Final Project, Part I

Due Feb 22 by 11:59pm **Points** 100 **Submitting** a file upload **File Types** zip and rar
Available Feb 6 at 10am - Feb 26 at 6am 20 days

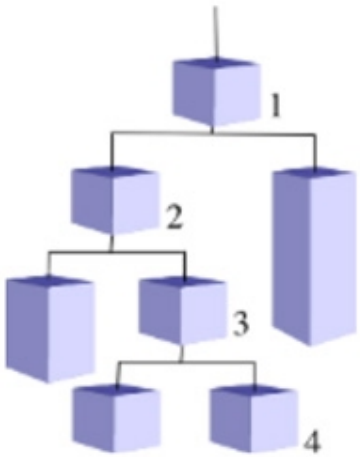
This assignment was locked Feb 26 at 6am.

Demo Video (https://video.wpi.edu/Watch/Tw96DaMr) (no audio)

Optional sphere.ply file: **sphere.ply**

Overview:

In this project, we will explore hierarchical modeling of 3D meshes. We will also explore lighting and shading. The goal of this part of the assignment is to familiarize yourself with transformations and hierarchical modeling in WebGL using a matrix stack. You will be modeling and animating a kinetic sculpture such as Alexander Calder's hanging mobiles. Example images and an example video are given below:



First Steps:

Your task is to model and animate a kinetic sculpture. In order to do so, you need to first construct a scene containing several objects, then you need to organize them hierarchically and simulate their motions according to the hierarchy you define. All models should be properly sized so that they do not look disproportionate or display awkwardly. All models should have approximately the same size. In the video, you should have noticed multiple axes of movement for each model, where the model may rotate about it's own axis while rotating around a larger arm off which it is hanging. Implement this effect!! To draw the arms connecting the meshes, you can just draw horizontal lines, and to connect the meshes to the ends of the arm, just draw vertical lines. Also, make sure you set up perspective projection

Gouraud Lighting (smooth shading): Draw all your meshes in your hierarchy as solid meshes (not wireframes). Implement shaders for Gouraud lighting (as demonstrated in class) and flat shading. You will switch between shading types using keystrokes below. Apply some interesting colors and material properties to your 3D meshes. Don't forget to enable depth testing and hidden surface removal.

Implement a spotlight: The spotlight is like a cone, and light only reaches objects inside the cone. Objects outside a certain angle--called the cutoff angle, phi--gets no light from the spotlight (although it may still receive ambient light from the environment). Make the cutoff angle a variable that can be controlled with a keystroke. Make sure the presence of the spotlight is obvious!

Matrix stack: You will need to implement a matrix stack, including stack.push() and stack.pop() routines. You may use a simple linked list to store the stack or use a more sophisticated tree structure.

Animation: Be mindful of performance issues while you implement this homework. As your scene size grows, avoid unnecessary re-initialization, file reopening, etc. that could slow down your program considerably.

Other Requirements: Your sculpture must consist of at least 6 meshes with interesting colors, including at least three cubes and at least three spheres. Each mesh should have a different color. Your sculpture must contain at least 3 levels of hierarchy. (See the second image at the top of the page.) The animation should move all parts of the sculpture. It should be physically motivated and make sense; that is, there are no magical, free-flying shapes or disjointed movements. Objects should remain connected in whatever way you connected them as you animate them.

All mesh models should rotate counter-clockwise about their own Y axis. The arms should rotate in the direction opposite of the level above it. For instance, if the arm of Level 1 of the hierarchy rotates clockwise, the arm of level 2 of the hierarchy should rotate counter-clockwise, the arm of level 3 should rotate clockwise, and so on. (You can have the level 1 arm rotate counter-clockwise and alternate from there, too.) Basically, the arms of consecutive levels of the hierarchy should alternate between rotating clockwise and counter-clockwise.

Additional Keystrokes:

Also implement the following additional keystrokes.

- **(User hits 'p'):** Increase spotlight cut off angle (increase cone angle)
- **(User hits 'P'):** Decrease spotlight cut off angle (decrease cone angle)
- **(User hits 'm'):** The scene is shaded using Gouraud lighting (smooth shading)
- **(User hits 'M'):** The scene is shaded using flat shading

Note that these keys are case-sensitive. If you do not want to worry about case-sensitivity, you are also welcome to implement "i" instead of "P" for "decrease spotlight cut off angle" and "n" instead of "M" for "the scene is shaded using flat shading.

Extra Credit:

For this project, you are allowed to include additional features above and beyond the basic requirements. The instructor has sole discretion over the point value of each feature, and the student may earn up to 10 points total. To earn any credit, *you must list each feature in the comments at the top of your main *.js file.*

To be eligible for extra credit, the additional features must be graphical in nature. This means that you show thought behind how something is presented to the user on screen or how the graphics are being processed behind the scenes. In addition, the additional features *may not be features you will implement as part of the base requirements for Part II of this project.*

Here are some example features to consider:

- Instead of basic geometric shapes, make some of the mobile models your .ply files from Project 2.
- Implement Phong instead of Gouraud shading. This means you will be implementing your shading on a per-pixel basis in the fragment shader.
- Implement a keystroke that draws a box around each mesh showing the extents of each mesh.
- Have the arms of the hierarchy follow a gentle sinusoid. i.e. in addition to going round clockwise or counterclockwise, the hierarchy arms (and objects) also go slightly up and down like on a merry go round. The sinusoidal movements of all levels of the hierarchy should have the same amplitude (i.e. move up and down gently by the same amount).

