Project 1

Blackjack Program

CSC-5

Section 46023

Summer 2014

Brynn Caddel

Introduction: Blackjack

The program presented here begins by prompting the user to select between playing Blackjack, reading the rules of Blackjack, or exiting the program altogether. The first option immediately allows the user to begin playing the game. The second option goes over a brief description of the game and the ways in which it is possible to win, lose, and draw, as well as the value of the cards. The third options exits the programs and discontinues the game.
Blackjack is a comparing card game between a player and dealer.The object of the game is to beat the dealer, which can be done in a number of ways:

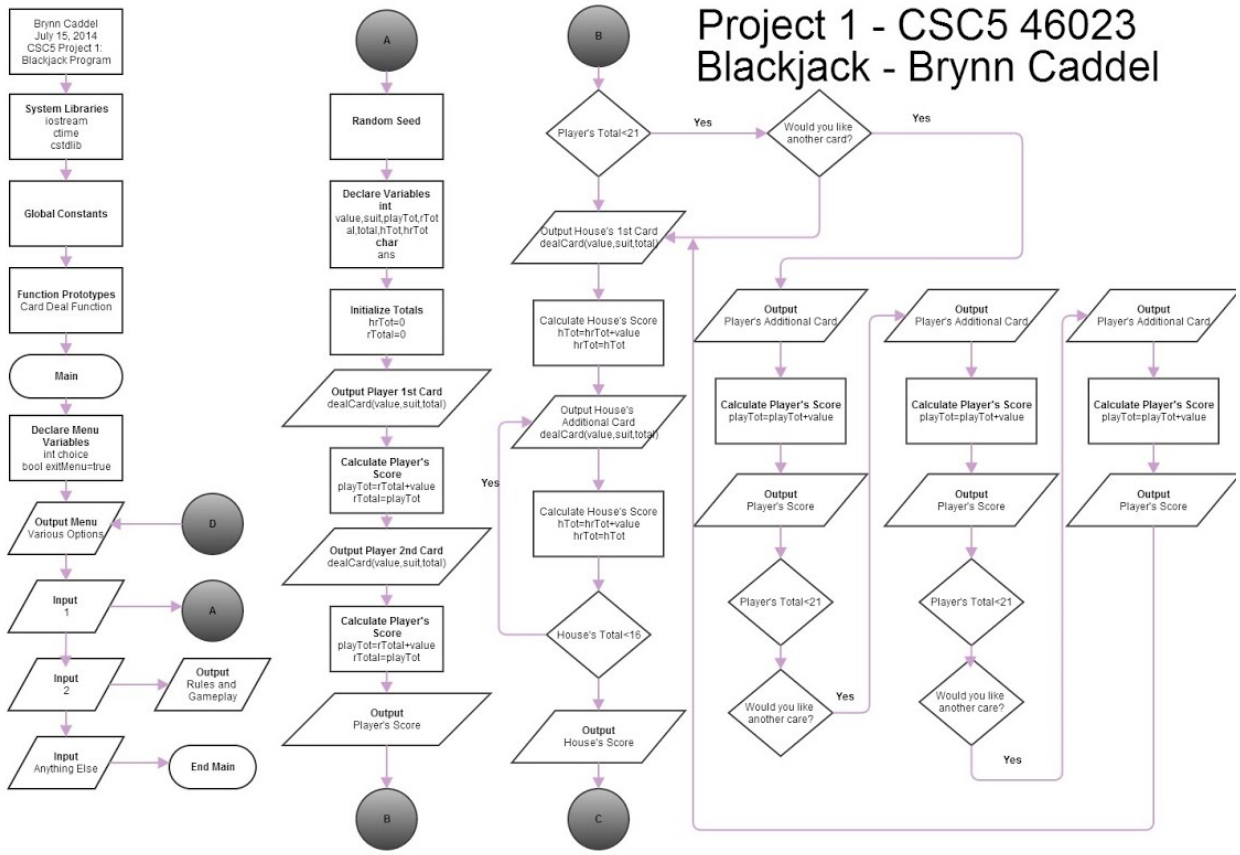
1) Get 21 points on the player's first two cards. This is called a blackjack.
2) Reach a final score higher than the dealer without exceeding 21.
3) Let the dealer draw additional cards until his or her hand exceeds 21.

