# CS148 Homework 5 Part 1

Homework Due: Jul 30th at 11:59 PM PST
Quiz Date: Tuesday Jul 26th

## 1 Assignment Outline

HW5 will be split into 2 parts. The first part, the Blender component, will have you explore what Blender has to offer for the concepts covered in Lecture 10 (Simulation and Animation). This part is included in this PDF and will be due the usual Saturday deadline.

Part 2 of HW5 will be the technical component and involve the concepts from Lecture 11 (Data-Driven Graphics), which will be given Tuesday July 26. As such, Part 2 will be released after class Tuesday and have a later deadline.

For this Part 1, you will do:

- `TODO Fluid Simulation` (1 point): Generate a fluid simulation in Blender and incorporate it into a scene. Then submit the ray traced render.
- `TODO Cloth Simulation` (1 point): Generate a cloth simulation in Blender and incorporate it into a scene. Then submit the ray traced render.
- `TODO Motion Blur` (1 point): Generate keyframes of a scene involving motion in Blender, then render the scene with motion blur.
- `TODO Volume Rendering` (1 point): Generate a volumetric object in Blender and incorporate it into a scene. Then submit the ray traced render.
- `TODO Project` (1 point): Render your current progress on your final project and submit the image. This does not have to look good (yet)!

These TODOs are marked with the **Action:** indicator in this PDF as well as TODO in the section headers.

Please download `HW5.zip`, save it in some course directory `$CS148_DIR` on your machine, and unzip the file in this directory. All the work will be done in the `$CS148_DIR/HW5/HW5.ipynb` Jupyter Notebook. When you are finished, print and save your Notebook as a PDF and submit to Gradescope.

## 2 Quiz Questions

- Explain how we can use the Explicit Euler method to compute the change from the initial state of a physical system at time $t$ to some new state at time $t + \Delta t$. Why does Explicit Euler sometimes cause our simulation to be unstable ("spiral out of control"), and what is the importance of the time step in avoiding this behavior?
- Describe the difference between the Lagrangian particle approach and the Eulerian grid approach to fluid simulation. How does each approach represent their fluid(s)? How may we mix both approaches and for what benefits?

- When modeling cloth as a grid of particles connected by a network of springs, why do we need three different types of spring connections? What is the purpose of each spring type?
- Explain how the concept of conservation of momentum allows us to evaluate the new states of two colliding particles after a collision in 1D. How can the concept of 1D collisions then be extended to the 3D case of a particle (aka a point) collision with a triangle (aka a plane)?
- Given a set of keyframes, how do we use cubic splines to smoothly interpolate the motion between the keyframes? Suppose we want 5 generated frames between our keyframes. How do compute them, and how would we ray trace motion blur with them?

## 3 Blender Fluid Simulation

Blender provides a relatively easy-to-use interface for generating workable meshes from fluid simulation(s). Let's try it out with a simple example. First, taking the default Blender scene:

- Scale the default cube to enlarge it. This large cube will become our fluid domain.
- Add a smaller cube mesh inside the default cube. This small cube will act as our collision object.
- Add an UV mesh above the smaller cube and inside the outer cube. This sphere will contain our liquid that gets advected in our fluid domain.

It may help to switch to wireframe view in the Viewport toolbar when setting this up. See Figure 1 for a visual of the full setup.

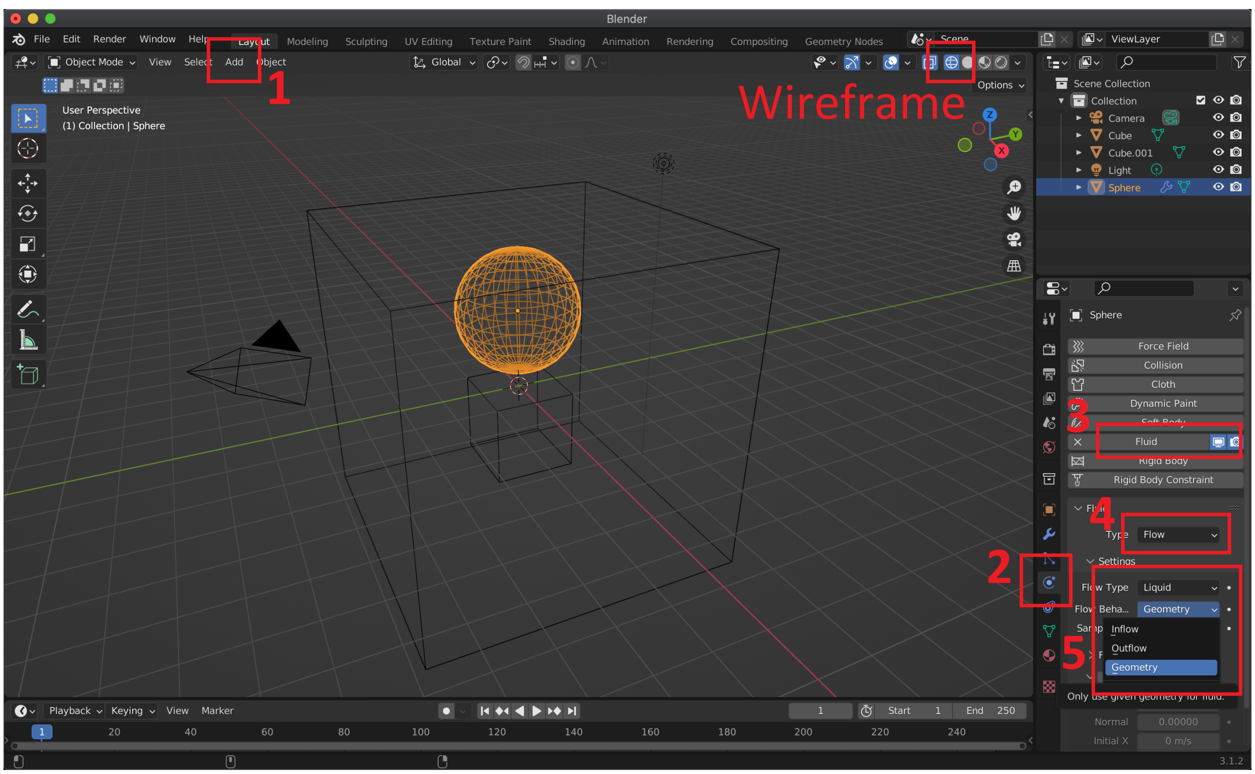


Figure 1

Start by selecting the sphere, then:

- Click on the **Physics Properties** panel of the Properties Editor (#2 in Figure 1) to access the physics options.
- Click on the **Fluid** option (#3 in Figure 1).
- Click on the **Type** dropdown menu and select **Flow** (#4 in Figure 1).
- Click on the **Flow Type** dropdown menu and select **Liquid** (#5 in Figure 1).
- Click on the **Flow Behavior** dropdown menu and select **Geometry** (#5 in Figure 1).

This sets up the sphere to be a giant collection of liquid particles, e.g. think a Lagrangian particle system. The **Geometry** option we set at the end tells Blender to have these liquid particles act as geometry when interacting with other objects in the scene (i.e. they will behave like actual mass when colliding with objects). In comparison, the **Inflow** and **Outflow** options will continuously add or delete fluid particles during the simulation, with the inflow simulating a running faucet, and the outflow simulating a drain with fluid disappearing upon interacting with the world.

Now, let's add a collision object to make the simulation more interesting. Select the smaller cube, then:

- Click on the **Physics Properties** panel of the Properties Editor.
- Click on the **Fluid** option.
- Click on the **Type** dropdown menu and select **Effector** (#1 in Figure 2).
- Click on the **Effector Type** dropdown menu and select **Collision** (#2 in Figure 2).

This sets up the cube to be an object that the sphere of fluid will collide with as it falls down in our simulation.

