - 10

Jack: 10

Queen: 10

King: 10

You have to bet a certain amount of money to buy in. Once the pot is set the dealer then deals you, other players, and themselves a hand. The main objective is to reach that score of 21 without going over and busting or else you lose that hand. If you are under the score amount you may ask the dealer to “hit” you to give you another card until you reach or get as close as possible to the score of 21 as possible. Whoever hits 21 or is closest to the score without going over wins the pot. Sometimes there will be multiple winners and then they will all split the winnings.

**Summary:**

Project size: 169 lines

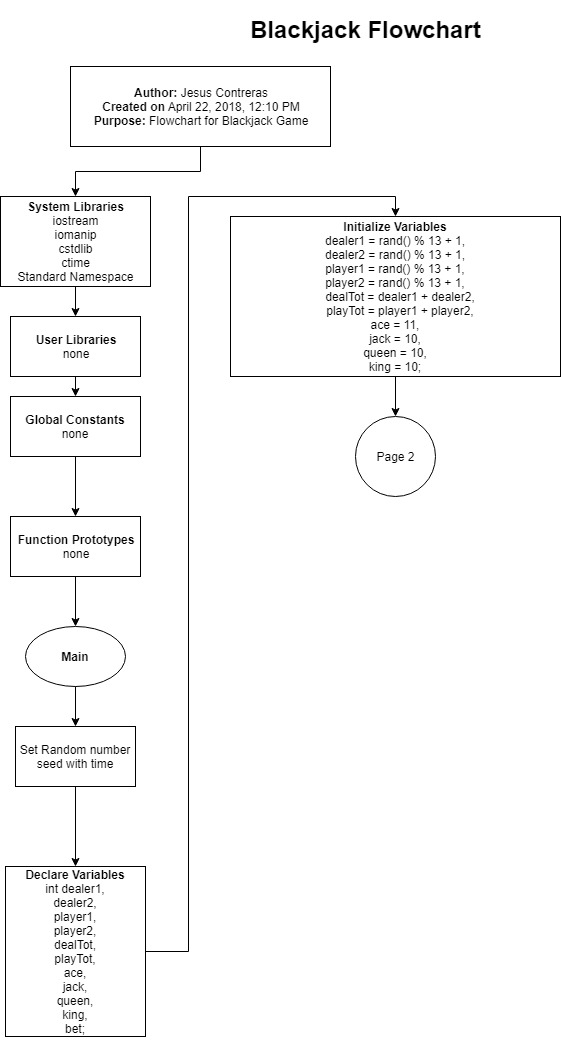
Variables: 11

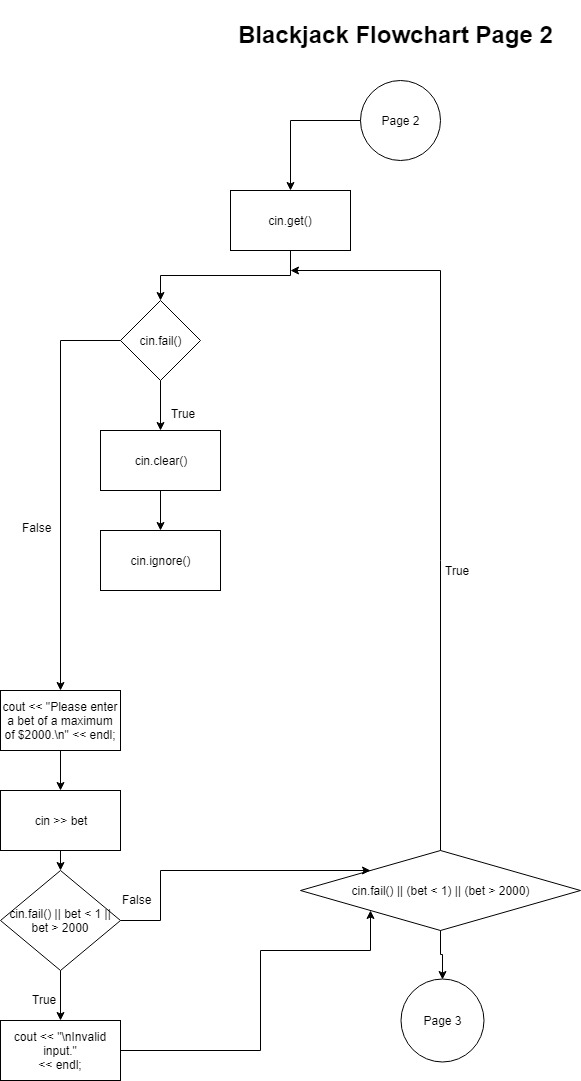
In this project I used some things that we learned so far. I made a loop to get the correct amount of money inputted by the user for a bet. The maximum bet I set was up to $2000. The range is between $1 to $2000 and if the user inputs something invalid it will loop until they put an amount that is allowed. After that I decided to use switch statements for the dealers cards and the players cards. With the use of srand and the seed that I used I was able to randomize the cards that the dealer and the players will be able to get. I was not able to make the user ask for a hit or stand option. Since it is a basic project without the use of arrays and other things I decided to go with something simple. Whatever cards you get and whoever has the most will be decided as the winner. So far from what I tested and ran it all checked out good. I did have a bit of trouble as this is my first project but I feel as though I may be progressing rather well. I will continue to practice and eventually by the second project I will be able to do much better.

