CS148 Final Project (Summer 2023)

By Romrawin Chumpu (Jin) (romrawin@stanford.edu)

Final Scene Submission

Variation A

Variation B

Inspiration

My final project is inspired by the lantern festival in my hometown (Chiang Mai, Thailand) when people gather on full moon nights in November to float lanterns up into the sky over the river. That day is usually coming around my birthday. This festival always makes me feel blessed. This scene also recalls Tangled's lantern scene. Tangled is also one of my favorite movies from my childhood. I would like to make a scene that combines these two styles for the final project.

Figure 1 (Left) lantern festival (right) Tangled's lantern scene.

Project Requirements

- *Leveraging the power of ray tracing:* light and reflection of lanterns in the water, beneath water paddles.
- *Main geometry from scratch:* lanterns and a character.
- *UV mapping and texturing from scratch*: almost all of them in the scene.
- *Blender/Cycles advanced feature:* volumetric rendering, geometry nodes, depth of field, and shader nodes.
- *Cite your sources:* all sources will be cited in attached links along in the procedure section.

Member duty – I did it all.

Procedure

The procedure for making this final project will be explained in two parts: *scene* and *character*.

Scene

1. *Lantern* – Five color lantern variations are made from scratch with each customized illuminated material. Shader nodes consist of noise texture, UV map, translucent, emission, gradient of color, and alpha map applied along the z axis (Figure 3).

Figure 2 (left) five color lantern variations (right) lantern texture.

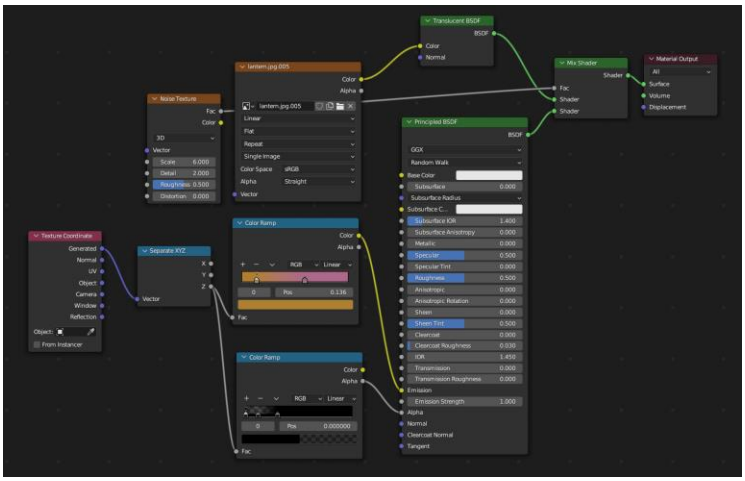

Figure 3 Shader node for lanterns.

2. *Distributed lanterns over background* – Lantern groups are duplicated and distributed above the sky by using geometry nodes. The process is generally based on mathematical equations (Figure 5). I studied each node from scratch by trial-and-error and found that this is visually simple and interesting.

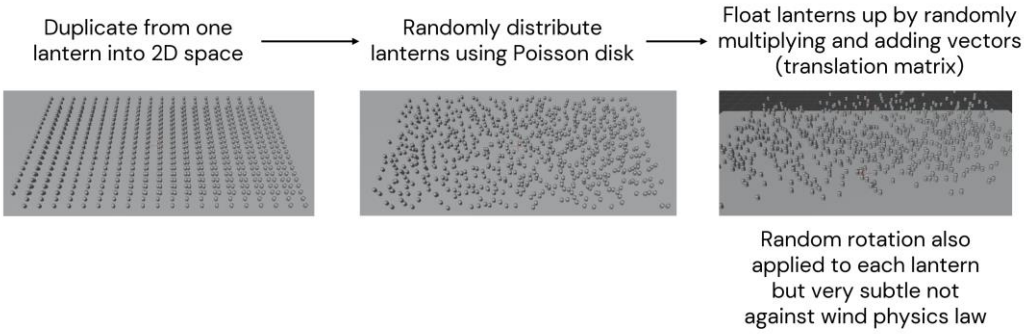

Figure 4 Floating lanterns process.

