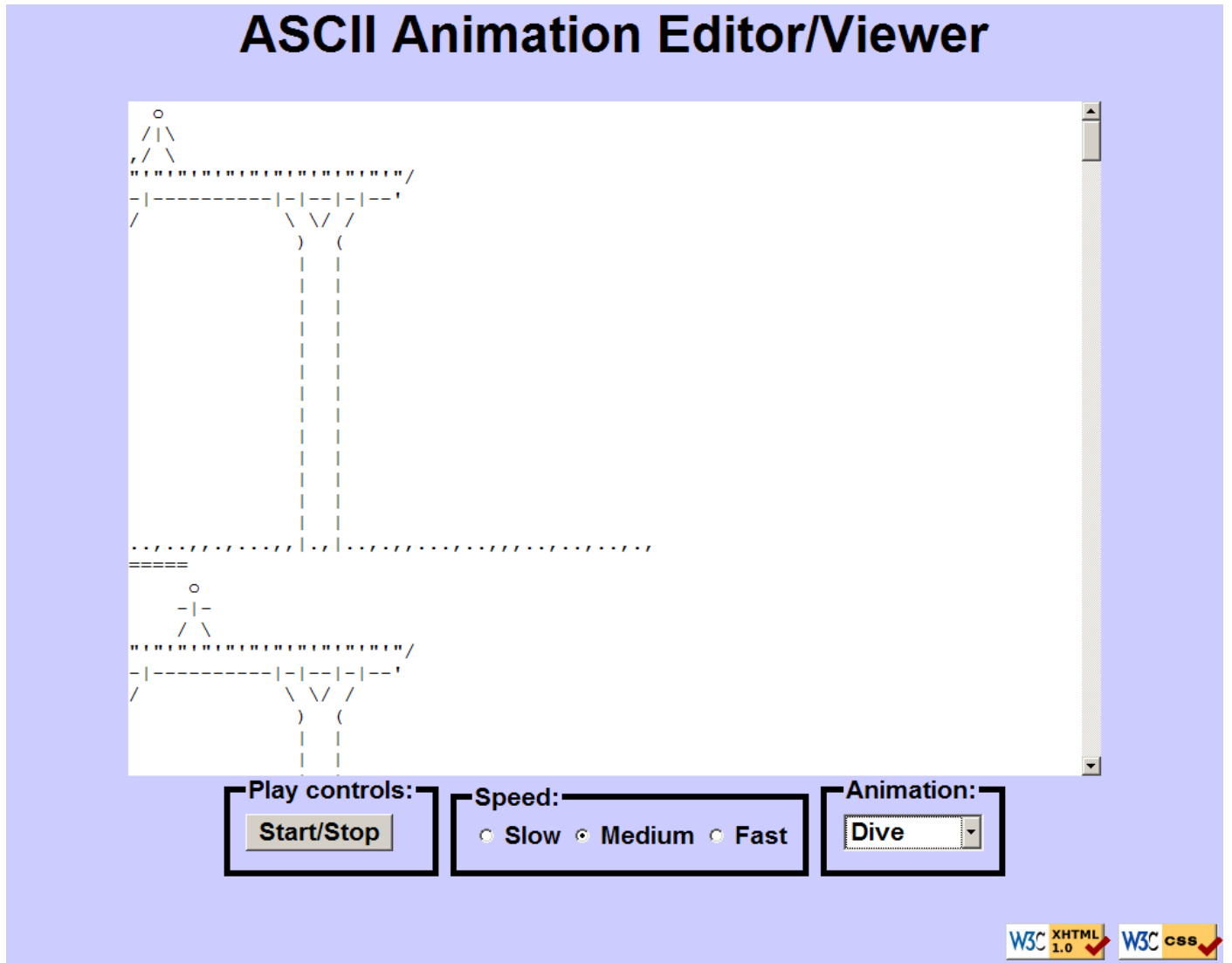


University of Washington, CSE 190 M, Spring 2007
Homework Assignment 3: ASCIImation
due Tuesday, April 17, 2007, 11:59pm electronically

Special thanks to Dave Reed for the idea of this nifty assignment.

This assignment tests your understanding of Javascript and its interaction with XHTML and CSS. You must match in appearance the following web page (between but not including the thick black lines):



Background Information:

[ASCII art](#) is the practice of drawing pictures that consist of text characters. ASCII art has a long history in computing as a way to draw pictures that could be printed by text-only printers or displayed on text-only computer monitors. One variation made possible on the web is animated ASCII art, or "ASCIImation." One group is even working to recreate the entire movies [Star Wars](#) and [The Matrix](#) as ASCIImations.

The main task for this assignment is to create a web page for viewing ASCIImations, stored in a file named `ascii.html`. This page must exactly match the appearance and behavior specified in this document.

The second part of your task is to create an ASCIImation of your own, stored in a file named

asciimation.txt. Your ASCIIimation must show non-trivial effort, must have multiple frames of animation, and must be entirely your own work. Be creative! We will put students' ASCIIimations on the course web site for voting in an anonymous contest. (No obscene animations, please.)