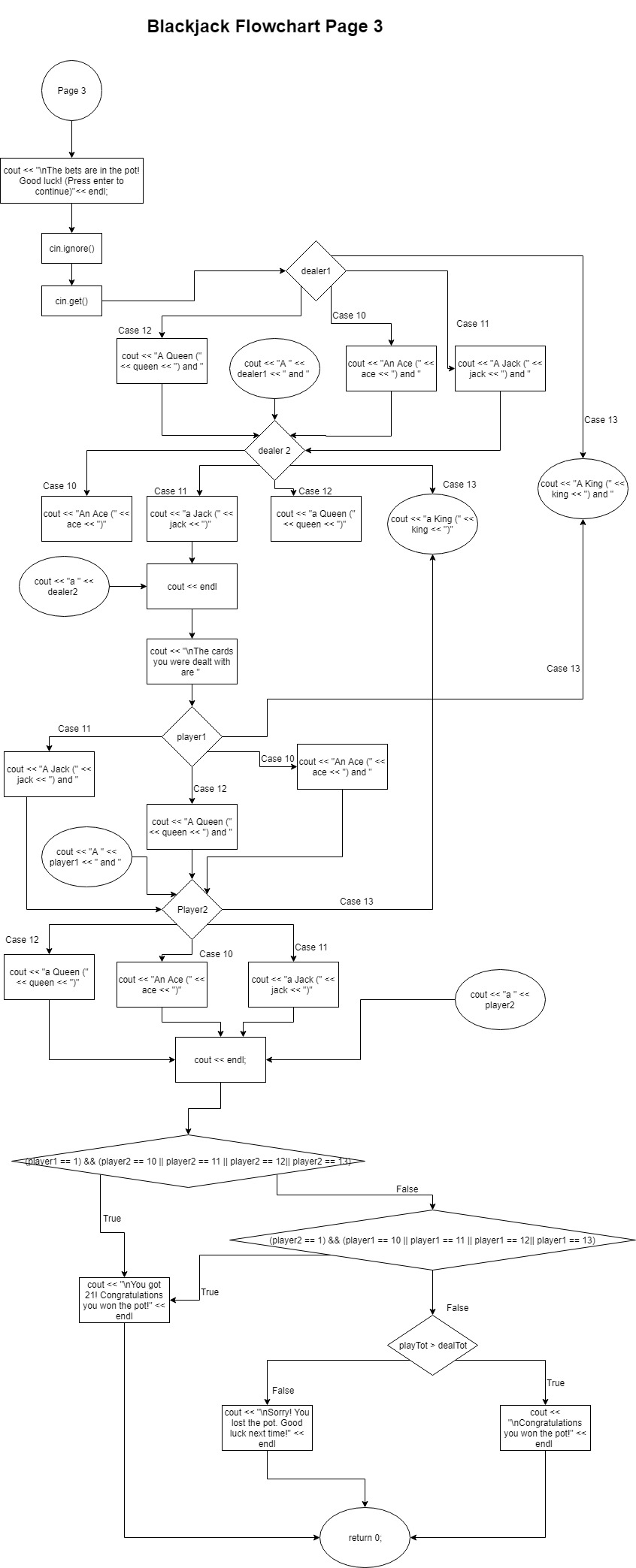
**Description:**

The main point for me in this project was to make a good enough loop and to randomize each card and get a set value from each. I did have trouble with completing the switch statements as that is what I took the longest with. I could not get the correct outcomes with the hands, but eventually I was able to fix it.

