Assignment 3

Visual Computing Fundamentals

Pablo Díaz Viñambres : *pablodi@ntnu.no*

October 3, 2022

Contents

1 Task 1: More polygons than you can shake a stick at

c) Normal vector surface coloring

To start with this task, we first loaded the mesh and modified the VAO function to support normal vectors. After passing it to the shaders, we could change the RGB values of the moon surface the normal vector information. Since most of the surface is flat, its normal vectors don't change that much, and that's why we predominantly see green. However, on the crater we can also see both blue and red since the normals change a lot more.

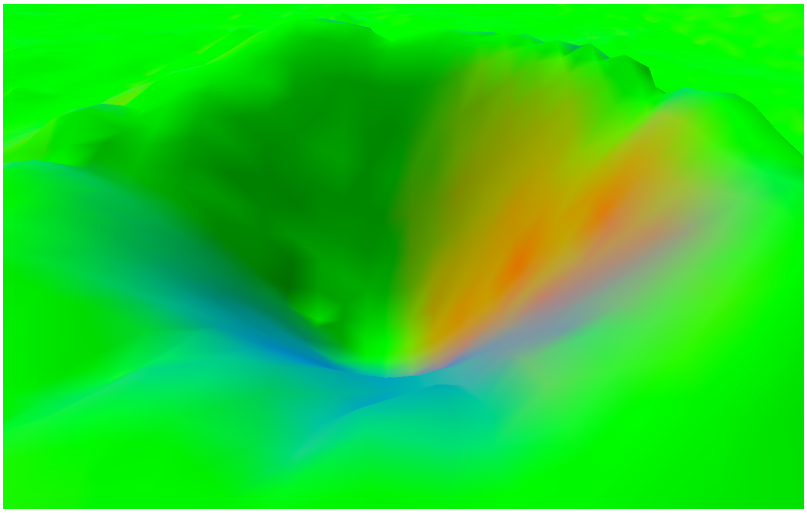


Figure 1: A crater on the moon with its RGB values set to the normal vector on each point.

d) Simple lightning

For this section, we implemented the given formula on the fragment shader and got a lightning model based on the Lambertian model. The results are very basic but you can definitely appreciate the depth and details of the surface.

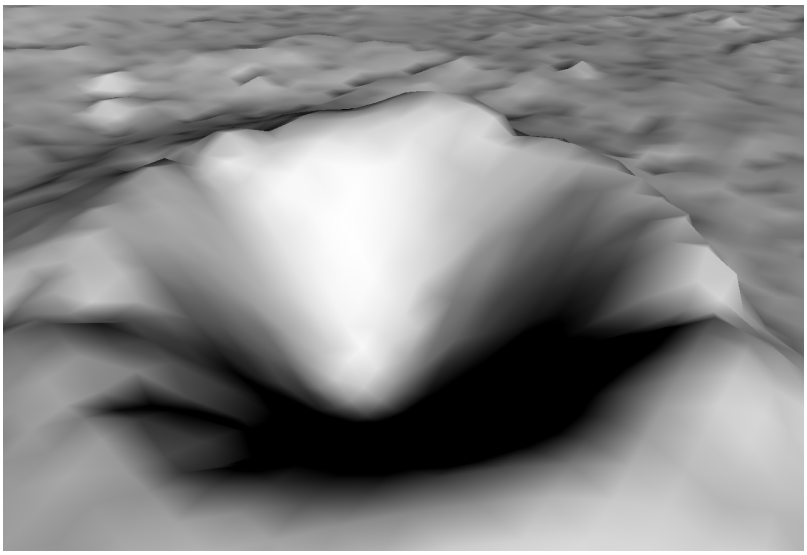


Figure 2: The moon surface with the simple lightning model implemented.

2 Task 2: Helicopter Parenting

c) Helicopter drawing

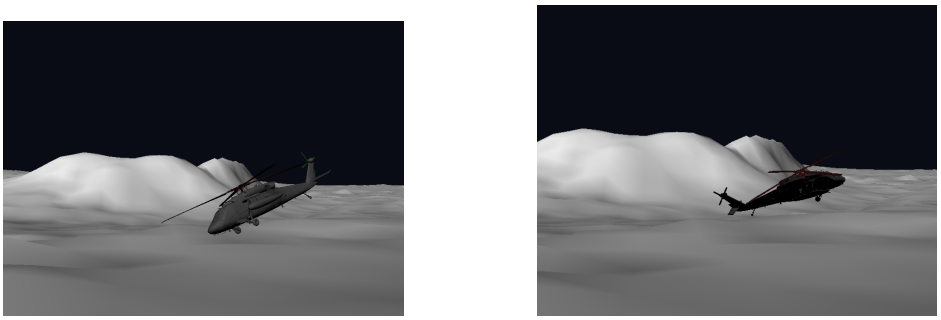
After building the Scene Graph, loading the new meshes, computing the transformations in the correct order (which was pretty hard to get right) and refactoring a lot of code, we could finally render the helicopter. It was also later encapsulated on a Helicopter class for making the next tasks implementation cleaner.

