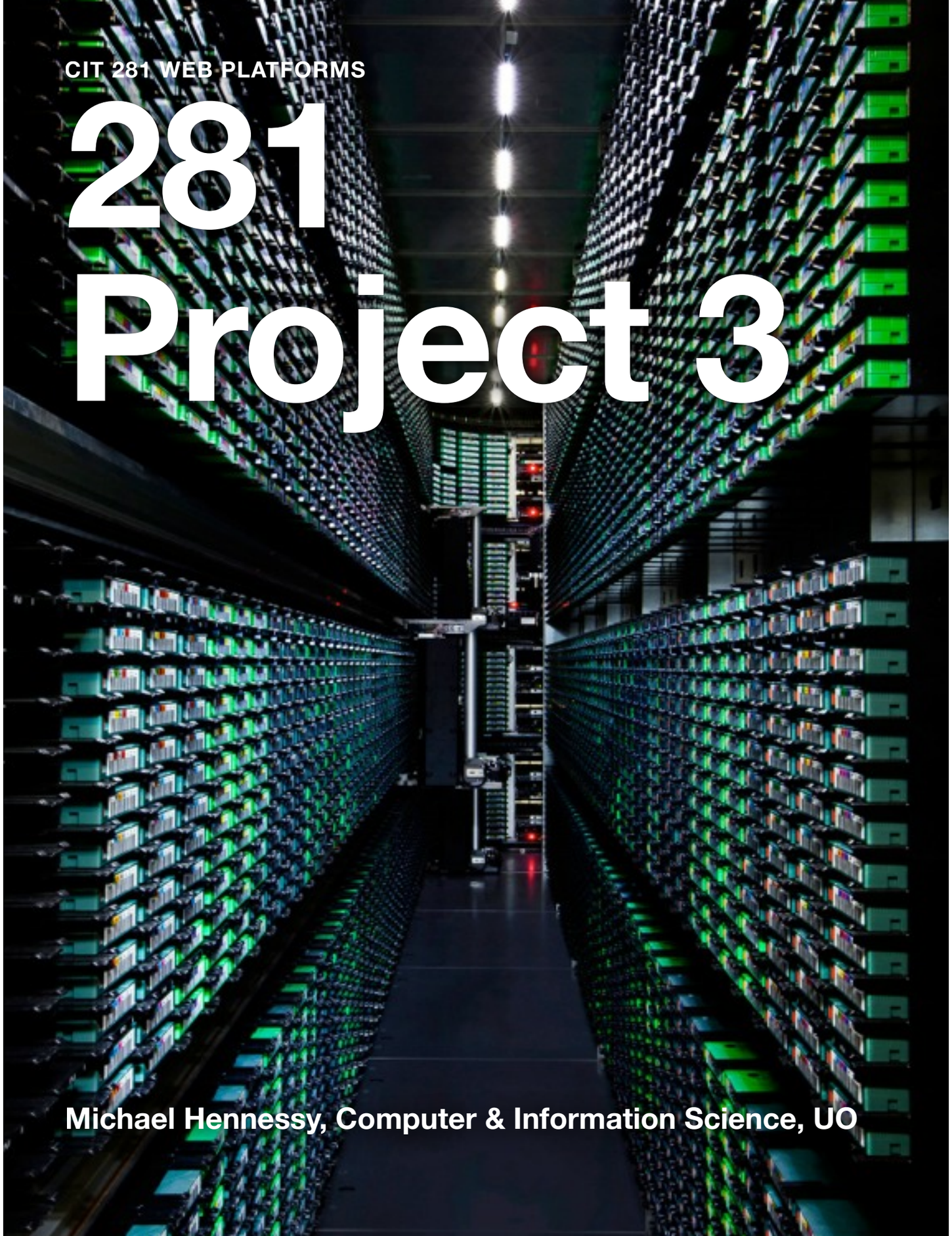


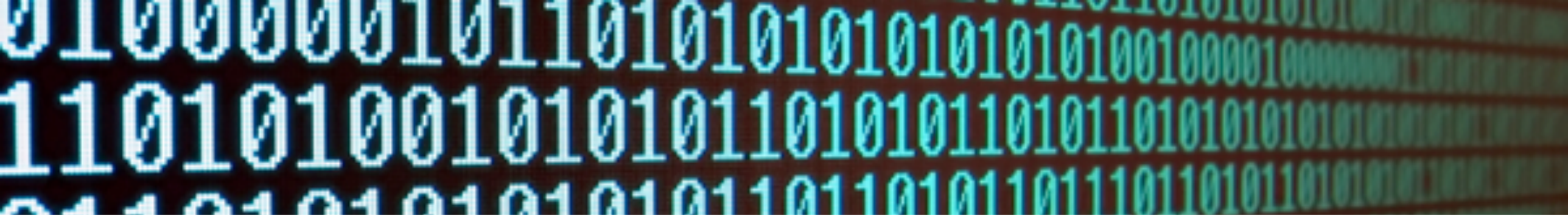
CIT 281 WEB PLATFORMS

281 Project 3

Michael Hennessy, Computer & Information Science, UO

CIT 281 The Web Platform

Project 3
Due 1700 M 8/11



Due: 1700 Mon 8/11.

PROJECT REQUIREMENTS

1. **[20 pts] 281/p3/roll-one-char.html, 281/js/roll-one-char.js, & roll-one-char.css**

   Learn how to use the Unicode Character Set to display special characters.

   Download random.js from dave-reed.com to 281/js/.
   Download the six dice images from dave-reed.com to 281/images/.
   Download roll-one-die.html, roll-one-die.js, and roll-one-die.css, from this Github Gist to 281/p3, 281/js/ and 281/css/, respectively.

   Preview the web app in Chrome. Debug, if necessary.

   Upload all files to your 281 website on the uoregon.edu server, and test the web app in Chrome. Debug, if necessary.