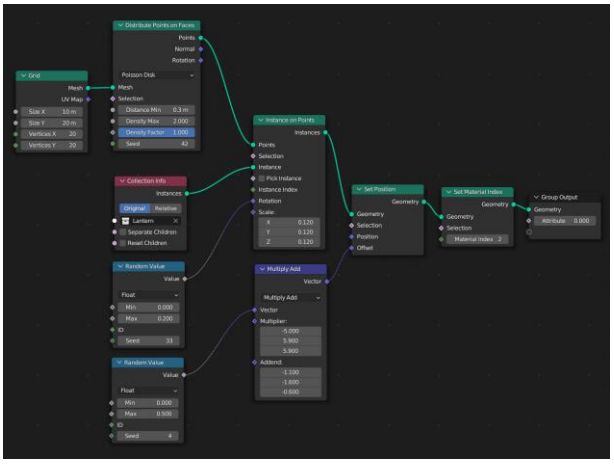

Figure 5 Geometry node of distributed lanterns.

3. *River* – River is made from a plane by a customized shader node with effects of reflection, refraction, transparency, Fresnel, and normal map.

Figure 6 (left) water normal map (middle) effect of water reflection (right) transparent effect.

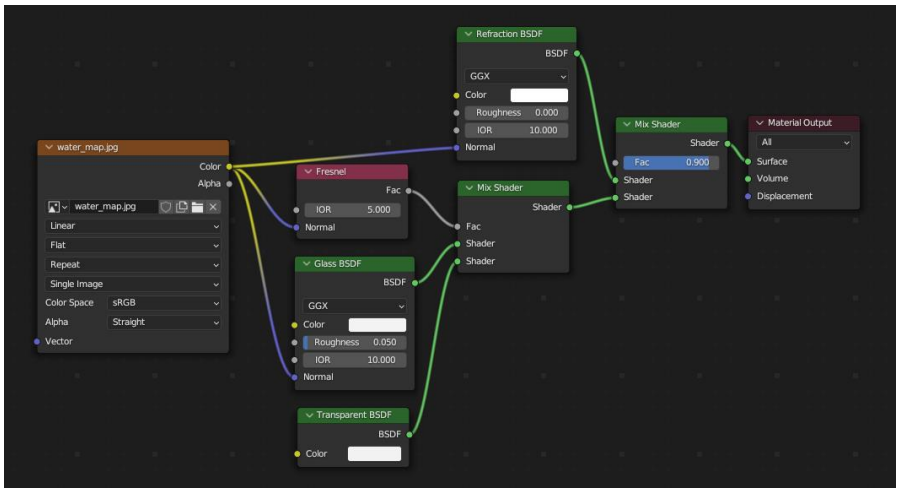

Figure 7 River shader node.

4. *Volumetric scattering* - In my scene, the volumetric feature is added behind the scenes setting. Volumetric scattering and rendering purposefully diffuse the lights and blur the lanterns in the background. The volume fog works well with a blurred background, but there is a limitation with the edge between water reflection and volume. I added a fader (alpha color ramp with vector mapping) along with the global coordinates of scene direction.

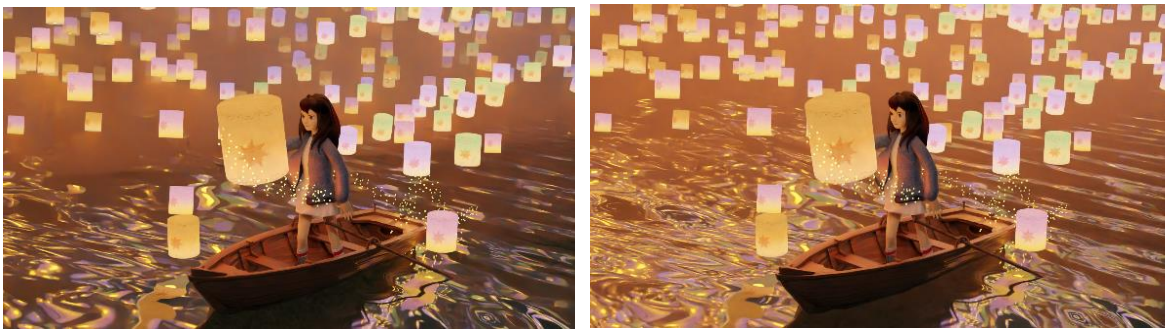

Figure 8 (left) before (right) after applying fader.

