Christopher Ejiofor Principles of Computing 27 April 2011

Term Project Design Description

*Set-Up*

The first thing I did when trying to understand how to best approach was to find out what kind of variable would I need to write a game to play poker I came up with: Suits(0-3 for each suit), Cards/Face(0-12 for each face), and Hand Type( 9-0 for Royal Flush, Straight Flush, Four of a Kind, Full House, Flush, Straight, Three of a Kind, Two Pairs, One Pair, Nothing/High Card). After I decided this I was advised by my CA to write poker in print for first then move on to graphics later. I began using the Three main variable above, but then I remembered using a getRandomDeck() function from hw3(Blackjack). I modified the function a bit and then decided I was going to used letters for the suits (CSDH) and a combination of letter and numbers for the faces (23456789TJKQA). To determine the hand Types, I looked at each possible hand and tried to break it down into simpler steps to that I could write a few helper functions that would return the same hand types. I concluded that my base functions would be:

1. quadsTripsPairs – counts the number of quad, trips or pairs present
2. isStriaght – to see if the hand is a straight
3. isFlush - to see if the hand is a flush
4. isRoyal – to see if the suits are royal suits

I decided to keep these functions in there on class so I created handEval and imported it into the main poker file.

*Game play:*

The game is written to be player by one player versus to computer players. However, I first wrote the game as three individual players planning against each other. The game would run from the start and continue for 3 rounds of betting. Dialog boxes would pop up asking user input for bets and typing in the indexes of the cards you want to trade it. This later evolved into click on each card that you want to give up and using a dialog box that properly handless malformed input and pressing the cancel button. From there It would evaluate the winners by comparing hand ranks, and if hand ranks are the same, comparing high cards. All this was done under playDrawPoker calling various helper functions such as playPokerRound, playAutoRound.

*Poker AI:*

To have the poker game lay against the computer, I wrote a poker AI file to hand actions of an automated player. If the hand rank was a straight or better, the hand was kept because there is a very low chance of getting a better hand once you hand is that high. However, the better the hand the more the bet. If the hand was ranked lower, I looked at every communication of cards from the hand with length of 4 or 3 or 2 and tried to see how many cards would need to be deleted to get that combination with a high rank. Then the option/combination that had a combination of the highest rand and lowest number of cards deleted was chose and the computer does that option.

*GUI:*

The GUI was built loosely on the sample board file, with a green background similar to most casino tables. I created a black half circle at the top for the deck and information on the player turn as well as the betting round. I then crate the three hand for the player, placed on top of a black square, for aesthetic appeal. In the middle there is a black oval the represents the money pot as well as the green text to the right of a player hand to represent the amount of money he has. I also created a welcome screen and rules screen implementing images and button to make to program look better.