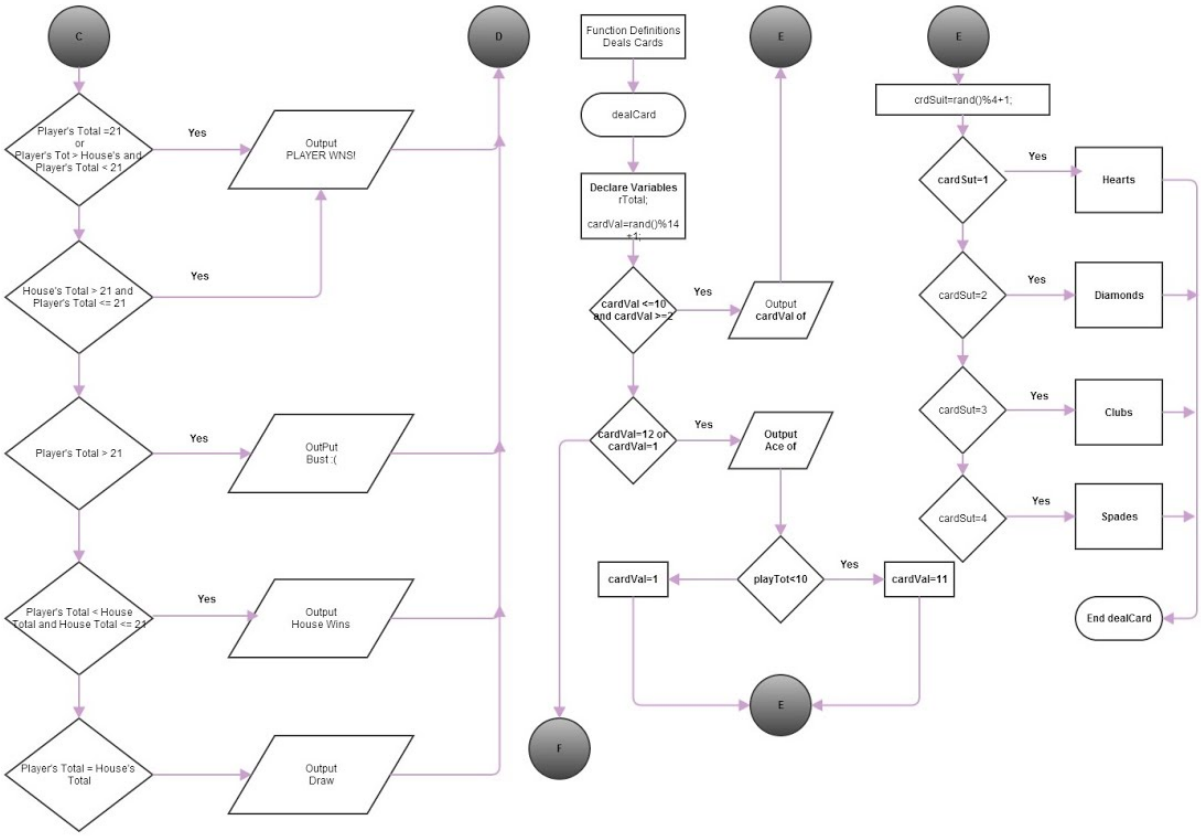
The player or players are dealt an initial two-card hand and add together the value of their cards. Face cards (kings, queens, and jacks) are counted as ten points. A player and the dealer can count his or her own ace as 1 point or 11 points, depending on the player's total. All other cards are counted as the numeric value shown on the card. After receiving their initial two cards, players have the option of getting a “hit,” or taking an additional card. Scoring higher than 21 (called “busting”) results in a loss. A player may win by having any final score equal to or less than 21 if the dealer busts.