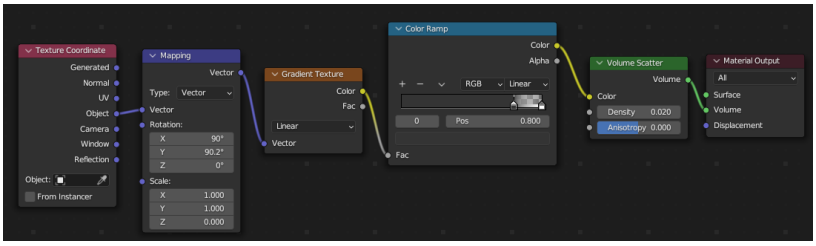

Figure 9 Volume scatter shader node.

5. *Background filter* – The purpose of the background filter is to layer the lanterns that are further away and dimmer the lights in the background. The material of the filter is black transparent glass with a 1.0 IOR.

Figure 10 (left) Background filters (middle) before (right) after applying background filtering.

Figure 11 Background filter shader node.

6. *Small light* – The light is created by myself using geometry nodes. This process is again based on mathematical equations. I made the emission points light by attaching them and following them along the curve object, randomly scaling up the size of the floating light points.

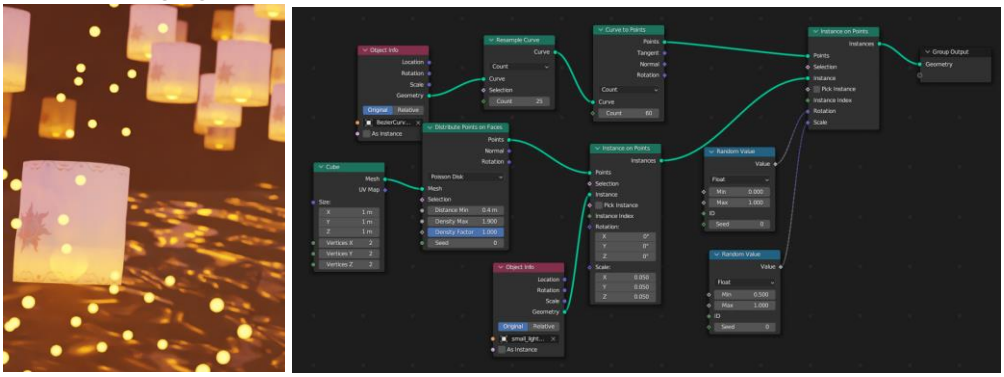

Figure 12 (left) Small light example (right) geometry nodes.

7. *HDRIs* – The mountain background is downloaded from Poly Heaven (Alps Field HDRI • Poly Haven). To make it look darker, I set the strength in the world to be 0.05.
8. *Boat* – The boat is downloaded from CGTrader (Wooden Boat free VR / AR / low-poly 3D model | CGTrader). The material of the boat is customized from HW2-3.

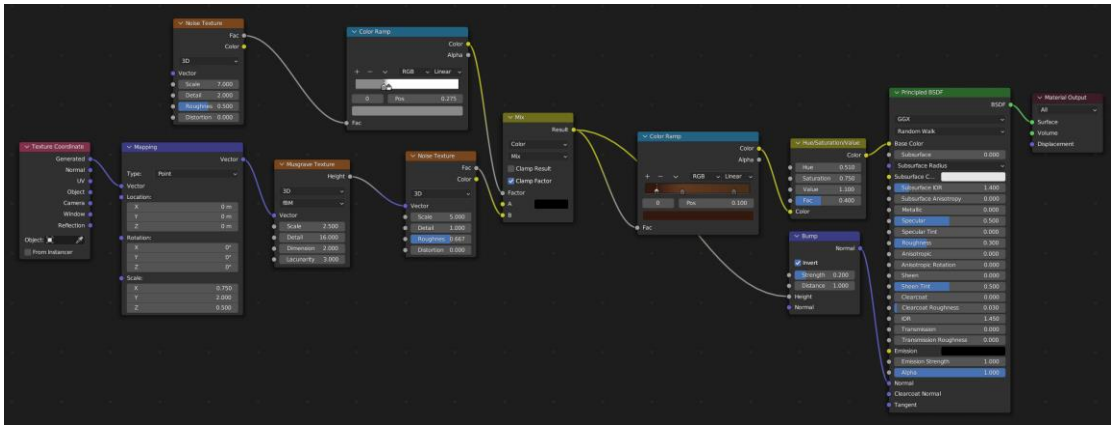

Figure 13 Boat wooden shader node.

Character

1. *Head and face* – I sculpted the human head and face from scratch by following this tutorial and customizing from this reference template. I created eyebrows and eyelids by following the same tutorial. I think the face is the hardest part.

