

Assignment 1: Painting (Easy)

1/19/2025

10/10 Points

Attempt 2



Review Feedback

1/19/2025

Attempt 2 Score:

10/10




Add Comment

Anonymous Grading: **no****Unlimited Attempts Allowed**

2/2/2025

▼ Details

Videos:

- Lab Section (Fall 21): [YouTube Playlist](https://www.youtube.com/playlist?list=PLbyTU_tFIkcNUOgEhJYtAERJwa7NTURVQ)  (https://www.youtube.com/playlist?list=PLbyTU_tFIkcNUOgEhJYtAERJwa7NTURVQ) || [Full Video](https://media.ucsc.edu/V/Video?v=3738321&node=12801874&a=286225505&autoplay=1) (<https://media.ucsc.edu/V/Video?v=3738321&node=12801874&a=286225505&autoplay=1>)
- Lab Section (Fall 20): [YouTube Playlist](https://www.youtube.com/watch?v=aNQrJTBxh6U&list=PLbyTU_tFIkcMLN0IkbfuHoJQgHZUOV0W)  (https://www.youtube.com/watch?v=aNQrJTBxh6U&list=PLbyTU_tFIkcMLN0IkbfuHoJQgHZUOV0W)
- Prof. James' Tutorial:
[YouTube Playlist](https://www.youtube.com/watch?v=rfyhLGeylGA&list=PLbyTU_tFIkcMK5FiV6btXxHQAy15p0j7X)  (https://www.youtube.com/watch?v=rfyhLGeylGA&list=PLbyTU_tFIkcMK5FiV6btXxHQAy15p0j7X)

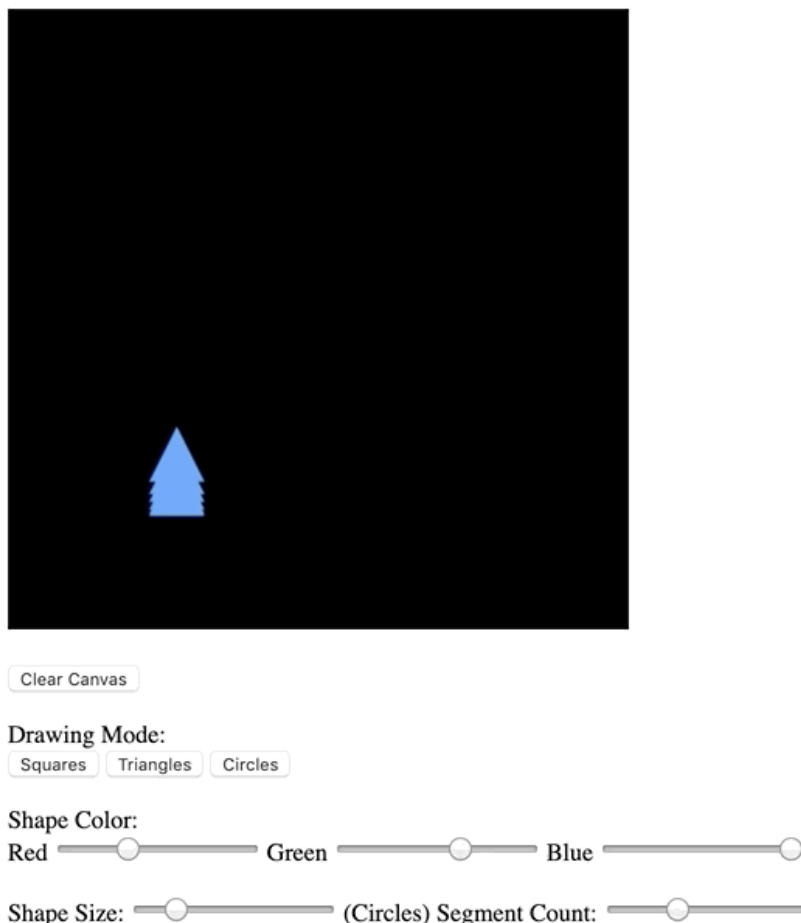
https://www.youtube.com/watch?v=rfyhLGeylGA&list=PLbyTU_tFIkcMK5FiV6btXxHQAy15p0j7X

Objectives:

To create a simple paint program that draws on the screen when you drag with your mouse.

