

UNIVERSITY OF TÜBINGEN

PROF. DR.-ING. HENDRIK P.A. LENSCH

CHAIR OF COMPUTER GRAPHICS

FAEZEH ZAKERI (FAEZEH-SADAT.ZAKERI@UNI-TUEBINGEN.DE)

LUKAS RUPPERT (LUKAS.RUPPERT@UNI-TUEBINGEN.DE)

RAPHAEL BRAUN (RAPHAEL.BRAUN@UNI-TUEBINGEN.DE)

ANDREAS ENGELHARDT (ANDREAS.ENGELHARDT@UNI-TUEBINGEN.DE)

PHILIPP LANGSTEINER (PHILIPP.LANGSTEINER@STUDENT.UNI-TUEBINGEN.DE)

EBERHARD KARLS  
UNIVERSITÄT  
TÜBINGEN



20. DECEMBER 2022

## COMPUTER GRAPHICS

### CHRISTMAS SPECIAL

**Submission deadline for the exercises:** 17. January 2023, 0:00

## 1 Render something nice (up to 50 Bonus Points)

Since we started, our framework has grown from a humble little mesh viewer to a somewhat capable path tracer and OpenGL real-time renderer.

Make use of its capabilities and render a nice scene of your own design.

You can find free meshes and textures on <https://polyhaven.com/>, <https://www.printables.com/>, <https://www.blendswap.com/>, and many other sites. In many cases, these assets come with a license that require you to name the author and license when you use it. If that is the case, please do.

When textures come in an EXR format, you can use GIMP to convert them to JPEG. (Convert the image precision from 32bit float to 8bit sRGB, then export as JPEG.)

For environment maps, please use the HDR (high dynamic range) format.

You can see a small example scene in Figure 1.

Points will be awarded as follows (up to 50 total):

- 10 Points for creating a new scene out of existing meshes, materials, and textures.
- 20 Points for creating your own non-trivial mesh or texture (e.g., in Blender).
- 20 Points for implementing a custom non-trivial shader (e.g., non-photorealistic rendering, noise, 3D textures, post-processing, animations).
- 40 Points for procedural generation (e.g. procedural textures, procedural geometry).
- 20 Points for implementing a new light source with support either in OpenGL or the path tracer. (E.g., environment map support in the path tracer or a spot light in OpenGL or the path tracer.)
- 10-50 Points for all computer graphics related topics and significant improvements upon the existing framework. Also describe the theory behind your implementation.

Hand in your code, at least one rendered image, and summarize your additions to the framework.

When you create new meshes or textures, make sure to point them out and show them individually.

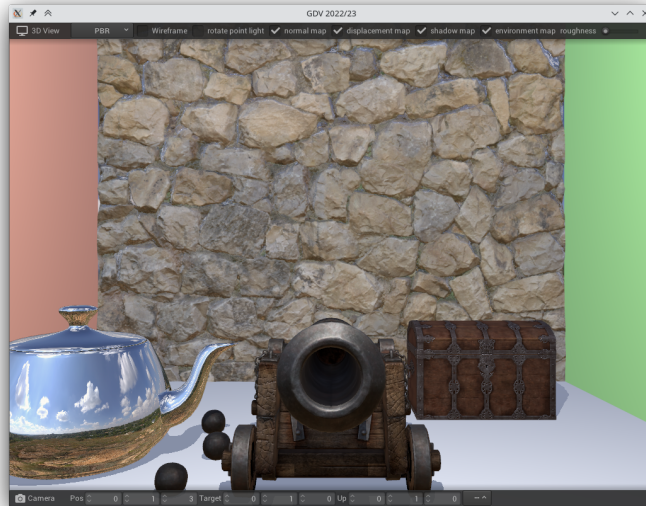


Figure 1: A small scene with physically-based rendering. The “roughness” slider allows you to change the roughness of otherwise perfectly specular materials, e.g., conductors like the silver teapot, to see the glossy reflection of the pre-filtered environment map.

## 2 Let it Snow (up to 50 Bonus Points)

With the updated framework, you can also find a small Christmas scene (see Figure 2). You have to enable it in the `main.cpp`. However, it still needs some falling snow to set the mood.

Implement some form of falling snow (20 Points). The snow needs to animate as time (`glfwGetTime()`) progresses (10 Points), have some depth – such that it can be hidden behind existing geometry (10 Points), and should be consistent even when the camera moves (10 Points).

Submit a picture and your code along with a short description of how you chose to implement it.

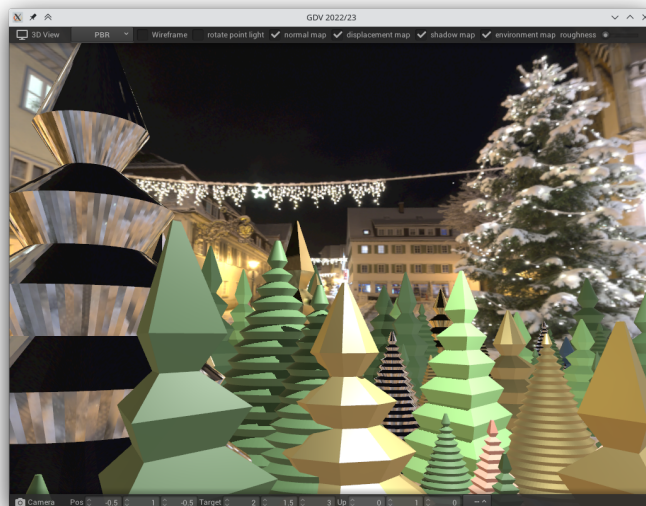


Figure 2: A randomly-generated scene with some Christmas trees CC-by mightynozzle.