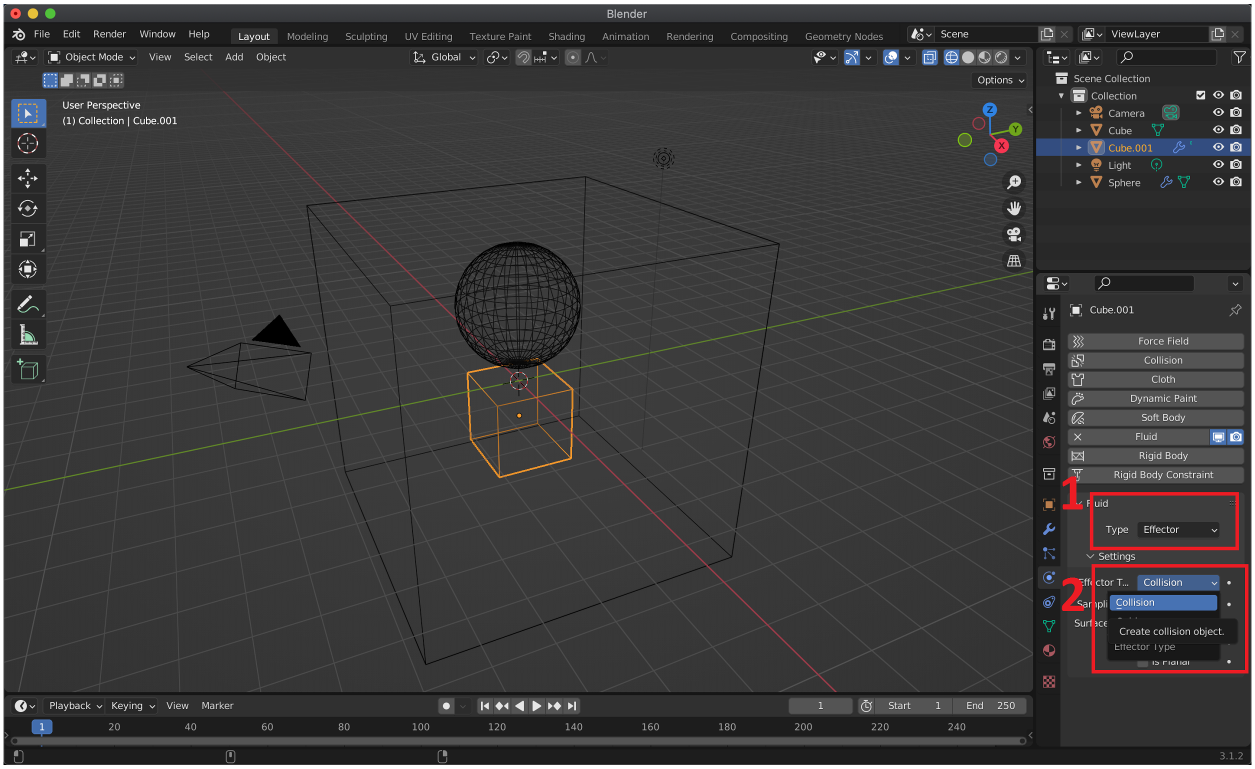


Figure 2

Finally, we need to set up the last cube in the scene. Select the large cube, then:

- Click on the **Physics Properties** panel of the Properties Editor.
- Click on the **Fluid** option.
- Click on the **Type** dropdown menu and select **Domain** to make this cube our fluid domain (#1 in Figure 3).
- Click on the **Domain Type** dropdown menu and select **Liquid** to tell Blender that this fluid domain will be used for simulating liquids (#2 in Figure 3).

This sets up the large cube to be our fluid domain that contains a gravity force field that will act on any liquid objects within. This includes the sphere of liquid, but not the collision cube. This means that when we run the simulation, gravity in our fluid domain will cause motion for the sphere of liquid and make it fall along the negative z-axis. But, the collision cube will stay in place. However, the sphere of liquid and the collision cube will interact, because we set the cube to be a collision object.

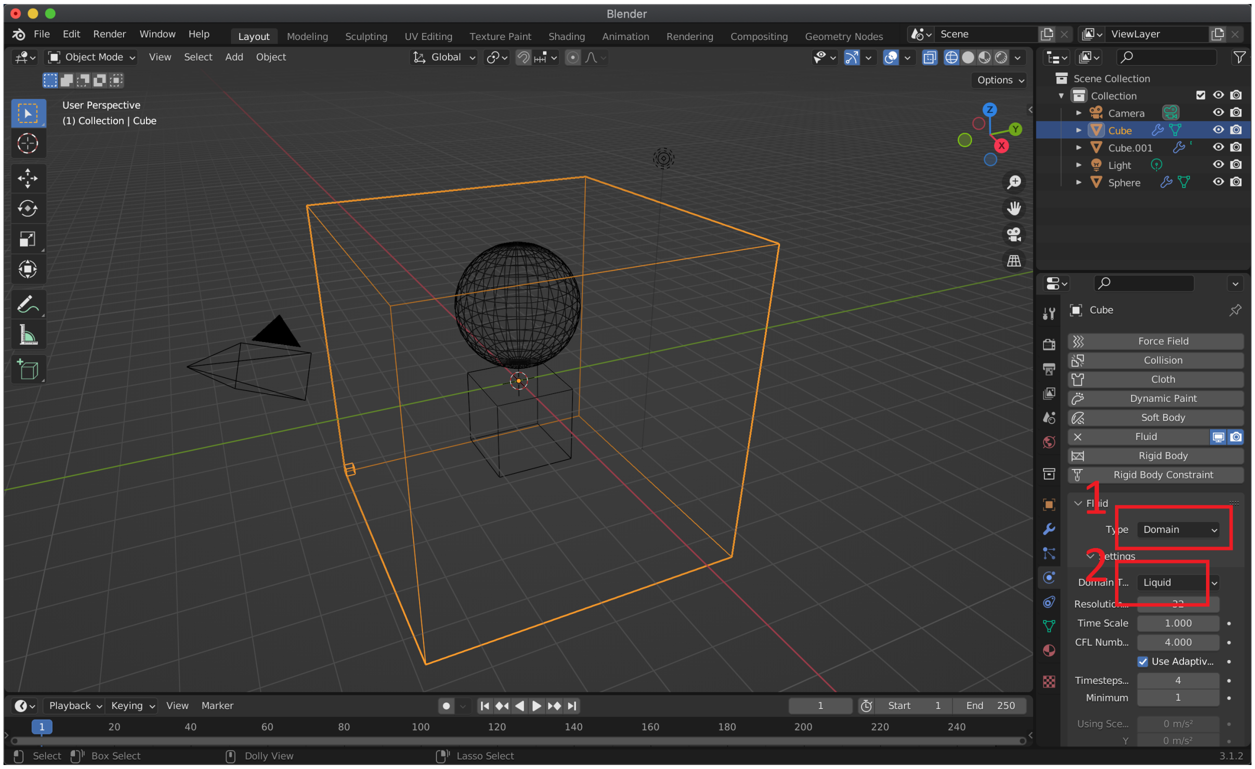


Figure 3

To actually run our fluid simulation, we need to set a few more parameters:

- Scroll down further in the **Physics Properties** panel of the large cube that we set to be our fluid domain.
- Set the **Cache** folder (#1 in Figure 4) to a folder where you want Blender to cache or temporarily save the files needed to model this fluid simulation.
- Set the **Type** for the simulation to **All** to tell Blender to generate geometry for all frames of the simulation (#2 in Figure 4).
- Check the **Mesh** checkbox to tell Blender to explicitly generate mesh geometry for the fluid simulation (#3 in Figure 4).
- Set the final **End Frame** of your fluid simulation in both the Properties editor and the **Timeline Editor** at the bottm of the Blender interface (#4 in Figure 4).
- Click **Bake All** when you're ready to generate your fluid simulation (#5 in Figure 4). It may take a few seconds for Blender to finish computing. There should be a progress bar at the bottom.