**Flowchart:**







**Pseudo Code**

*Inititalize*

*Get user input to begin game*

*Ask user to input bet amount*

*If invalid input loop until correct amount inputted*

*While correct amount is inputted Start dealing cards*

*Dealer’s card are outputted randomly on screen*

*Player’s card are outputted randomly on screen*

*If player gets 21 they instantly win the pot*

*Else if player gets a higher score they win the pot*

*Else the dealer wins the pot*

**\*\*NOTE\*\***

\*I don’t know if I did the pseudo code part correctly. Is this how it is?\*

**Major Variables:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Variable Name** | **Description** | **Location** |
| Integer | dealer1 | Dealer’s first card |  |
| Integer | dealer2 | Dealer’s second card |  |
| Integer | player1 | Player’s first card |  |
| Integer | player2 | Player’s second card |  |
| Integer | dealTot | Dealer’s total card count |  |
| Integer | playTot | Player’s total card count |  |
| Integer | ace | Ace card value is equal to 11 |  |
| Integer | jack | Jack card value is equal to 10 |  |
| Integer | queen | Queen card value is equal to 10 |  |
| Integer | king | King card value is equal to 10 |  |
| Integer | bet | Player’s input for wager amount |  |

**C++ Constructs:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Chapter | Section | Topic | Where Line #’s | Pts | Notes |
| 2 | 2 | cout | 54, 55, 56, 57, 68, 73, 79, 85, 86, 90 92, 94, 96, 98. etc. |  |  |
|  | 3 | libraries |  | 8 | Iostream, iomanip, cstdlib, ctime |
|  | 4 | variables/literals |  |  |  |
|  | 5 | identifiers |  |  |  |
|  | 6 | integers | 29 - 39 | 3 |  |
|  | 7 | characters |  | 3 |  |
|  | 8 | strings |  | 3 |  |
|  | 9 | Floats no doubles |  | 3 |  |
|  | 10 | bools |  | 4 |  |
|  | 11 | Sizeof \*\*\*\* |  |  |  |
|  | 12 | Variables 7 characters or less |  |  |  |
|  | 13 | Scope \*\*\*\* No global variables |  |  |  |
|  | 14 | Arithmetic operators |  |  |  |
|  | 15 | Comments 20%+ |  | 5 |  |
|  | 16 | Named Constants |  |  |  |
|  | 17 | Programming Style \*\*\* Emulate |  |  |  |
|  |  |  |  |  |  |
| 3 | 1 | cin |  |  |  |
|  | 2 | Math expression |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 4 | Mixing data types |  |  |  |
|  | 5 | Overflow/Underflow |  |  |  |
|  | 6 | Type casting |  | 4 |  |
|  | 7 | Multiple Assignment |  |  |  |
|  | 8 | Formatting Output |  | 4 |  |
|  | 9 | Math library |  | 4 |  |
|  | 10 | Hand tracing |  |  |  |
|  | 3 | Strings |  | 3 |  |
| 4 | 1 | Relational operators |  |  |  |
|  | 2 | if |  | 4 |  |
|  | 4 | if-else |  | 4 |  |
|  | 5 | Nesting |  | 4 |  |
|  | 6 | if-else-if |  | 4 |  |
|  | 7 | Flags |  |  |  |
|  | 8 | Logical operators |  | 4 |  |
|  | 11 | Validating user input |  | 4 |  |
|  | 13 | Conditional operator |  | 4 |  |
|  | 14 | Switch |  | 4 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | 1 | Increment/Decrement |  | 4 |  |
|  | 2 | While |  | 4 |  |
|  | 5 | Do-While |  | 4 |  |
|  | 6 | For Loop |  | 4 |  |
|  | 11 | Files input/output both |  | 8 |  |
|  | 12 | No breaks in loops |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| \*\*\*\*Not | required | To show | Total | 100 |  |

