Project 2 - Javascript

Paul Linton

CS 316 Program 2 JavaScript Fall 2017
=====================================

Oh no! Xelkalai's attempts to return home has failed! A miscalculated conversion due to a programming mistake in the Xelkalai Conversion web app has damaged his system severely.

He cannot repair his system with the materials he has on hand and he has no Terran currency. He has a few trinkets of personal value.

A Terran collector that specializes in collecting otherworldly items has made a bargain with Xelkalai. In exchange for Terran credit Xelkalai agreed to give his trinkets to the Terran. The Terran, however, must agree to give Xelkalai a chance to acquire them back in a game of chance from his planet.

Xelkalai presents his game to the Terran and the Terran has hired you to write the game. Xelkalai has learned from his previous mistake and does not wholly trust Terran programmers. Therefore he has agreed to your hiring as long as the program is written in JavaScript, which he can scrutinize before playing. After all, gambling establishments are notorious cheats around the galaxy! He trusts *you*, just not the Terran. The game is called, Xarnker. Xelkalai gives you the following game rules followed by coding requirements he insists you use.

Xarnker rules:
=============

0) The game is card-based.
1) There are (4) suits of cards and each suit has 10 elements. Xelkalai is completely shocked that your planet does not have any Xarnker cards. You suggest an alternative representation of common playing cards ace (aka 1) through 10. Each suit will replace the 4 suits Xelkalai describes in detail. He accepts your substitution until he can acquire better replicas.
2) The game is involves two people head-to-head.
3) Each player is dealt 3 cards. One card is dealt face up to start the discard pile. The rest of the deck is "face down".
4) There are 3 rounds. Each round a player can choose to pick up the top card from the pile, or the top card from the deck. If the player doesn't want a card, they must pick the card from the pile, or the next card from the deck and simply discard it.
5) After choosing, a card must be placed on the discard pile.
6) After 3 rounds the player with the highest scoring 3 cards wins the hand.
7) Each game consists of 5 hands.

Scoring a hand (highest to lowest):

50 - (3) of a kind - any 3 cards with the same value
40 - if the 3 cards values are an ace, a 2, and a 3,
of any suits.
31 - if the 3 cards values are a 3, an ace, and a 4 of
any suit.
X - where X is the sum of cards with the same suit only.
So if you have the 10, 9, and 6 of clubs, X = 25.

8) Each game must go to 5 hands even if someone already won 3 hands.

===

Xelkalai's Programming Requirements:
0) You shall submit your JavaScript file to Canvas named as
"Lastname_p2.html", or "Lastname1Lastname2_p2.html" if you
work in a team.
1) You shall properly comment your code including putting your
name(s) at the top.
2) You shall follow Dr. Finkel's checklist for good programming:
http://www.cs.uky.edu/~raphael/checklist.html

3) You shall write a JavaScript program to play Xarnker with the
following requirements:
- Your program shall ask the player for their name via a prompt
box.
- Your program will then begin a loop of (5) games following the
scoring above.
- For each game you shall shuffle the deck with a proper random
function using the deck format below **.
- You shall deal two (2) hands of (3) cards and turn the 7th card
up onto the discard pile.
- You shall allow the player to choose their move as they are player 1.
- The player can click on the top card of the discard pile, or
the top card from the deck. Add the card to their hand.
- The player must then choose which card to discard. Add that
card to the top of the discard pile, face up.
- After the player, then the computer will play as player 2.
Your program shall execute a function decide() (written by you)
to decide what to do. Your function will use the following
decision:
- if all 3 cards are the same suit, attempt to get a better
card of the same suit.
OR
- if 2 cards are the same suit then attempt to draw a 3rd card
of that suit (either from the pile top or a new card).
OR
- if 2 cards are the same value, attempt to draw a 3rd card of
that same value.
OR
- if 2 cards are in the set (ace,2,3) then attempt to draw the
third card of the set.
OR

- if 2 cards are in the set (3,ace,4) then attempt to draw the
third card of the set.
OR
- [this means 3 cards of different suits]
if the discard card is 8, 9, or 10 and matches one of our suits
choose it, discarding the lowest valued card of the other 2
cards.
otherwise draw a new card, if it matches one of our suits,
discard one of the other 2 cards with the lowest value. If
if doesn't match, simply discard the lowest value card of
the 4.
- After each hand, compute the best score for each and display.
- After the 5th hand, your program shall determine the winning
hand. You shall reveal the computer's cards to the player.

- Your program shall increment the win total of the
player if they win. Ties go to the player, not the computer.
Increment the total number of games no matter what.

Starter code
==
Deck Format: **
When your program starts, it randomly shuffles a deck of 40 cards.
This is simulated by randomizing the numbers 1-40 kept in an array.
You must use this datastruture!

Each number represents a playing card. The cards are displayed during
the play. There are graphic files supplied for each card:

1.png ace of clubs
2.png 2 of clubs
3.png 3 of clubs
4.png 4 of clubs
....
10.png 10 of clubs

11.png ace of spades
12.png 2 of spades
13.png 3 of spades
14.png 4 of spades
....
20.png 10 of spades

21.png ace of hearts
22.png 2 of hearts
23.png 3 of hearts
24.png 4 of hearts
....
30.png 10 of hearts

31.png ace of diamonds
32.png 2 of diamonds
33.png 3 of diamonds
34.png 4 of diamonds

....
40.png 10 of diamonds

So in the card array, 1 represents the ace of clubs, 2 the 2 of clubs,
31 the ace of diamonds, 40 the 10 of diamonds.

The images for the cards are on Canvas->Files->Assignments->Project2

Start with the the source code in:

Canvas->Files->Assignments->Project2->P2_starter.html

BUT replace our name in the title line with your name.

The text boxes are updated:

Computer/my score: computed value of current the hand.

[Details to be added]

Your web page must run on, and will be tested on,
Firefox and Chrome (not Internet Explorer).

Some JavaScript functions that you may need:

length	length of a string variable
Math.random	random number generation
Math.floor	integer floor of a number (e.g. floor of 2.1 is 2)
splice	take a slice of an array

Restrictions:

You must use the following data-structures. If you cannot, you need to talk to me well in advance, and be ready to explain "why not?".

card[] is an array of numbers, the numbers are indicators of the card. See above "Deck Format" **

Create three arrays "playerhand", "computerhand","discardpile" that contain the cards dealt to each, and the discard pile.

You should also use a variable "newcard" which is either the top discarded card (if selected) or the first card off the deck.

"Push" the player/computer's discarded card onto "discardpile".

Teams:

You have the option of working in teams of 2. Each team member must contribute equally to the project.

Search entries or author Unread Subscribe

Reply

(https://uk.instructure.com/courses/1898125/users/5249034)

Paul Linton

(https://uk.instructure.com/courses/1898125/users/5249034)

Tuesday

I went ahead and uploaded the JS examples files in the Files/Project 2 folder.

- Paul

Reply