Assignment4

Textures

Francesco Andreussi
francesco.andreussi@uni–weimar.de

9 January 2020

Deadline

Wednesday, 22 January 2020 at 23:55.

Task

- Load a texture for each of the planets in your `application_solar.cpp`. (35%)
- Use these textures in your fragment shader as a source for your diffuse color. (35%)
- Comment the code extensively. (10%)
- **Additional Task 1**: implement a Skybox, creating a box in OpenGL and assigning it some suitable textures (10%).
- **Additional Task 2**: implement a Normal Mapping for at least one planet (10%).

Tips & Suggestions

- Load png or tga files with `texture_loader::file()` function, overwrite an instance of the `pixel_data` structure; this will contain all the useful information.
- Modify in the `model_loader::obj()` function the last parameter in "`model::NORMAL | model::TEXTCOORD`".
- You can find the textures at this link.
- It is possible to derive from a grey-scale map a normal map thanks to this webpage.
- A Skybox texture is a collection of 6 textures you can use *binding* a `GL_TEXTURE_CUBE_MAP` and then defining, as texture data, 6 different textures using the `glTexImage2D` function, with the appropriate *binding point*.

- For the normal mapping you can use the following function (ONLY if you can explain it in the comments)

```
#extension GL_OES_standard_derivatives : enable

vec3 perturbNormal( vec3 vertex_pos, vec3 surf_norm ) {

vec3 q0 = dFdx( vertex_pos.xyz );
vec3 q1 = dFdy( vertex_pos.xyz );
vec2 st0 = dFdx( uv.st );
vec2 st1 = dFdy( uv.st );

vec3 S = normalize(  q0 * st1.t - q1 * st0.t );
vec3 T = normalize( -q0 * st1.s + q1 * st0.s );
vec3 N = normalize( surf_norm );

vec3 mapN = texture2D( normalMap, uv ).xyz * 2.0 - 1.0;
mapN.xy = normalScale * mapN.xy;
mat3 tsn = mat3( S, T, N );
return normalize( tsn * mapN );

}
```