Project

Three-Dimensional Visualization and Animation
IST 2022/2023

Group 2

Afonso Ribeiro - 89400 | Diogo Ravasco - 89434 | Bernardo Jordão - 95540

Trailer video

Github link to project

This project aims to to recreate a Mars rover mission scenario. The idea is to model both the rover and the terrain, in such a way that an interactive animation can be produced.
First a basic model of the scene and rover was created. This had no special effects, having the scene consist of nothing more than simple primitives and the aesthetic aspect going no further than the application of textures to said primitives then special effects and other techniques were employed to enrich the scene. Scene was rendered using Blinn-Phong Shading Model.

1 Assimp library to load objects

The rover model was loaded through the assimp library, which converts the obj file into a mesh we can render.

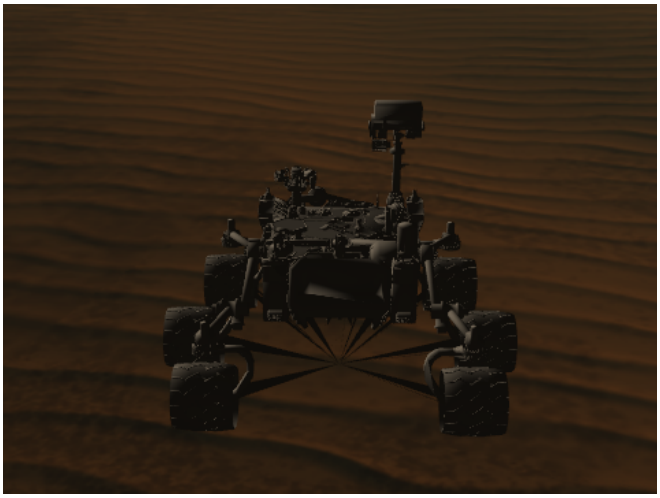

Figure 1: Rover object loaded with Assimp

2 Rear View

Parallel to the standard camera, a second camera was introduced, but while the first "shoots" what the user intends to see, by controlling it with the mouse, the second is pointed directly backwards from the rover.

In order to display both cameras the stencil method was employed. In this implementation the cleared stencil buffer was set to ones and when creating the stencil they are replaced with zeroes, more about this choice ahead in the Planar shadows and reflections section.

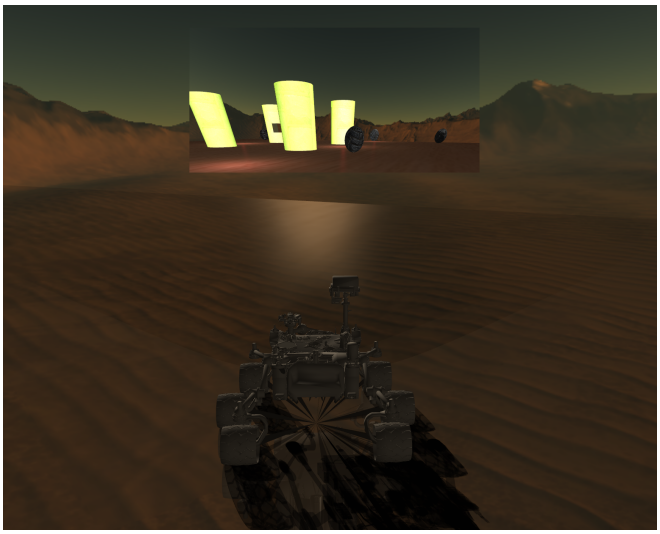

Figure 2: Showcase of rear view camera

3 Billboards

The billboard technique is used to implement a mesh that is always facing the camera. This gives the illusion of a 3D object without compromising the polygons budget. In our project it was implemented in the form of clouds in the sky, as shown below.

Figure 3: Cloud billboards above the scene

4 Particle system

The particle system was implemented in the form of fireworks, that show up in the scene at every 10 points. Each firework is composed of approximately 1500 particles, which are basically quad meshes with a bright texture. When each particle is created a life time is associated to it and after each frame this value is reduced, until it get to 0 and the particle "dies".

