4Leaf Co.

TESTING

Jordan Quick (PM)

Frances Coronel

Calvin Chambers

Anesha Passalacqua

Table of Contents

4.0 Tracking and Control Mechanisms

1.0 Introduction

1.1 Overview

4Leaf Co. has been tasked with implementing a program that will correctly teach a user how to play Blackjack. This program will be used as both a learning experience and interactive activity to provide users with a better understanding of this fun game. Applying various tools and a Database with an Interactive Desktop Environment will result in the creation of an efficient and interactive game of Blackjack.

2.0 Test Plan

2.1 Software To Test

The software to be tested is a program that teaches a user the correct way to play blackjack. There are four components that will be tested: database, a program that calculates the probability of winning based on n hands, a program that calculates the overall probability of winning a game and the blackjack game itself. A series of tests will be performed to confirm that each portion of the program is working the correct way.

2.2 Testing Strategy

Outlined below are the ways that this program will be tested to see whether it passes or fails. Any component that fails a test will be fixed and re-tested until it passes and remains as passing.

2.2.1 Unit Testing

Unit testing is the software development process in which the smallest testable parts of an application are individually and independently scrutinized for proper operation. Unit testing is time-consuming and tedious.

2.2.2 Integration Testing

Integration testing is when program units are combined and tested as groups in multiple ways. This test can expose problems with the interfaces among the program components before trouble occurs in real-world program execution.

2.2.3 System Testing

System testing is when there are a variety of tests that are performed on a system to explore functionality or to identify problems. This is where we test the system to the point where it malfunctions or returns incorrect results.

2.2.4 Functional Testing

Functional testing is tested for acceptability and assess whether it is acceptable for delivery. The result from this test is either a pass or fail; either it completes each function as it should or it does not.

2.3 Test Resources

This program is solely a desktop application. In order to complete the functionality of this program, all components of this application must work together from the database to the calculation programs onto the actual blackjack game.

2.4 Test Metrics

Metrics for each test will vary for each component tested. Success (or passing) will be determined by the component performing the correct function, producing accurate results and remains and/or remains as passing. If a component fails to remain passing, it will be re-evaluated until it's passing again and remains as passing.

2.5 Test Schedule

Throughout the development of the Blackjack program, tests will be conducted on a regular basis to ensure the product is on time for final delivery. As we get closer to final delivery, there will be an increase in the amount of tests that are conducted.

3.0 Tested Items and Status

3.1 Test Log

NAME OF TEST	DATE EXECUTED	RESULTS (PASS/FAIL)

Database		
SQL Server Connection Test	2/8/2016	Pass
Displays Random Hand Combinations	2/8/2016	Pass
Displays Variable Constants for Cards	2/8/2016	Pass
Sets Number Of Deck Sizes	2/29/2016	Pass
Randomizes *n* Possibilities	3/1/2016	Pass
Connects to Frequency Table	2/28/2016	Pass
Statistics Program 1: Frequency of n hands occurring		
Displays User and Dealer	2/28/2016	Pass

Cards		
Displays User and Dealer Totals	2/28/2016	Pass
Displays Frequency Table	3/1/2016	Pass
Displays Correct Random Frequencies	3/1/2016	Pass
Randomizes 1 Million Possibilities	3/2/2016	Pass
Statistics Program 2: Probability of winning due to each action during the game		
Creates Strategy Card Text File	3/31/2016	Pass
Formulates Statistics of Winning For Each Action	3/31/2016	Fail
Simulates Game in Program for Each Action	4/27/2016	Pass
Creates Strategy Card Output	4/27/2016	Pass
Randomizes 1 Million	3/31/2016	Pass

Times Per Action		
Game Console Program		
Displays Template for Strategy Card	4/25/2016	Pass
Displays Way to Use Different Number of Decks	3/1/2016	Fail
Hit Method	3/1/2016	Pass
Stand Method	3/1/2016	Pass
Double Down Method	4/25/2016	Pass
Split Method	4/25/2016	Pass
Displays Game Output	3/1/2016	Pass
Reads Strategy Card from input file	3/31/2016	Pass
Editable Strategy Card	3/31/2016	Pass