Introduction:

You will create a web application to draw simple geometric shapes made out of points and triangles. Your application should look like the following example:



Basic features: Your application is required to have the following basic features. There are lots of tutorial videos on the *basic* assignment, and it builds directly from the book.

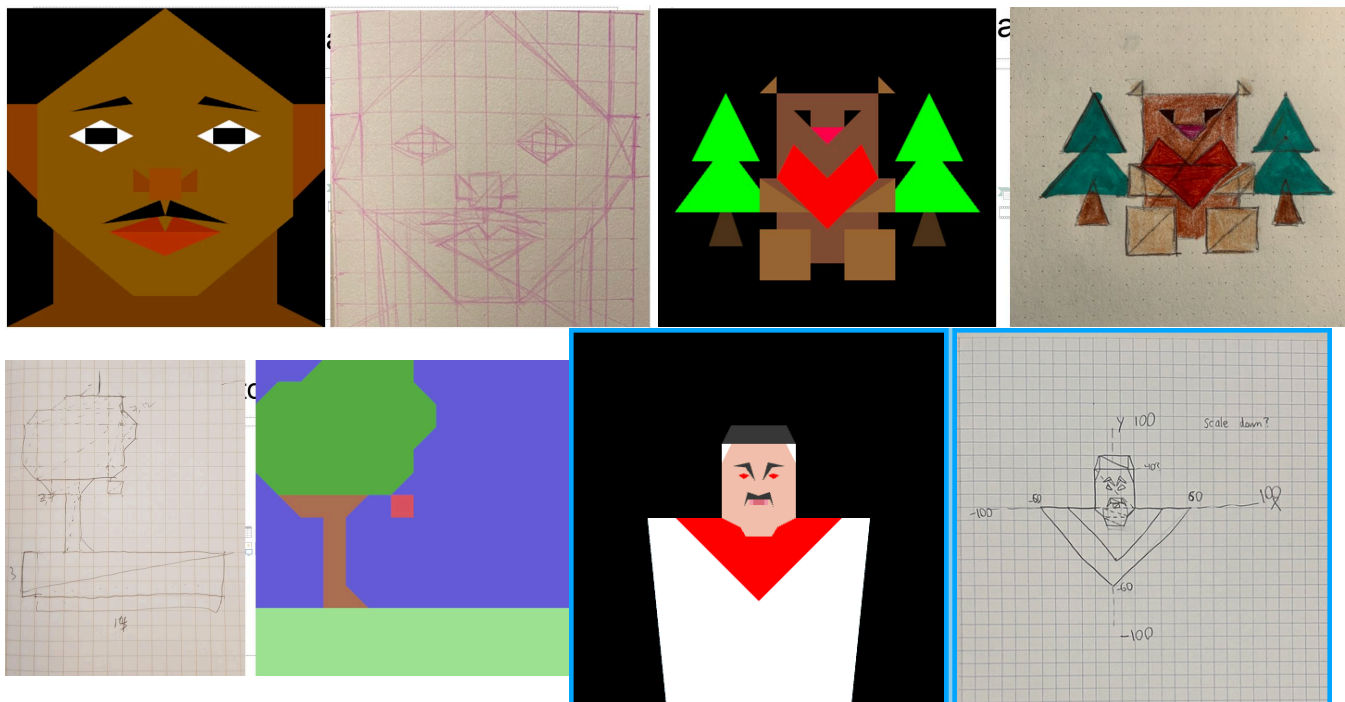
- A canvas where the user can click to draw simple geometric shapes.
 - When the user clicks on the canvas, a shape (square, triangle or circle) is drawn on the clicked position.
 - When the user clicks and drags the mouse on the canvas, shapes are drawn along the dragged direction.
- Buttons to choose between points, circle and triangle brushes.
- A slider to determine the number of segments used to create a circle.
- A button to clear the canvas.
- A slider to set the brush/shape size.
- Three sliders to control the shades of red, green and blue of the brush/shape color

The next part of the assignment you need to do something that does not have 'helper videos', and isn't directly given in the book. For this lab you will draw a picture out of colored triangles. (See the directions below).

Beyond the basics: In theory you now understand getting triangles on the screen, basic interaction, buttons, sliders. The next part ask you to keep applying this without explicit helper videos.

