CS148 Homework 2 - Triangles & Transformations

Grading on Monday, Oct 10th

1 Assignment Outline

This assignment document is a mix of instructional tutorials and exercises. The exercises are marked with **Action:** and are what you’ll demonstrate during the grading session. **We recommend reading the Action items first, then read the instructional tutorials for more context.** As with the first assignment, the complete list of what to submit are at the end of the document.

2 Collaboration Policy

You may work with one other partner and attend the grading session together as one group. **Both partners must be present for grading though.** Discussion of the exercises and quiz questions across groups on Piazza, etc is allowed as long as the answers (i.e. any numbers, screenshots, explanations, etc) are not explicitly posted or given out.

3 Office Hours

Office Hours start this week, and the logistics have been posted on Canvas. There is a mix of both in-person and online office hours.

For in-person office hours, you simply just need to show up to the location listed on Canvas at the scheduled time. The CA(s) for those office hours will give further instruction at the location.

For online office hours, enter the Zoom call listed on the Canvas calendar at the scheduled time. You’ll be placed in a waiting room, and the CA(s) will pull you into the main session when they are ready. **Note that the Zoom calls for online office hours are not the same as the Zoom call used for the grading sessions!**

4 Virtual World: Camera

In lecture, we talked about camera distortion based on distance and focal length. In this section, we will use Blender to explore this phenomenon (also known as the *dolly zoom* or vertigo effect) firsthand.

**Action:**

- First, we are going to set up the scene. Open a new Blender file, delete the cube, and add a monkey.
• Make sure the transformation sidebar is visible by toggling **View → Sidebar**. Select the camera in the outliner (upper right of the interface), and set the camera location to \((0, -4, 0)\) and rotation to \((90, 0, 0)\). Press the camera icon within the viewport (circled in red) to go to the camera view.

![Camera View](image)

• In the **Properties Editor** (lower right), change the focal length to 25 mm. Render and save this scene as an image. (If you’ve forgotten how to generate a render, review HW1. In particular, remember to set both the start and end frames to 1.)
• Change the camera location to \((0, -8, 0)\) and focal length to 50 mm. Render and save this scene as a 2nd new image.

• Change the camera location to \((0, -16, 0)\) and focal length to 100 mm. Render and save this scene as a 3rd new image.

• Compare the 3 images and discuss the effect of changing the focal length.

As a fun fact, common focal lengths for real-world cameras (prime lens) are: 24mm, 35mm, 50mm, 85mm, and 105mm.

5 Virtual World: Geometric Transformations

In this section, you will be applying the concept of geometric transformations (e.g. translating, rotating, and scaling points in a coordinate space) as discussed in lecture.

5.1 Transforming Points by Hand

**Action:** Consider a point located at \((1, -1, 1)\) in world space. Using the transformation matrices from lecture, compute the new coordinate of this point for each of these 4 separate cases:

1. rotating it about the x-axis by 45 degrees, then rotating it about the y-axis by 45 degrees.
2. rotating it about the y-axis by 45 degrees, then rotating it about the x-axis by 45 degrees.
3. translating it along the x-axis by 1 unit, then rotating it about the y-axis by 45 degrees.
4. rotating it about the y-axis by 45 degrees, then translating it along the x-axis by 1 unit.

You may use a calculator, Mathematica, or even your own written code, etc to compute the transformations, but you must be prepared to explain the math. That is, you must be able to describe the equations you’d use for each case. For instance, let \(R_x\) be the rotation matrix about the x-axis, \(R_y\) be the rotation matrix about the y-axis, and \(\vec{x}\) be the point – in what order do you need to multiply \(R_x, R_y, \vec{x}\) to compute the answer for case 1 above?
5.2 Transforming Objects in Blender

To transform an object in Blender, you can use the buttons in the left toolbar of the 3D viewport: Move, Rotate, Scale, and Transform. Selecting an object with one of these buttons enabled will prompt an orange dot to appear to indicate the object origin, as well as red, green, and blue lines to show the degrees of freedom in which you can move the object.

The keyboard shortcuts are g (for grab/translate), r (for rotate), and s (for scale). You can press x, y, or z afterward to lock the transform to a specific axis. For example, if you press g, then x, and then move your mouse, then the object will only translate along the X-axis. You can also type in numbers after pressing the keys to specify the exact values you want to move, rotate, or scale. For example, if you press g, z, 2, then hit Enter, then the object will translate 2 units along the Z-axis.

