Programming Project 2

Crazy – 8s

NAME:__________________________________ DEMO DATE:______________

DUE DATE: Wednesday, 14th March, 2012

Crazy-8's: A basic description

In this game, the user "plays" the computer. Both user and computer are dealt seven cards, but neither player has knowledge of the other's cards. The next card in the deck becomes the "in-play" card. Starting with the user and alternating between user and computer, the player must select and put down a card that matches the in-play card in either face value or suit. The newly placed card becomes the new in-play card. If the player is unable to match the in-play card, that player must draw cards from the deck until a match occurs and can become the in-play card. The first player to exhaust all of their cards wins. If the deck is exhausted and no player has won, the discard pile is reshuffled and put back into play.

<u>For a grade of "C", the following is required</u>:

1. Create a C program with proper documentation and indentation
2. Print out a description of the game. Print out historic game statistics.
3. Ask the user if he/she wants to play or exit the game.
4. At the start of the game, a new deck of cards is used.
5. 7 cards each are dealt to the user and computer to populate their respective hands (alternate cards as if they were being dealt).
6. Display the user hand but do not display the computer's hand– just display the total number of cards the computer has in its hand. [*Note: when trouble shooting your code, go ahead and display both hands so you can check that the computer's decisions are correct. I suggest having a game mode and a demo mode that is available only to you the developer and not the user*]
7. Display the in-play card (1st card after the user and computer's hands have been dealt).
8. User goes first and then the computer
9. Player (user or computer) turn:
    a. Once the player plays a card, that card becomes the in-play card and is displayed.
    b. Make sure to update the display on the player hand once a card has been played.
    c. If the player has no matching cards then they must draw cards from the deck till a match is found.
10. Remember, the user cards must be displayed but the computer cards are not to be displayed (you must inform the user how many cards were drawn by the computer to find a match).
11. If the deck is exhausted, reshuffle the discard pile and go to step 7.
12. A game is won when either the player or the computer have discarded all their cards.
13. Track and display the number of times the computer and user have won.
14. Track and display the game statistics – be creative, don't just track number of user wins and computer wins, think of some other cool stats.
15. After a game is over, you should query the user to continue playing or exit the program. If the user chooses to continue playing, go back to step 4.

<u>For a grade of "B", you must do the above and add the following:</u>

- At the start of step 9. Allow the user to exit the game and save all the cards for the user and computer. Allow the game to restart at the point in the future. You will have to also save the discard pile and the draw pile.

For a grade of "A", you must do the above and add the following:
NO points will be given to this section unless you complete all of sections B and C.

- Implement the Crazy-8 rule – i.e. 8s are wild and the player may put an eight down when there are no other playable cards in the hand. The player that puts down an eight must pick a suit.
- You are to write an Artificial Intelligence computer player, which makes the computer make reasonable decisions when playing. Have some fun with this requirement. I suggest having a cheat mode; have the computer make decisions based upon the user's cards. **To earn full credit for this section, the computer must do something challenging and interesting.**

Extra credit (5 points):
If you demo on or before March 12, you will receive 5 extra credit points.

Project Tips:
The look and feel of the game is up to you (aside from the functional requirements stated above). Use `printf` statements to design a game logo and an attractive front end game display. Have stopping points in the game so the display can be read without scrolling backwards. I suggest you have a friend (someone not in this class) play your game and give you some feedback. This really is an excellent way to get suggestions for improvement before the instructor plays the game. Don't wait until the demonstration of your program to ask questions. Start early and clear up any confusion or questions before that point.

Turn in Requirements (in this order) in lab:

- Hard copy of your source code (output not required) and be ready to demo in lab. Remember to attach this handout as a cover sheet.

Your program will be demonstrated in lab. IMPORTANT: I will have about 5 minutes per student to check out your program. When your name is called for demo, you must be ready to go.

Tentative Rubric:

Basic operations of the game, flow of the game -displays in-play card -displays number of cards in computer hand -reshuffles discard pile if deck is exhausted -displays entire user hand -count number of wins and display game stats	55 pts
Save game	10 pts
Crazy-8s rule	10 pts
Computer AI	5 pts
Good design code (module, functions)	10 pts
Format, comments, etc	10 pts