1. You will design a drawing on paper first, and draw it using triangles. Remember that you can put the vertices of the triangle wherever you want, and you have a function to draw a triangle from the basic assignment. It may be useful to draw axes for your sketch to determine where your triangles

will be drawn on the canvas. Don't just use the basic painting triangle shape, move those vertices around to where you need them. Here are some samples from past classes:



2. Do something awesome. This is very underdefined on purpose. Creativity is required. Grading is at the discretion of the grader and gets 0.0-1.5pts in this lab. Examples of the kinds of things that might make you awesome are below. Include a note on your webpage to tell us what is awesome if you are trying to get these points.

- A better paint program
 - A better brush shape that looks more like paint
 - Aligning the brush shape the the stroke direction
 - Filling in the gaps during mouse drag (Consider GL_LINES)
 - Learning about and using 'alpha' to get 'transparent' paint
- A simple game of your own design
 - e.g. your mouse motion controls a 'mouse', and there is a 'cat' that chases your mouse
 - or e.g. pong

Instructions:

The following will walk you through each point of the basic rubric. There are videos to demonstrate what to do. The following is a playlist with all videos:

https://www.youtube.com/playlist?list=PLbyTU_tFlkcMK5FiV6btXxHQAy15p0j7X 
 (https://www.youtube.com/playlist?list=PLbyTU_tFlkcMK5FiV6btXxHQAy15p0j7X)

1. Have an HTML Page with a Canvas to Draw on

- Read Matsuda through HelloPoint1.js (up to page 40)
- Watch Videos for 1.1
- Hooray! You got 0.5 pt with no new code. HelloPoint1.js is good enough.

2. Draw a shape when the mouse is clicked

- Read Matsuda through ColoredPoints.js (up to page 66 – Chap 2)
- Grab the example code for ColoredPoints.js
- Hooray! Another 0.5 pt with no new code.

3. Organize your code with specific functions for handling `setupWebGL()`, `connectVariablesToGLSL ()`, `handleClicks()`, and `renderAllShapes()`.

- It turns out the book code is organized for explanation in book form. However its not organized for building larger programs with 100s of lines of code. We want to organize it a bit, and make sure we understand all the function calls. This will save 75% of the bugs that you will have eventually
- Watch Videos for 1.3
 - These videos walk you through this code cleanup, starting from ColoredPoints.js. You need not do it exactly how I did it. You need not give the functions exactly the same names and parameters. If you prefer a different coding style, and don't like my way, that's fine. The point is to understand and clean up this book code to something which is more clear.
- Make explicit functions for the following, and put the appropriate code in them.
 - `setupWebGL()` – get the canvas and gl context
 - `connectVariablesToGLSL()` – compile the shader programs, attach the javascript variables to the GLSL variables
 - `renderAllShapes()` – based on some data structure that is holding all the information about what to draw, actually draw all the shapes.
 - `click()` – there should be a specific function that handles a click, but doesn't have extra code that doesn't belong there (like initialization and rendering)
 - Hooray another point! No new functionality. Just understanding and cleaning up the book example.

4. Have HTML sliders for choosing the RGB color to paint

- The book does not show how to make buttons and sliders. However this is standard HTML and javascript functionality. See the links to resources for this assignment.
- This point is meant to make sure you know how to connect UI elements from HTML to javascript. You can add more user interface elements as needed later.
 - Watch videos for 1.4

5. Have an HTML slider for choosing the shape size

- Watch videos 1.5
- Go to the vertex shader and make sure you know how to change the size of the point being drawn. Just hardcode to check what controls this.
- Add a slider to HTML.
- Add js code to handle the slider motion, saving the current selected size
- Add a uniform variable to the vertex shader for passing this data, and use it instead of the hard coded size.
- Add the js code necessary to connect js to the new uniform variable. `gl.getUniformLocation()`
- Add the code necessary to the `click()` function to save the size for each point in an array (same as `g_points` and `g_color`)

- Add the code to the `renderAllShapes()` function to set the GLSL uniform variable `gl.uniform1f()`, before drawing the point (`gl.drawArrays()`)

6. Have a single variable which contains the list of all shapes that need to be drawn. Have a class which contains the information for a Point.

