# Graphics programming

# Exercise 11 – Discrete techniques

Henrique Debarba

IT University of Copenhagen

# Exercise 11

- Learning objectives
  - Implement reflection and refraction effects using a environment mapping
  - Implement normal mapping effect

# Exercise 11 - Additional resources

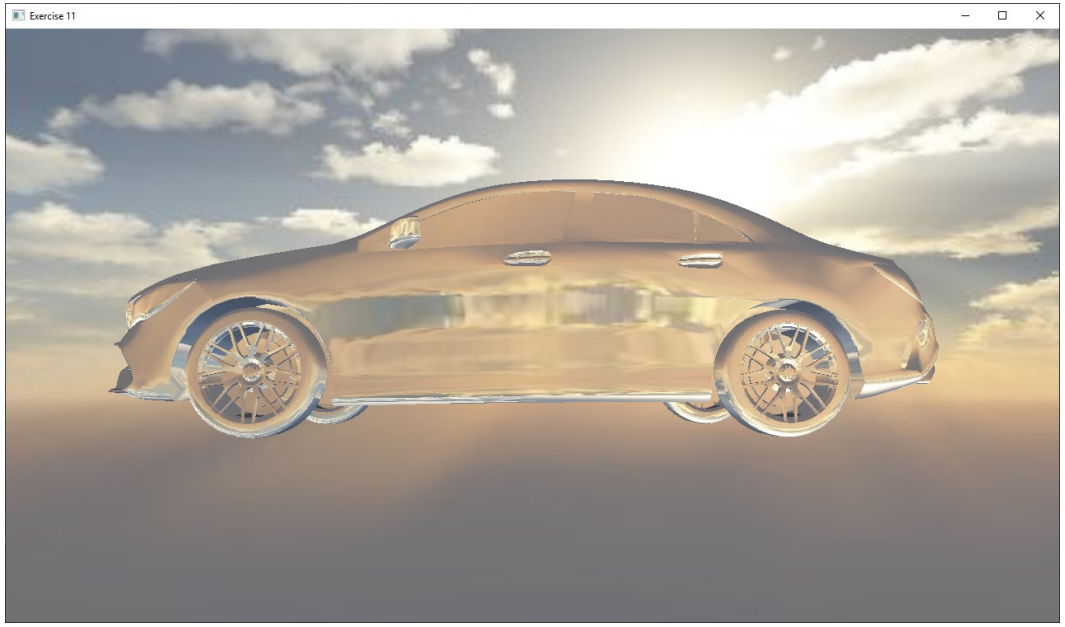
- LearnOpenGL
  - https://learnopengl.com/Advanced-OpenGL/Cubemaps
  - https://learnopengl.com/Advanced-Lighting/Normal-Mapping
- Other tutorials
  - https://developer.download.nvidia.com/CgTutorial/cg_tutorial_chapter07.html
  - https://developer.download.nvidia.com/CgTutorial/cg_tutorial_chapter08.html
  - https://docs.cryengine.com/display/SDKDOC4/Tangent+Space+Normal+Mapping

# Exercise 11.1 - reflection

- Implement reflections with environment mapping:
  - In the fragment shader, **reflect** the eye to fragment position incident vector;
  - Sample the skybox texture using the **reflected vec3**.
  - Remember that, as I mentioned in class, we sample the cubemap using a vector. That is because, for all practical purposes, we assume that the image in the cubemap is infinitely far from the objects in the scene (this is not true, but the objects rendered in a skybox are so far from the observer that the error is negligible)