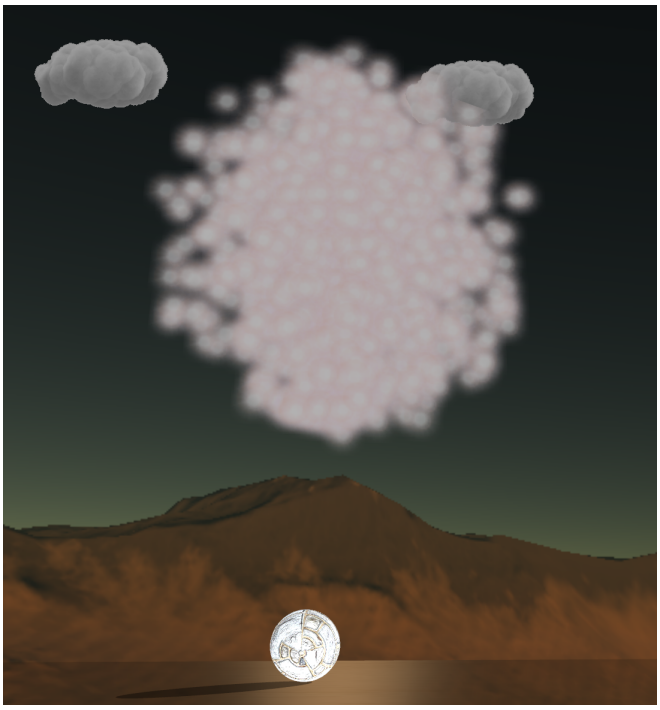

Figure 4: Particle system

5 2D lens flare

The lens flare is the optical effect created by the scattering of light in a lens system. This is approximated in our project by using several 2D objects running across a line that intersects the center of the screen and the light position, as is shown in the provided example.

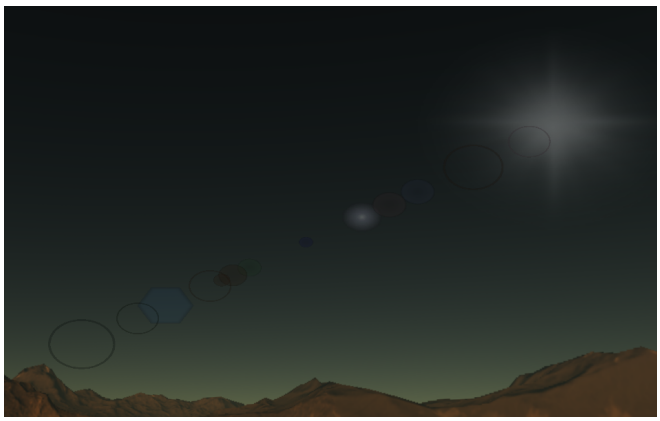

Figure 5: Lens flare with light position some distance from view port center

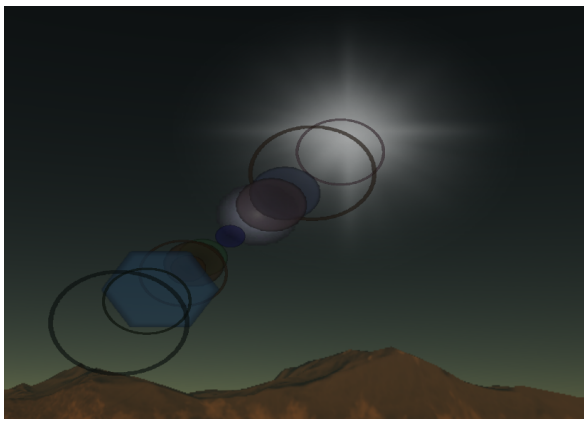

Figure 6: Lens flare with light position close to view port center

6 Planar shadows and Planar reflections

Both these features were implemented, using the techniques taught in the lectures.

- Planar reflections were obtained by creating a stencil in the shape of the "floor", rendering the scene with the objects and lights mirrored by the "floor" followed by a rendering of the floor quad with blending enabled.

- Planar reflections were obtained by first rendering the floor quad and then darkening the color of a pixel in the color buffer, if the projection of a given object on said floor quad overlaps with that particular pixel.

In the rear view camera we discussed our choice of which values we used for the stencil buffer, we made the conscious choice of not having planar reflections and shadows in the rear view camera, as we saw it as a valuable insight into what the scene would like with and without those techniques applied concurrently. If, however, we were to have those effects replicated in both cameras our choice would then be different.
In order to have that effect our buffer would have a initial and clearing value of 2. When creating the rear view camera the values would be replaced by 0, those that fall within that rear view camera view port that is. Then when the planar techniques were being employed we would render the floor quad and increment the values in the stencil buffer, this would mean:

- Values within the rear view go from 0 to 1
- Values outside the rear view go from 2 to 3

Depending on which camera we were rendering at the time we would have the stencil test for the planar reflections be either EQUAL to 1 - rear view camera - or EQUAL to 3 - standard camera.
Then after this step we would then again render the floor and decrement the stencil buffer, thus basically rolling back the stencil buffer to what it was at the instant of the first rear view camera stencil making.