- The book `ColoredPoints.js` example has `g_points` and `g_colors`. We're going to change this into a single variable. Many graphics packages have a 'scene graph'. We just need a simple list (`shapesList`).
- Make a class (object oriented) that stores the data needed for a shape. For the current points, perhaps the object is named `Point`. The object has a `Point.render()` function that knows how to draw itself.
- These are code cleanup steps that is going to save you much pain later.
- Watch Video 1.6

7. Have a button to clear the canvas

- Just add a button, and have it set `shapesList=[]` and then call `renderAllShapes()`
- Watch video 1.7

8. Draw a shape when the mouse is held down and there is mouse motion

- `canvas.onmousemove = click();` // In addition to the `onmousedown` you previously had.
- The 'event' has a field that specifies if the mouse button is down. You can use `ev.buttons==1` to check that.
- If you are having performance issues, check out `performance.now()` to get the current time so you can test the time a certain function (like rendering) is taking
- Also about performance, on some browsers, changing the call to get the gl context to be like the following will help.
 - `gl = canvas.getContext("webgl", { preserveDrawingBuffer: true});`
 - Without this change I start to lag at around 100 dots, with this change I start to lag at around 5000 dots.
 - Watch video 1.8

9. Have a button to draw triangles

- Watch videos for 1.9
- Go the `Chap3 HelloTriangle.js` and reorganize the code needed to make a triangle into a function `drawTriangle([x1, y1, x2, y2, x3, y3]);`
- Verify this function works by drawing several triangles at different locations
- Make a class `Triangle.js` file, and put it in your `ColoredPoints` project, but copy over the `drawTriangle()` function you made to make the new class work
- Add HTML buttons to choose point vs triangle
- Hook up the buttons to a variable to keep track of the state
- Change `click()` to create the correct kind of object

10. Have a button to draw circles

- Make a new class for `Circles`
- Set up the buttons, and variables to store `Circles` in your `shapesList`

- Design the render() function for circles
- Watch videos 1.A

11. Have a slider to determine the number of segments in the circle

- Add a slider and connections and variables
- Add a field to your Circle class to store the number of segments

12. Draw a picture

- No more helper videos (oh no!)
- Get a piece of graph paper and draw a picture using triangles. Use colored pencils, pens, crayons, or whatever. Use at least 20 triangles to make a detailed picture. (Enjoy being in grade school again!) (The top of this assignment page has examples)
- Take a picture of your drawing with your phone, and add this to your web page as a reference.
- Add a button to your webpage, so that when its clicked, computer graphics is used to recreate your drawing using WebGL triangles. (Make sure the painting features still work, we just want to add the picture when the button is clicked)
- You do not need to integrate your picture with your painting. The picture should stay on the screen, but you dont need to make this function put all the triangles in your painting point list (unless you want to).

13. Awesomeness!

- Impress us! Example ideas of a mini-game, or improving the paint program are given at the top of this page, but you are free to do something else. Just make sure to tell us what you did on the webpage so the grader knows what to look for.

Resources:

Readings:

- (WebGL) Needed for basic concepts of using WebGL Matsuda/Lea: Ch01, Ch02.
- (WebGL) The chapter that this assignment is built from: Ch03 (67-91).
- (HTML/Javascript) Mozilla's "Getting started with the web": https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web  (https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web)
- (HTML/Javascript) How to create HTML buttons: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/button>  (<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/button>)
- (HTML/Javascript) How to create HTML sliders: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/range>  (<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/range>)

(%24CANVAS COURSE REFERENCE%24/file_ref/gaa68d350951bf96b83681a987bb395f3/download)

What to Turn in:

(%24CANVAS COURSE REFERENCE%24/file_ref/gaa68d350951bf96b83681a987bb395f3/download)

1. Canvas Submission

Zip your entire project and submit it to Canvas under the appropriate assignment. Name your zip file "[FirstName]_[LastName]_Assignment_0.zip" (e.g. "Lucas_Ferreira_Assignment_0.zip").

2. Live Hosted Submission

You will upload your submission to GitHub Pages (or any other service of your choosing). If you use GitHub Pages, [click here \(https://canvas.ucsc.edu/courses/78931/pages/submission-guide?module_item_id=1490434\)](https://canvas.ucsc.edu/courses/78931/pages/submission-guide?module_item_id=1490434) to learn how to set it up.