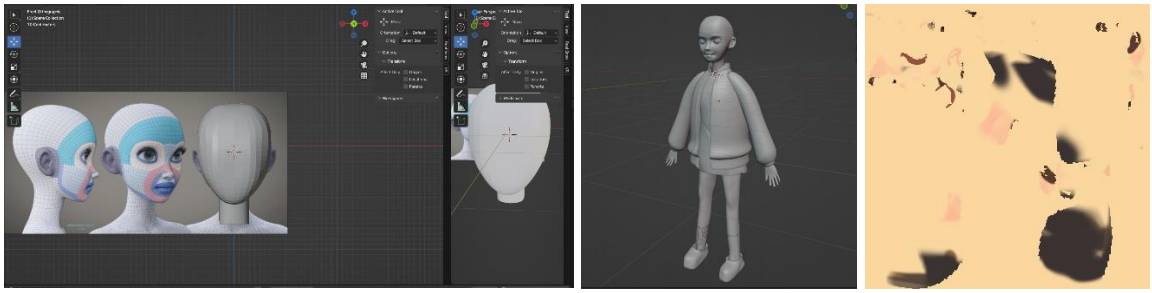

Figure 14 (left) Head sculpture at the beginning of the process (middle) first human prototype (right) face UV texture.

Face texture is created using UV unwarp and then painted with texture painting.

2. *Body* – I sculpted the human body from scratch by adding cubes, spheres, and cylinders and went randomly through trial-and-error using my sense of sculpturing. The material of the body and skin areas is set to have subsurface scattering.

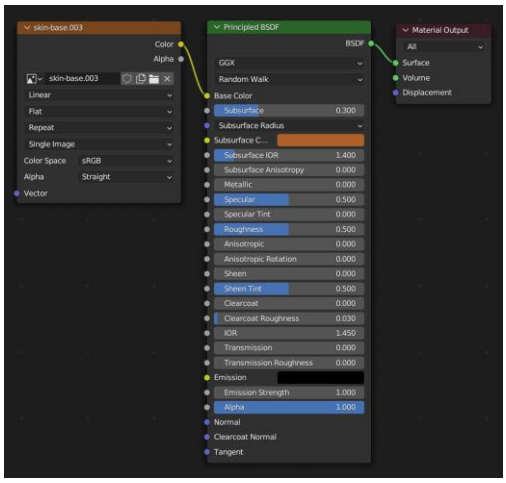

Figure 15 Skin shader node.

3. *Clothes* (Figure 13 on the right)
    a. *Cardigan*
        i. This cardigan was made by following this tutorial and customizing it in my own style.
        ii. Cardigan fabric texture images are downloaded from ambientCG (Fabric 018 on ambientCG). I painted it blue and created the shader node myself. I learned and understood how to add normal maps together.

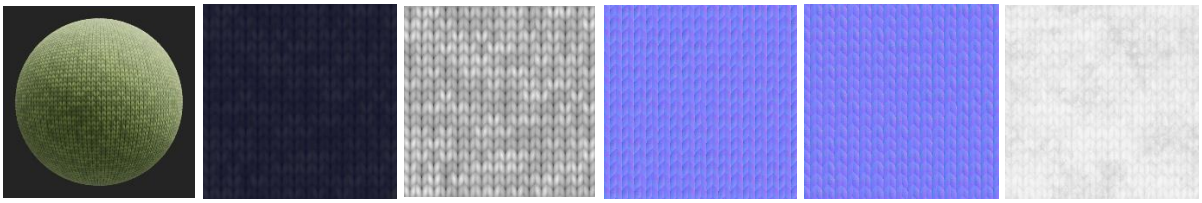

Figure 16 Cardigan image texture.

