

Assignment 0: Vector Library (Very Easy) S 4 Points Possible

1/11/2026

Attempt 1



In Progress

NEXT UP: Submit Assignment



Add Comment


Unlimited Attempts Allowed

1/11/2026

Details

Videos:

- Lab Section: [Full Video \(https://media.ucsc.edu/V/Video?v=3707844&node=12671712&a=1743617027&autoplay=1\)](https://media.ucsc.edu/V/Video?v=3707844&node=12671712&a=1743617027&autoplay=1) || [YouTube Video Playlist](https://www.youtube.com/playlist?list=PLbyTU_tFlkcNpIHMXN_G4sB0wjimJuRpz) 
(https://www.youtube.com/playlist?list=PLbyTU_tFlkcNpIHMXN_G4sB0wjimJuRpz)
- Lab Section (Fall 20): [YouTube Video Playlist](https://www.youtube.com/watch?v=CkbO7_jYAmM&list=PLbyTU_tFlkcOUaZ9kLznqF6Eyy4jUwgmS) 
(https://www.youtube.com/watch?v=CkbO7_jYAmM&list=PLbyTU_tFlkcOUaZ9kLznqF6Eyy4jUwgmS)

Note that the videos might mention using Atom, Sublime, etc. You're free to use whatever IDE you wish, but we recommend Visual Studio Code. See [here](https://ucsc-cse-160.github.io/docs/setup/)  (<https://ucsc-cse-160.github.io/docs/setup/>) for how to set up your local development environment, Github Pages, and Github repo.

Objectives:

Extend the matrix library provided by the textbook to support vector operations such as addition, subtraction, multiplication, cross product, dot product, etc. With this assignment, you will learn:

- How to create objected oriented graphics projects in Javascript.
- How to draw to a `<canvas>` element using a 2D context.
- Review fundamental concepts of Linear Algebra.

Introduction:

The textbook provides a matrix library called *cuon-matrix.js* which contains functions to create 4x4 matrices and operate with them. We will use this matrix later in this course for transforming (translate, rotate, scale) geometries. However, for now, we want to use this library for reviewing basic concepts of Linear Algebra as well as this library is limited, since it only supports *matrix x vector* operations, but does not support a full range of vector x vector operations, such as vector addition, multiplication, subtraction. Your job in this assignment is to extend the provided library to support most basic vector operations.

Reference Book: <https://sites.google.com/site/webglbook> 

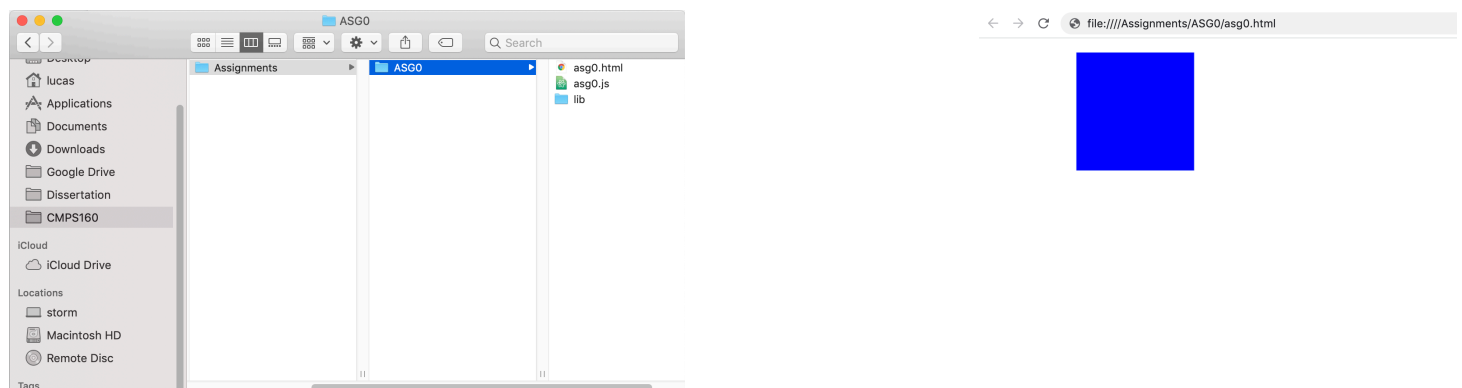
(<https://sites.google.com/site/webglbook>)

Instructions:

1. (1 point) Setup the DrawRectangle example from the book with our matrix library.

