CSE 264 – Web Systems Programming, Fall 2014
Class Project – Step 2

Due Wednesday, 11/19/14

- **Objective**
  The project will bring together a number of different topics we have covered in class and provide additional implementation experience.

- **Description**
  We will be playing a variation of the card game Uno using a standard deck of playing cards. (See http://www.instructables.com/id/How-to-play-UNO-with-regular-playing-cards/)

- **Layout of the Table**

  1. The hand of cards that you implemented in Step 1 should be displayed horizontally near the bottom of the screen.
  2. In the upper right area of the screen there is a textbox and a login button for logging the user into the game.
  3. The last played card should appear roughly in the middle of the screen in portrait orientation.
  4. In the center area of the screen, somewhere near the last played card are several sets of buttons. These buttons should all be visible but grayed out and deactivated when not needed.:
     a) Spades, Hearts, Diamonds, Clubs: these are for when a player plays a King and needs to pick a suit.
     b) Reverse, Skip: These are for when a player plays a Jack and needs to pick an action.
     c) One: This is for when a player has only a single card left.
     d) Draw: This is for when a player cannot play a card and must draw.
  5. Somewhere towards the right side of the screen is a block with a shaded background for displaying messages from the server. This area must have 140 characters of space.
  6. On the left side of the screen is a vertical list of the names of the players in this game and their current scores. The name of the player who is currently playing should be highlighted.

- **Play of the Game**
  1. The game starts by a player typing in a login name into the text box and clicking the login button.
  2. Next the player list is loaded with the top name on the list highlighted and the initial card is displayed.
  3. The current player plays a card if possible by dragging the card to the played card pile, or, clicks the Draw button if s/he has no playable card.
  4. If the player plays a 2-9, then their score gets updated and play moves to the next person.
  5. If the player plays a Jack with a matching suit to the current card, the Reverse and Skip buttons are activated. Then the player clicks one of the two buttons the appropriate change in the next player occurs. The buttons are immediately deactivated.
  6. If the player has a Queen and either the suit matches the current card or another Queen was just played, and the player plays the card, then 2 cards are added to the next player's hand and the play progresses skipping the player who drew the cards.
  7. If the player plays any King the suit buttons are activated and the player clicks the desired button. The play then progresses to the next player and the suit buttons are deactivated.
  8. If the player plays any Ace, then 4 cards are added to the next players hand and the play progresses skipping the player who drew the cards.
  9. If the player has 2 cards left, then they must click the One button before playing their penultimate card. If they fail to do this before the card hits the played card pile, then they forfeit their turn.
  10. When someone plays their last card and goes out...

- **Instructions**

1. Open your project from Step 1.
2. If you haven't already, update your code so that it displays the card you dragged to the played card pile.
3. Add the following script tag to your page after the tag that loads the jQuery library:

```
<script src="http://tertullian.cse.lehigh.edu:3200/socket.io/socket.io.js"></script>
```

4. Add all the additional elements to the page: buttons, table for players, message area, login text.

5. Create a script tag at the bottom of your page for all your JavaScript code. Add the following definition:

var serverUrl = “http://tertullian.cse.lehigh.edu:3200/”;

6. Add (and fully implement) the following js functions to the script tag:

(a) function displayPlayers(players)

Players is an array of objects like the following:

{ name: “Bob Smith”, score: 25, one: false }

The function should empty the player list and refill it from the array. The name and score go in separate columns. Highlight the row if the one property is true.

(b) function displayCard(card)

Card is one of the card numbers from Step 1. The function displays the given card in the last played card pile (also the drop location for your drag and drop).

(c) function enableSuits(true/false). The function (un)grays and (de)activates the suit buttons (makes then (un)clickable). function enableReverseSkip(true/false), function enableDraw(true/false), function enableOne(true/false).

(d) function displayMessage(message). Set the html content of the message area to the string message. Note: the message string may include html tags.

7. Add the following function stubs to the script tag and hook them up to the appropriate buttons:

(a) function setSuit(suit) where suit = [“C”|”H”|”S”|”D”].

(b) function setReverse()

(c) function setSkip()

(d) function setDraw()

(e) function setOne()

8. Add debugging code to test all of the buttons and other functions.

9. To submit your assignment, do a clean on your NetBeans project, ZIP it up and upload the zipped file to Coursesite.