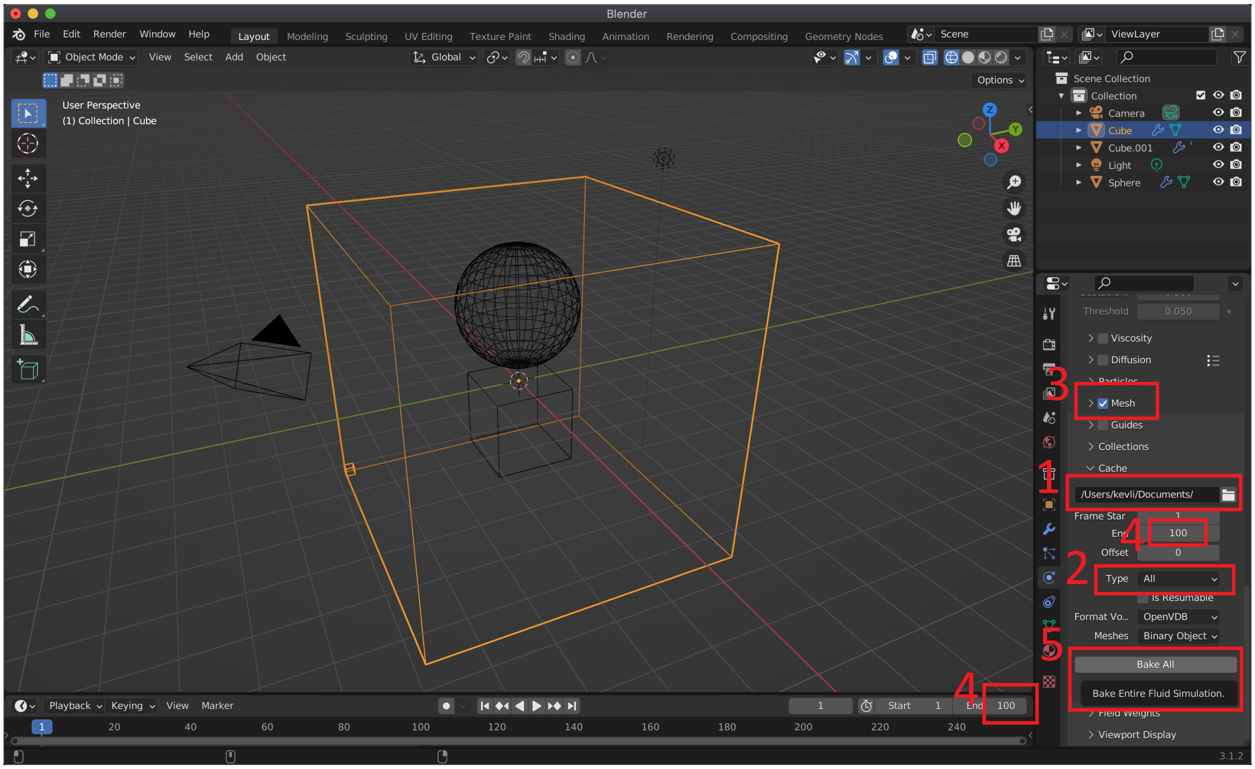


Figure 4

To see your fluid simultion in action, move the **Timeline Editor** at the bottom of the Blender interface up until you see the blue line clearly (see Figure 5). You'll see a timeline starting at 1 and ending at the **End Frame** you set earlier. You can scrub through the timeline by clicking and dragging the blue line to see the fluid simulation at different frames. For instance, Figure 5 shows us looking at the 50th frame of our simulation of 100 frames.

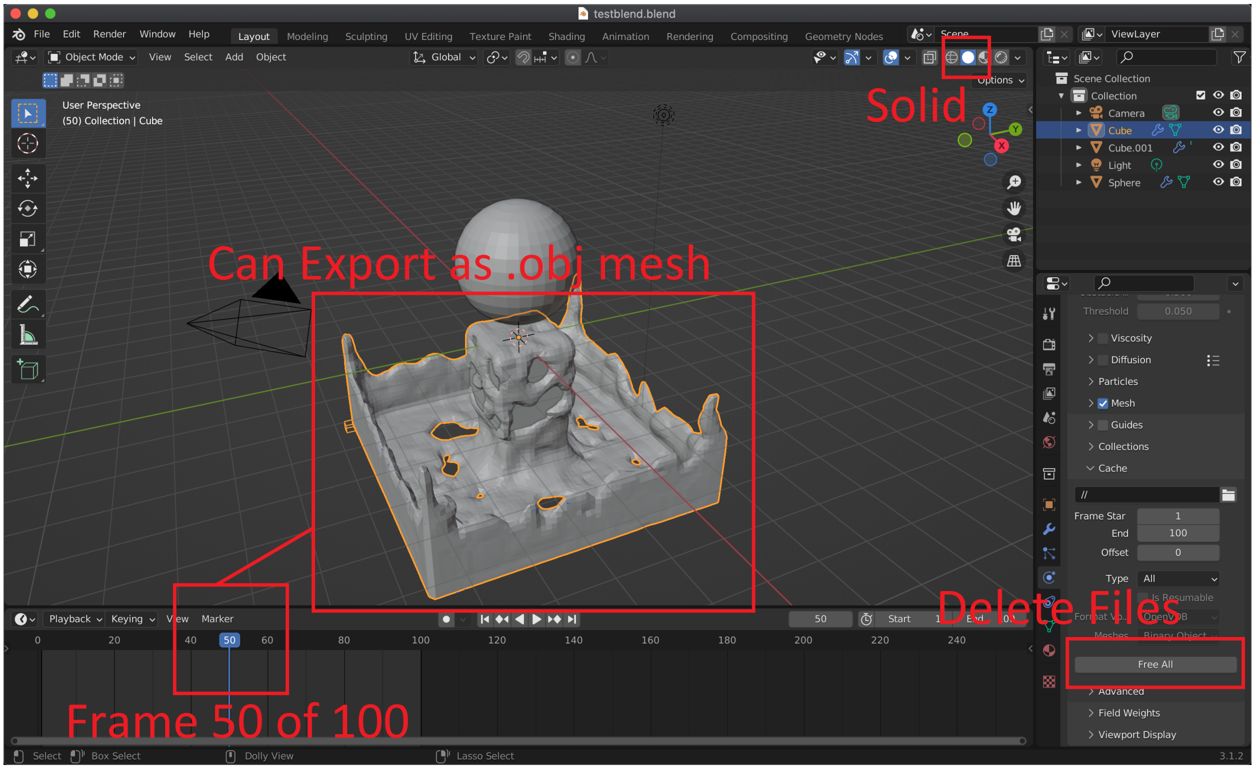


Figure 5

You may want to switch to solid view in the Viewport toolbar to get a better sense of the geometry of your fluid. If you want to redo your fluid simulation with a different set up, then you need to first delete the cache files for the current simulation (see Figure 5) before modifying your scene.

When you find a frame in your simulation that you're happy with, you can export all the geometry in that frame as an .obj mesh. Simply use the **File → Export** option as usual. Then, you can import that same .obj into another scene and work with it like any other geometry. For instance, in Figure 6, we show how we can transform the generated liquid mesh just like any other object. You can also add materials, textures, etc to your liquid mesh.

For more info on the parameters and the various options that Blender provides for fluid simulation, see their official manual.

