

# Assignment 1: 3D Geometry

## Description

Write a single-page, responsive web application to render various 3D geometries (cone, cylinder, sphere and torus) to a <canvas> and to manipulate their properties using native HTML, CSS, JavaScript and the given WebGL library. Your application should use the full device width and height and render properly in landscape or portrait mode on a mobile device. You should also dynamically add the corresponding UI components to the page when a user selects a 3D geometry from the combo box.

### Landscape Mode (width > height)



### Portrait Mode (width < height)



## General Requirements

- Must include your name, date and short description at the top of each file
- Must use your own HTML/CSS/JavaScript (no third-party libs are allowed)
- Must be unobtrusive (no inline CSS and JavaScript)
- Must be validated (no errors or warnings in HTML/CSS/JavaScript)
- Must be responsive (use full device width & height)
- Must use CSS flex layout and media query for the responsive design
- Must dynamically include the associated HTML and JavaScript files into the main page when a 3D geometry selected by a user
- Must use addEventListener() to register all user events
- Must import Smal.js, webgl.js and Logger.js into your HTML page

## References

Note 01, test\_cube.html

**NOTE:** test\_cube.html contains the base HTML/JS/CSS codes to begin with. Modify it and add your own codes to it.

## Deliverables

An archive file, **Assignment1\_<yourname>.zip**, which contains all your files (HTML, CSS and JavaScript). It should be runnable on any system after download it.

**NOTE:** You must include the file header at the beginning of each file. The header must contain a short description, your name, email, date, etc.

## Submission and Due Date

Submit your deliverables to SLATE/Assignments/Assignment1 by **Sunday, May 26, 11:59 PM**.

You may submit multiple versions, but only the latest version will be evaluated.

**NOTE:** Late submission will be deducted 10% per day. (max. 3 days)

**NOTE:** Partial implementation will be accepted.

**NOTE:** This assignment is individual work and subject to Sheridan Academic Integrity Policy.

## Tasks & Evaluation (Total 100 points)

### Task 1: index.html, main.js & main.css (20 points)

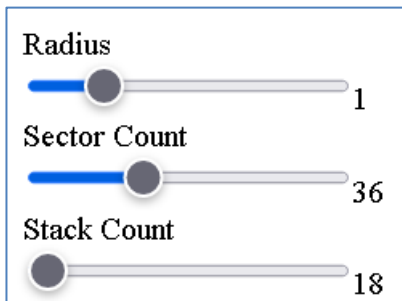
1. Implement **index.html**
  - a. Construct the static DOM elements and a placeholder for the dynamic content
  - b. <canvas> element for the 3D geometry
  - c. The name of the 3D geometry at the top-left corner
  - d. <select> element to choose a geometry with 4 options; cone, cylinder, sphere & torus
  - e. Common UI components; 3 checkboxes ("*Show texture*" is checked by default) and a button
  - f. Copyright block at the bottom-right corner, e.g. © 2024, [Song Ahn](#)
  - g. A placeholder for the dynamic HTML content, e.g. <div id="includeblock">
2. Implement **main.css**
  - a. Dynamically resize the whole page to fit in any mobile device
  - b. Display 2 columns horizontally in the landscape mode
  - c. Display 2 columns vertically in the portrait mode
3. Implement **main.js**
  - a. Implement the event handlers for the <select>, checkboxes and button
  - b. Implement a function to include a HTML & JavaScript files dynamically
  - c. Use the following constructors to create 3D geometries
 

```

          = new Smal.Sphere(gl, radius, sectors, stacks, gl.smooth)
          = new Smal.Cone(gl, baseR, height, sectors, stacks, gl.smooth)
          = new Smal.Cylinder(gl, baseR, topR, height, sectors, stacks, gl.smooth)
          = new Smal.Torus(gl, majorR, minorR, sectors, stacks, gl.smooth)
          
```

### Task 2: sphere.html & sphere.js (20 points)

1. Implement the sphere specific UI components and event handlers
2. It is a partial HTML block, not complete HTML page (do not include <html>, <body>, etc.)
3. Use IIFE to execute the JavaScript code immediately after loaded

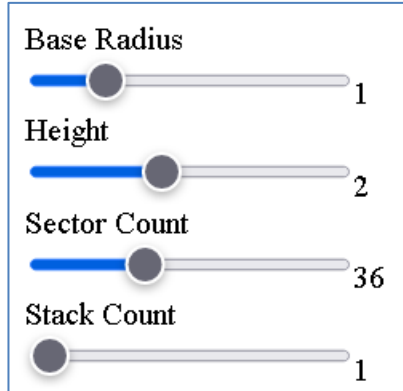


Parameter	Default value	Range (min ~ max)	Step
Radius	1	0.1 ~ 5	0.1
Sector Count	36	3 ~ 100	1
Stack Count	18	2 ~ 100	1

