Gretel Rajamoney

1/25/2020

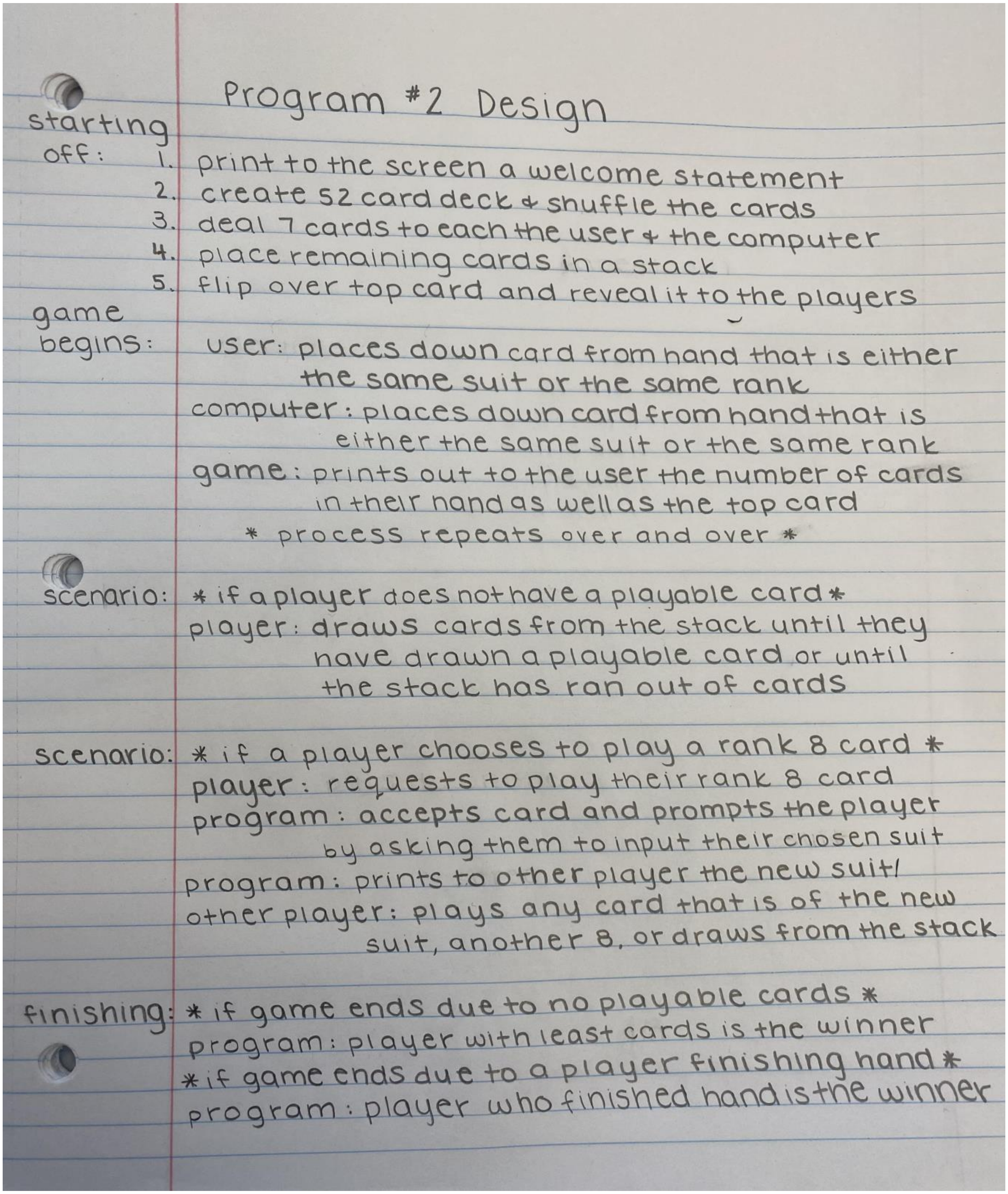
Design Document #2

Understanding the Problem:

