**MEMO**

Date:

Subject:

To:

From:

Introduction:

This project attempts to design and implement a Java application that allows the user to play the game of Blackjack.

Key Points:

* This project is designed applying the Object Oriented concepts of Java.
* GUI is implemented using SWING and AWT.
* Project is packaged using Java Web Start (JAWS).
* The system does not use any database.
* External resources and development tools used in the project are open-source.
* Implements almost all the major conventional rules applicable in a game of Blackjack (‘Split’ is not implemented).

Discussion:

Analysis:

The game of Blackjack is one of the most widely played casino banking games in the world (source: wikipedia.org). The requirements for this project are obtained from the information on Internet (http://www.pagat.com/banking/blackjack.html and Wikipedia). The system implements the object-oriented design. It is abstracted into several modules and specific functionality has been defined for each module.

Since the project is packaged using JAWS, the application can be easily downloaded by hitting the host url and run in local systems installed with Java. The application also checks for the latest version each time it is launched to keep it updated with latest changes.

The main module starts the application and prepares the GUI with separate panels for dealer, player and choices using SWING. On reception of the player’s bet, cards are dealt both to the dealer and the player. Based on the choices made by the player, the game progresses to decide a winner based on standard BlackJack rules and pays out money. The system implements several rules like bet, hit, stand etc. to handle player’s choices in the game.

The player starts the game by placing bets from his available money. If the bet amount is more than or equal to the minimum bet, the player is dealt the first two cards of his hand. He then chooses either to hit or stand in order to get as closer to the BlackJack value (21) as possible. He might get busted in the due course. Once he decides to stand, the dealer’s turn is started.

The Dealer has a pre-defined set of rules for hitting. He has to hit until his hand is greater than or equal to 17. He might get busted and the player wins automatically. If his hand has reached 17 or more, he can hit no more and stands for result.

After the player and dealer have finished their turns, the system gets the best possible value of each hand and compares their values to decide a winner and pays out the bets. Tied values result in a Push and the bets are returned. The game restarts if the player runs out of money or the deck is fully dealt. The application exits if no bets are placed. All the resources and tools used in developing this project are open-source.

Number of hours needed to get the code working:

Number of hours spent preparing submission:

Challenges faced:

1. The game is played with different set of rules in different parts of the world. A prominent variant was researched to implement consistent rules.
2. Some effort was required to develop a good look and feel interface using Swing.
3. The advantages of packaging using JAWS were understood while developing multiple versions of the application.

Recommendations:

1. The current system does not implement Split, Doubledown rules of Blackjack. This can be worked upon as an enhancement.
2. It is also desirable to support multiple players in the application.

Conclusion:

Blackjack game has been successfully implemented using the specified technologies.