Figure 3: The helicopter being drawn in our scene.

3 Task 5: Help! My lighting is wrong!

a) Incorrect lightning in animated objects

Since the lightning model only considers the normal vectors of the original model, whenever we rotate or move our helicopter on the scene our shading doesn't change. This makes the original dark part of the helicopter to still stay dark even when it's facing the light source.

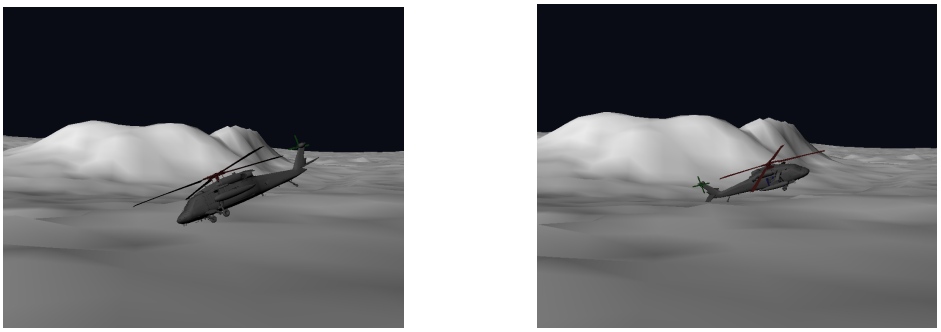


(a) Correct illumination on one of the faces. (b) The other face is still on the dark.

Figure 4: Incorrect lightning when animating the helicopter, the original dark side stays dark even when facing the light source.

c) Approximated Normal Matrix fix

After applying the fix, the illumination becomes dynamic and much more realistic.



(a) Correct illumination on one of the faces. (b) Now the other face is also reflecting light.

Figure 5: The lightning now displays correctly on both helicopter faces, the parts facing the light source correctly reflect its light.

4 Task 6: Time to turn this thing up to ~~11~~ 5

a) Multiple helicopter drawing

For this task, we made a vector of helicopter objects and animated them on a loop with some offset so they don't overlap. Personally, I don't recommend setting the variable `num_of_helicopters` to very high values, I tried 1001 and my system froze for 2 minutes.

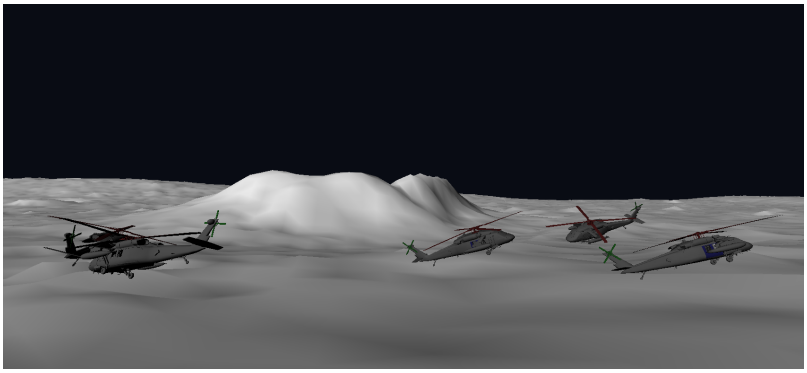


Figure 6: 5 helicopters flying around the path, the animation offset is set to 10.0.

5 Task 7: Optional Challenges

d) Make the door animated

For this effect, instead of simply sliding the door in the Z axis, I wanted it to rotate on its hinge realistically. For accomplishing this, I first located the door mesh inside the helicopter model and tried on setting its first vertex point as the door node reference point. Surprisingly, this worked perfectly (since it's probably the upper or lower left corner vertex) and rotating the door on the Y axis led to the desired motion.

For controlling the rotation, I added two methods on the Helicopter class. The first one, `open_door`, behaves like the animation function, being called on each frame to smoothly change the rotation angle. The second one, `toggle_door`, changes the door status from opening to closing. It is **called upon pressing the key 'o'**, but can only be done so every 12 frames, to avoid twitching.

The door can only open up to a $\frac{\pi}{5}$ angle since going further would make a collision with the right side of the helicopter.

Figure 7: The door on its open position.

f) Find the easter egg

I'm sure this extra 0.01 points will be essential for me passing this course!

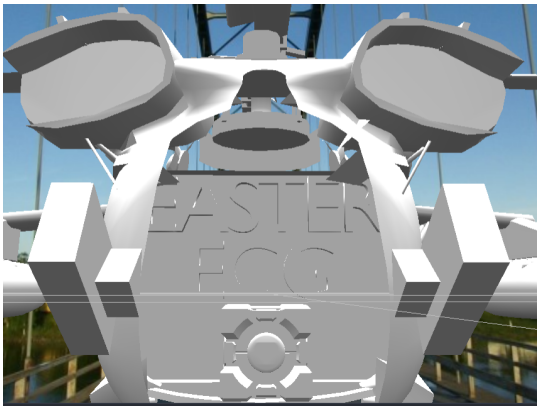


Figure 8: I love Portal as well :)