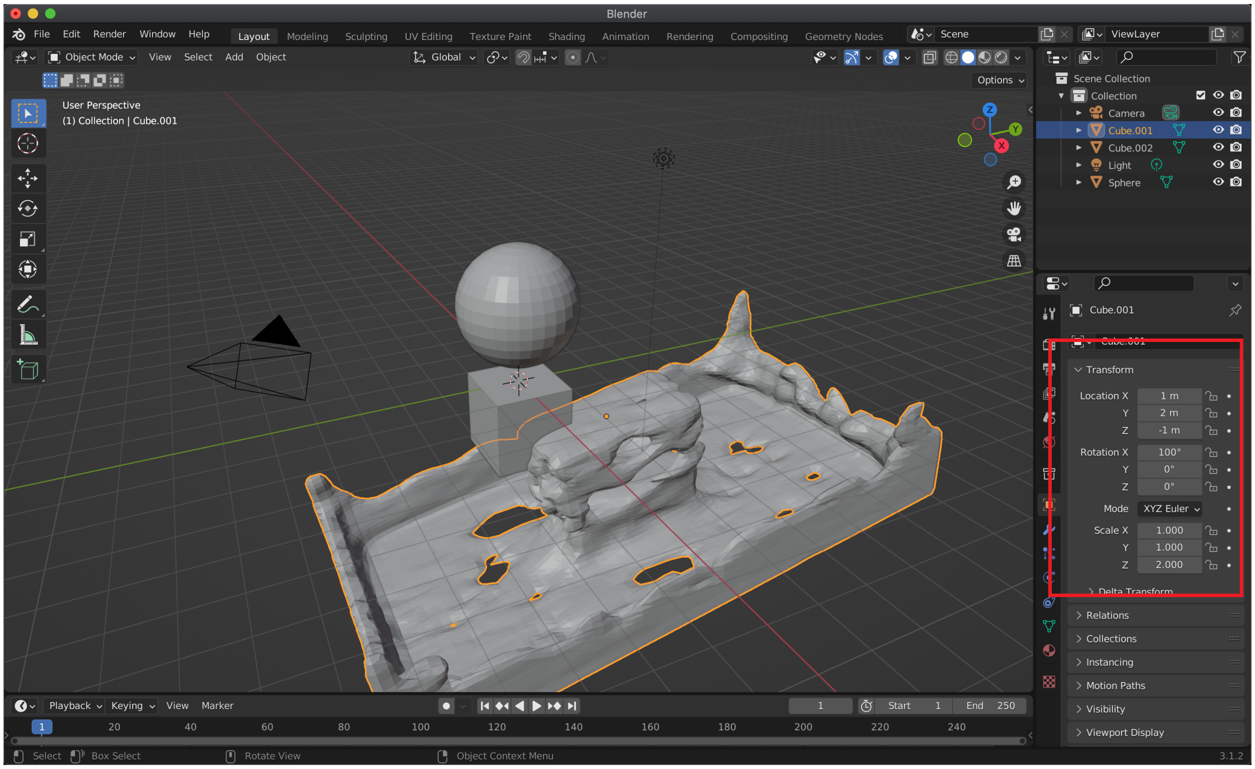

Figure 6

## 4 TODO Fluid Simulation

**Action:**

Use Blender's fluid simulation interface to generate as simple or as complex of a fluid simulation as you'd like. If you're planning to have fluids in your final project scene, then we recommend spending some time here to make something that you'll use later. When you're satisfied with your fluid simulation, change the **Render Engine** to **Cycles** for ray tracing. Then, render your scene as an image as you've done in previous homeworks.

When the Blender Render window finishes, save the result to the **$CS148_DIR/HW5/images** directory. Find the **TODO Fluid Simulation** cell in the **$CS148_DIR/HW5/HW5.ipynb** Jupyter notebook, and edit the cell to display your saved image.

## 5 Blender Cloth Simulation

Blender also provides a straightforward interface for deforming meshes with cloth simulation(s). Let's see with a simple example. Create a new scene, delete the default cube, and add a plane mesh plus the built-in **Monkey** mesh. Position the plane above the monkey head (see Figure 7) The plane mesh will become our cloth in our cloth simulation, and we'll have it collide with the monkey head.

Select the plane, then:

- Go into **Edit Mode**, then **Modifier Properties** in the Properties Editor (#1 in Figure 7).

- Click **Add Modifier** and add a **Subdivision Surface** modifier to allow us to subdivide the plane into a finer grid.
- In the modifier options, click **Simple** to use Blender's simple subdivision algorithm, which doesn't round out the edges of our plane (#2 in Figure 7).
- Right click the plane and click **Subdivide**. This will bring up a **Subdivide** control panel in the bottom left (#3 in Figure 7).
- Change the **Number of Cuts** to 50 to make the plane a 50x50 grid of cells.

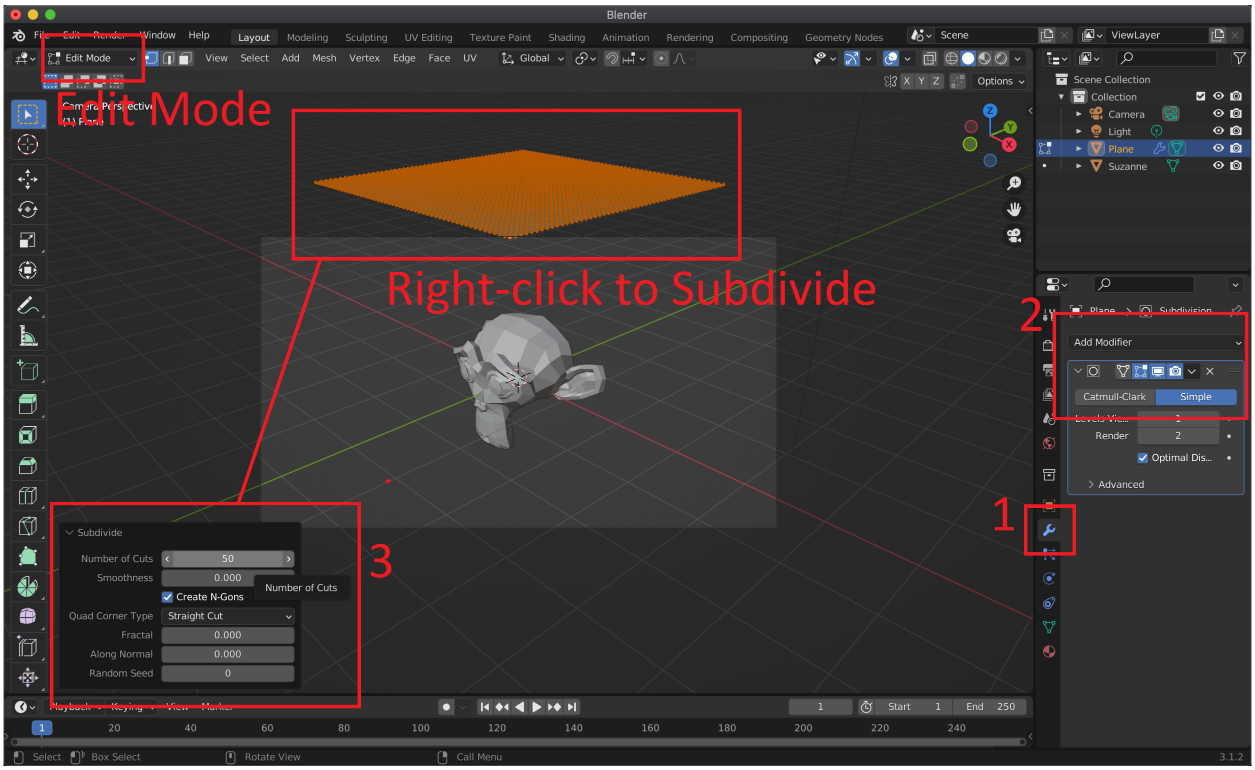


Figure 7

Now, we need to set the plane to act as a cloth object and the monkey head to act as a collision object.

- Select the plane (you want to go back to **Object Mode**), then click on the **Physics Properties** panel of the Properties Editor. Select the **Cloth** option. This tells Blender to model our plane as a network of particles and springs to model a piece of cloth.
- Select the monkey head, then click on the **Physics Properties** panel of the Properties Editor. Select the **Collision** option. This tells Blender to have our monkey head interact with any objects that collide with it.

