Introduction to Computer Graphics

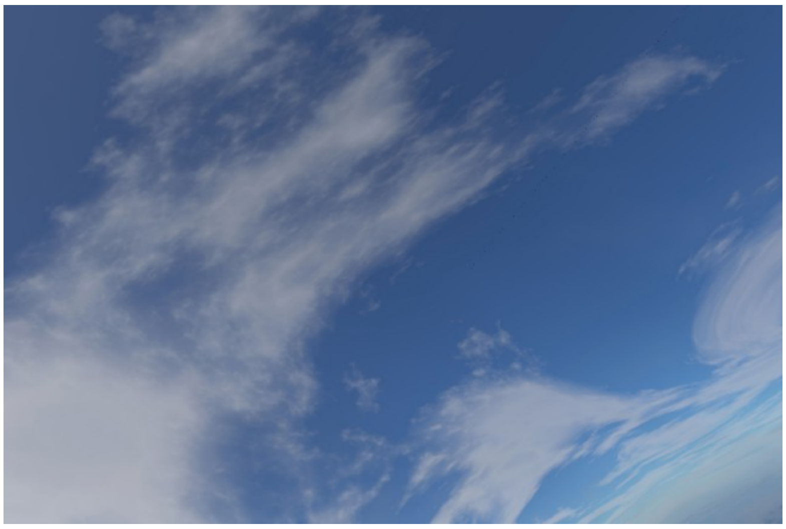
Project report (part 2)

Kim Lan Phan Hoang, Tim Nguyen, Lucie Perrotta

Required

Sky modeling – Skybox

The implementation of the skybox was straightforward. We implemented a cube just like in the homework and replaced the ".tga" texture with a sky texture. The cube is not very large but large enough to englobe both the mountains and the camera. Keeping the cube small avoid useless computation. However, we want the sky to be in background, for that purpose we disable the GL_DEPTH_TEST to be sure the sky will always be in the back.
Here is the skybox as seen from its inside.


A part of the Skybox

Required

Water modeling

We create a new framebuffer to contain the reflection image, which contain the grid (cut until a water level) and the sky drawn using both inverted camera position and lookup. Once done it can be mapped an a plate grid. The opacity of the obtained water is a little reduce so that it is possible to see the root of the mountains through it.
We also added some movement to the water by computing sin-cos waves multiplied by the time (as seen in the homework).

A lake

Optional

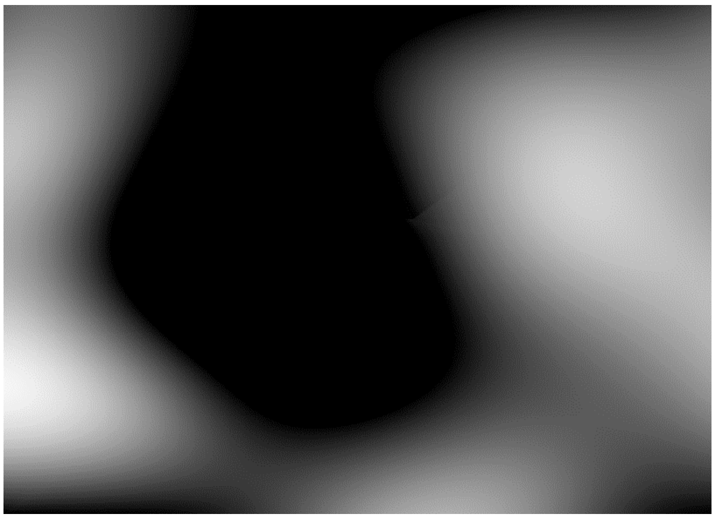
Keyboards controls

To complete the trackball, keyboard controls had been implemented. Directionnal keys are moving the camera position (and the camera lookup) to be able to move around the mountains. AWSD keys are moving the lookup of the camera so that we can change the view. XY add the possibility to zoom in and out (to see the mountains closer).

Optional

Simplex noise

We have implemented a simplex noise, as a function of a uv vector returning a black and white value, just like we did for the Perlin noise. We simply compute for each point whether it is in the upper or the bottom simplex (which are triangles in the 2d case). For this code, which needs many heavy computations, such as square roots and such, we chose to use numerical approximations, as suggested on glsl blogs. Typically, we used the approximations proposed by Stefan Gustavson on his blog, to help the code run clearly faster.



The simplex noise

Optional

Multifractal motion and ridged fBm

In extension to the simplex noise, we also implemented the ridged fBm and the multifractal motion, giving a nicer grain than the fBm.

The multifractal motion

Optional

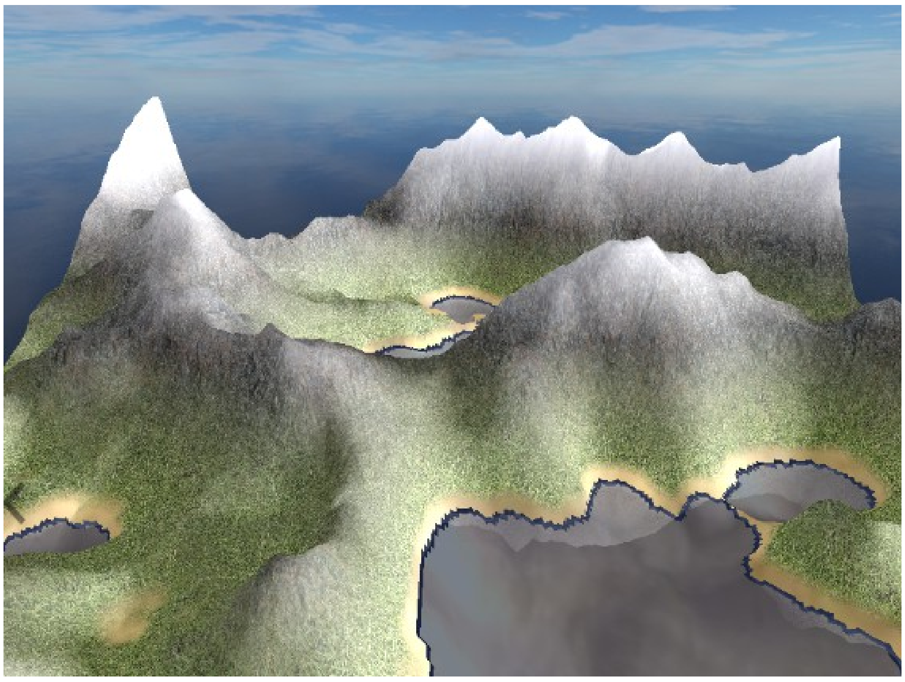
Trackball

We also took the trackball from the homework to have a better idea of what's going on.

Required

Improved texturing

To improve our textures, we first load some images representing the sand, grass, rock, and snow in tga files. The dimension of these images should be a multiple of 4. Then the color of each pixel is mapped to our textures, depending on the height. We used the mix() function to blend the transition between the textures.
We also scaled down our textures by using a scale_factor on our uv coordinates to make them smaller and repeat the textures so they seem realistic with the sizes of our mountains.


Our texturized landscape

Implementation

Keyboard controls – Kim
Textures - Tim
Water modeling – Kim & Tim
Improvements of the perlin noise – Lucie
Trackball – Lucie
Skybox – Lucie

Fraction of the workload of each group member

Kim : 33%
Lucie : 34%
Tim: 33%