

Computer Animation I

Assignment 5: Otherworldly Still Life

In this assignment, you are to design, model, shade, and render an otherworldly still life. You will be given a scene file that has a camera and a three-light setup to use for rendering your image.

This assignment is intended to give you experience with:

- building models by moving the vertices, edges, and/or faces of polygonal primitives,
- using the parameters of the Phong shading model to achieve particular effects, and
- creating image-based texture maps and attaching them to one or more Phong attributes.

Assignment requirements:

- Your still life must contain at least three objects, each with its own version of a Phong shader.
- You must use at least three different texture maps to vary the parameters of at least three parameters of the Phong shaders you are using.
- You need to hand in both a rendered TIFF image and a project folder (which contains your final Maya ASCII file and your texture images).
- You have to build your objects from scratch, though you can use any texture images you wish.
- You are free to interpret “otherworldly” as you see fit.

The lighting and camera rig lives in the file named “a5lights.ma.” It contains a camera named `main_camera` that you should use for final rendering. This camera has three lights as children: a key light, a fill light, and a rim light. They are already configured with colors and shadows, and the scene is ready to render using the Mental Ray renderer.

Note: because the a5lights.ma file is already configured, you should import your model(s) into this file for rendering (as opposed to the other way around). It would also be okay to build your models and shaders entirely in your own copy of the a5lights.ma file, too.

Step 1: get everything you need to do the assignment, run Maya, load the scene file

- The project folder is available on the course website.
- Follow the instructions from previous assignments to copy this folder to your local machine's desktop, rename it, run Maya, set the project appropriately, etc.

Step 2: build the geometry for your still life

- Unlike in the last assignment, in this assignment I encourage you to explore the variety of polygonal modeling you can do in Maya instead of planning out carefully in advance. Grab some polygonal primitives, and move around some of their components. See what forms intrigue you, and respond to what you're seeing.
- Don't forget that you can use a transformation hierarchy if you want, as well as the object-level scale, rotate, and translate channels. The focus in this assignment is getting some intriguing objects out in front of the camera for you to shade and render.
- Explore beyond moving the components on the basic primitives. With the **polygons** menu active, grab your object and go to **Mesh->Smooth**, see what that does. You might also want to grab some edges and try **Edit Mesh->Bevel**, or grab some faces and use **Edit Mesh->Extrude**. Many of these tools can completely destroy your model, of course, so you should save often and remember that Undo (Z) is usually your friend.

Computer Animation I

Assignment 5: Otherworldly Still Life

Step 3: build the Phong shaders for your objects, with texture maps

- With an object selected, create a new Phong shader for it by going to the **rendering** menu, then selecting **Lighting/Shading->Assign New Material->Phong**. This will create a basic gray Phong shader and assign it to the selected object.
- In the attribute editor (on the right), you can see the various attributes of the Phong shader you just created. Explore what these do by clicking on them and setting different values, then rendering an image from the `main_camera`.
- Things will get more interesting when you begin to assign image maps to certain attributes of the Phong shaders. To better understand how your 2D images will map to your 3D objects, you should start by creating a texture map for the **Color** attribute for one of your shaders. You do this by clicking on the little checkerboard to the right of the attribute for that Phong shader. This will pop up a choice of all sorts of Render Nodes to create. The one you want for an image-based texture map is near the top, under the **2D Textures** section, called **File**.
- This will open up yet another node, this time a “file” node, which has an attribute called **Image Name**. Pick a JPEG image you want to map to this object, make sure you copy it into YOUR COPY of the project folder, then click on the little file folder icon to the right of **Image Name**. This will open a browser, and you should select the image that you just copied into your project folder.
- Once you’ve done that, your JPEG will be texture mapped to your shape. If you started with a basic polygonal primitive, the mapping will (hopefully) be fairly sensible. Hit render and see what you get!
- You are free to change this texture (in Photoshop or some other editing program) to better suit your object and scene, or remodel the object, or pick an entirely different image, or all of the above. The point here is for you to get experience seeing how 2D images can be used to modify attributes across a 3D object.
- Once you’ve seen how this works, you should try creating texture maps for other Phong parameters. Transparency, for instance, is an interesting one. So is Reflectivity. How do the colors/values in your texture images alter these non-color-based attributes?
- Continue changing your models, images, and/or shaders until the render of your still life looks the way you want it to. Despite all the playful exploration that is inherent in this assignment, try to make an overall image that coheres in at least some ways!

Step 4: hand in your work

- When you’re done, make sure you export a final TIFF image from the render view window (see assignment 1), and hand in the TIFF along with your complete project folder. I won’t be able to open your projects if you don’t copy the JPG files you’ve used into your project folder, so please be sure to do that. Don’t forget to name things unambiguously.

DUE Wednesday October 15th at the beginning of class