

University of Arizona, CSc 337

Homework Assignment 9: Paint

Special thanks to Marty Stepp of Stanford and Jessica Miller

This assignment tests your understanding of using the HTML Canvas element. You must match the appearance and behavior of the following web page:



You will be creating a paint program. The user should be able to select different drawing tools, different sizes and different colors to create artwork.

No skeleton files are provided. Your page should link to a style sheet you'll write named [paint.css](#) for styling the page. After creating your page, you must make the UI interactive by writing JavaScript code in [paint.js](#) so that clicking the UI controls causes appropriate behavior. Your HTML page should link to your JS file in a `script` tag.

In total you will turn in the following files:

- [paint.html](#), your web page
- [paint.css](#), the style sheet for your web page
- [paint.js](#), the JavaScript code for your web page

Our screenshots were taken on Windows in Firefox, which may differ from your system.

Appearance Details:

The page should have a title of **CSC 337 Paint** centered horizontally, 36px sized with letter spacing of 3px. The overall page has a background color of #222222. The preferred font for all text on the page is Helvetica in white. If that is not available it should use the default sans-serif font available on the system.

Under the page's heading is a canvas with a white background, a width of 700px and a height of 500px and a 100px wide column of drawing controls. These are centered horizontally. The controls are divided into three sections: a color picker, pen size, and the mode buttons. The color picker and pen size sections should be 100px tall and have a 15px margin on all sides. The mode button section should have a margin of 15px, each mode button should have a margin of 5px and each mode button should be 80px wide.

Behavior Details:

When the page is opened the pen mode should be selected, a color of black and a pen size of 1.

Modes:

Pen: if the user clicks on the canvas, holds down the mouse and drags, a line should appear on the canvas in the mouse's path.

Circles: circles are drawn along the path of the mouse.

Squares: squares are drawn along the path of the mouse.

Lines: lines are drawn from the upper left hand corner of the canvas to where the mouse is.

Clear: the contents of the canvas are cleared leaving it blank and white.

In all modes, the color used should be the color last selected by the user and the size should be the size last selected by the user.

Pen Size:

The pen size controls contain a plus button that should make the pen size increase by one and a minus button that should make the pen size decrease by one. It also contains a 50px by 50px canvas that shows a black circle representing the pen size at its center. This circle should change size whenever the pen size changes.

Color Picker:

The top control is a color picker to allow the user to pick the pen color. This should be made using a 100px by 100px canvas.

To create the colors on the color picker canvas use two gradients. First create a gradient that shows colors red, blue, green, yellow, etc horizontally and then put another gradient on top that is partly transparent and goes from white to black vertically. The colored gradient should have the following stops:

| Place | red | green | blue |
|-------|-----|-------|------|
| 0 | 255 | 0 | 0 |
| 0.15 | 255 | 0 | 255 |
| 0.33 | 0 | 0 | 255 |
| 0.49 | 0 | 255 | 255 |
| 0.67 | 0 | 255 | 0 |
| 0.84 | 255 | 255 | 0 |
| 1 | 255 | 0 | 0 |

The black and white gradient should have the following stops:

| Place | red | green | blue | opacity |
|-------|-----|-------|------|---------|
| 0 | 255 | 255 | 255 | 1 |
| 0.5 | 255 | 255 | 255 | 0 |
| 0.5 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |

Development Strategy and Hints:

1. Write the basic **HTML** content including the proper UI controls.
2. Write your **CSS** code to achieve the proper layout.
3. Write the JS code for drawing a line when the user clicks and drags the mouse on the canvas
4. Add the size change options
5. Add the color change options
6. Add the other modes

Implementation and Grading:

Submit your assignment online to Gradescope. All of your HTML, CSS, and JavaScript code should follow the good style rules we have discussed in class.

Implement your page using **HTML5** as taught in class. Your page must pass the W3C HTML5 **validator**. Choose appropriate tags to match the structure of the page content. Do not express style information in the HTML page itself, such as inline styles or presentational HTML tags such as **b** or **font**. Use **unobtrusive JavaScript** so that no JavaScript code, **onclick** handlers, etc. are embedded into the HTML code.

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, slides, or textbook.

Format your HTML, CSS, and JS to be readable, like to the examples in class. Place a comment header atop each HTML/CSS/JS file. Your JavaScript should have more descriptive **comments**, including a header on each function (including anonymous) and complex sections of code describing the relevant code, the function's behavior, etc.

You should follow reasonable style guidelines similar to those of a CSc 110 programming assignment. In particular, avoid redundant code, and use parameters and return values properly.

You should use strict mode and have no global variables. Minimize the use of **module global variables**. Do not ever store DOM element objects, such as those returned by the **document.getElementById** function, as global variables. As a reference, our own solution has four global variables, mostly related to the set of frames to draw, which frame is currently displayed, the delay between frames, and so on.

Format your code similarly to the examples from class. Properly use whitespace and indentation. In your HTML, do not place more than one block element on a line or begin a block element past the 100th character. In your JavaScript, properly space/indent your code, and do not write any lines of code longer than 100 characters.

You should not use any external **JavaScript frameworks or libraries** such as jQuery to solve this assignment.

Do not place a solution to this assignment on a public web site