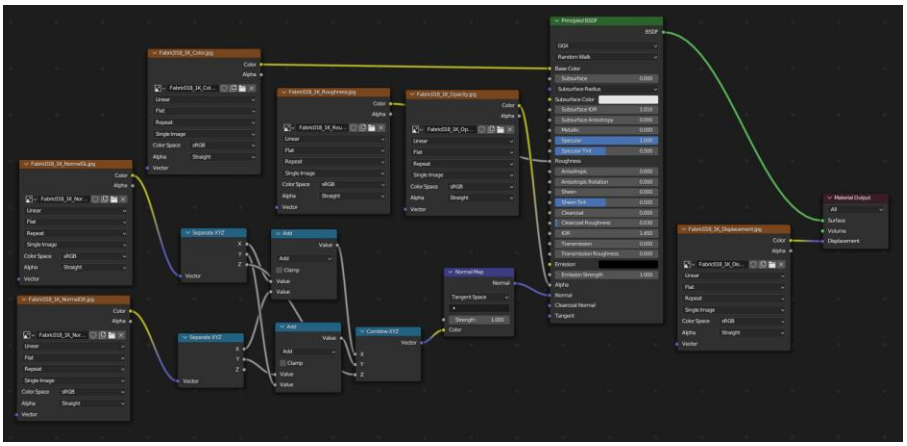

Figure 17 Cardigan shader node.

b. *Dress*

i. I followed this tutorial and customizing it in my own style.

ii. The material is customized by myself to have a fabric texture supported by the incoming light.

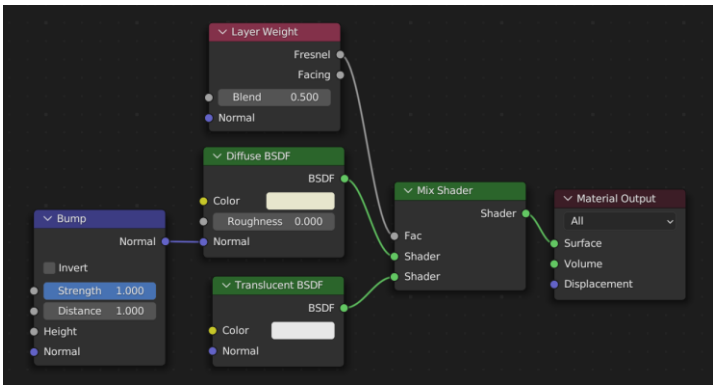

Figure 18 Dress shader node.

c. *Socks and shoes* – These parts are using UV unwarping and texture painting.

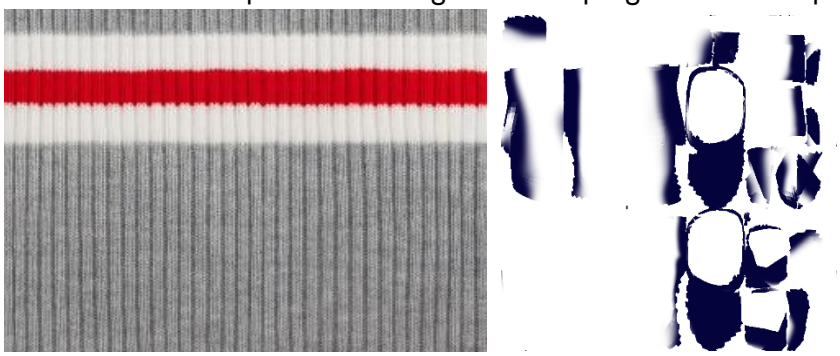

Figure 19 Socks and shoes texture.

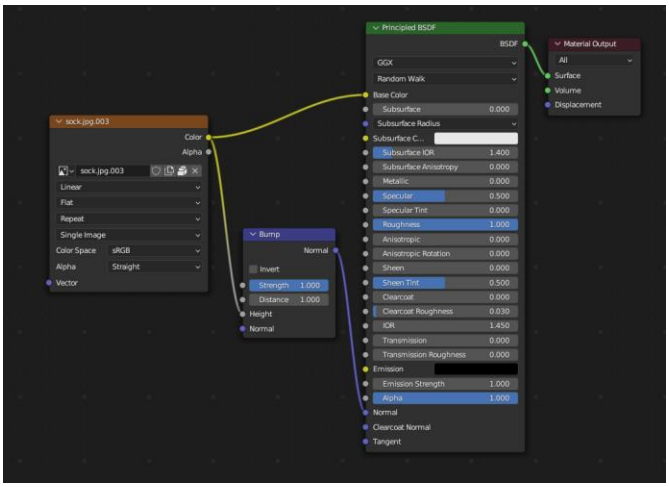

Figure 20 Socks shader node

4. *Hair* – Originally, the set of hair was downloaded from CGtrader (Stylized hair free 3D model | CGTrader). I customized it to fit the shape of my own human head. The hair texture was downloaded from this source and painted a darker color. The shader node is customized and adds more texture and subsurface scattering for reflecting light onto the hair and making it look more like lines.