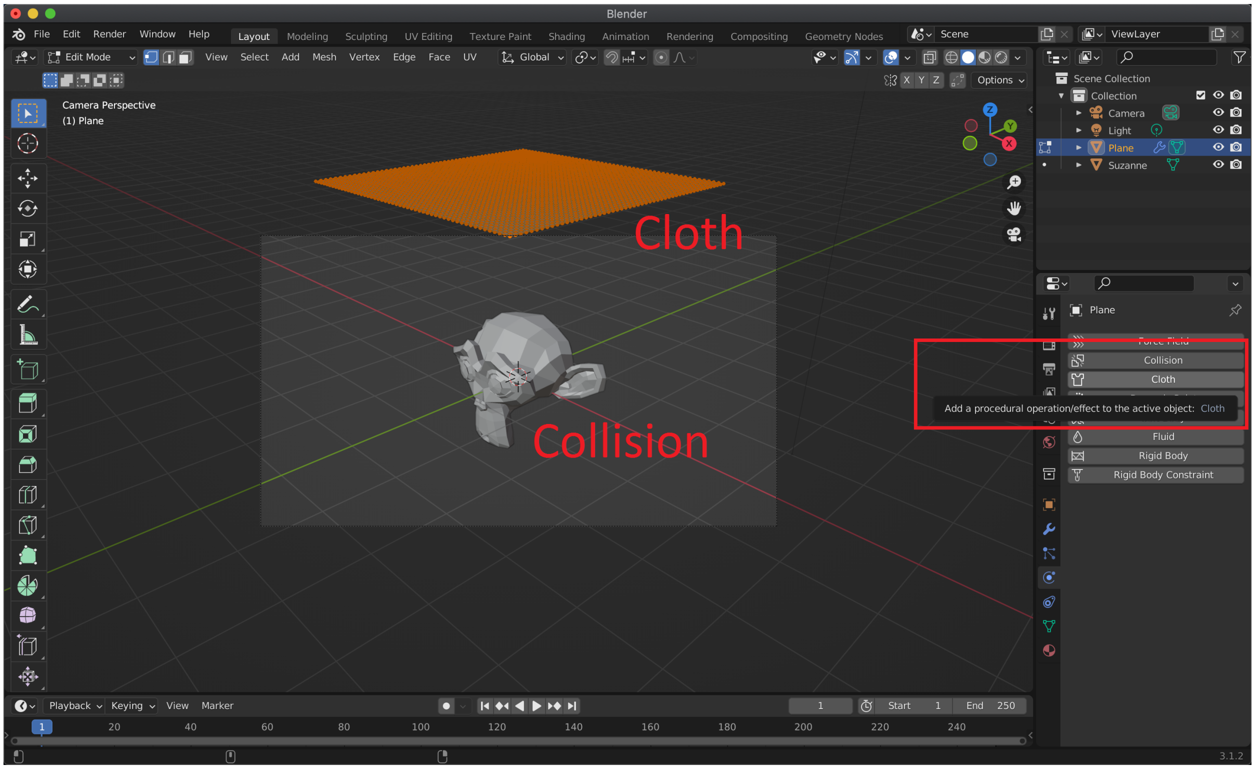


Figure 8

To see your cloth simultion in action, move the **Timeline Editor** at the bottom of the Blender interface up until you see the blue line clearly (see Figure 9). You'll see a timeline starting at 1 and ending at 250 (you can change this to another number like you did with the fluid simulation). You can scrub through the timeline by clicking and dragging the blue line to see the cloth simulation at different frames. For instance, Figure 9 shows us looking at the 31st frame of our cloth simulation.

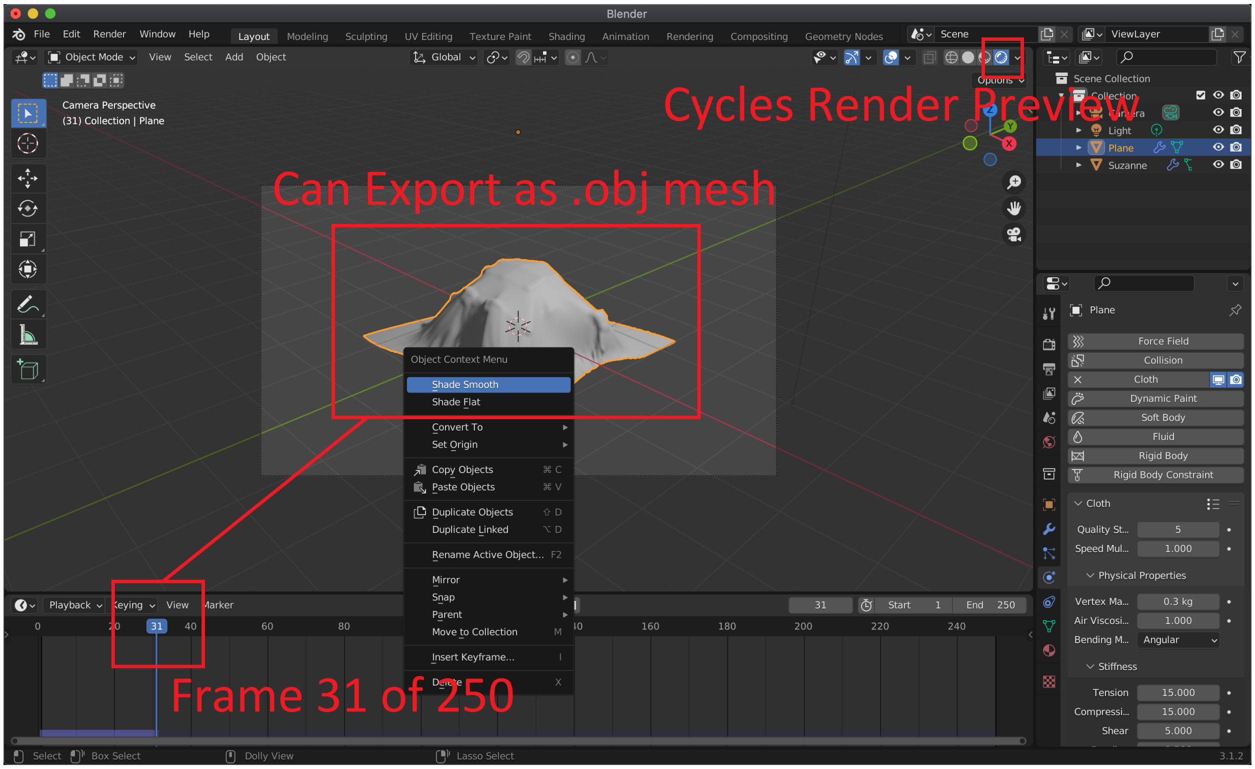


Figure 9

To get a better sense of how your deformed plane looks like as a cloth, you can turn on `Shade Smooth` when right-clicking it in `Object Mode`. Make sure to change the `Render Engine` to `Cycles` for ray tracing. Then click the cycles render preview on the viewport toolbar (see Figure 9).

When you find a frame in your simulation that you're happy with, you can export all the geometry in that frame as an .obj mesh. Simply use the `File → Export` option as usual. Then, you can import that same .obj into another scene and work with it like any other geometry.

For more info on the parameters and the various options that Blender provides for cloth simulation, see their official manual.

## 6 TODO Cloth Simulation

**Action:**

Use Blender's cloth simulation interface to generate as simple or as complex of a cloth simulation as you'd like. If you're planning to have cloth in your final project scene, then we recommend spending some time here to make something that you'll use later. When you're satisfied with your cloth simulation, change the `Render Engine` to `Cycles` for ray tracing. Then, render your scene as an image as you've done in previous homeworks.

When the Blender Render window finishes, save the result to the `$CS148_DIR/HW5/images` directory. Find the `TODO Cloth Simulation` cell in the `$CS148_DIR/HW5/HW5.ipynb` Jupyter notebook, and edit the cell to display your saved image.

# 7 Blender Motion Blur

Blender provides a very simple way to set up and ray trace motion blur. Let's take a look with a simple example. First, taking the default Blender scene:

- Translate the cube to some location that you want to act as our first keyframe.
- Move the `Timeline Editor` at the bottom of the Blender interface up until you see the blue line clearly (see Figure 10).
- Right click the cube and click `Insert Keyframe...` (see Figure 10), then click `Location` (see Figure 11). This will mark the current scene as our first keyframe in the timeline and also tell Blender that we would like to interpolate location aka position values for our animation.
- Move the blue line in the `Timeline Editor` to a later frame like frame 10.
- Go back and translate the cube to another location that you want to act as our second keyframe.
- Then, using the same steps as above, insert a new location keyframe.

