

SLOT MACHINE

CSC-5 40375



February 4, 2014

ABEL SALAZAR

Asalazr25@student.rcc.edu

INDEX

PREFACE page2

INSTRUCTIONS page2

VARIABLES page3

FUNCTIONS page3

FLOWCHART page6

Currently the game is optimized for Netbeans but cstdlib has been added for some older machines. Bloodshed will not run from a plug and play.

Copyright - February 4, 2014

**Preface**

Originally I had constructed just the slots game. With the constructs, I had to expand the concept of just having a simple slots game and decided to make a small casino simulator. The program includes Blackjack and craps. It was supposed to include 5 card poker as well as Texas Hold’em, but they had too many bugs to present at the unveiling of Abel’s Casino version 2.1. Needless to say version 3.0 will include them and any of the unknown bugs presently in the game will be fixed.

**INSTRUCTION MANUAL**

First off, you are given the option to load your saved game. Safeties aren’t built into the load game function so you have to be careful to enter “1” or “2”. There is no option to leave game until you pass this section. After that you choose from the 3 games or the exit number.

**SLOTS**

The slots game is very simple in my model. The program prompts you to continue or exit after every section. The game will then prompt you to make a bet. The bets are preset to 3 specific bets. You must choose between $1, $5, and $25 but selecting the corresponding numbers 1, 2, and 3. Characters are randomly generated several times and the last set are the ones that are used to determine if you won. You must get all of the same character on the same line vertically or horizontally. Version 3.0 will include wild cards and the ability to win diagonally. You win your bet times 4 when you win and it is added to your game after every pass.

**BLACKJACK**

Blackjack is played with you against the computer A.I. You place your initial bet and it is subtracted from you total money. Two cards are then dealt to you and the dealer. The values of the cards are the same for all card values between 2 and 10. Cards with face values of J, Q, and K are all worth 10. Aces or “A” have a hard and a soft value. Hard is worth “11” and soft is worth “1”. The computer automatically adds the values for you. After the initial draw you are given the option to hit or stand. The goal is to have a sum higher than the dealer while not going over 21. So you hit as many times as you need in order to get your value up to 21. The dealer will hit if he has less than 17 so keep that in mind. If you beat the dealer you get your money back times 2. If your cards and the dealer’s cards are equal, you push. Basically, you just get your initial bet back as if nothing happened.

**CRAPS**

Craps is played with a pair of 6 sided dice. In the first round, if you roll a 7 or an 11 you automatically win and do not move to the second round. If you roll a 2, 3, or 12 in the first round you lose automatically and you do not move to a second round. In the event that you do roll a (2,3,7,11, or 12) you must continue rolling until you roll the same number you rolled in the first round. BUT in the second round if you roll a 7, you lose. When you win, you receive an amount equal to double your initial bet. Currently the game is optimized for Netbeans but cstdlib has been added for some older machines.

**Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **Function** | **Description** | **Location** |
| Int | Whole numbers | 4 bytes | Every function |
| Unsigned short | Whole numbers | 2 bytes | Used with random number generator  int main(); |
| Long | Whole numbers | 8 bytes | Global variable “money” |
| Char | Character | 1 byte | int convertString(string);  int convertNum(char);  void cvn\_char (int [], int, int); |
| String | Character Array | Depends on the size of the array | int convertString(string);  string loadGame(); |
| Bool | True / false |  | bool ifSave(); |
| Float | Numbers with decimal | 4 bytes | Money |
| Double | Gets you an F | Self-explanatory  8bytes | N/A |
| Array | Stores ints and chars | Stores data in for my slot grid | Main |
| vector | array |  | Bets |

Functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **Function** | **Description** | **Location** |
| void randGen (int [], int); | Generates random numbers for an array | Uses do while loop to fill 9 sections of array  a[ counter ]. Uses modulus 6 + 1 to ensure value between 1 and 6. | Line 228 |
| void prntArray(int [], int, int); | Outputs array that has been populated | Used to output the values of arrays used in program. Uses modulus 3 to end each line for formative of grid | Line 218 |
| void cvn\_char (int [], int, int); | Outputs letters for corresponding numbers | Uses for loop to sort through array and output in the form created by prntArray() | Line 193 |
| void prntSpc(); | Prints empty space | Uses for loop to output 25 end lines.  Used to clear the screen after inputs | Line 186 |
| bool is\_Winner(int [], int); | Tests columns horizontally and vertically for matching numbers | Tests populated array with corresponding patterns. Uses else if statements to figure out if each pattern is a winner. Winner returns true | Line 163 |
| bool ifSave(); | Determines if you want to load a saved game | If user decides to use a saved game, if returns a true value | Line 237 |
| string loadGame(); | Loads saved game value | Uses fstream library to load numbers store in “saved\_game.txt” Uses getline to read in the string stored in the file | Line 255 |
| int convertString (string); | Converts string to character array while adding up values from the file | Uses string.size to determine size of the string. The only files being stored are numbers. The string is plugged into a character array with a for-loop. To save time the function calls another function to convert the numbers while they go through the for-loop | Line 271 |
| int convertNum (char); | Used to change character to an int | Uses an int datatype in function. When a number between 0 and 9 occur it returns the integer | Line 296 |
| int thouConvert(int, int); | Converts single digits to their actual value | Uses 10 and n – 1 as the power. Uses size to determine n and multiplies the new total with the int passed through as a parameter. Works with large integers | Line 343 |
| void saveGame(int); | Saves your money | Opens “saved\_file.txt” and overwrites the old number with the new number. Closes the file | Line 352 |
| int placeBet(); | Reads in the bet you want to place | Outputs bet options and gets input as an integer. Uses switch case to return the value of the target bet | Line 141 |
| int prize(int); | Modifies bet | Uses modifier to multiply bet because you only win 1 every 5 times. So it is set to return 4 times the amount of money bet. | Line 136 |
| Craps(int); | Runs craps program | Calls functions and is the main driver for craps. Calls randGen to fill array of random die rolls. |  |
| isPos(float, float) | Detersmines if true | Takes bet and the total money and checks if it is positive and also greater than 0. |  |
| Blackjack(int) | Runs black jack program | Calls functions and is the main driver for blackjack |  |
| firsthand(int [], float) | Draws 2 pairs for cards | Calls randgen and prompts user to make a bet. Outputs first cards. |  |
| hit(int [], float &, float&) | Asks user if hit? | Takes total of cards and adds another card to the total. If the total is over 21 it ends the loop |  |
| compHit(int []) | Gives coms hand cards | If the comps hand is less than 17 it gives the comp cards cards until the value is over 17. If the total is over 21 the dealer busts |  |

FLOWCHART