For this assignment I will be writing a program that allows the user to play a game of Crazy Eights against the computer. Crazy Eights is a card game that utilizes all 52 cards in the deck. To be more specific, starting from the Ace, the numbers 2-10, the Jack, the Queen, and lastly the King. These 13 cards will be duplicated amongst 4 different suits; the Clubs, the Diamonds, the Hearts, and lastly the Spades. Within each of these ranks and suits, there exists only one of each, resulting in a grand total of 52 cards. Before the game begins, the entire deck of cards is shuffled, next each of the two players are distributed 7 cards. The remaining 38 cards are placed face down in a stack on the table, this stack is where the players will be drawing from as the game proceeds. The top card of this pile will be turned over and be placed face up on the table for the players to view. A player will begin this game by placing down a card from their hand that is either the same suit or rank as the card that has been flipped over onto the table. The card that was placed down by the player will take place as the new top card. The next player will then play a card from their hand that is either the same suit or the same number as the card that is on top. This newly placed card will replace the prior card as the new top, this process will continuously repeat. If a player is unable to put down a card that is either the same suit or number, the player will draw cards from the face down stack on the table, the player will continue to draw cards until they draw a card that is playable or until there are no more cards left to draw from in the stack. The card with the rank 8 is considered a wild card regardless of its suit, there are four of these in the entire 52 card deck. This card is useful, because it can be played on any turn and the player has the ability to select the suit that they would like the other player to place down. This game continues a repeating cycle until either one of the players runs out of cards in their hand, or until neither player has playable cards and the stack is empty. When the game finishes the player that put down all the cards from their hand is crowned the winner, if the game finishes because neither player has playable cards, then the player with the least amount of cards in their hand is crowned the winner.

When this program is being created it will need a Card class, Deck class, Hand class, Player class, and a Game class. The Card class will contain all 52 cards, the suits and ranks of all these cards will be represented using the int data type. This class will also ensure that each of these cards will represented with a string that is understandable to the user as they are playing. The Deck class will be the source of all the cards. In this class the cards will be transferred to the players hand and will also deal cards from the deck and then to the player. The Hand class will be responsible for storing all the cards in a player’s hand. The class will also be responsible for deleting and adding to the array size as cards will be frequently added and removed as the game plays out. The Player class will represent the player as well as their hand, and it will keep track of all the cards the player has laid down. This player class must be representative of both the user as well as the computer. Lastly, the Game class will represent the entire game. This class will keep track of when the game has been finished as well as whether the players have executable moves that can be made. The program as a whole is required to create the deck of 52 cards correctly and shuffle them. It must also be able to deal 7 cards to both players and execute the game of Crazy Eights correctly. Throughout the game, the program must print to the user the current state, which includes the cards held by the user and the card that is on top of the pile. The computer player’s card total will be kept hidden from the user. Once the game is finished the program must print out the winner, as well as request whether the user would like to play again. The game must contain no memory leaks.

Devising a Plan/Design:



Program #2 Design

starting off:

1. print to the screen a welcome statement
2. create 52 card deck & shuffle the cards
3. deal 7 cards to each the user & the computer
4. place remaining cards in a stack
5. flip over top card and reveal it to the players

game begins:

user: places down card from hand that is either the same suit or the same rank

computer: places down card from hand that is either the same suit or the same rank

game: prints out to the user the number of cards in their hand as well as the top card

* process repeats over and over *

scenario: * if a player does not have a playable card *

player: draws cards from the stack until they have drawn a playable card or until the stack has ran out of cards

scenario: * if a player chooses to play a rank 8 card *

player: requests to play their rank 8 card

program: accepts card and prompts the player by asking them to input their chosen suit

program: prints to other player the new suit/

other player: plays any card that is of the new suit, another 8, or draws from the stack

finishing: * if game ends due to no playable cards *

program: player with least cards is the winner

* if game ends due to a player finishing hand *

program: player who finished hand is the winner

Looking Back/Testing:

Actual	Expected	Actual Meets Expected
user plays a wild card	accept the card and then prompt the user to input what suits they would like to select	
user does not have any playable cards to put down	draw cards from the stack until they draw a playable card or until the stack is empty	
if the stack becomes empty and the player has no playable cards	the player will pass their turn and the opposing player will continue and put down a card	
if neither player has playable cards in their hand	the game finishes and then the player with the least number of cards in their hand is declared the winner of the game	
if one of the players finishes the cards in their hand	the player who finished their cards is declared the winner	
when the game finishes	declare who the winner of the game was and then prompt the user by asking them whether they would like to play the game again	
if the user requests to play a card which they don't have	print to the user that the card is invalid and to try again	
if the user attempts to play a card that isn't playable	print to the user that the card is not playable and to try again	
if the first flipped over card is a wild card	flip over another card until the top card is not a wild card	