Additional libraries

We used FreeImage to import texture image files. (http://freeimage.sourceforge.net/)

We implemented our own object loader.

Implemented Effects

Shadow Mapping (depth)

We began with dual paraboloid shadow mapping. However, we did not manage to map the texture coordinates correctly, which is why we opted for cube mapping. The scene is projected onto a cubemap (shadowMap) by rendering the scene from six different perspectives and a field of view of 90°. By using a geometry shader, all of this can be done in one rendering step instead of 6. The (now 3-dimensional) coordinates are then read by the main shader (vbo_vao) to create the shadows. It should be seen especially with the palm trees, but there is a bug that prevents the shadows from appearing.

[1] WILLIAMS, Lance. Casting curved shadows on curved surfaces. In: ACM Siggraph Computer Graphics. ACM, 1978. S. 270-274.

[2] HEIDRICH, Wolfgang; SEIDEL, Hans-Peter. View-independent environment maps. In: Proceedings of the ACM

SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware. ACM, 1998. S. 39-ff.

[3] VANEK, Juraj, et al. High-quality shadows with improved paraboloid mapping. In: International Symposium on Visual Computing. Springer, Berlin, Heidelberg, 2011. S. 421-430.

Environment Mapping

We implemented reflective and refractive cube mapping. To achieve this effect, the scene is projected onto a cubemap (EnvironmentMap) using a special camera (EnvironmentMapCamera) that has a 90 degree field of view and is positioned at the center of the reflective object. In our scene, a sphere in the kitchen is reflective. The sphere inside a room refract and reflects light. We render two environment maps per frame, each centered at one of the spheres, so that both locations reflect / refract realistically.

Once the environment map is rendered, the reflected color is accessed in the shader (reflectiveShader, refractiveShader) using the reflection of the vector pointing from the camera to the vertex. The reflected color is then mixed with the color obtained from the model's texture and phong shading.

We used Schlick's approximation [5] to mix reflection and refraction in the transparent sphere.

[4] GREENE, Ned. Environment mapping and other applications of world projections. IEEE Computer Graphics and Applications, 1986, 6. Jg., Nr. 11, S. 21-29.

[5] SCHLICK, Christophe. An Inexpensive BRDF Model for Physically-based Rendering. In: Computer Graphics Forum. 13 (3): 233, 1994

Tools Used to create the Models / Sources

We created the walls and doors of the apartment in Blender. The other models (except for the kitchen) are from https://free3d.com. The kitchen model is from https://www.cgtrader.com and was modified in Blender.

Controls

The camera can be (re-)started and stopped using the space bar. There is also a debug camera available that can be controlled with WASD.

Graphics Cards

Nvidia GPU