In total you will turn in the following files:

- `ascii.html`, your web page
- `ascii_style.css`, the style sheet for your web page
- `ascii_script.js`, the Javascript code for your web page
- `asciimation.txt`, your ASCII animation

Since the Javascript containing the animations is long, you may want to put this into a separate `.js` file.

Appearance Details:

The overall page has a background color of `CCCCFF`. The preferred font for text on the page is the default sans-serif font available on the system, in size 16pt.

The top of the page contains a heading in 32pt bold text, centered horizontally within the page. There is no gap between the heading content area and the top of the page.

Under the heading is a text box, positioned 10% from the top of the page and centered horizontally within the page. Its width is 80% of the page size and its height is 70% of the page size. It has no border and uses a 12pt monospaced font.

Below the text entry box is a set of controls. The controls are grouped into three boxes with 5px black borders and labels. The controls and their behavior are described in the Behavior section of this document.

The bottom of the page has a right-aligned paragraph containing a copyright notice along with two links to the W3C validators. These are the same images and links as used in the previous assignments. The area containing these images should always remain in the very bottom-right corner of the page, with no gap between its content and the edge of the browser window. The images should not have borders.

All other style elements on the page are subject to the preference of the web browser. The screenshots in this document were taken on Windows XP in Firefox 2.0, which may differ from your system. When the browser is resized, the percentages described above should remain constant. You may assume that the browser window is large enough to accommodate its contents.

The page's title text is ASCIIimation. The page should also have a "Favorites icon" of <http://www.cs.washington.edu/education/courses/cse190m/07sp/homework/3-asciiir>

Behavior Details:

The following are the groups of controls at the bottom of the page and each control's behavior:

- Play controls:
 - Start/Stop: When clicked, toggles the state of the page between animating and idle. When the page is idle, all frames of the animation are visible. Frames are separated by a line consisting entirely of 5 equals signs in a row: =====
When animation starts, the text currently in the text entry box is used to produce frames of animation. One frame is visible within the text entry box at any given moment, starting with the first frame. When animation is stopped, the text that was present in the box before animation began is returned to the box.
- Speed:
 - Contains three radio buttons. When a button is clicked, sets the speed of animation to the given rates:
 - Slow: 1 frames per second
 - Medium: 5 frames per second

- Fast: 20 frames per second

Initially the Medium speed is selected. Only one speed button can be selected at a time. The radio buttons always remain enabled regardless of whether the animation is currently playing. If the animation is playing and one of these buttons is clicked, the speed changes immediately.

- Animation:

A drop-down list of ASCII animations. When a choice is clicked, the text of the text entry box updates to show all frames of this animation. The following choices are available in the box:

- Blank
- Exercise
- Juggler
- Bike
- Dive
- Custom

Initially the Blank animation is selected and no text is showing in the text entry box. The drop-down list should be enabled when the page is idle but should be disabled while the animation is playing.

The Custom choice should show your own custom ASCII animation that you have created. The text of the other animations is available in the Homework section of the course web site.

Implementation and Grading:

Submit your assignment online from the turnin area of the course web site. Because the turnin system does not work well with .js files, you should package your files into a .zip archive and turn in this archive. For reference, our solution has 63 lines of HTML, 58 lines of CSS, and 82 lines of Javascript.

Implement the content of your web page using XHTML 1.0 Strict as taught in class. For full credit, your page must successfully pass the W3C XHTML 1.0 Strict validator. Choose appropriate HTML tags to match the structure of the content on the page. Add metadata tags describing the page's content type, author, and a description of its content. Do not express stylistic information in the HTML page itself, such as inline styles, CSS code in the HTML page's header, or presentational HTML tags such as `b` or `font`.

Express all stylistic information on the page in CSS using your style sheet file. For full credit, your style sheet must successfully pass the W3C CSS validator. You should not use HTML or CSS constructs that have not been discussed in lecture or the slides during the first three weeks of the course.

You should use absolute and fixed positioning in your style sheet to achieve the appearance specified. Please note that you must specify your HTML body to have a width and height of 100% for percentage-based absolute positions to work properly in Firefox.

Format your HTML and CSS so that they are as readable as possible, similarly to the examples shown in class. Also place a comment in each file containing your name and section.

Implement the behavior of each onscreen control using Javascript event handlers defined in your script file. For full credit, you should not embed Javascript code in your HTML file, other than to set event handlers.

Your Javascript code should follow reasonable stylistic guidelines similar to those you would follow on a CSE 14x programming assignment. In particular, you should avoid redundant code, minimize the number of global variables, utilize parameters and return values properly, correctly use indentation and spacing, and place a comment header on top of every function explaining that function's behavior.

Please do not place a solution to this assignment online on a publicly accessible (not password protected) web site. Doing so is a violation of the course academic integrity policy. Please see the course web site for directions on setting up password protection on your web space.