You are not limited to these ideas. If there is something else you would like to do but are not sure if it would qualify for extra credit, please contact me.

Submitting Your Work:

Make sure to double-check that everything works before submitting. Put all of your files (JavaScript, HTML, etc.) into a folder and zip it. Please include your *.ply files for ease of grading. Upload it to Canvas. Do not email me your program or submit it via a third-party cloud storage account.

Create documentation for your program and submit it along with the project inside the zip file. Your documentation can be either a pure ASCII text or a Microsoft Word file. The documentation does not have to be long. Briefly describe the structure of your program and what each file turned in contains. Also, please indicate whether your program uses .ply files, and please describe the rough location and direction of the spotlight in your scene. Name your zip file according to the convention *LastnameFirstname_Final1.zip*.

Additional Notes:

- You are free to consult any resources you need to help you complete this assignment, including your classmates, the TA, the instructor, the class text, and any other books and websites. However, the code you turn in must be your own. ***Any evidence of plagiarism will result in an automatic 0 for this assignment.***
- You are welcome to use any class coding examples (posted under "Modules") in your program.

FAQ:

Q. I'm trying to load the sphere.ply file, but the browser won't let me load the file unless a user selects it. What do I do?

The sphere.ply file is optional. I encourage you to create spheres using subdivision surfaces as shown in one of the class examples. However, if you want to use the *.ply file, it's perfectly fine to keep the file upload box that you used in Project 2. Just be sure to indicate in your accompanying documentation that the user needs to upload the accompanying *.ply file to start the program. Also, please include the *.ply file.

Q: The wrong faces on some of my meshes are being culled and/or my meshes are being lit incorrectly.

It's possible that the face normals are being calculated incorrectly. If the shaders receive the vertex information in the wrong order, it's possible they may be calculating a face normal that's pointing in the wrong direction, making the culling or lighting routines think the polygon is back-facing instead of front-facing (and vice-versa for the back-facing polygons).

Try reversing your vertex order for each polygon in your sphere, then see what happens. If that doesn't work, try changing the way you load the vertices and normals into the arrays from a,b,c to a,c,b.

If it's a lighting issue, also make sure that the light position of your point light has a 1 in the fourth position (the w position). This makes it a point light. A light with w = 0 is an area light, which has different properties. Check out the section "Positioning Directional Lights" on this page for more details: **https://tinyurl.com/yckyqmmg** **(https://tinyurl.com/yckyqmmg)** .

Final Project, Part I Rubric

Criteria	Ratings		Pts
Objects (3 pts each) - At least six objects are on the mobile, not including hangers. - Objects are either read in from a *.ply file or constructed in the program. - All objects are properly sized, similar in size, and are not disproportionate or awkward in display. - Each object is a distinct color. - Back-face culling and depth testing implemented.	**15.0 pts** **Full Marks**	**0.0 pts** **No Marks**	15.0 pts
Hierarchical Modeling (4 pts each) - Program uses hierarchical modeling to maintain parent-child relationships. - Children automatically inherit the animations of their parents. - Stack with push() and pop() used to maintain these relationships. - At least 3 levels of hierarchy are present. - Objects remain connected throughout the course of the animation.	**20.0 pts** **Full Marks**	**0.0 pts** **No Marks**	20.0 pts
Animation (5 pts each) - Each object rotates counter-clockwise about its own y-axis. - Rotation is visible. - The arms rotate in the direction opposite the level above them. - Program is not considerably slowed down as a result of coding inefficiencies.	**20.0 pts** **Full Marks**	**0.0 pts** **No Marks**	20.0 pts
Lighting (10 pts) - Both Gouraud and flat shading are implemented. - Spotlight is implemented and is obvious.	**20.0 pts** **Full Marks**	**0.0 pts** **No Marks**	20.0 pts
Keyboard All keyboard commands implemented correctly.	**15.0 pts** **Full Marks**	**0.0 pts** **No Marks**	15.0 pts
Documentation and Organization (5 pts each) - Code is clean, well-commented, and easy to read - Documentation file describing structure of program and what it contains. File also indicates whether the program uses .ply files, and it describe the rough location and direction of the spotlight in the scene.	**10.0 pts** **Full Marks**	**0.0 pts** **At least one of documentation file and code comments is missing**	10.0 pts
Extra Credit	**0.0 pts** **Full Marks**	**0.0 pts** **No Marks**	0.0 pts
Late Submission	**0.0 pts** **Full Marks**	**0.0 pts** **No Marks**	0.0 pts
			Total Points: 100.0