Figure 21 (left) Hair lighting test (right) hair texture.

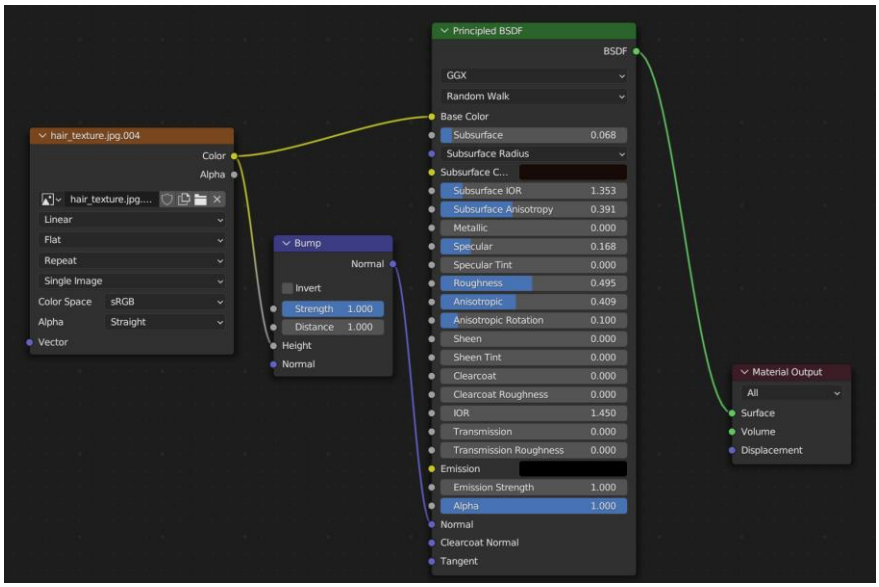

Figure 22 Hair shader node.

5. *Eyes* – Eyes consist of two layers: UV texture and glass (Figure 19). The eye texture is downloaded from this source. I painted the iris a darker color to match human hair.

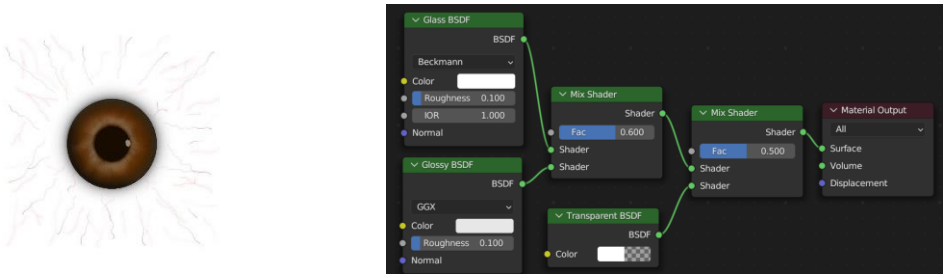

Figure 23 (left) Eye texture (right) glass layer shader node.

6. *Rig* – I followed this tutorial for generating human poses.

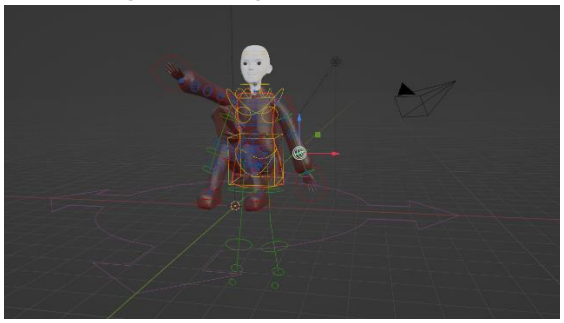

Figure 24 Rigging process.

7. *Posture* – I designed and sculpted the human pose by myself after rigging. The human posture will be the main part of the story in the scene. I made it appear as if this character is trying to raise the lantern.

Lessons learned

- Python scripts work only 2 – 50 objects. If there are 1000 objects, we should use geometric nodes with mathematical equations. Because geometric nodes are designed for repetitive object embedding under the Blender backend.
- Just the corner of the mouth can change the whole face's expression.

- Automatic rigging in Blender should start with the T-pose model.
- Every solving technique in my scene that I can think of comes from simple math!
- Be patient, especially when sculpting a model and rendering an image.
- The direction of little light matters! On the left it looks like a lantern is floating down and on the right one it looks like the is floating up.

- I should add a disclaimer - this is not an AI-generated image!