02561 Computer Graphics

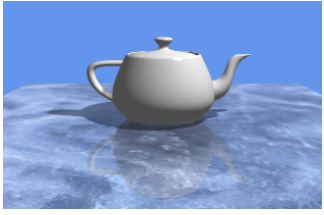
Project suggestion

Jeppe Revall Frisvad

November 2024

Default project: Planar reflector

- Use a reflection matrix to draw a reflected version of the scene.
- Blend reflector and the reflected scene drawn behind it.
- Use a stencil buffer to ensure that reflected objects are only drawn where the reflector will be drawn afterwards.
- Use oblique near plane clipping to ensure that submerged objects do not result in reflected objects appearing in front of the reflector.
- A *project initiator* worksheet is available.



Material Captures (MatCaps) and user interaction

MatCaps project

- Create a program that uses MatCaps for shading 3D models.
  (`https://github.com/nidorx/matcaps`)
  (`http://pixologic.com/zbrush/downloadcenter/library/`)
- MatCaps are really just tightly cropped images of spheres/hemispheres.
- Invent interesting ways of creating/editing MatCaps.
- Discuss the limitations of MatCaps.

UI project

- Consider interfaces that offer interesting opportunities for user interaction (e.g. device orientation: `https://web.dev/articles/device-orientation`).
- Create an interactive application that exploits the interface of interest.
- Discuss the relation between this interface and more conventional approaches.
- See if you can find a way of comparing user performance when using the new interface with performance when using a more conventional interface.

Other projects

1. Extend the 2D drawing program to include rational quadratic Bézier curves.
2. Render animated 3D objects (using simple physics or key frames or skeleton).
3. Model a plane with different movable parts and provide a user interface that enables interaction with the different parts (W03P4).
4. Load *uv*-coordinates and texture(s) together with an OBJ model and render a textured model (W06P4).
5. Use a noise-based procedural solid texture to do bump mapping of a surface or other texturing effects (`https://www.shadertoy.com/view/ldscDM`, `https://www.shadertoy.com/view/NsjBDy`). [**R**: 6.3]
6. Work with depth-dependent techniques (like exponential attenuation due to absorption) and use this for volume rendering. [**R**: 14.3.1]
7. Use a ping-pong rendering technique to render multiple reflections [**R**: 11.6].
8. Work with rendering of glossy materials by look-up into blurred versions of the environment map.

Other projects

9. Using a physics engine for physics-based interaction.
   - `ammo.js` is a port of a physics engine for WebGL. https://github.com/kripken/ammo.js/
   - Tutorials for use with `three.js` are available.
   - Based on this, I made an example using regular WebGL.

     https://courses.compute.dtu.dk/02561/physics/
10. Collect texture assets from https://polyhaven.com/textures with layers such as diffuse, displacement, normal, roughness, ambient occlusion, and use these layers to render a sphere (and perhaps another object as well) with a high level of detail.
11. Draw a subdivision surface (like the Newell teapot). [**R**: 17.5]
    Take inspiration from OpenSubdiv: https://graphics.pixar.com/opensubdiv/,

    https://github.com/takahito-tejima/takahito-tejima.github.io

Recommended hand-in procedure

- Make a pdf with your project report and a link (with no public references) to your lab journal webpage and project implementation.
- Make a zip file with a copy of all the code, libraries, and data from the lab journal and the project.
- Put your link in your pdf report and in the text box for comments when submitting on DTU Learn (under Assignments). Submit the pdf as the main document and the zip-file as a supplement.

- If you collaborated with other students, state this explicitly and include the student id of your collaborator(s).
- If you wrote lab journal and/or report together in a group, carefully read and follow the guidelines on the course webpage regarding report individualization.
- Even if you hand in an individualized report as a group, each group member must still submit it to DTU Learn (one submission for a group of people is not allowed).