- Download the **DrawRectangle** (Listing 2.1, page 11) example from the book. This example is composed of two files: **DrawRectangle.html** and **DrawRectangle.js**.
- Rename these two files to **asg0.html** and **asg0.js**. In your new asg0.html, make sure to update the line that loads *DrawRectangle.js* to load asg0.js.
- Create a directory called **lib**.
- Download our modified matrix library [cuon-matrix-cse160.js](https://canvas.ucsc.edu/courses/88811/files/11924934/download) (<https://canvas.ucsc.edu/courses/88811/files/11924934/download>) inside the lib directory.
- In **asg0.html**, load this library . Hint: use the <script> html tag.

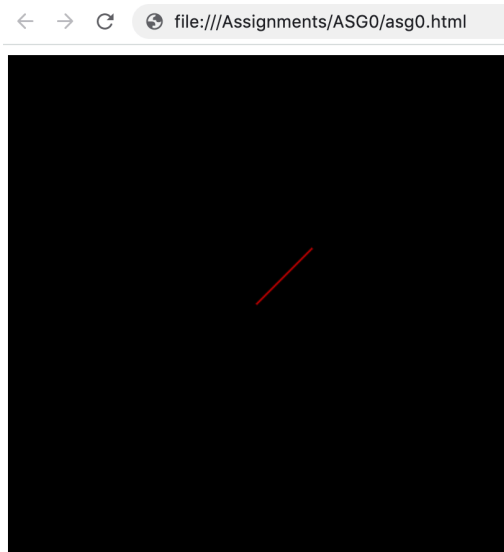
Expected result:



2. (1 point) Draw a red vector v1 on a black canvas instead of the blue rectangle. The origin of the vector should be the center of the canvas.

- In **asg0.js**, instantiate a vector v1 using the Vector3 class from cuon-matrix.js library (set the z coordinate to zero).
- In **asg0.js**, create a function drawVector(v, color) that takes a Vector3 v and a string color (e.g. "red"). Inside this function, use the builtin javascript function lineTo() to draw the vector v1. The resolution of the canvas is 400x400, so scale your v1 coordinates by 20 when drawing it. This will make it easier to visualize vectors with length 1.
- Call drawVector(v1, "red") in the main() function.

Expected result with $v1 = (2.25, 2.25)$:



3. (1 point) Add to your webpage an interface for the user to specify and draw the v1 vector.

- In **asg0.html**, use one `<input>` html element to specify the x coordinate and another one for the y coordinate. Use a third `<input>` html element for the draw button.
- In **asg0.js**, create a function named **handleDrawEvent()** that is called whenever a user clicks on the draw button. Inside `handleDrawEvent()`:
 1. Clear the canvas.
 2. Read the values of the text boxes to create v1.
 3. Call `drawVector(v1, "red")`.

Expected result:

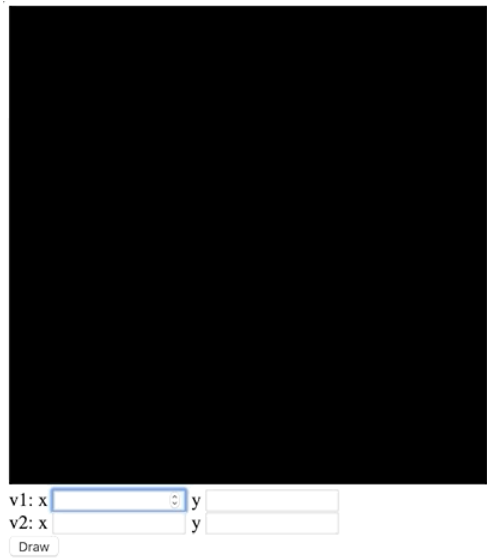


4. (2 points) Add to your webpage an interface for the user to specify and draw a second vector v2.