**Reference:**

1. Textbook - Starting Out with C++ from Control Structures to Objects 8th Edition By Tony Gaddis

**Program:**

/\*

\* File: main.cpp

\* Author: Jesus Contreras

\* Created on April 20th 2018, 6:10 PM

\* Purpose: Blackjack Game

\*/

// System Libraries

#include <iostream> // I/O Library -> cout, endl

#include <iomanip> // Format Library

#include <cstdlib> // Rand/Srand

#include <ctime> // Time

using namespace std; // namespace I/O stream library created

// User Libraries

// Global Constants

// Math, Physics, Science, Conversions, 2-D Array Columns

// Function Prototypes

// Blackjack Game

int main(int argc, char\*\* argv)

{

// Set the random number seed

srand(static\_cast<unsigned int>(time(0)));

// Declare Variables

int dealer1, // Dealer's first card

dealer2, // Dealer's second card

player1, // Player's first card

player2, // Player's second card

dealTot, // Dealer's total card count

playTot, // Player's total card count

ace, // Ace card is equal to 11

jack, // Jack card is equal to 10

queen, // Queen card is equal to 10

king, // King card is equal to 10

bet; // Player inputs wager amount

// Initialize Variables

dealer1 = rand() % 13 + 1,

dealer2 = rand() % 13 + 1,

player1 = rand() % 13 + 1,

player2 = rand() % 13 + 1,

dealTot = dealer1 + dealer2,

playTot = player1 + player2,

ace = 11,

jack = 10,

queen = 10,

king = 10;

cout << "|---------------------|" << endl;

cout << "|Welcome to Blackjack!|" << endl;

cout << "|---------------------|" << endl;

cout << "\n(Press enter to begin!)" << endl;

cin.get();

do

{

if(cin.fail())

{

cin.clear();

cin.ignore();

}

cout << "Please enter a bet of a maximum of $2000.\n" << endl;

cin >> bet;

if(cin.fail() || bet < 1 || bet > 2000)

cout << "\nInvalid input."

<< endl;

}

while(cin.fail() || (bet < 1) || (bet > 2000));

cout << "\nThe bets are in the pot! Good luck! (Press enter to continue)"

<< endl;

cin.ignore();

cin.get();

cout << "\nThe dealer deals the cards." << endl;

cout << "\nThe dealer has ";

switch (dealer1)

{

case 10: cout << "An Ace (" << ace << ") and ";

break;

case 11: cout << "A Jack (" << jack << ") and ";

break;

case 12: cout << "A Queen (" << queen << ") and ";

break;

case 13: cout << "A King (" << king << ") and ";

break;

default: cout << "A " << dealer1 << " and ";

break;

}

switch (dealer2)

{

case 10: cout << "An Ace (" << ace << ")";

break;

case 11: cout << "a Jack (" << jack << ")";

break;

case 12: cout << "a Queen (" << queen << ")";

break;

case 13: cout << "a King (" << king << ")";

break;

default: cout << "a " << dealer2;

break;

}

cout << endl;

cout << "\nThe cards you were dealt with are ";

switch (player1)

{

case 10: cout << "An Ace (" << ace << ") and ";

break;

case 11: cout << "A Jack (" << jack << ") and ";

break;

case 12: cout << "A Queen (" << queen << ") and ";

break;

case 13: cout << "A King (" << king << ") and ";

break;

default: cout << "A " << player1 << " and ";

break;

}

switch (player2)

{

case 10: cout << "An Ace (" << ace << ")";

break;

case 11: cout << "a Jack (" << jack << ")";

break;

case 12: cout << "a Queen (" << queen << ")";

break;

case 13: cout << "a King (" << king << ")";

break;

default: cout << "a " << player2;

break;

}

cout << endl;

if ((player1 == 1) && (player2 == 10 || player2 == 11 || player2 == 12

|| player2 == 13))

{

cout << "\nYou got 21! Congratulations you won the pot!" << endl;

}

else if ((player2 == 1) && (player1 == 10 || player1 == 11 || player1 == 12

|| player1 == 13))

{

cout << "\nYou got 21! Congratulations you won the pot!" << endl;

}

else if (playTot > dealTot)

{

cout << "\nCongratulations you won the pot!" << endl;

}

else

{

cout << "\nSorry! You lost the pot. Good luck next time!" << endl;

}

return 0;

}