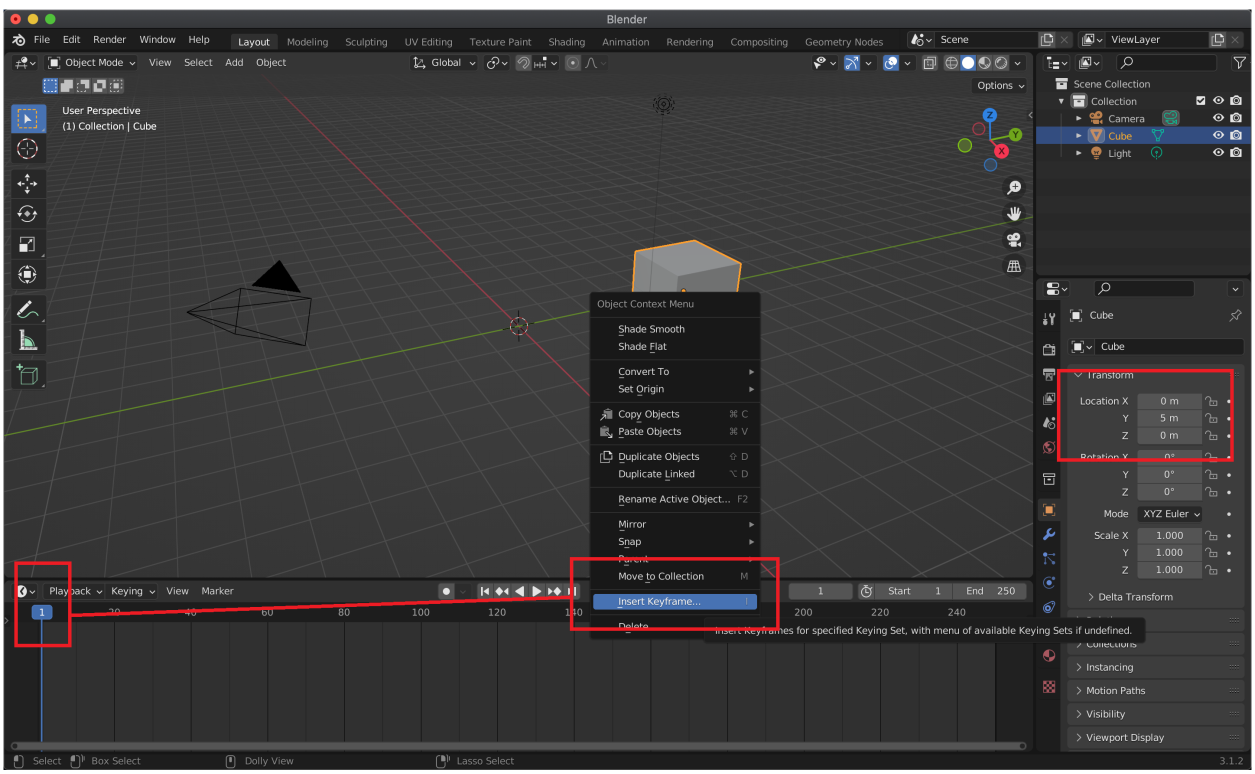


Figure 10

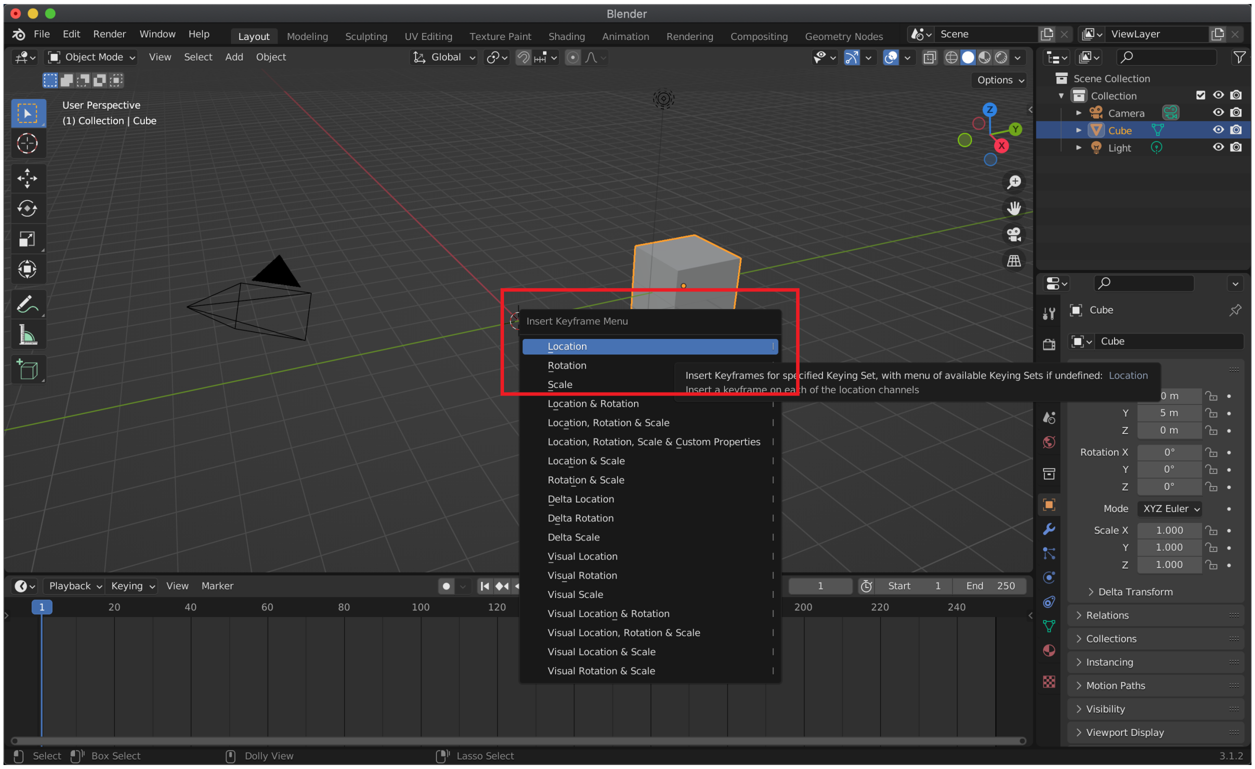


Figure 11

Now, if you scrub through the timeline by clicking and dragging the blue line from frame 1 to frame 10, you'll see your cube move smoothly from its position in your first keyframe to its position in your second keyframe. Figure 12 shows our example with the cube at frame 7 of the animation and its interpolated position values.

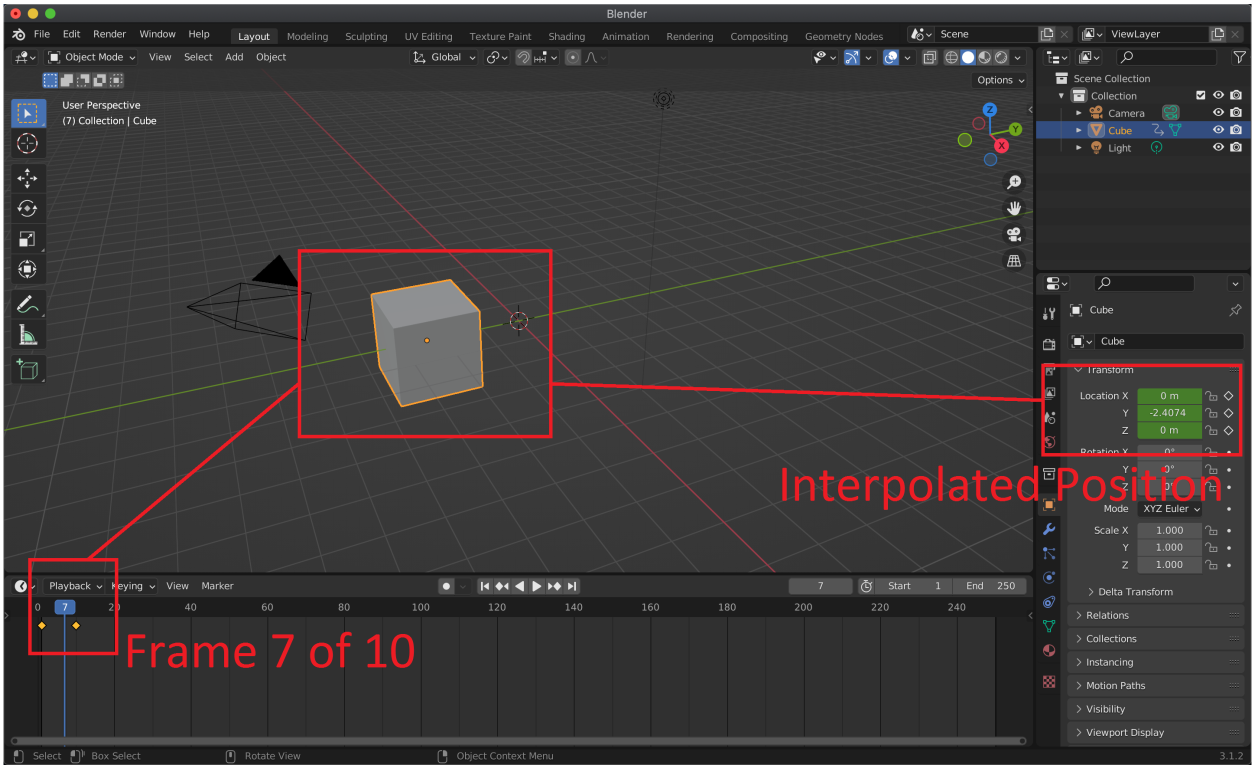


Figure 12

To have Blender ray trace your scene with motion blur, first go to **Render Properties** as usual to change the **Render Engine** to **Cycles** for ray tracing. Then scroll down in this Properties Editor to find the **Motion Blur** checkbox (see Figure 13). Just check the box, and Blender will do motion blur!