   Save roll-one-die.html as roll-one-char.html.
   Save roll-one-die.js as roll-one-char.js.
   Save roll-one-die.css as roll-one-char.css.

   Modify the web app to display a Unicode die face rather than an image. Note that the codepoints for die faces are the range x2680 .. x2685.

2. **[10 pts] 281/Chapter5/Example1/**

   Upload your ~/Documents/repos/Projects/Chapter5/Example1/ folder to your 281 website on uoregon.edu.

   Test the AJAX-powered web app in Chrome, without turning off Chrome's security restrictions. Debug, if necessary.

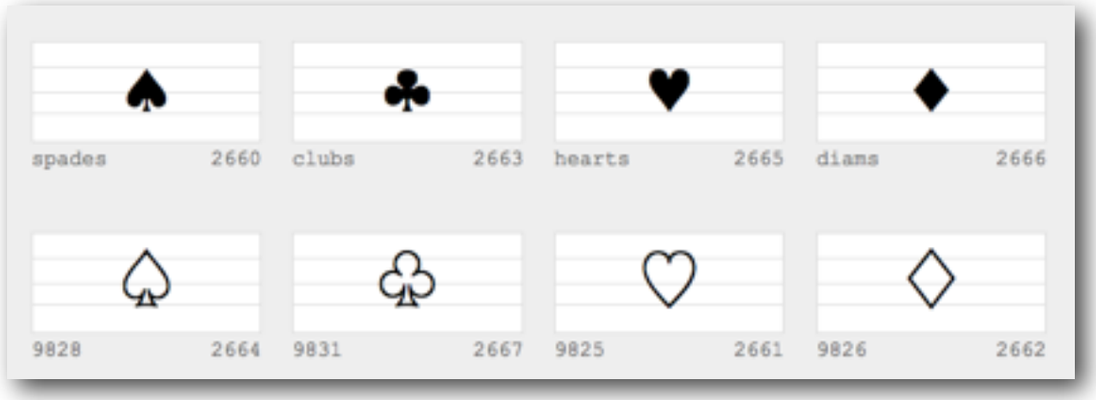
3. **[20 pts] 281/Chapter5/Example2/**.

   In this exercise, you will use JavaScript to display playing card glyphs (numbers and suits), instead of English words.

   Remember that JavaScript uses "\uFFFF" hex notation for displaying Unicode special characters.

