Assignment 2: Blocky 3D Animal (Medium)

2/2/2025

10 Points Possible

Attempt 1

2/2/2025

NEXT UP: Review Feedback

Attempt 1 Score:
N/A

Add Comment

Unlimited Attempts Allowed

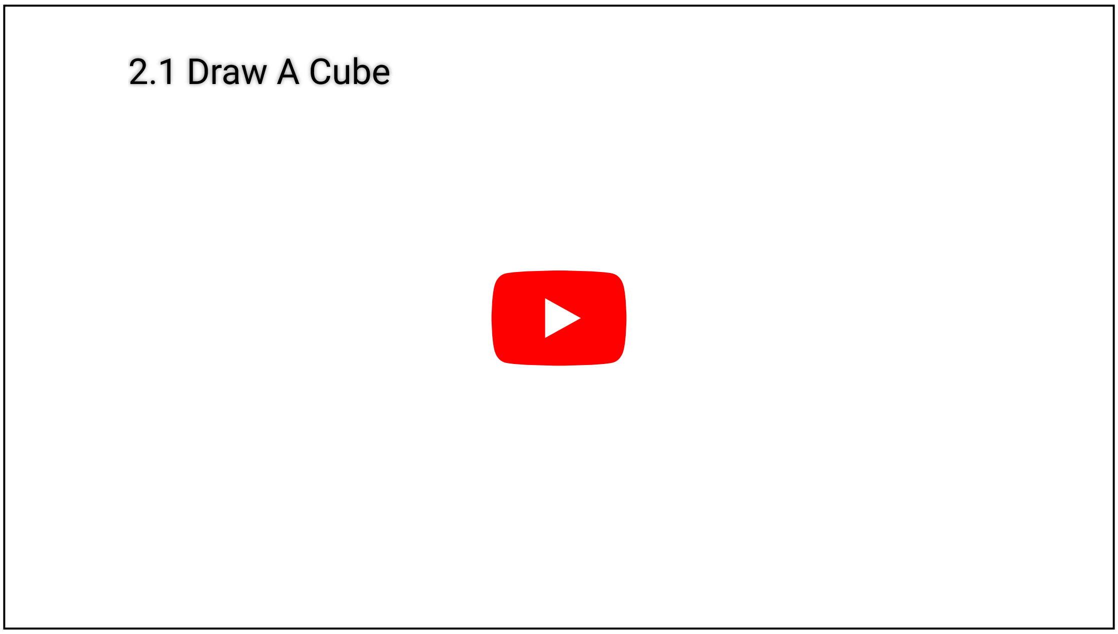
2/16/2025

Details

Videos:

These videos may be helpful, but they are not required.

- Prof. James' Tutorial: **YouTube Playlist (https://www.youtube.com/watch?v=8xgejUPmGgo&list=PLbyTU_tFIkcPmW6JknG_z9jdWDpJt7DYS)**
-
- Videos from TA intro to Lab in past quarters
  - Lab Section (Fall 2021): **Full Video (Week3) (https://media.ucsc.edu/V/Video?v=3767678&node=12883126&a=345868255&autoplay=1)** || **Full Video (Week 4) (https://media.ucsc.edu/V/Video?v=3798366&node=13002713&a=535111528&autoplay=1)** || **YouTube Playlist (https://youtube.com/playlist?list=PLbyTU_tFIkcOKv804fCDxzetIwjHIP8Rg)**
  - Lab Section (Fall 2020): **YouTube Playlist (https://www.youtube.com/watch?v=iASTFJ9OErg&list=PLbyTU_tFIkcP_TKnSbT03YjUt5r88Q791)**



Objectives:

To transform (translation, rotation, scaling, etc) simple 2D geometric shapes and 3D objects using matrices.

Introduction:

See this video for a sample blocky animal. You don't need to make one this complex.

https://www.youtube.com/watch?v=ka8prwshnuo (https://www.youtube.com/watch?v=ka8prwshnuo)

(https://www.youtube.com/watch?v=ka8prwshnuo)

The basic assignment: Create a simple 3D animal. You will create this out of simple geometric objects, like cubes, spheres, and cylinders. The animal must have at least 8 parts, e.g. body, upper legs (x4), lower legs(x4), head OR whatever. The sample above has 20+ parts. There needs to be at least one chain of parts that extends at least two elements away from the body. For example, there should be a bend at the knee so the leg has two parts. At least one part should animate. There should be a way (slider or mouse) to rotate the object and see the animal from all sides. There should be slider control of at least two joints in a chain.