Reference: [http://www.songho.ca/opengl/gl\\_sphere.html#webgl\\_sphere](http://www.songho.ca/opengl/gl_sphere.html#webgl_sphere)

**Task 3: cone.html & cone.js (20 points)**

1. Implement the sphere specific UI components and event handlers
2. It is a partial HTML block, not complete HTML page (do not include <html>, <body>, etc.)
3. Use IIFE to execute the JavaScript code immediately after loaded

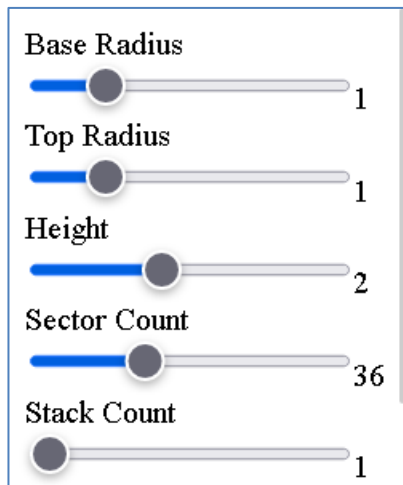


Parameter	Default value	Range (min ~ max)	Step
Base Radius	1	0.1 ~ 5	0.1
Height	2	0 ~ 5	0.1
Sector Count	36	3 ~ 100	1
Stack Count	1	1 ~ 100	1

Reference: [http://www.songho.ca/opengl/gl\\_cone.html#webgl\\_cone](http://www.songho.ca/opengl/gl_cone.html#webgl_cone)

**Task 4: cylinder.html & cylinder.js (20 points)**

1. Implement the cylinder specific UI components and event handlers
2. It is a partial HTML block, not complete HTML page (do not include <html>, <body>, etc.)
3. Use IIFE to execute the JavaScript code immediately after loaded

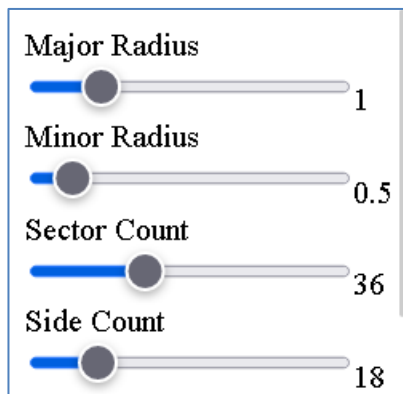


Parameter	Default value	Range (min ~ max)	Step
Base Radius	1	0.1 ~ 5	0.1
Top Radius	1	0.1 ~ 5	0.1
Height	2	0 ~ 5	0.1
Sector Count	36	3 ~ 100	1
Stack Count	1	1 ~ 100	1

Reference: [http://www.songho.ca/opengl/gl\\_cylinder.html#webgl\\_cylinder](http://www.songho.ca/opengl/gl_cylinder.html#webgl_cylinder)

**Task 4: torus.html & torus.js (20 points)**

1. Implement the cylinder specific UI components and event handlers
2. It is a partial HTML block, not complete HTML page (do not include <html>, <body>, etc.)
3. Use IIFE to execute the JavaScript code immediately after loaded