Note: make sure your cursor is in the 3D viewport area when pressing the keys. Blender uses the cursor location to determine which area you want to send your keystrokes to.

You can view and alternatively change the local transform of an object in two other places in the Blender interface. The first is the sidebar of the 3D viewport by pressing n. The second is in the Properties Editor in the bottom right under Object Properties, represented by an orange square. Under Transform, you should see the local location (translation), rotation, and scale of the selected object.

5.2.1 Rotating Objects about the Global World Origin

You may notice that if you try to rotate an object using aforementioned tools in Blender, then it only rotates around its own origin point rather than the world space origin. For instance, if you translate the default cube e.g. 5 units along x, then try to rotate about z, then the cube ends up rotating about its own center, not the global z-axis. This is because by default, Blender does local transforms about the object’s own origin.

To make it rotate about the global axes, go to the Object dropdown menu near the Object Mode indicator, then Snap → Cursor to World Origin. If you’re in Edit Mode instead, then it will be under the Mesh dropdown menu instead in the same spot. From there, set the transform pivot in the dropdown menu next to the Global indicator to 3D Cursor. Now, if you use the rotate tool or the r keyboard shortcut, you’ll see the rotations go about the world origin.
5.2.2 Viewing the Global Coordinates of a Vertex

Enter **Edit Mode** and select a vertex. From there, press the **n** key-board shortcut to open the side-bar of the 3D viewport. This by default displays the local coordinates of the vertex, relative to the object origin. Toggle to **Global**, and you'll see the global coordinates of the vertex, relative to the global origin.
5.2.3 Order of Operations

**Action:** Create a new Blender scene with the default cube, and find the vertex at \((1, -1, 1)\) in *Edit Mode*. Then pull up the viewport sidebar with the *n* hotkey, and toggle to show the global coordinates of the vertex. For each of the following transformations, take a screenshot of the viewport with the vertex’s **global coordinates** at the end as shown in Blender.

You will likely want to do the transforms in *Object Mode* first, then swap to *Edit Mode* to view the vertex. For sanity’s check, the transformed coordinates of this vertex that you see in Blender for each of these separate cases should match those you computed in Section 5.1.

1. Rotate the default cube about the **global** x-axis by 45 degrees, then rotate it about the **global** y-axis by 45 degrees.

2. Rotate the default cube about the **global** y-axis by 45 degrees, then rotate it about the **global** x-axis by 45 degrees.

3. Translate the default cube along the x-axis by 1 unit, then rotate it about the **global** y-axis by 45 degrees.

4. Rotate the default cube about the **global** y-axis by 45 degrees, then translate it along the x-axis by 1 unit.

6 Triangles: The OBJ file format

One of the most commonly used file formats to store graphics data is the Wavefront .obj format. To get a sense of how the file format works, let’s examine an .obj file firsthand. Open up a new Blender scene, and delete the default cube. Import the example .obj file that came with this homework (*File → Import → Wavefront (.obj)* and navigate to the example file, e.g. $CS148_DIR/HW2/sphere_example.obj$).
The sphere .obj should now appear as a mesh in the Blender GUI. Select the sphere, and enter Edit Mode. You should see an assortment of vertices and faces along the surface of the sphere. To look at the .obj file at an even lower level, open up the sphere example file in your favorite text editing program (e.g. Notepad++, TextEdit, Sublime Text, etc). Scrolling through the file, you may notice the following file structure:

```
v x1 y1 z1
v x2 y2 z2
v x3 y3 z3
...
v xm ym zm
f face0v1 face0v2 face0v3
f face1v1 face1v2 face1v3
f face2v1 face2v2 face2v3
...
f facenv1 facenv2 facenv3
```

This is an example of the most basic form of the .obj format, which contains a list of vertices and a list of triangular faces that together specify the geometry of a 3D object mesh. The coordinate space in which the vertices are specified is the object space of the object as discussed in lecture.

The lines that begin with a v contain vertex data, while the lines that begin with a f contain face data. Each “vertex line” starts with a v and follows the v with three floating point numbers: the x, y, and z coordinates (in that order) of a vertex in the described 3D model. Each “face line” starts with a f and follows the f with three integers specifying the three vertices that make up a triangular face in the 3D model. For example, the following line:

```
f 1 8 37
```

specifies a face that is made up of the first, eighth, and thirty-seventh specified vertices in the file. Note that this means the vertices are 1-indexed; i.e. the first specified vertex in the file is referred to as the first vertex, rather than the zeroth vertex. This is simply due to convention.

