Noah Phillips

CS400, Fall 2019

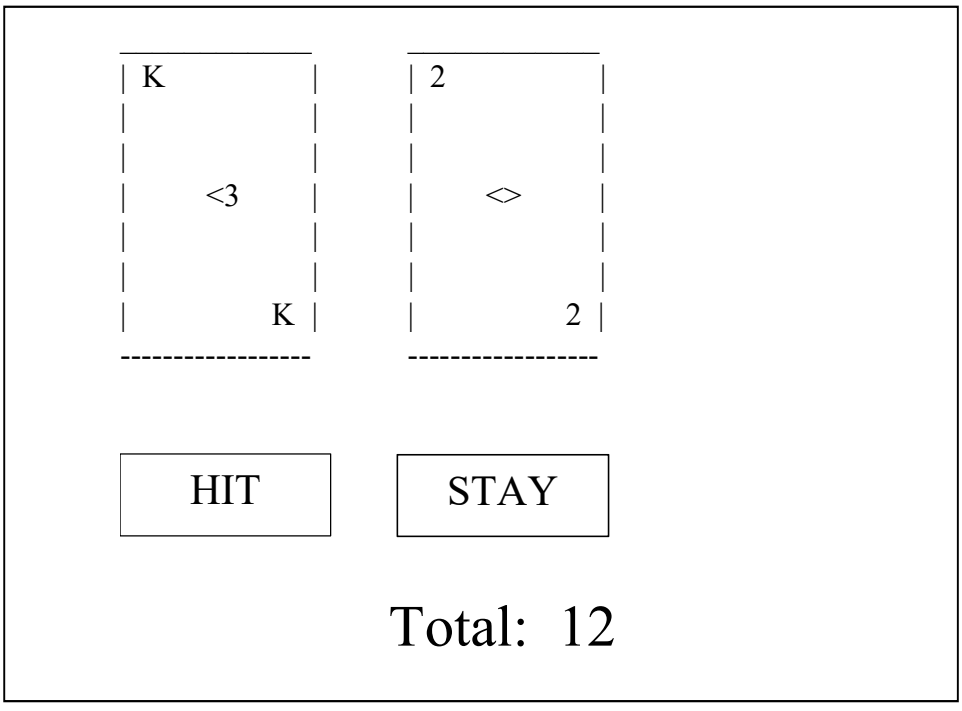
x4 Individual Member Project Proposal

Title: Array-based Blackjack

Problem: This game will both alleviate boredom as well as serve as a training tool to learn how to play and develop strategies for Blackjack.

Primary Stakeholder: Anyone can enjoy playing this game, although the primary stakeholders would be 1. Anyone trying to learn the basics of Blackjack and 2. Anyone trying to refine specific strategies through repetition of simulated situations.

Graphical User Interface: The GUI will display the cards in the user's hand. Then, the user will be given the option to either "hit" and receive another card, or stay with the cards they have, in line with the standard Blackjack rules. The user will make this decision by clicking one of two buttons labeled "HIT" or "STAY". The GUI will also display the numerical summation for the cards in the user's hand. The interface will give the user feedback on whether their decisions were good, or else give advice on what should have been done instead. At the end of each hand, the user can choose to play another hand or to exit the program.



Data: This program will utilize a class Deck, which will hold 52 objects of type Card. These 52 Cards will be stored in an array, and each Card will hold two primary fields, suit and value, which together represent the card (4 of spades, Queen of hearts, etc.). Rather than "shuffling" the deck, cards will be randomly retrieved through the use of a random number generator to access a random index within the Deck array. That index will then be marked as invalid until the end of

the hand, to prevent duplicate cards. Once cards are retrieved, they will be placed into a List which represents a player's "hand" of cards.

This program can also create customized sets of Cards by reading from a data file. This data file allows the user to determine the symbols used to represent the suit and value of the cards. By default, for example, the king of hearts is represented by the one-character symbol K and the two-character symbol <3, however a user could instead set the character for King to be '%' and the two characters for hearts to be "@~". By reading and parsing this file, the graphical interface will use the user-supplied symbols when the cards in their hand are "drawn" to the screen. The input file can be a variant of a standard text file. If no input file is given, the program will execute using the default symbols.