- In **asg0.html**, use one `<input>` html element to specify the x coordinate and another one for the y coordinate. Use the same v1 draw button to draw v2.
- In **asg0.js**, modify your **handleDrawEvent()** function to draw the vector v2 in blue:

1. Read the values of the text boxes to create $v2$.
2. Call `drawVector(v2, "blue")`.

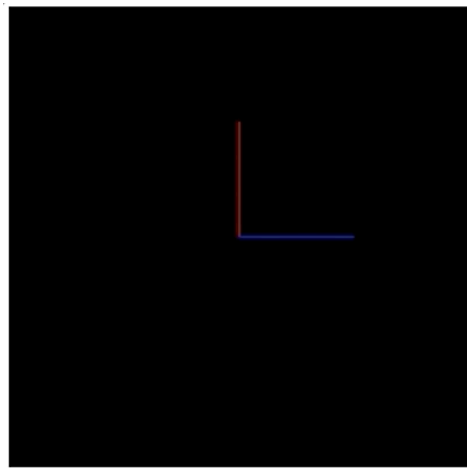
Expected result:



5. (2 points) Add to your webpage an interface for the user to perform and visualize the results of add, sub, div and mul operations.

- In `lib/cuon-matrix.js`, implement the functions **add**, **sub**, **div** and **mul** of the **Vector3** class.
- In `asg0.html`, use the html `<select>` element to add an operation selector. Use the html `<input>` element to add a number input textbox and. Use another the html `<input>` to add a draw button for the user to perform operations.
- In `asg0.js`, create a function named **handleDrawOperationEvent()** that is called whenever a user clicks on this second draw button. Inside **handleDrawOperationEvent()**:
 1. Clear the canvas.
 2. Read the values of the text boxes to create $v1$ and call `drawVector(v1, "red")`.
 3. Read the values of the text boxes to create $v2$ and call `drawVector(v2, "blue")`.
 4. Read the value of the selector and call the respective Vector3 function. For add and sub operations, draw a green vector $v3 = v1 + v2$ or $v3 = v1 - v2$. For mul and div operations, draw two green vectors $v3 = v1 * s$ and $v4 = v2 * s$.

Expected result:



v1: x 0 y 5
v2: x 5 y 0

Draw

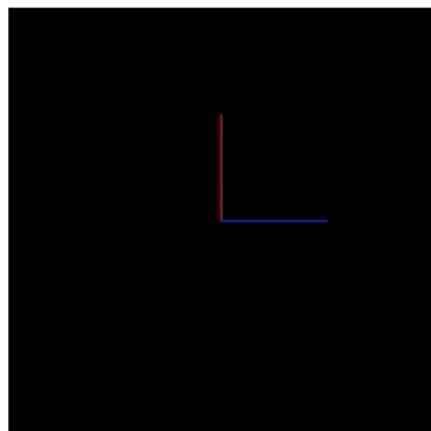
Operation: Add Scalar:

Draw

6. (1 points) Add to your webpage an interface for the user to perform and visualize the results of magnitude and normalize operations.

1. In **lib/cuon-matrix.js**, implement the **magnitude** function.
2. In **lib/cuon-matrix.js**, implement the **normalize** function using the **magnitude** function you just implemented.
3. In **asg0.html**, include these two options in your html selector.
4. Print the magnitude results of this operation to the console. Hint: use javascript builtin `console.log()` function and open your browser's console.
5. Draw normalized v1 and v2 in green.

