Exercise 02561-09 **Hardware Reflections**

Readings B1: Chap. 8.10-11. Help slides

Introduction Like shadows, reflections are almost trivial to compute in Ray Tracing and more difficult when we use OpenGL. But again, if we simplify the situation and consider only planar reflectors, it becomes rather simple to draw the reflection. Imagine that we render an object in front of a reflecting planar surface. To draw the reflection, we can simply mirror all the vertices of the object in the plane, and render the mirrored object.

Purpose To understand and implement planer reflections using OpenGL.

Part 1 The same framework is used for this exercise as for the previous (shadow mapping). This time put the code in the `draw_mirror` function.

Part 2 The plane we are reflecting about is incident on (0,0,0) and has normal (0,1,0). Now draw the teapot in such a way that it is also reflected in this plane. That is easy – just scale using (1,-1,1) and remember to change the orientation of frontfacing polygons from counter-clockwise to clockwise.

Question: Why is the change of orientation necessary?

Finally, to verify that your implementation works, disable the drawing of the normal teapot and the textured plane with the shadow. (We will reenable both things later)

Part 3 Set up a second light source that is reflected about the mirror plane. This light source should be less bright and only enabled when we draw the reflection.

Part 4 Reenable the shadowed plane from the last exercise. To make sure that it works with the reflection, we must make some changes. In the display function, `draw_mirror` is called before `draw_proj_shadow`. `draw_proj_shadow` first draws the shadow to the alpha buffer but it also affects the colour buffer. This would ruin our reflection. Therefore, you must disable drawing to the colour buffer in `draw_proj_shadow`. This is done by calling `glColorMask`. Remember to enable drawing to the colour buffer again before drawing the plane with texture.

If everything is done corretly, you now have the shadow in the alpha channel and the reflection in the colour channels.

Finally, the textured plane should be blended with the reflection of the teapot. Replace

```
glBlendFunc(GL_DST_ALPHA, GL_ZERO)
```

With

```
glBlendFunc(GL_DST_ALPHA, GL_ONE)
```

Part 5 A minor annoyance: The reflection of the teapot is visible even outside of the reflecting square. To fix this we need to draw the reflecting plane to the stencil buffer.

Before drawing the reflection of the teapot, draw the reflecting plane in such a way

that the colour buffer is not changed (just use the colour black) and the depth buffer is not changed (`glDisable(GL_DEPTH_TEST)`), but the stencil buffer is set to 1 for all pixels covered by the plane. You need to use the functions `glStencilFunc` and `glStencilOp` for this. Then change the setup so that we only draw where the stencil is 1 and not where it is 0.

Part 6

Change the path of the teapot so that it penetrates the reflecting surface (imagine it is water). Now, we have a new problem, the reflection pops up over the surface. To fix this, set a user defined clipping plane that is coincident with the reflecting plane and clips away everything “above the surface”. Only enable this plane when drawing the reflection.