Computer Graphics Project

Karavias Dimitrios Emmanouil - s141278

As a project I have implemented a random terrain generator. The terrain is rendered using triangle strips to generate the surface and perlin noise to calculate the heigh for each vertex. The project implements the following noticable features.

1. Animation

I have created an "infinite" loop that handles user input, updates the variables and renders the objects. This loop has been created using the glut timer callback function. The animation includes the movement of the camera for navigation which updates the model view projections based on the direction and position of the camera. This is happening in C++ code and the updated views are used in the shaders. One more animation includes an animation of the position of the vertices of the map which is calculated in the vertex shader. The main code sends a float uniform value to the vertex shader that is actually a counter of the step of the animation. When the map is generated the counter starts counting from 100 and below. When the value of the counter is highter than the y position of the vertex, the vertex uses the value of the counter combined with a sine wave to generate a wavy effect. When the value of the counter is less than the y position of the vertex, the the vertex uses its original y position. This way I wanted to generate an effect of a paper map that when falls down creates the real map. When the user changes the map attributes and a new map is generated, the counter starts counting from 0 to 100 generating the reverse effect.

2. Map lighting, mirroring and skybox.

I decided to split the rendering of the sea and the land part of the map into two different drawings. So the shader for the land discards the vertices for the sea, and the shader of the sea discards the land vertices. I did this so i can use the water surface to mirror the mountains and the skybox used in the scene. The mirroring has been done with the stencil buffer. On the other hand the land shader implements the diffuse shading while the water shader doesn't use the light at all.

3. User Input.

User may use the keyboard to navigate through the scene and/or change the map.
For navigation the keys are:
Q: Move up
E: Move forward
A: Move left
S: Move backward
D: Move right

The user may change the terrain attributes with the following keys:
1, 4: Decrease/increase the amplitude of the terrain.
2, 5: Decrease/increase the frequency used by the perlin noise library.
3, 6: Decrease/increase the octaves used by the perlin noise library, generates more smooth or rough terrains.
-, +: Decrease/increase the seed used by the perlin noise.

Changing any of the above attributes will result a new map to be generated.

Screenshots from rendering the project:

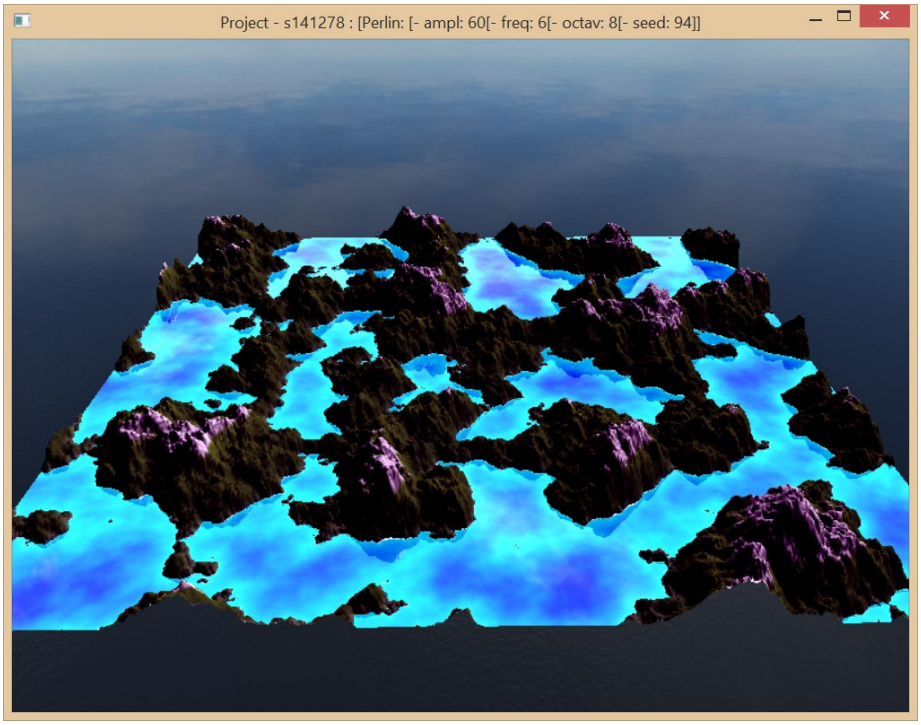

Figure 1. When the application starts you will see the dark side of the map.