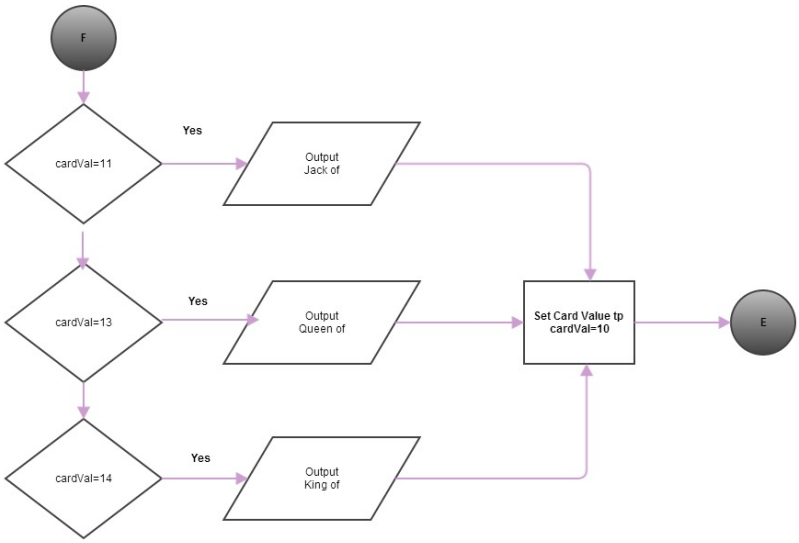
Development Summary

Before I decided to write this program, I had actually already written up a turn-based Pokemon battle that I planned on submitting as my project. However, I didn’t think that the program showcased all of the ideas that we had learned up to this point in the course, so I decided to try and write this Blackjack program instead. Admittedly, I had never played Blackjack before, but my Grandma happens to manage several casinos in Las Vegas, so even though I could have easily Googled how to play Blackjack, he was able to help me understand the basic jist of the game. While it’s easy to understand, the logic behind programming it took a while. Before we had covered functions during lecture, I found it difficult to implement the carding dealing without calling on functions. Even after lecture, I continued to read my *Problem Solving with C++ 8th Edition by Walter Savitch* textbook and read various forums on sites like (http://cpluspluscom) for more clarification and insight into ways to utilize functions. When I first began writing the code, I tried to do so by directly typing it into NetBeans. A couple days after I began, I realized that it was a jumbled mess and instead took a different approach that included flowcharting it out by hand first. I thought flowcharts were a little unnecessary before this project, and now I definitely see how useful they can be. It still took many revisions to get the program to run the way I wanted it to, but I’m finally satisfied with the program.

Project 1 - CSC5 46023
Blackjack - Brynn Caddel






F
cardVal=11
Yes
Output
Jack of
cardVal=13
Yes
Output
Queen of
Set Card Value tp
cardVal=10
E
cardVal=14
Yes
Output
King of

Variables Used:

Type	Variable Name	Purpose
Integer	value	Holds the dealt card value
	suit	Holds the suit of the card
	playTot	Holds the player's score
	runTotal	Used to keep a running total of the player's score
	total	Used to hold the value of a card
	hTotal	Holds the house's total score
	hrTotal	Used to keep a running total of the house's score
	cardVal	Holds the card value within the function
	crdSuit	Holds the card value within the function
	choice	User input selection
Character	answer	Used to allow the player to continue the game
Boolean	exitMenu	Used in Case 3 to end the program

Concepts Used:

Textbook Used: Problem Solving with C++ 8th Editions by Walter Savitch

2.1	Variables and Assignments
2.2	Input and Output
2.3	Data Types and Expression
2.4	Simple Flow Control
2.5	Program Style
3.1	Using Boolean Expressions
3.2	Multiway Branches
3.3	More About C++ Loop Statements
3.4	Designing Loops
4.1	Top-Down Design
4.2	Predefined Functions
4.3	Programmer-Defined Functions
4.4	Procedural Abstraction
4.5	Scope and Local Variables
5.1	Void Functions
5.2	Call-By-Reference Parameters

Other Sources of Information:

Functions: http://www.cplusplus.com/doc/tutorial/functions/
Boolean Expressions: http://www.cplusplus.com/forum/beginner/15438/
Boolean Operators: http://www.cplusplus.com/doc/tutorial/operators/
Void Functions: http://www.cplusplus.com/forum/beginner/113032/
Call-By-Reference Parameter: http://www.cplusplus.com/forum/general/7990/
Flow Control/Loops: http://www.cplusplus.com/doc/tutorial/control/