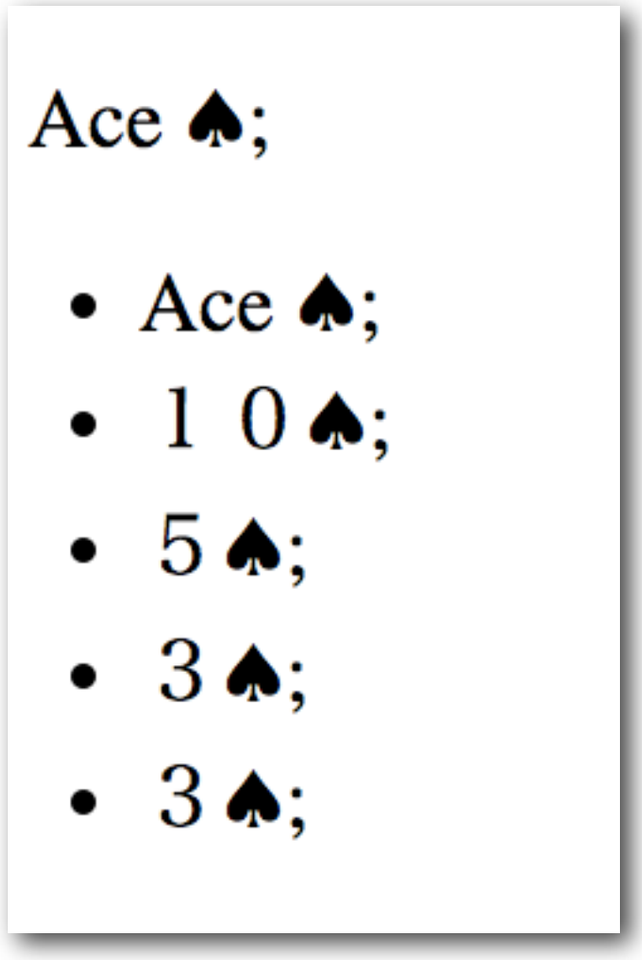
   To identify the codepoints for full Unicode full-width digits, view the full-width code chart.

   The following diagram shows the codepoints for playing-card symbols:



   To begin, duplicate your ~/Documents/repos/Projects/Chapter5/Example1/ folder, and name the copied folder Example2.

   Modify the web app to display glyphs rather than English words, as shown in this illustration:



   Your web app should be able to handle any arbitrary hand of cards described in the .json files:

Jack ♦;

- King ♣;
- Queen ♥;
- Jack ♠;
- 1 0 ♣;
- 9 ♦;

When complete upload your Example2 folder to 281/Chapter5/ on the uoregon.edu server, and test it.

4. **[20 pts] 281/Chapter5/Example3/**.

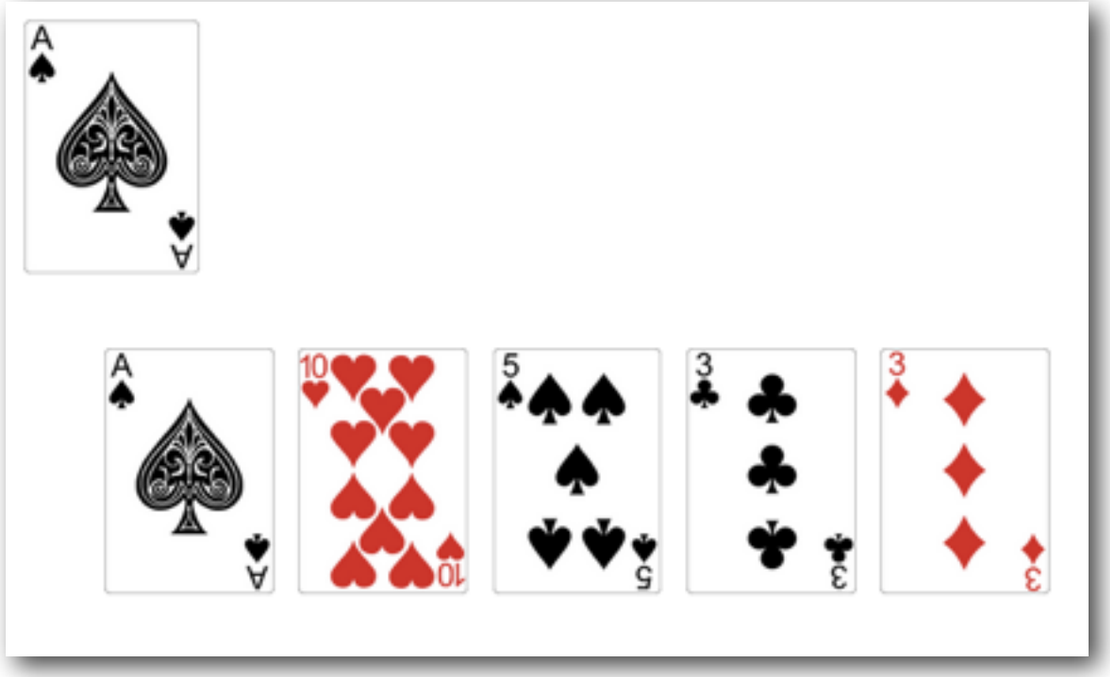
   In this exercise, we are back to using images.