Hard stuff for max points is described in bullet 11 below: Color, full animation, third level joint, non-cube primitive, second animation, mouse rotation, performance.

Instructions:

Here are the steps (which match the rubric).

1. First get a function working that can drawCube(), This function should build the buffer needed to draw a cube out of triangles. Don't have this spread out all over in your code, get a single function that does this. Its just like the circle, except that you have to place the vertices to form a cube instead. Actually you could even start with a drawCircle() function instead of drawCube(), you'll just have a very flat animal, but the principle is the same.
    - Since you will be drawing 3D geometries in this assignment, you will have to enable depth test on your WebGL context. To do that, you can execute the following command just after retrieving the gl context:

        -
          ```
          gl.enable(gl.DEPTH_TEST);
          ```
    - Once that is enabled, you also have to clear the DEPTH_BUFFER when you clear your screen:

        -
          ```
          gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
          ```

2. Modify this function to take a Matrix parameter drawCube(Matrix M), that will apply this matrix when drawing the cube. The Matrix should get passed the the vertex shader as a uniform variable before calling drawArrays(). The vertex shader should have a line similar to glPosition = u_ModelMatrix * a_position; Test this function with some simple translate and rotate matrices and see that your cube does move around on the screen.
3. Have a slider that sets a global variable for the rotation angle gAnimalGlobalRotation; Also attach this to a uniform variable that gets passed to GLSL. Modify the vertex shader to be glPostion = u_GlobalRotation * uModelMatrix * a_position; Check that you can rotate your single cube with this slider.
4. Have a function renderScene() which is going to draw your whole scene. All the drawing happens in this function. Initially its just going to have a few lines {glClear(); M=scale(2,1,2); drawCube(M);}
   Call the renderScene() function at the end of main(), and also after you update a slider. (And later when you are animating or updating the window. The key is having all the drawing isolated in one place so you can just call that function to draw).
5. Now expand your renderScene() function to add more cubes to your animal, placed in the right locations. e.g. {glClear(); M=scale(2,1,2); drawCube(M); M=identity(); M.translate(1,2,3);M.rotate(30);drawCube(M)}, Keep adding cubes until you have a whole animal, with at least one limb that is two levels or more deep.
6. Make some joint angles controllable with sliders. Probably store the slider angles in global variables, like all UI elements. Edit the render() function to make use of these angles.
7. Now add slider control to a second level joint in the same limb that already has first level control. The real challenge comes from dealing with parts not directly attached to the body. The foot connects to the calf that connects to the thigh that connects to the body. These lower parts need to reflect the motion of the parts higher in the tree or they won't stay connected. Start with moving the thigh. Then figure out the stack of matrices that will get the calf to move connected. Then get the foot to stay connected to the calf. Hint: The foot rotation occurs in the calf coordinate frame, so first apply all the xforms you applied to the thigh, then those for the calf, then finally those for the foot. Dont try to just rotate the foot independently.
8. Add a tick() function (as discussed in the book). Create a global variable for the time, or frame of your animation. You are going to use this to update the animation. Whenever there is a tick() event, then you need to {g_time=theNewTime(); renderScene()} Just to test if your tick() is working, pick any cube in your animal and modify its matrix with M.rotate(g_time); in drawScene() before calling drawCube(M); Its going to move in some crazy way, but we just want to make sure tick() is doing something before moving on.
9. Add a button(s) to turn animation on and off, and move the animation code to a updateAnimationAngles() function that will automatically set the joint angle of at least one part. If its the same part you used previously, use the animation function instead of the slider values when animation is on. Try to keep the animation logic out of your render() function.
10. Thats it, you're done with the basic assignment (the helper videos get you up to this point). -
11. Additional points (which are harder to get) can be obtained for the features below. There are no help videos, but the concepts directly build on what you have already done.