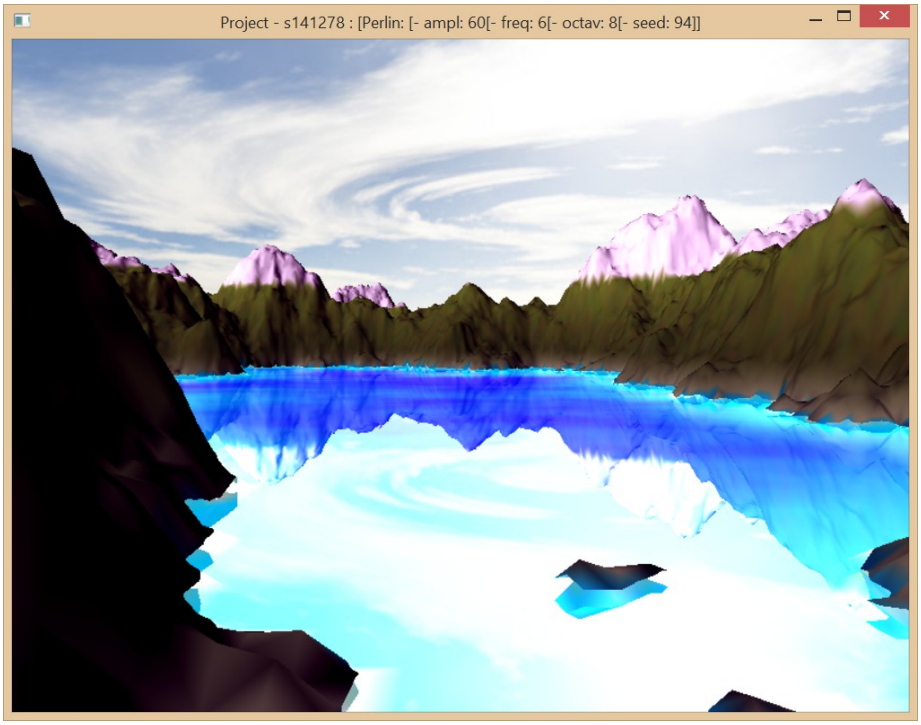

Figure 2. A nice view of the water mirroring the mountains and the skybox.

Figure 3. A view of the map animated while falling down to take the original shape.

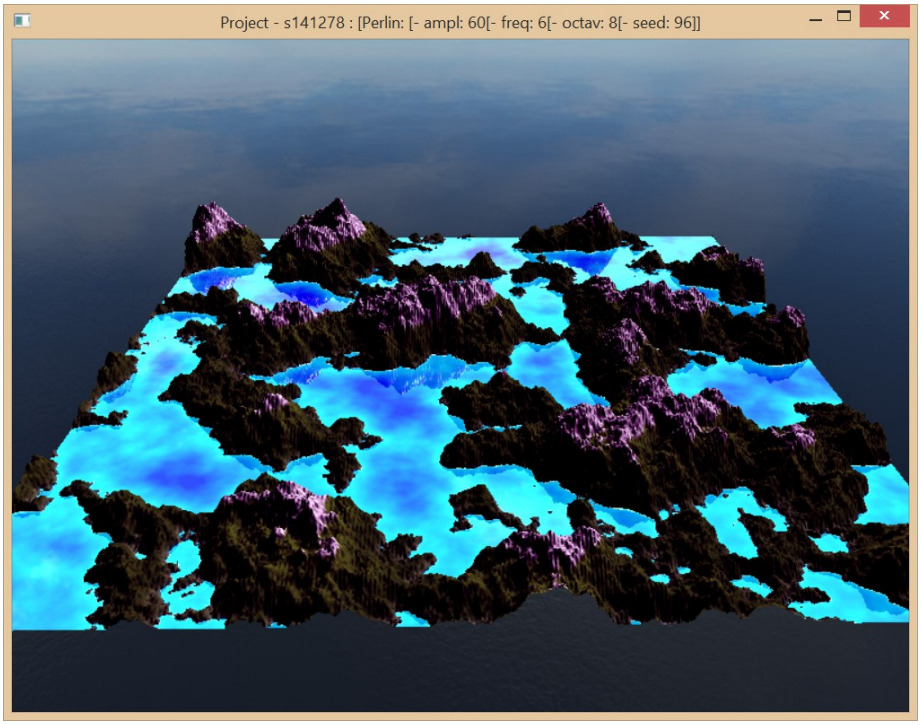

Figure 4. A completely new map by changing the seed.