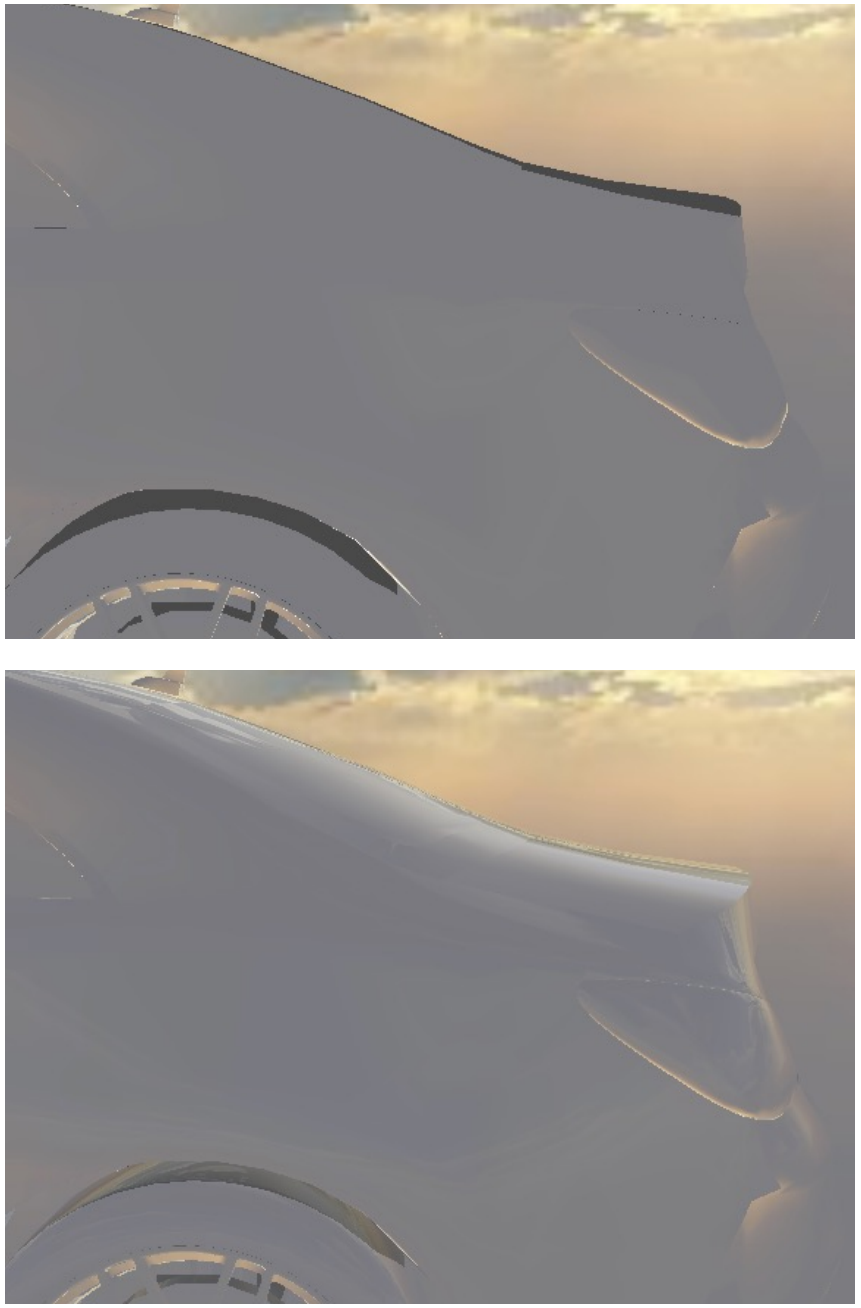

# Exercise 11.2 - refraction

- Implement refraction with environment mapping:
  - Using **Snell's law** and the **refractive index** of different materials;
  - In the fragment shader, **refract** the eye to fragment position incident vector;
  - Sample the skybox texture using the **refracted vec3**.

# Exercise 11.3 – Fresnel effect

- Implement the fresnel effect:
  - In the fragment shader, set the variable "**reflectionProportion**" using Schlick's approximation of the Fresnel equations.
  - Mind tha you need to compute the angle between the **normal** and the **view** vectors



# Exercise 11.4

- Normal mapping
  - In the vertex shader:
    - Build a **TBN matrix** that transforms from **world space** to **tangent space**;
  - In the fragment shader
    - **Sample the normal map** texture and use it for light calculation;

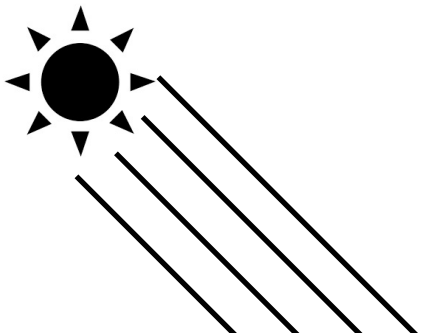

# Exercise 11.5 - reflection

- In the fragment shader
  - Transform the normal from Tangent space to World space (make a copy, or you will mess up with lighting computation)
  - **reflect** the eye to fragment position incident vector according to that **normal**;
  - Sample the skybox texture using the **reflected vec3**.

# New light caster

- Point source
  - Model with position and color

- Directional light
  - Distant source = infinite distance away (parallel rays)
  - No attenuation



More at https://learnopengl.com/Lighting/Light-casters