Expected result:

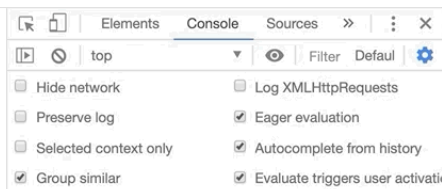


v1: x 0 y 5
v2: x 5 y 0

Draw

Operation: Add Scalar:

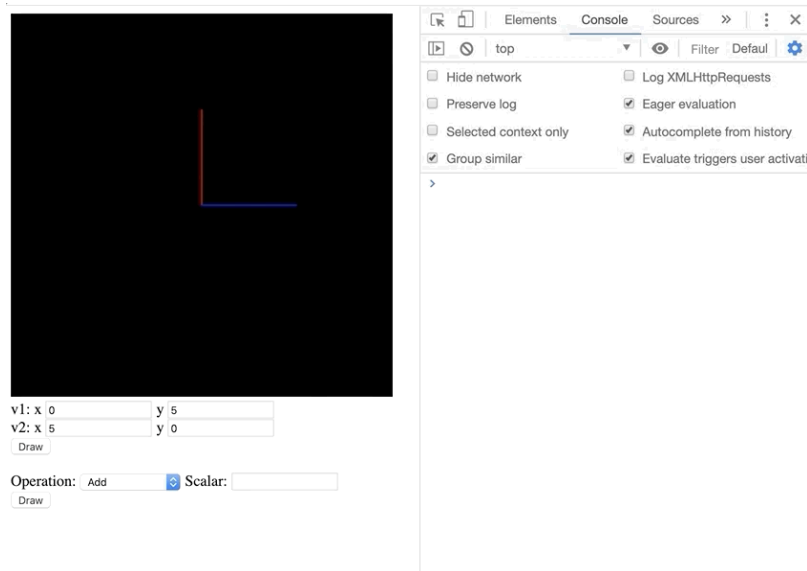
Draw



7. (1 points) Add to your webpage an interface for the user to calculate and visualize the angle between the vectors v1 and v2.

1. In **lib/cuon-matrix.js**, implement the static **dot** function.
2. In **asg0.html**, include the option "Angle between" in your html selector.
3. In **asg0.js**, write a function **angleBetween(v1, v2)** that uses the **dot** function to compute the angle between v1 and v2. Hint: Use the definition of dot product $\text{dot}(v1, v2) = \|v1\| * \|v2\| * \cos(\alpha)$.
4. Print the result of this operation to the browser console.

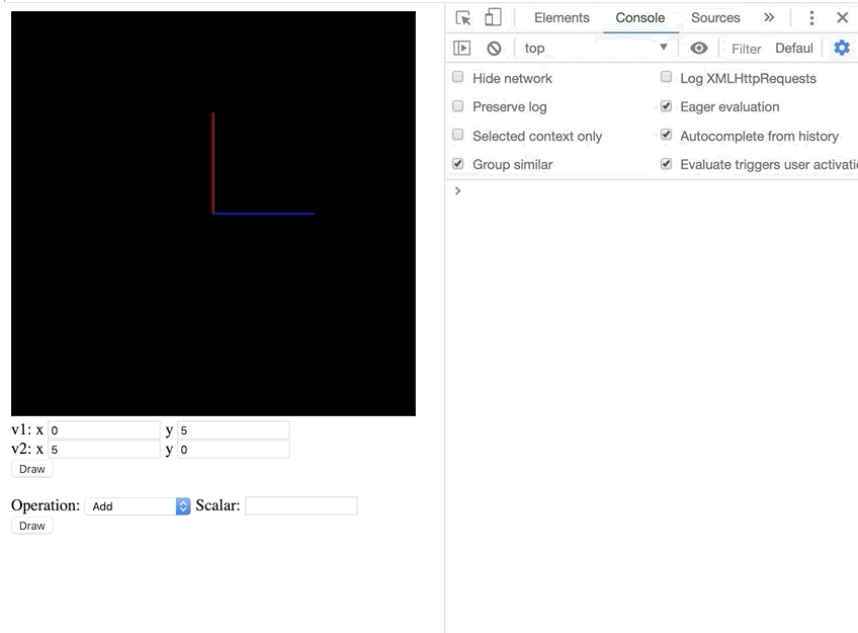
Expected result:



8. (1 points) Add to your webpage an interface for the user to calculate and area of the triangle formed by vectors v1 and v2.