6.0.1 Reflections

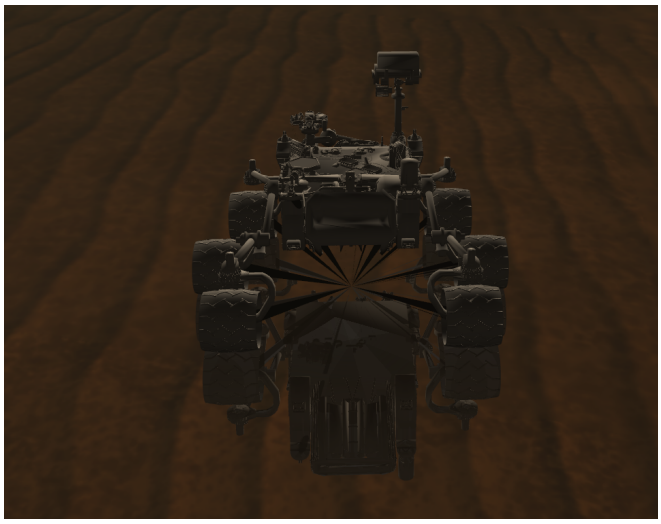

Figure 7: Planar reflection of rover with planar shadows disabled

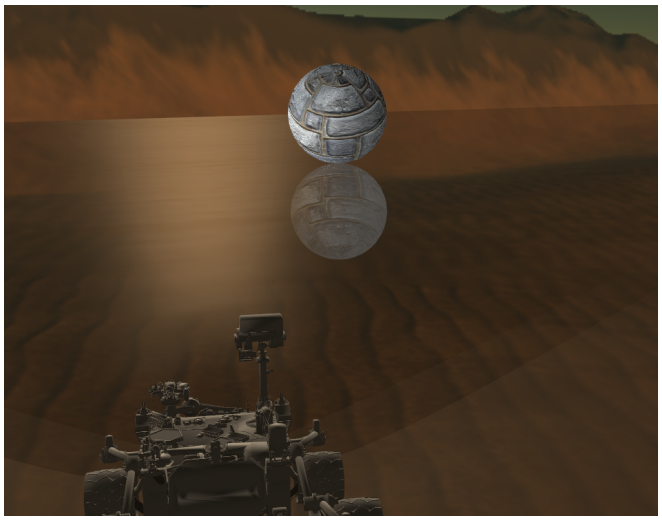

Figure 8: Planar reflection of a rock with planar shadows disabled

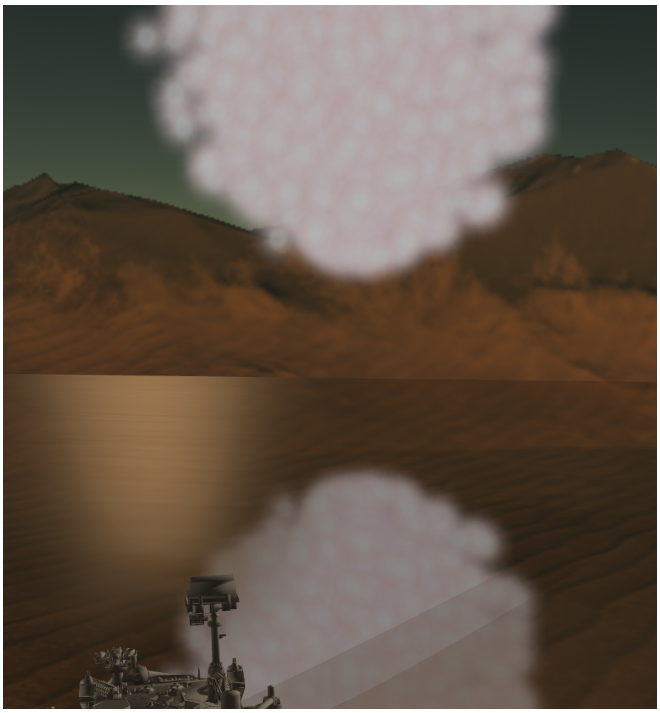

Figure 9: Planar reflection of particle system with planar shadows disabled

6.0.2 Shadows

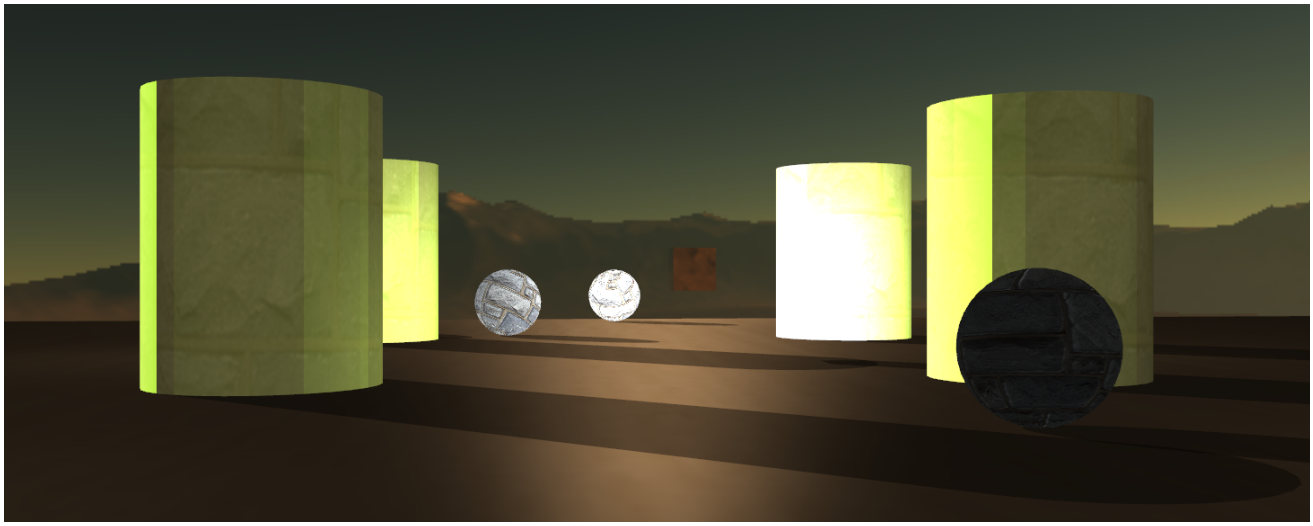

Figure 10: Planar shadows with planar reflections turned off

7 Bump-mapping

This technique maintains the shading model, but instead of using the interpolated normals it looks up the normal at a given point from a texture file. We decided to apply this technique to the rocks, their inherent movement makes them better objects through which to display the technique.

Figure 11: Bump maps when only the directional light is turned on

Figure 12: Bump maps when only the point lights are turned on