Corrects user if makes wrong move	3/31/2016	Pass

PLEASE SEE TESTING PLAN CHECK OFF FOR FURTHER TESTS

3.2 Test Descriptions

Database

- *SQL Server Connection Test*
    - The SQL Server Connection Test was conducted using the MySQL Workbench. This enabled the database to connect and import data into tables and to store values of the cards.
    - Description on how to run test:
        - To run this test, first start by opening up MySQL Workbench. In the workbench, click on the connection information in the lower left hand corner. If you can see the status then the this test is completed.
- *Displays Random Hand Combinations*
    - The “Displays Random Hand Combinations” Test was tested to see if the database could be randomized to display a random possible hand of the first 20 cards. This test succeeded with the implemented algorithm.
    - Description on how to run test:

        - To run this test, formulate SQL code to view the content of each table in the database. Configure the following code: SELECT * FROM blackjackdatabase ORDER BY RAND() LIMIT 20; to formulate a random hand of 20 cards for one deck.

- *Displays Variable Constants for Cards*
    - The “Displays Variable Constants for Cards” Test was tested to see if the database could display the values for the constants in the database. This test worked for both Java Code and for commands in MySQL Command Line. MySQL Command Line Commands proved more organized than Java Code but produced similar results.
    - Description on how to run test:
        - In order to run this test, you can use the MySQL Workbench or the MySQL Command Line Client. Access the database and display the set of cards set in each table for the set of hands.
- *Sets Number Of Deck Sizes*
    - The “Number Of Deck Sizes” Test was a test to display the number of decks used in the game for the possible hand combinations. This test failed to produce the number of decks

and could not be implemented past 1 deck for random combinations.

    - Description on how to run test:
        - In order to complete this test, you will need to access the tables in MySQL Workbench. From there you are able to change the sizes of the decks through SQL code. The test is a success ones the number of decks have been changed.

- *Randomizes n Possibilities*
    - The "Randomizes *n* Possibilities" Test was to display *n* possibilities for the random 20 set of hands for the game. This test failed due to the lack of a proper loop to iterate the random class *n* times.
    - Description on how to run test:
        - In order to complete this test, you will need to be able to randomize the constraints of the deck. Configure the following code: SELECT * FROM blackjackdatabase ORDER BY RAND(); to randomize the deck.
- *Connects to Frequency Table*
    - The "Connects to Frequency Table" tests out the ability of the database being able to connect to the java code and displaying the frequency table.

- Description on how to run test:
  - In order to complete this test, you will need to access the Frequency table and check if you are able to pull card constraints from the database. If the test works, then you have completed it.

Statistics Program 1: Frequency of Hand Occurrence

- *Displays User and Dealer Hands*
  - The “Displays User and Dealer Hands” Test displays the user hand and the dealer hand which was taken from the database.
  - Description on how to run test:
    - In order to complete this test, you will need to have completed the “Connects to Frequency Table” Test. You would need to capture the variable constants of each hand and return the user and dealer cards.
- *Displays User and Dealer Totals*
  - The “Displays User and Dealer Totals” Test is a test to display the totals for the user and the dealer which defines its location in the frequency table.
  - Description on how to run test:
    - In order to complete this test, you will need to total the user and dealer cards. Once this is done and the totals

display the correct integer values, then the test is complete.

- *Displays Frequency Table*
    - The “Displays Frequency Table” Test was executed to display the format of the frequency table to show if it was able to be displayed.
    - Description on how to run test:
        - In order to complete this test, you will need to display a frequency table within your Frequency Code that displays the number of occurrence of each hand. This is done automatically within the code when it is ran.
- *Displays Correct Random Frequencies*
    - The “Displays Correct Random Frequencies” test was executed to check that the random frequencies are producing accurate results. A hand calculated frequency table was derived and checked alongside the code. This will show if the possibility of the occurring hands are accurate.
    - Description on how to run test:
        - In order to complete this test, you will need to display the random frequencies of hands from the database. This is done within the database code.
- *Randomizes one million hand possibilities*

- The “Randomizes One Million Hand Possibilities” Test was conducted to make sure that one million hands can be shown and recorded into the frequency table. Each hand combination will be tallied up into the frequency table showing the number of occurrences.
- Description on how to run test:
  - In order to complete this test, you will need to create a loop around the assignment of each rand hand combination. This will loop and create hands 1 million times.

