

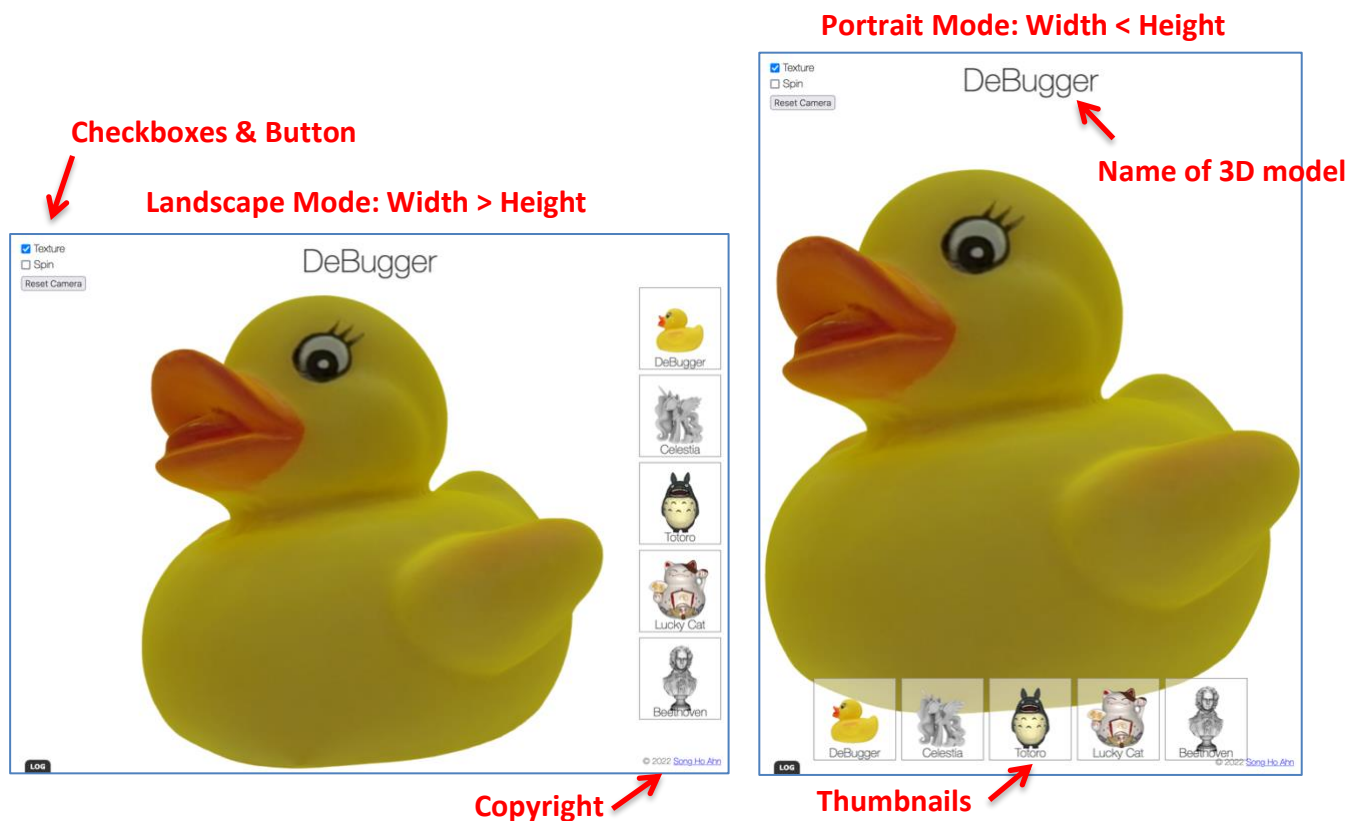
Assignment 1: Responsive DeBuggers

Description

Write a web application to display a list of 3D models and interact with them using your own JavaScript and CSS styles. The application must be responsive; the sizes of thumbnails and font are dynamically resized based on the device dimension, and the thumbnails are positioned vertically in landscape mode and horizontally in portrait mode.

The thumbnail images and 3D media (models/textures/JS) are provided. The thumbnail images are available at SLATE/Assignment1 folder. And, 3D media and the base JavaScript files are available remotely (Use external links to reference them);

<http://ejd.songho.ca/syst24444/>



General Requirements

- Must be responsive width, height, font size, etc. (Must use the full device width and height)
- Must be unobtrusive (no inline CSS and JavaScript)
- Must define a separate JavaScript file for the main driver (e.g. main.js)
- Must define a separate CSS file to define styles (e.g. main.css)
- Must use your own CSS and JavaScript (no third-party libraries or frameworks)
- Must be validated (no errors or warnings in HTML, CSS & JavaScript)
- Must use flex CSS styles for the container of thumbnails
- Must use HTML elements for the static contents, title, checkboxes, thumbnails, etc.
- Must display the selected 3D model when a thumbnail is clicked (DeBugger is default)
- Must toggle on/off displaying texture and spinning 3D models (texture on, spinning off by default)
- Must use `addEventListener()` to register user events (Do not use "onclick" property)
- Must import **Smal.js**, **webgl.js** and **Logger.js** into your HTML page

References

Note 01, 02, 03, Lab 01, 02, 03

Deliverables

An archive file, **Assignment1-<yourname>.zip**, which contains all the files (HTML, CSS, JavaScript).