To see, render your scene as an image as you've done in previous homeworks. You should see something like that in Figure 13 pop up in your Blender Render window.

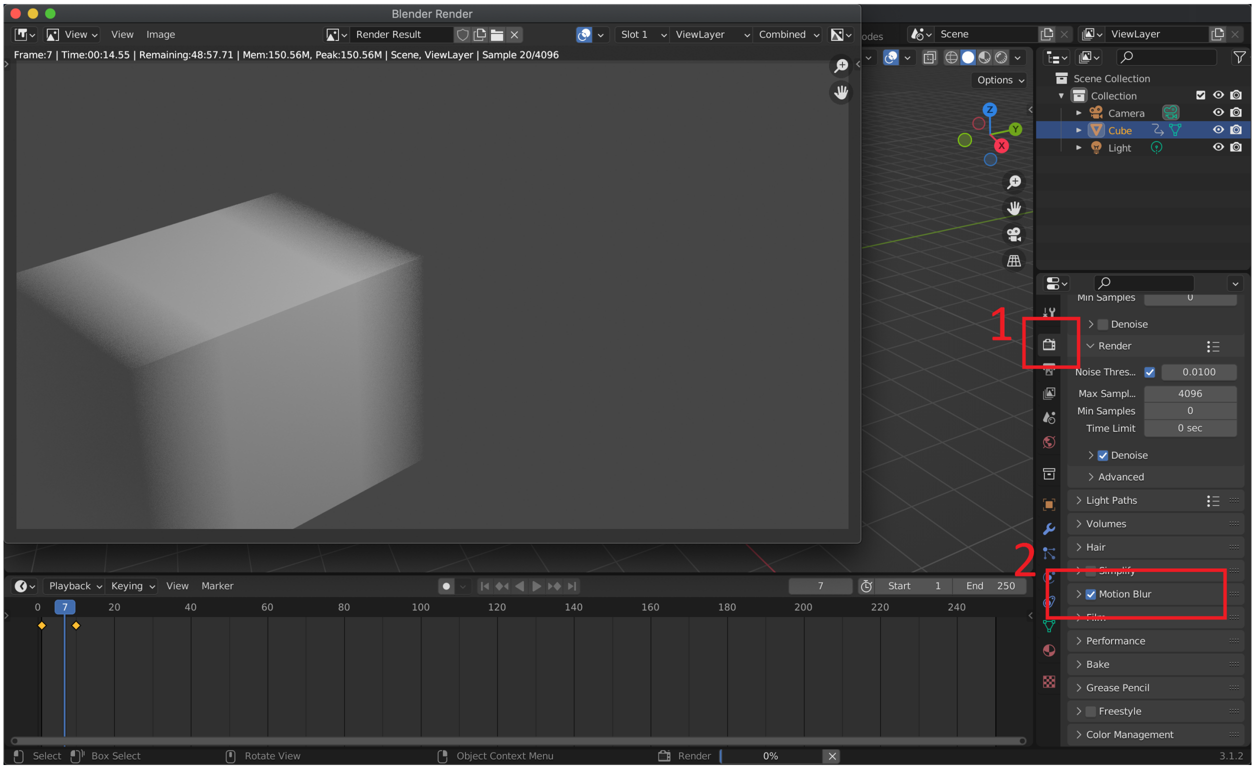


Figure 13

# 8 TODO Motion Blur

**Action:**

Use Blender to ray trace as simple or complex of a scene you want with motion blur. If you're planning to have motion blur in your final project scene, then we recommend spending some time here to make a scene that you'll use later. When you're satisfied, change the `Render Engine` to `Cycles` for ray tracing. Then, render your scene as an image as you've done in previous homeworks.

When the Blender Render window finishes, save the result to the `$CS148_DIR/HW5/images` directory. Find the `TODO Motion Blur` cell in the `$CS148_DIR/HW5/HW5.ipynb` Jupyter notebook, and edit the cell to display your saved image.

# 9 Blender Volumetrics

Volume rendering is an advanced topic that we haven't yet convered in lecture, but may be something that you want to use for your final project. It is a technique that can achieve effects that cannot be represented by surface meshes alone. More specifically, volume rendering is designed to render **volumetric objects** like smoke, fire, fog, and clouds.

We'll try out Blender's volume rendering capabilities with a simple example. For more in-depth info on all the options that Blender has to offer for volume rendering, see its manual.

For our example, we're going to set up fog. First, create a new default scene and enlarge our default cube. Then:

- Go to `Material Properties` in the Properties Editor (#1 in Figure 14).

- Click on the **Surface** option (#2 in Figure 14), then **Remove** (#3 in Figure 14) the default Principled BSDF material.
- Scroll down to the **Volume** options (#1 in Figure 15).
- Add a **Principled Volume** in the **Volume** option.
- Then, set the **Density** (of our fog) to 0.2 (#2 in Figure 15)
- Place another cube inside our enlarged cube. You may want to switch to wireframe view when doing this.

