Jane Wang
ICS3U1
Ms. Ma
Apr.30.2014

Culminating Activity

Part A

DESCRIPTION:

The program will perform a game called “Black Jack”, also known as “Twenty-one”, it is a text-based game involving one player to play with the computer, and the aim of this game is to get close to 21 points without exceeding 21 points. During the game, users will be given information on his/her cards in terms of images and texts. At the end of game, player will be given money status, result and feedback.

RULES:

1. Face cards (kings, queens, and jacks) are counted as 10 points, ace is counted as 1 point, other cards are counted as the numeric value shown on the card, Jokers are excluded.
2. Each player will be given two random cards as initial cards, all cards are faced down, only the player himself/herself knows what the cards are. The score are counted by adding the points of the each card the player has.
3. After receiving the initial cards, player can choose to get an additional card (also called as “hit”), or to stop receiving cards for on round (also called as “stand”). The dealer has to take hits until his or her cards total 17 or more points.
4. If no players are asking for a “hit”, then the game ends, the result of failure or victory will be made at the end of the game.
5. Player wins if he/she does not bust and has a total that is higher than the dealer's. The dealer loses if he or she busts or has a lesser hand than the player who has not busted. If the player and dealer have the same point total, this is called a "push", and the player typically does not win or lose money on that hand.
6. At the beginning of the game, the player has 10 dollars as initial money, and can wage from 1 to 10 dollars. If the player wins without the dealer busting, he/she will receive double money based on his/her wager. If the player wins with the dealer busting, he/she will receive triple money based on his/her wager. If the player loses, he/she will lose double money based on his/her wager.

PROGRAMMING CONCEPTS:

Decision: if-else statements. Used when indicating player’s choice on whether or not picking a card and indicate the result of the game.
Repetition: while loops, do loops and for loops. Used when repeating output the information and questions, displaying the pictures of the cards that player has and focusing player on specific replies to some questions.
Array: string array which storing the feedback for the player at the end of the game.

Graphics: pictures of 52 cards. Applied when displaying the cards that the player has.

REASERCH: Further concepts on graphics and array, especially on how to store a graph as an image.

IPO CHART:

INPUT	PROCESSING	OUTPUT
1.Need user to enter his/her name **String name;** **2.** Need information from user that indicate his/her options on whether or not picking an additional card **char option;** **3.**need information from the player that indicate whether or not he/she wants to start another game **char continue;**	**1.** Declare "name" to null and read in what user has entered. **2.** display a brief rules for the user **3.** Declare player's cards and dealer's cards with array scalled "playerCard[]" and , "dealerCard[]", then initialize them to null. **4.** Declare and initialize the name and points of the 52 cards by using cardName[52], and cardPoint[52]; **5.** Prepare 52 images for each card and store them in a array called "image[52]". **6.** Randomly give the player and dealer two cards by using math method, Math.random(). **7.** And then parse them to playerCard[] and dealerCard[], display player's cards each round but do not display dealer's cards. **8.** Randomly give the dealer a card each round until the dealer's score is greater than 17. **9.** Ask player whether or not to pick a card. **10.** if the player choose to pick a card, then randomly give he/she a card from the rest of cardName[] using Math.random() **11.** Initialize player's score and dealer's score to int playerScore;	**1.** output the name of the game, "Black Jack" and its brief rules **2**.ask and output player's name **3.** output card names and their images for each round **4.** player's total score **5**. feedback at the end of the game **6.** dealer(computer)'s cards at the end of the game **7.** ask player whether or not to start another game **8.** the format of the output should be as following: • User's name • Names and images of the cards that the player has at the end of the game • names and images of the cards that the dealer(computer) has at the end of the game • his/her total score • the money status of the player • feedback about the game • ask player if he/she wants to play another game

	and int dealerScore; and then calculate their scores and compare. Example: playerScore += cardPoint[i]; **12.** Display the results (including money status) of the player and ask him/her whether or not to start another game. Repeat the game if the player choose “y”.	