Statistics Program 2: Probability of winning during game

- *Creates Strategy Card Text File*
  - Creates Strategy Card in a Text File Through Code to be imported by Console Game.
  - Description on how to run test:
    - In order to complete this test, you will need to use the probability program to gather the best move for each hand scenario. With this calculation, organize and sort the action (choice) with the sample strategy guide to make it easily transferable to the console game.
- *Formulates Statistics of Winning for Each Action*
  - Formulates Statistics of Winning For Each Action.

- Description on how to run test:
  - In order to complete this test, you will need to formulate the statistics of winning for each action by totalling each of the action's occurrences and creating statistics based off of the totals. If a proportional number occurs then the test is complete.

- *Simulates Game in Program for Each Action*
  - Simulates The Game and Every Action For Each Action Chosen by the User.
  - Description on how to run test:
    - In order to complete this test, you will need to check the code used to simulate the game based off the hands. If the game returns an action then the test is complete.
- *Creates Strategy Card Output*
  - This Test is executed to create the strategy card output within the program.
  - Description on how to run test:
    - In order to complete this test, you will need to display the organized set of actions within the strategy card output. This output is similar to the design of the frequency table. Once this table is devised, then the test is completed.
- *Randomizes 1 million Times Per Action*

- The Program will go through each action and create a probability based on of the action's results one million times.
- Description on how to run test:
  - In order to complete this test, you will need to formulate a loop that will form the output for each action and pick the action that appears the most for each card combination.

Blackjack

- *Displays Template For Strategy Card*
  - Displays strategy card from input file for user to reference during the game.
  - Description on how to run test:
    - In the console app code, find the line that says "System.out.println(line);" and take out the line comment that may be in front of it. Run the console and the strategy card from the input file will be displayed. Go back to the console app code and find that same line and comment that line. Run the console app again and notice that the strategy card is not displayed for the user to reference during the game
- *Displays Way to Use Different Number of Decks*

- User is able to input the number of decks they would like to use during the game
- This feature will be implemented with future development

- *Hit Method*
  - The user has the option to hit their current hand. A hit refers to taking another card from the dealer.
  - Description on how to run test:
    - Run console game. With the hand that is given, choose hit. If that was not the correct move, an error message will display and if it was the correct move then the user will continue on.
- *Stand Method*
  - The user has the option to stand their current hand. Stand refers to not taking any more cards.
  - Description on how to run test:
    - Run console game. With the hand that is given, choose stand. If that was not the correct move, an error message will display and if it was the correct move then the user will continue on.
- *Double Down Method*

- The user has the option to double down their current hand. Double down refers to doubling a bet after seeing one's initial cards, with the requirement that one additional card is drawn.
- Description on how to run test:
  - Run console game. With the hand that is given, choose double down. If that was not the correct move, an error message will display and if it was the correct move then the user will continue on.

- *Split Method*
  - The user has the option to split their current hand. Split refers to if a player's first two cards are a pair (two cards of the same value), then the player can choose to split or move a second bet equal to the first into an area outside the betting box of the original bet (using the points as coins). The game will separate the cards to create two new hands, placing one bet into each hand. The player can then play two separate hands.
  - Description on how to run test:
    - Run console game. With the hand that is given, choose split. If that was not the correct move, an error message will display and if it was the correct move then the user will continue on.
- *Displays Game Output*

- Displays the appropriate game messages depending on what action the user takes throughout the game.
- Description on how to run test:
  - Run console app and play the game. The game is played correctly and incorrectly (deliberately making the wrong) to test to see if the console alerts the user of the wrong move

4.0 Tracking and Control Mechanisms

4.1 Document Revision History

Version	Implemented By	Approved By	Date	Reason
1.0	Calvin Chambers Jordan Quick Frances Coronel Anesha Passalacqua	Jordan Quick	1/21/16	Iteration 1
2.0	Anesha Passalacqua Calvin Chambers Frances Coronel	Jordan Quick	2/11/16	Iteration 2
3.0	Calvin Chambers Anesha Passalacqua	Jordan Quick	3/3/16	Iteration 3

	Frances Coronel			
4.0	Anesha Passalacqua	Jordan Quick	3/31/2016	Iteration 4
5.0	Anesha Passalacqua Calvin Chambers	Jordan Quick	4/26/2016	Iteration 5