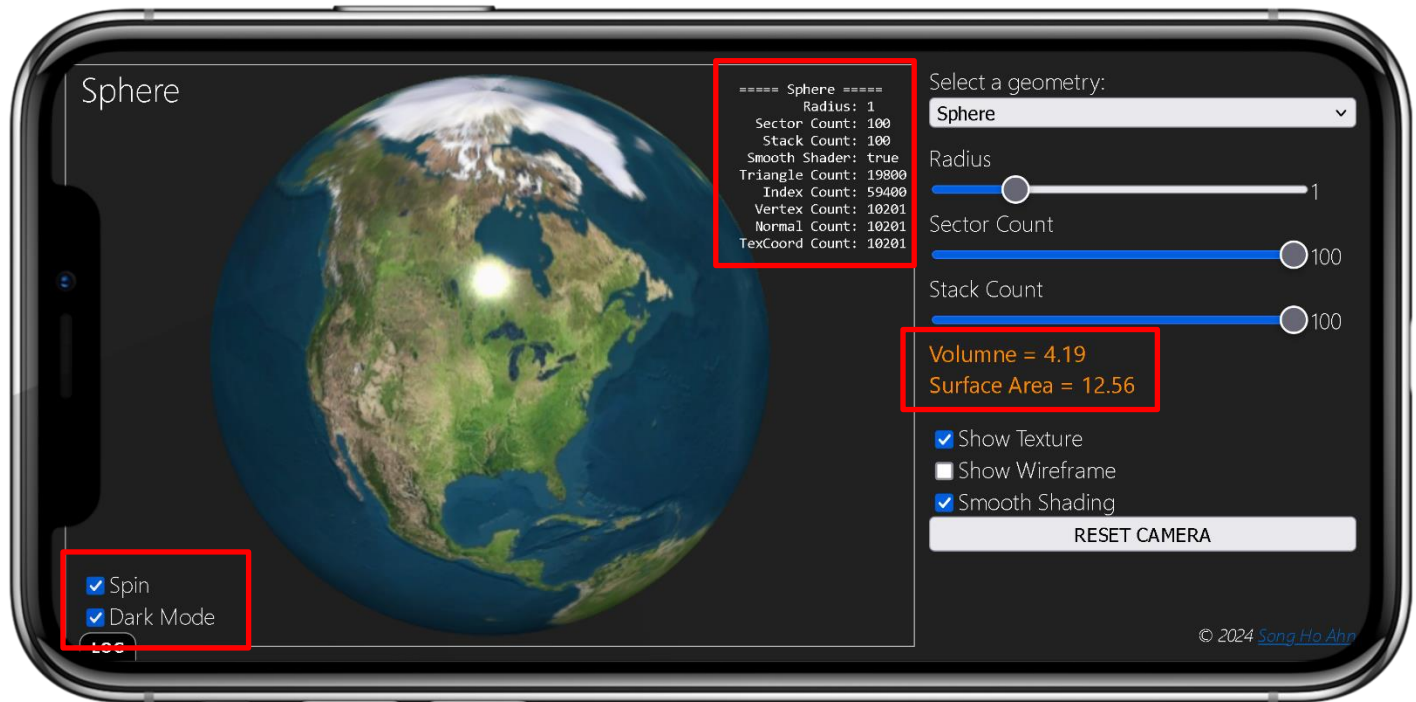


Parameter	Default value	Range (min ~ max)	Step
Major Radius	1	0.1 ~ 5	0.1
Minor Radius	0.5	0.1 ~ 5	0.1
Sector Count	36	3 ~ 100	1
Side Count	18	3 ~ 100	1

Reference: [http://www.songho.ca/opengl/gl\\_torus.html#webgl\\_torus](http://www.songho.ca/opengl/gl_torus.html#webgl_torus)

**Bonus (20 points max)**

- Display the information of the geometry (5 points)  
**HINT:** Use `gl.model.toString()` to get the geometry info as string
- Toggle camera rotation animation (5 points)  
**HINT:** Use `enableSpin()` & `disableSpin()` (See the line# 674 of `webgl.js`)
- Toggle the dark mode (5 points)
- Compute & display the volume & surface area of the geometry (5 points)
- Add additional features/styles to enhance your application (5 points)

**Q & A (Extra Notes)**

**Q1. How to change the column layout vertically in the landscape mode (width < height)?**

**A1.** Use **max-aspect-ratio** media query and **flex-direction** style.

```
/* CSS media query */
@media (max-aspect-ratio: 1/1) {
  .flexblock {
    flex-direction: column;
  }
  .rightblock {
    flex-basis: auto;
  }
}
```

**Q2. How to dynamically include HTML and JavaScript to the main page?**

```
////////////////////////////////////
// dynamically add html and js
// PARAMS: domId: the container DOM ID
//          html : URL to html file
//          js   : URL to js file
```

```
function includeHtml(domId, html, js)
{
    fetch(html)
    .then(response => response.text())
    .then(text =>
    {
        // success
        console.log("loaded HTML: " + html);
        let dom = document.getElementById(domId);
        dom.innerHTML = text;

        // load JS after html is loaded
        loadJavaScript(js);

        // resize canvas manually because page was updated
        handleResize();
    })
    .catch(e =>
    {
        // failed
        console.log("Failed to load " + html);
    });
}

function loadJavaScript(file)
{
    if(!file)
        return;

    let jsId = btoa(file); // encode to base64 string
    let script = document.createElement("script");
    script.id = jsId;

    // add script to dom
    if(!document.getElementById("jsId"))
        document.head.appendChild(script);

    // callback onload
    script.onload = () =>
    {
        // parsing JS is completed, safe to execute it here
        console.log("loaded JS: " + file);
    };

    // callback onerror
    script.onerror = e => console.log(e);

    // start to load
    script.src = file;
}
```