- Keep adding animation to most of the other cubes so that the whole creature animates. Try to make the creature move in a natural way.
- Add color using a uniform variable, same as assignment 1. e.g. modify your drawCube(M, color) to take a color parameter so you can set this uniform before drawing the cube. Fix up your creature to be colored nicely.
- Add a third joint (e.g. not just upper arm and lower arm, now get the hand too). Its the same idea, but really makes sure you have your matrices worked out.
- Use a non-cube primitive (e.g. cylinder, dodecahedron, torus, cone) for one or more segments. You will likely use a function to generate the triangles in this shape, similar to how you made a circle in the prior assignment, but its ok to specify all the vertices by hand as well if that works for you.
- Add mouse control to rotate your animal. You learned to detect mouse clicks in the prior assignment and you already have rotation control in this assignment. It is sufficient to have simple control, just map x-position to x-rotation and y-position to y-rotation, not necessary for true trackball rotation unless you want to.
- Add a 'poke' animation sequence. Whenever the user holds shift and clicks, have the animal do a special animation different than the normal animation. for example - cry, or fall down, or explode, or wink. Be creative. (We are using shift-click since a normal click is likely connected to your rotation. Check on mouseEvent.shiftKey)
- Performance at 10fps at least, with a performance indicator on your webpage. If you followed my video your performance stinks, because I didnt worry about how often I reallocated memory, or buffers, or anything like that. One of the lab activities covers some performance hints, but the main idea is to not do things a zillion times that really only need to happen once. You can include the performance indicator from the lab activity or you can just check wall clock time like I did in the videos. This becomes a much bigger issues in Programming Assignment 3, so we're giving it a point here to make sure you are thinking about it. You will probably have to change some functions I suggested making (Bad professor!) Web browsers and platforms differ in performance. The grader will use a common browser like Safari, Firefox, Chrome, but you don't know which.

Resources:

- Readings:
  - (WebGL) Matsuda/Lea Ch3 (91-113) and Ch 4 (Appendix C)
  - Optional Chap 9 (p323-344) in Matsuda discusses hierarchical objects (however it has concepts you arent introduced to yet, and attempting to copy sample code from Chap9 will likely be a poor choice)

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) 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) for further explanation on how to submit your assignment.

View Rubric

Assignment 2 Rubric

Criteria			Points
Draw a Cube.	Full Marks 0.5 pts	No Marks 0 pts	/0.5 pts
Function that can draw a cube using a given matrix. drawCube(Matrix M);. view longer description	Full Marks 0.5 pts	No Marks 0 pts	/0.5 pts
The model can rotate on a global axis when using a slider. This is passed explicitly as a uGlobalRotation to the vertex shader.	Full Marks 0.5 pts	No Marks 0 pts	/0.5 pts
There is a single renderScene() function that handles all the drawing.	Full Marks 0.5 pts	No Marks 0 pts	/0.5 pts
There is a complete animal, with at least 8 parts.	Full Marks 0.5 pts	No Marks 0 pts	/0.5 pts

At least one joint angles is controllable with sliders.	**Full Marks** **0.5 pts**	**No Marks** **0 pts**	/0.5 pts
Slider control of a second level joint. (Without messing up the control of the first level joint)	**Full Marks** **1 pts**	**No Marks** **0 pts**	/1 pts
A tick() function works and animates at least one part.	**Full Marks** **1 pts**	**No Marks** **0 pts**	/1 pts
Button(s) to turn animation on and off	**Full Marks** **0.5 pts**	**No Marks** **0 pts**	/0.5 pts
Color on most of the body in a way that looks nice.	**Full Marks** **0.5 pts**	**No Marks** **0 pts**	/0.5 pts
Animation on most of the animal, and it moves in a natural way	**Full Marks** **0.5 pts**	**No Marks** **0 pts**	/0.5 pts
Place your site link as a comment of the submission.	**Full Marks** **0.5 pts**	**No Marks** **0 pts**	/0.5 pts
Third level joint.	**Full Marks** **0.5 pts**	**No Marks** **0 pts**	/0.5 pts
Non-cube primitive	**Full Marks** **0.5 pts**	**No Marks** **0 pts**	/0.5 pts
Poke with shift-click for a different animation	**Full Marks** **0.5 pts**	**No Marks** **0 pts**	/0.5 pts
Mouse control to rotate	**Full Marks** **0.5 pts**	**No Marks** **0 pts**	/0.5 pts

Performance optimization view longer description	Full Marks 0.75 pts	No Marks 0 pts	/0.75 pts
Performance indicator view longer description	Full Marks 0.25 pts	No Marks 0 pts	/0.25 pts

Preview Unavailable

Thomas_Wong_Assignment_2.zip

Download

(https://canvas.ucsc.edu/files/10283887/download?download_frd=1&verifier=PSqeJ8B5T6jhFF2xOD52ktbX7ER9WrOS0VPsnYNX)

New Attempt