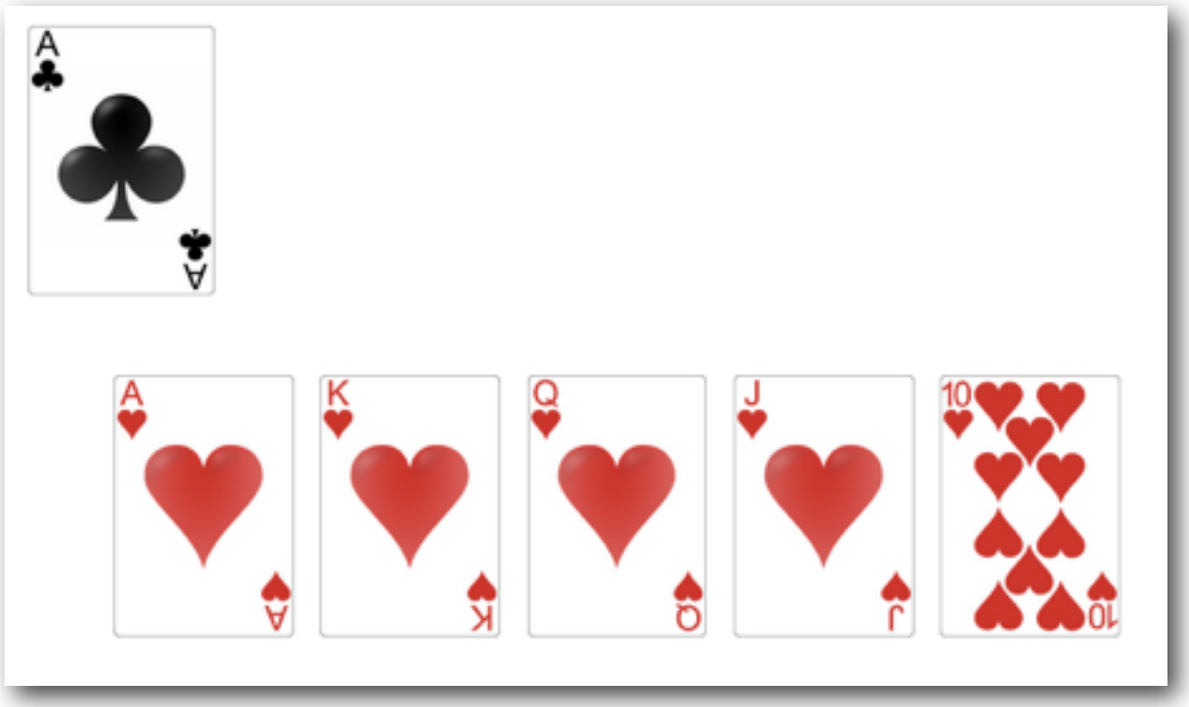
   To begin, duplicate your ~/Documents/repos/Projects/Chapter5/Example2/ folder (not Example1, and name the copied folder Example3.

Download Google Code's .zip file of playing card images (PNG format). Decompress the .zip file, and rename the folder as "cards".

Create a new folder named images in ~/Documents/repos/Projects/Chapter5/Example3/images/. Move the cards folder into the images folder.

Modify the web app to display playing card images rather than English words, as shown in these illustrations:





Preview in Chrome by disabling security restrictions. When correct, upload both web apps to uoregon.edu. Restart Chrome, and then test the web apps on the server. Debug, if necessary.

When your web app is complete, upload it to 281/Chapter5/Example3/ on uoregon.edu, and test it. Debug, if necessary.

5. **[30 pts] 281/Chapter5/Flickr/.**

   See the 281 schedule, week 3, for the two Flickr web apps.

   a) getFlicker-v1.html, app.js, and style.css.
   No changes are necessary.

   b) getFlicker-v2.html, app-v2.js, and style.css.
   Complete this web app by replacing the ??? with JavaScript.

Meeting the Deadline

How to Handle the 17:00 Deadline

- Start working on your project early. Do not delay.
- Turn in what you have by the deadline-- partial credit is better than none.

[Instructors] are a Superstitious Sect, Great Keepers of Set Times and Places.

-- from *Poor Richard's Almanac*