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 **Thursday, Jun 02, 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 and Evaluations (Total 100 points)

Task 1: HTML (30 points)

1. Construct static DOM elements and placeholders in HTML page
2. A block **<canvas>** for 3D content using full device width/height
3. A block for title (name of 3D model)
4. A block for the copyright, e.g.: © 2022 [Song Ho Ahn](http://ejd.songho.ca) (Replace with your name)
5. A Block for controls (texture on/off, spin on/off and reset camera)
6. Blocks for thumbnails (each thumbnail contains an icon and caption)
7. The name of 3D models are “DeBugger”, “Celestia”, “Totoro”, “Lucky Cat” and “Beethoven” in order.
8. Import **Smal.js**, **webgl.js** and **Logger.js** within **<script>** from <http://ejd.songho.ca/syst24444/>

Task 2: CSS (40 points)

1. Create a separate CSS file and define the styles for the DOM elements
2. The content page uses the full device width and height
3. The title is centered at the top of the page
4. The copyright text is positioned at the bottom-right corner
5. The control block is positioned at the top-left corner
6. Use a **flex** layout for the container block of the thumbnails
 - a. The thumbnail container is vertically centered at the right corner if width > height
 - b. The thumbnail container is horizontally centered at the bottom if width < height
 - c. Each thumbnail has the background image (available at SLATE) and caption for the 3D model
 - d. The caption for each thumbnail is horizontally centered at the bottom of the thumbnail
 - e. Each thumbnail has mouse hover effect; changing mouse pointer and change background colour
 - f. The width/height of each thumbnail is **15vh** if w > h, and **15vw** if w < h
 - g. Each thumbnail is responsive; dynamically resize width/height and font size based on the device width/height

HINT: Use CSS **transform** with [translate\(\)](#) function to position a block centered

e.g.: `transform: translate(-50%, 0); /* move left 50% of its width */`

HINT: Use [calc\(\)](#) function to resize the font size based on the device width

e.g.: `font-size: calc(1em + 2vw); /* base + variant */`

HINT: Use **max-aspect-ratio** media query to determine landscape /portrait modes

e.g.: `@media (max-aspect-ratio:1/1) { ... } /* portrait mode, w/h < 1 */`

Task 3: JavaScript (30 points)

1. Invoke `initWebGL("your_canvas_ID")` when the DOM content is loaded, "DOMContentLoaded" event. This function will load and display "DeBugger" 3D model into your <canvas> block.
2. Add event handlers for controls; toggle ON/OFF texture and spinning, and reset the camera
3. Must use `addEventListener()` to register the user events (Do NOT use "onclick" property)
4. Must initialize the texture checkbox ON, spin checkbox OFF by default when the page is loaded
5. Invoke `toggleTexture(flag)` to enable/disable texture
6. Invoke `toggleSpin(flag)` to enable/disable spinning
7. Invoke `resetCamera()` to reset camera transformations
8. Add event handler for each thumbnail to load the selected 3D model and to change the title
9. Invoke `loadModelByName(name)` to change the 3D model. The available model names are "DeBugger", "Celestia", "Totoro", "Lucky Cat" and "Beethoven".
10. Use efficient and declarative codes wherever possible

HINT: You can put all your JavaScript codes inside `main()` function

```
// main entry point
document.addEventListener("DOMContentLoaded", main);
function main()
{
    // put your code inside main()
    initWebGL(...);
    ...
}
```

Bonus (20 points max)

- Add additional CSS style for visual enhancements (5 points)
- Add "keydown" and "keyup" event handlers (5 points)
- Add a separate help/description page/overlay (5 points)
- Add additional feature to improve functionality (5 points)
- Other JavaScript functions you may use for the bonus implementation:
 1. Shift Left : `startShiftCamera(1,0)`
 2. Shift Right : `startShiftCamera(-1,0)`
 3. Shift Up : `startShiftCamera(0,1)`
 4. Shift Down : `startShiftCamera(0,-1)`
 5. Zoom In : `startZoomCamera(1)`
 6. Zoom Out : `startZoomCamera(-1)`

Q & A**Q1. How to set a <div> with 100% device height?**

Set the height property to **100vh** and line-height property to 0.

```
body { margin:0; padding:0; }
.fullblock { height:100vh; line-height:0; }
```

Q2. How to get the model name when a thumbnail is clicked?

You can attach a custom data to a <div> using "data-*".

```
<div class="thumb" data-name="DeBugger">
  
  <div class="caption">DeBugger</div>
</div>
```

```
// in JS
let name = thumb.dataset.name; // OR, thumb.getAttribute("data-name")
```