Figure 13: Bump maps when only the spotlights are turned on

8 Skybox

The Skybox provides the illusion of a far away 3D background. This is implemented in the form of a big textured cube that envelops the whole scene. The only caveat is that the cube must appear to be at the same distance, regardless of where we are in the scene, to give the proper illusion of the far away background.

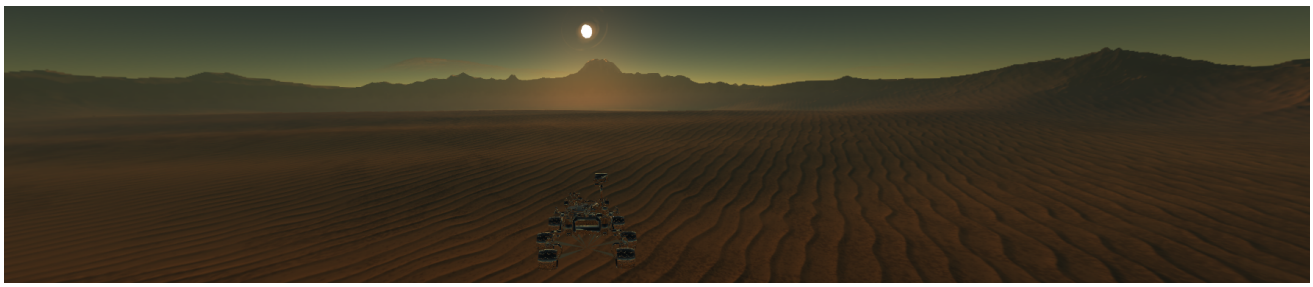

Figure 14: Skybox

9 Environment Cube Mapping

Environment mapping is a technique used to simulate the reflection of the scene in an object. In our project it is implemented in the form of a reflective cube, which reflects the whole Skybox.

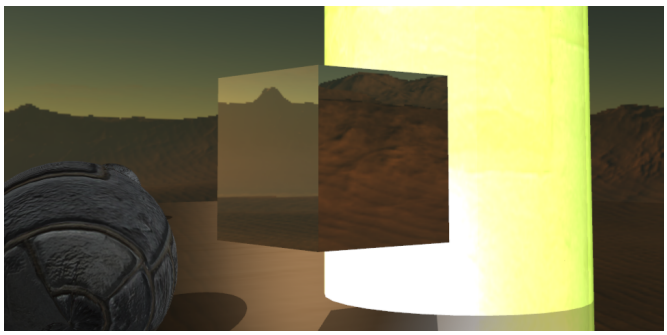

Figure 15: Environment cube mapping

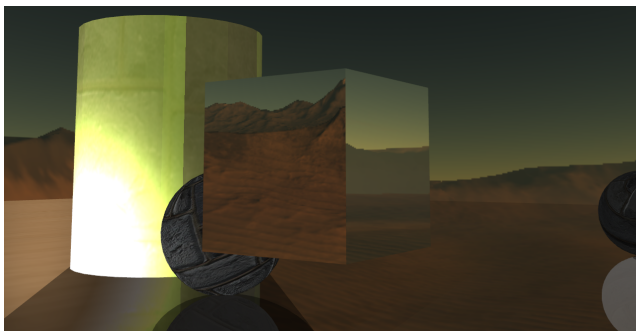

Figure 16: Environment cube mapping from different angle