1. In **lib/cuon-matrix.js**, implement the static **cross** function.
2. In **asg0.html**, include the option "Area" in your html selector.
3. In **asg0.js**, write a function **areaTriangle(v1, v2)** that uses the **cross** function to compute the area of the triangle created with v1 and v2. Hint: Remember $\|v1 \times v2\|$ equals to the area of the parallelogram that the vectors span.
4. Print the result of this operation to the browser console.

Expected result:



9. (recommended) Test your vector function implementations

1. Download [vectorTests.html](https://canvas.ucsc.edu/courses/88811/files/11924902?wrap=1) (https://canvas.ucsc.edu/courses/88811/files/11924902/download?download_frd=1) and place it in the same folder as **cuon-matrix-cse160.js**.
2. Open **vectorTests.html** which lists 9 tests stating either "Passed" or "Failed."

Resources:

- WebGL Matsuda/Lea: Ch01, Ch02 (pages 9-16).
- Eloquent Javascript (use as a reference to Javascript): <https://eloquentjavascript.net>
- HTML <input> element: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input>
- HTML <select> element : <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/select>
- How to create HTML buttons: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/button>

What to turn in:

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 \(https://canvas.ucsc.edu/courses/88811/pages/submission-guide?module_item_id=1869747\)](https://canvas.ucsc.edu/courses/88811/pages/submission-guide?module_item_id=1869747) 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 \(https://canvas.ucsc.edu/courses/88811/pages/submission-guide?module_item_id=1869747\)](https://canvas.ucsc.edu/courses/88811/pages/submission-guide?module_item_id=1869747) for further explanation on how to submit your assignment.

✓ View Rubric

ASG0 Rubric			
Criteria	Ratings		Points
Setup the DrawRectangle example from the book with our matrix library. view longer description	Full Marks 0.5 pts	No Marks 0 pts	/0.5 pts
Draw a red vector v1 on a black canvas instead of the blue rectangle. The origin of the vector should be the center of the canvas.	Full Marks 0.5 pts	No Marks 0 pts	/0.5 pts
Add to your webpage an interface for the user to specify and draw the v1 vector.	Full Marks 0.5 pts	No Marks 0 pts	/0.5 pts
Add to your webpage an interface for the user to specify and draw a second vector v2.	Full Marks 0.5 pts	No Marks 0 pts	/0.5 pts
Add to your webpage an interface for the user to perform and visualize the results of add, sub, div and mul operations.	Full Marks 0.5 pts	No Marks 0 pts	/0.5 pts

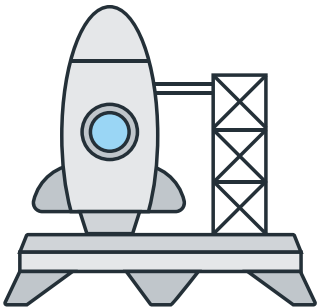
Criteria	Ratings		Points
Add to your webpage an interface for the user to perform and visualize the results of magnitude and normalize operations.	Full Marks 0.5 pts	No Marks 0 pts	/0.5 pts
Add to your webpage an interface for the user to calculate and visualize the angle between the vectors v1 and v2.	Full Marks 0.5 pts	No Marks 0 pts	/0.5 pts
Add to your webpage an interface for the user to calculate and area of the triangle formed by vectors v1 and v2.	Full Marks 0.5 pts	No Marks 0 pts	/0.5 pts
Place your site link as a comment of the submission. view longer description	Full Marks 0 pts	No Marks 0 pts	/0 pts

Choose a submission type

Upload

⋮

More



Choose a file to upload

or

 Webcam Photo

 Canvas Files

Submit Assignment