FINAL REPORT

Thevie Mortiniera, Melik Thévoz, Patrik Wagner

Here are the features implemented beyond the minimum requirement :

- Water reflection : We flipped the camera along the horizontal plane to get a mirrored texture of the terrain and the sky that we use to texture the water by mixing it with a phong shading.

- Wave like animation : We use our perlin noise implementation to simulate waves that move according to the time.

- Infinite map : We build a big grid, each time we arrive to a limit close to the borders of this grid, we regenerate a new grid at this position. In our first implementation, we had an array with multiple grids to give the impression of an infinite size.

- Day/Sunset/Night : We can switch between three different day times by using the keyboard.

- Texture mixing : We mix our different textures in proportions depending on the height/normals to get a smoother transition between the layers and a more natural general feel.

This is how we split the work among us :

- Patrik took care of the Bezier algorithms implementation, and everything related to the camera and infinite map.
- Melik and Thevie took care of the texturing (including the sky box) and the water (reflection) implementation
- The rest of the work (Perlin, ...) and most of the debugging was done all together.

External code references :

- Perlin noise was implemented by us based on the algorithm presented in the slides, and not using the available source code. However we used a piece of code from this page :
http://stackoverflow.com/questions/4200224/random-noise-functions-for-glsl as a random generator for our Perlin.

- Ridged multifractal algorithm was found in these websites :
https://engineering.purdue.edu/~ebertd/texture/1stEdition/musgrave/musgrave.c
https://github.com/jdupuy/fractalTerrain/blob/master/terrain.glsl

- We only used the slides of the course for the Bezier curve implementation.

- Finally, we looked at online OpenGL libraries for the functions we used in our code. (khronos, opengl.org ...)