

---

# Fast Realistic Rendering

Pere Pau Vazquez

[ppau@cs.upc.edu](mailto:ppau@cs.upc.edu)

Marc Comino Trinidad

[mcomino@cs.upc.edu](mailto:mcomino@cs.upc.edu)

## PROJECT STATEMENT - PHYSICALLY BASED RENDERING

In this assignment, we will deal advance rendering techniques. The objective of this assignment is to get familiar with the elements that are under the hood of physically based rendering. In order to do so, you have to implement a method based on the **microfacet** approximation to non-optically flat surfaces.

In order to get familiar with this theory, you can check the following resources:

1. Siggraph **Physically Based Shading in Theory and Practice** course slides:

[http://blog.selfshadow.com/publications/s2014-shading-course/hoffman/s2014\\_pbs\\_physics\\_math\\_slides.pdf](http://blog.selfshadow.com/publications/s2014-shading-course/hoffman/s2014_pbs_physics_math_slides.pdf)

2. Siggraph Physically Based Shading in Theory and Practice course notes:

[http://blog.selfshadow.com/publications/s2013-shading-course/hoffman/s2013\\_pbs\\_physics\\_math\\_notes.pdf](http://blog.selfshadow.com/publications/s2013-shading-course/hoffman/s2013_pbs_physics_math_notes.pdf)

3. GPU Gems on **Image Based Lighting**:

[https://developer.nvidia.com/gpugems/GPUGems/gpugems\\_ch19.html](https://developer.nvidia.com/gpugems/GPUGems/gpugems_ch19.html)

Particularly, we ask you to:

1. Implement an environment reflection shader using a cube texture.
2. Implement an image based lighting approach based on:
  - a. Approximating the implicit geometry function using the microfacet theory.
  - b. Cubemaps that contain precomputed cosine weighted specular convolutions. The material roughness must be a parameter for the precomputation.
  - c. The schlick approximation to Fresnel reflectance.
3. You will need some application to precompute the filtered cubemaps for (b). You can either implement your own or use one of the following:
  - a. The trial version of Lys: <https://www.knaldtech.com/lys/>

- b. Cubemapgen:

<https://seblagarde.wordpress.com/2012/06/10/amd-cubemapgen-for-physically-based-rendering/>

## FURTHER READING

You can find more details on CubeMaps and Image Based lighting here:

1. <https://learnopengl.com/Advanced-OpenGL/Cubemaps>
2. <https://learnopengl.com/PBR/IBL/Diffuse-irradiance>
3. <https://learnopengl.com/PBR/IBL/Specular-IBL>

You can find more Cube Maps here:

1. <http://www.humus.name/index.php?page=Textures>

## DELIVERY

Please upload a single zip file by **March 26th**, named after your username. For instance: marc.comino.zip

The zip file should contain:

- A compilable and executable project. This includes:
  - All the required .c, .cc, .cpp, .h, .hpp, .ui, etc. files needed to compile your application.
  - All the code used for your shaders.
  - All the used texture files.
  - A Makefile, CMakeLists or similar script that is able to compile and generate an executable file out of your source files.
  - For **linux** submissions: A list of the dependencies needed to compile your application.
- A short report explaining the implemented shaders.
  - The report could consist on a very well documented version of your shaders. It should be clear which lines of the code relate the different parts of the algorithms.
  - I personally recommend to elaborate the report using Microsoft Word or Latex or Google Docs.
- A live presentation of your project. We will later arrange how to do this.