**Action:** Be prepared to give a quick (< 1 min) explanation of how the .obj file format stores an object’s vertex and triangle face data.

## 6.1 More Complex OBJ Files

You may notice while importing the sphere example file in Section 6 that there was also an export option in Blender. If you try exporting e.g. the default cube as an .obj file and then examining it in a text editor, then you will see a lot more data in the file than just the vertices and faces.

We will discuss this other data later in the class. But if you’re curious, then you might find this website a good reference.
7 Triangles: Modeling Geometry in Blender

Now that you’re familiar with the .obj format, you can try your hands at modifying object meshes in Blender to create anything you’d like! Two approaches that we’ll discuss are polygon modeling and sculpting.

**Action**: Use any of the following polygon modeling or sculpting methods to make as simple (or as complex) of an object as you’d like. Whatever you make does not have to be anything remarkable (yet), as we’ll work more with geometric modeling in a later homework.

7.1 Polygon Modeling

Polygon modeling is the basis of every 3D modeling package. With modeling, we manipulate polygonal meshes by moving around vertices, edges, and faces to create more complex and interesting objects. Modeling is ideal for creating hard-surface objects (i.e. man-made objects with sharp angles) due to the precision of selecting and moving individual vertices. On the flip side, modeling is not so great for organic objects - if you’re hoping to create a more organic effect, then sculpting might be the better technique to use.

![Figure 1: Dining table object made from modeling. Source: ArtStation](image1.jpg)

![Figure 2: Stormtrooper helmet object made from modeling. Source: ArtStation](image2.jpg)

To access the modeling tools in Blender and make changes to objects at the vertex/edge/face-level, we will want to be in **Edit Mode**. Once in **Edit Mode**, you will notice that 1) new tools have appeared in the sidebar, and 2) you can now see individual vertices, edges, and faces.
7.1.1 Selection

Simply click on a vertex to select it. If you want to select multiple vertices, hold Shift, and click on other vertices that you want to select. You can also toggle between Vertex Select, Edge Select, and Face Select in the upper-left, or alternatively with the 1, 2, and 3 shortcut keys.

7.1.2 Deletions

To delete a vertex / edge / face, simply press X while the vertex / edge / face is selected. This will bring up the “delete” menu, which you can use to specify exactly what you want to delete.

7.1.3 Extrusions

A basic but powerful mesh editing tool is the Extrude Region tool. To extrude a face, first select the Extrude Region tool in your toolbar. Then, select the face you want to extrude (switch to Face Select and click on the face). Then, click and drag the yellow + symbol. A keyboard shortcut is also available by pressing E after selecting the face to extrude. We can also extrude an
entire region composed of multiple faces. Simply select the faces you want to extrude (hold Shift and click to multi-select), and drag the yellow + symbol.

7.1.4 Resources

- This Youtube video series is a great basic intro series to modeling on Blender’s official channel. It was created as part of Blender’s 2.8 fundamentals series, but works in later versions as well.

- Other common techniques for modeling are with booleans and bevels - see this beginner’s Youtube video tutorial.

- Blender has a free add-on for performing boolean operations. It already comes with Blender, but see here for how to enable the Bool Tool and configure its preferences plus keyboard shortcuts.

- For more in-depth information on how to use any of Blender’s modeling tools, see the full Blender documentation on editing meshes.

- There are many, many modeling tutorials on Youtube, so if there’s a specific object you’re interested in creating, try searching on Youtube for a video tutorial that can help get you started. Donut tutorials for instance tend to be a very popular start for students.
7.2 Sculpting

Sculpting is best suited for organic shapes, and uses brushes to deform the object. Note that the resulting mesh from sculpting will likely have a high polygon count, so you will need decent computing power if you plan to sculpt fine details.

Figure 3: Sculpting a realistic face. Note that human faces are really hard to get right! Source: FlippedNormals

Figure 4: Sculpted creature. Source: ArtStation

Access Sculpt Mode by directly clicking the Sculpting tab on the top or through the Object Mode menu. Once in Sculpt Mode, you will notice that new tools have appeared in the sidebar.