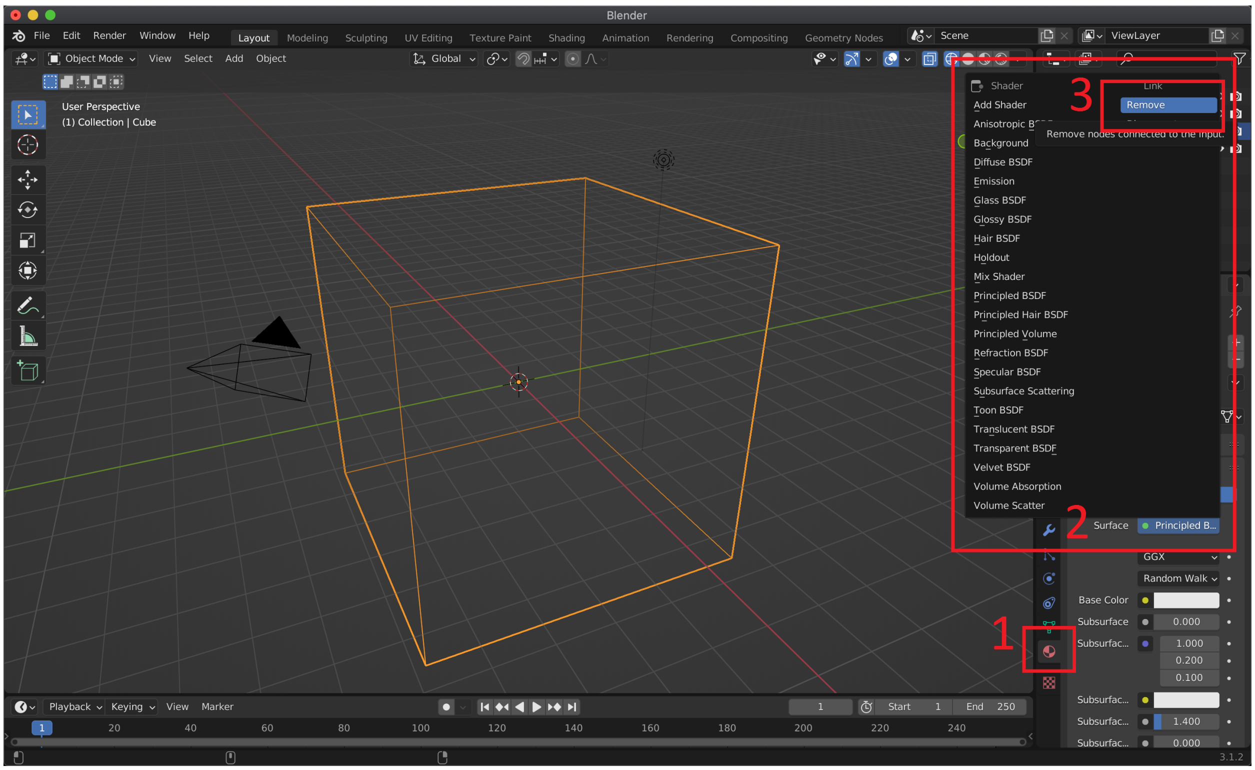


Figure 14

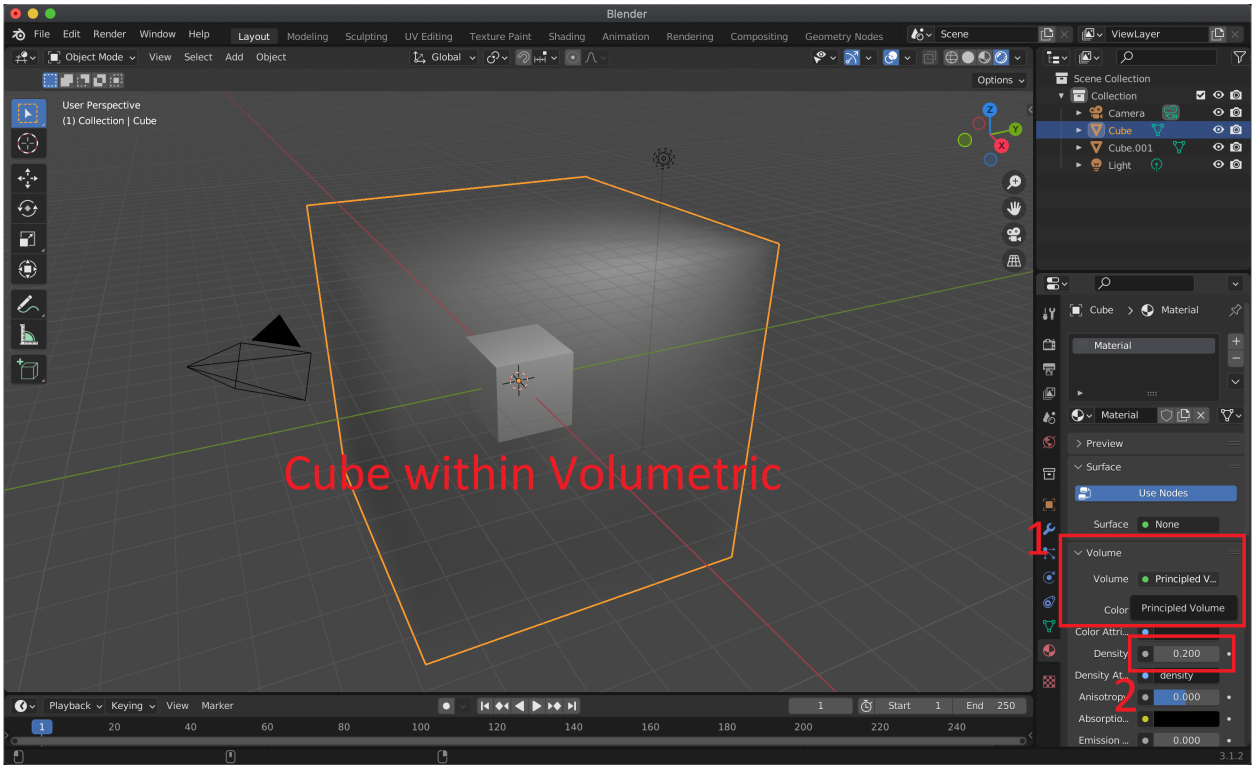


Figure 15

From here, set Blender to **Cycles** to enable ray tracing, then toggle on the cycles render preview in the viewport toolbar. You should see a fog like effect inside the outer cube surrounding your inner cube. If you render the image, then you'll see something similar to Figure 16.

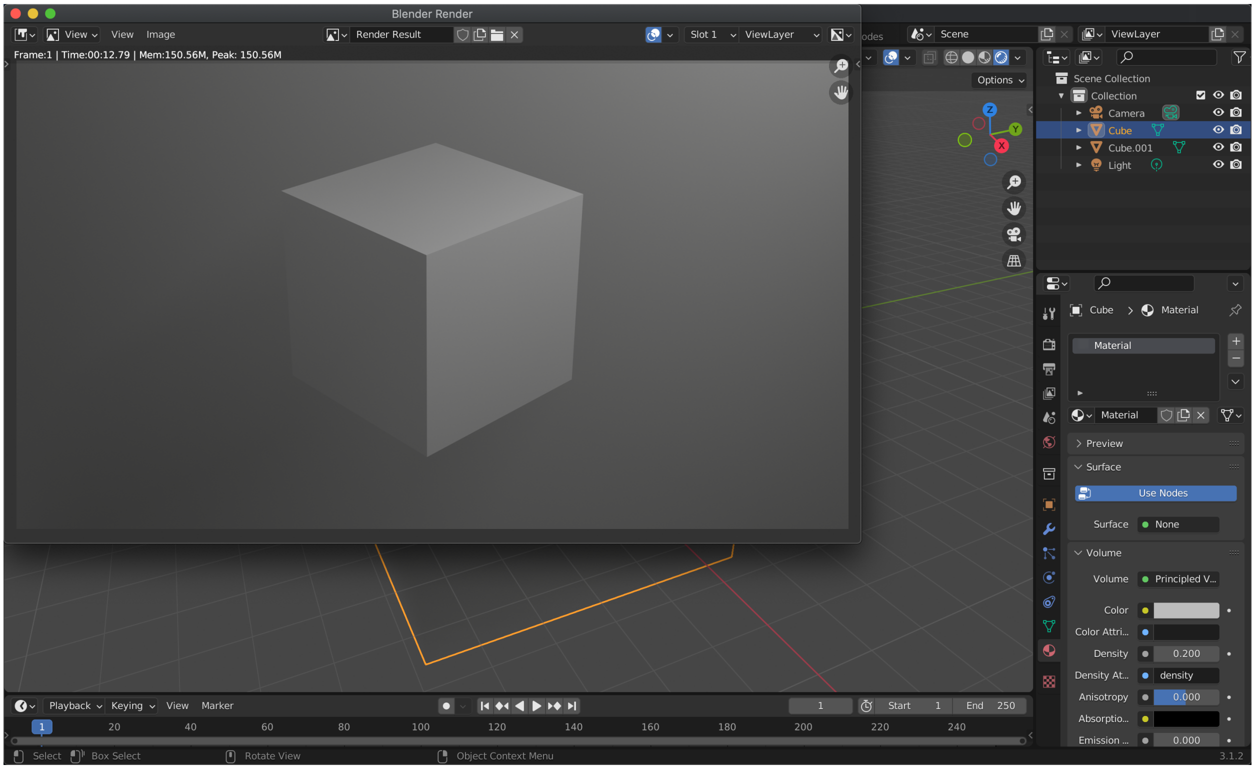


Figure 16

# 10 TODO Volume Rendering

**Action:**

Use Blender to render a scene with as simple or as complex of a volumetric effect (like fog, smoke, fire, etc) as you'd like. If you're planning to have volumetrics in your final project scene, then we recommend spending some time here to make something that you'll use later. When you're satisfied, change the `Render Engine` to `Cycles` for ray tracing. Then, render your scene as an image as you've done in previous homeworks.

When the Blender Render window finishes, save the result to the `$CS148_DIR/HW5/images` directory. Find the `TODO Volume Rendering` cell in the `$CS148_DIR/HW5/HW5.ipynb` Jupyter notebook, and edit the cell to display your saved image.

# 11 TODO Project

**Action:**

Render your current progress on your final project using the Blender `Cycles` ray tracer. Save the result to the `$CS148_DIR/HW5/images` directory. Find the `TODO Project` cell in the `$CS148_DIR/HW5/HW5.ipynb` Jupyter notebook, and edit the cell to display your saved image.

Note: this does not have to look good (yet)!