WHEN SUBMITTING YOUR PROJECT ON CANVAS, PLACE YOUR SITE LINK AS A COMMENT OF THE SUBMISSION.

Read the [Submission Guide \(https://canvas.ucsc.edu/courses/78931/pages/submission-guide?module_item_id=1490434\)](https://canvas.ucsc.edu/courses/78931/pages/submission-guide?module_item_id=1490434) for further explanation on how to submit your assignment.

[\(%24CANVAS COURSE REFERENCE%24/file_ref/gaa68d350951bf96b83681a987bb395f3/download\)](https://canvas.ucsc.edu/courses/78931/pages/submission-guide?module_item_id=1490434)

✓ View Rubric

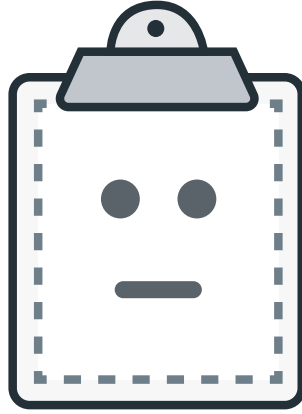
Select Grader

Harry Scapellati (TA)



ASG1 Rubric			
Criteria			Points
Have an HTML page with a canvas.	<div>Full Marks</div> <div>0.5 pts</div>	<div>No Marks</div> <div>0 pts</div>	0.5/0.5 pts
Draw a shape when the mouse is clicked	<div>Full Marks</div> <div>0.5 pts</div>	<div>No Marks</div> <div>0 pts</div>	0.5/0.5 pts
Organize your code with specific functions for handling setupWebGL(), connectVariablesToGLSL(), handleClicks(), and renderAllShapes().	<div>Full Marks</div> <div>0.5 pts</div>	<div>No Marks</div> <div>0 pts</div>	0.5/0.5 pts
Have HTML sliders for choosing the RGB color to paint	<div>Full Marks</div> <div>0.5 pts</div>	<div>No Marks</div> <div>0 pts</div>	0.5/0.5 pts

Have an HTML slider for choosing the shape size	<div>Full Marks</div> <div>0.5 pts</div>	<div>No Marks</div> <div>0 pts</div>	0.5/0.5 pts
Have a single variable which contains the list of all shapes that need to be drawn. Have a class which contains the information for a Point.	<div>Full Marks</div> <div>0.5 pts</div>	<div>No Marks</div> <div>0 pts</div>	0.5/0.5 pts
Have a button to clear the canvas	<div>Full Marks</div> <div>0.5 pts</div>	<div>No Marks</div> <div>0 pts</div>	0.5/0.5 pts
Draw a shape when the mouse is held down and there is mouse motion	<div>Full Marks</div> <div>1 pts</div>	<div>No Marks</div> <div>0 pts</div>	1/1 pts
Have a button to draw triangles view longer description	<div>Full Marks</div> <div>0.5 pts</div>	<div>No Marks</div> <div>0 pts</div>	0.5/0.5 pts
Have a button to draw circles	<div>Full Marks</div> <div>1 pts</div>	<div>No Marks</div> <div>0 pts</div>	1/1 pts
Have a slider to determine the number of segments in the circle	<div>Full Marks</div> <div>0.5 pts</div>	<div>No Marks</div> <div>0 pts</div>	0.5/0.5 pts
Awesomeness! view longer description	<div>Full Marks</div> <div>1.5 pts</div>	<div>No Marks</div> <div>0 pts</div>	1.5/1.5 pts
Place your site link as a comment of the submission.	<div>Full Marks</div> <div>0.5 pts</div>	<div>No Marks</div> <div>0 pts</div>	0.5/0.5 pts
Draw a picture view longer description	<div>Full Marks</div> <div>1.5 pts</div>	<div>No Marks</div> <div>0 pts</div>	1.5/1.5 pts



Preview Unavailable

Thomas_Wong_Assignment_1-1.zip

↓ [Download](#)

[https://canvas.ucsc.edu/files/10202580/download?
download_frd=1&verifier=ZgmXCtomlg2sNGLm3oc34vrbW9MFpfMFfri63FA9](https://canvas.ucsc.edu/files/10202580/download?download_frd=1&verifier=ZgmXCtomlg2sNGLm3oc34vrbW9MFpfMFfri63FA9)