7.2.1 Brush Settings

The brush settings are on either the top or side toolbar. You can change the size of the brush using Normal Radius and the intensity using Hardness. On the top right, you can change symmetry between x,y,z. The brush, at default, is symmetrical around x.

Two useful settings to know are the Accumulate and Front Faces Only options. Toggling Accumulate on causes brush strokes to stack on top of each other, while toggling Front Faces Only on makes it so that the brush only affects the vertices that you see from your viewpoint.

Also note that toggling ctrl/cmd while using the brush will change the direction that the brush affects the geometry. Furthermore, holding down ctrl/cmd subtracts from the object while using the brush shape.
7.2.2 Dyntopo

Dyntopo (short for Dynamic Topology) is a useful tool that can help you add or remove geometry, allowing you to construct more complex shapes out of a simple mesh. You can find it in the upper right when in Sculpt Mode. If you need more detail in a model, then enabling Dyntopo will cause Blender to dynamically tessellate your mesh into more vertices and faces for finer sculpting.

The above edit to the default monkey mesh (Object Mode → Add → Mesh → Monkey) was done with a Clay Strips brush with Accumulate and Dyntopo enabled to make the top of the head larger.
7.2.3 Resources

- For an introduction to sculpting and an explanation of the various brushes, see the official tutorials on Blender Sculpting Fundamentals and Sculpting in Blender with Complex Models.
- There’s also this [post](#) for first-time sculptors if you’re into character modeling.
- The Youtube channels YanSculpt and CGBoost are good places to find advanced sculpting videos.
- For more in-depth information on how to use the sculpting tools, see the full Blender documentation on Sculpting.

8 Quiz Questions

During the grading session, the CA will choose one of the following questions at random to ask. Please be prepared to give a brief (~1 min) answer based on the lecture material.

- What is the purpose of homogeneous coordinates in the context of geometric transformations? How can we use them to composite rotation, scaling, and translation operations into 1 matrix operation?
- Suppose you see two objects of equal size in front of you; the one farther away will appear smaller. How does the idea of perspective projection (i.e. transforming a viewing frustum into an orthographic volume) achieve this effect for the virtual world?
- Explain why the order of the vertices for a triangle face matters when it comes to rendering. For which orientation of the vertices do we not render the triangle?
- How do we deal with overlapping triangles so that we may appropriately color the pixels of our image when rendering?
- How can we use barycentric weights to describe a point inside a triangle based on the triangle’s vertices?

9 How to Submit / How to get Graded

On **Monday, October 10**:

- if you are assigned the 2-3 or 3-4 PM time block, then please join this Zoom call.
- if you are assigned the 5-6 or 6-7 PM time block, then please join this other Zoom call.

These are different Zoom calls! Please join the right one! The links are embedded into the blue text. Those with alternative times as scheduled via email should join the 2nd (5-7) Zoom call at the same alternative time window.

We won't be using QueueStatus anymore, as we saw it couldn’t handle the high number of students for the HW1 grading session. Instead, you simply just need to join the Zoom, and a CA will get to you within the hour. For HW1, every student who joined the Zoom call was addressed within the hour block they joined.

If you are working with a partner, then both of you may get graded together at a single time block. **Both of you need to be in the Zoom call to get graded!** When one partner is let in, they should tell the CA to also let the other partner into the same room.
10 What to Demo / Submit

During the grading session, please screenshare with your CA and have all of the following ready to show beforehand:

- (1 pt) Show the 3 images from Section 4 of the monkey head at various distances and focal lengths, and discuss the effect that changing the focal length has on the image. Hint: look closely at what happens to the monkey head’s ears and forehead across the images.

- (1 pt) Show your computed transformation of the point from Section 5.1 and explain the order of matrix operations for each case. Hint: you should get a different transformed point for each case.

- (1 pt) Show the screenshots from Section 5.2.3 of your Blender transformations for the cube. Make sure the coordinate of the specified vertex is clearly visible in the screenshots. Hint: the numbers should match those you got from Section 5.1.

- (0.5 pt) Explain how .obj files store the vertex and triangle face geometry data for objects. Hint: see Section 6.

- (0.5 pt) Show off an object that you modeled or sculpted. As mentioned in Section 7, this object does not have to be anything remarkable (yet). (It can be as simple as drawing some brush strokes on a sphere.)

- (1 pt) Answer the quiz question given by the CA correctly.