Graphics – Assignment 3
Mikkel Svagård

Repetition

The Z buffer stores the currently closest distance (the Z coordinate) of every pixel, hence giving the oportinity to determine wheter there are something in front of the fragment in question (known as a depth test). In openGl this buffer is known as a depth buffer, whereas the test regarding this bufffer is usually done after the Fragment Shader. If the test fails the fragment is discarded, if succeedes the depthbuffer gets updated with the fragments depth along with the framebuffer. T

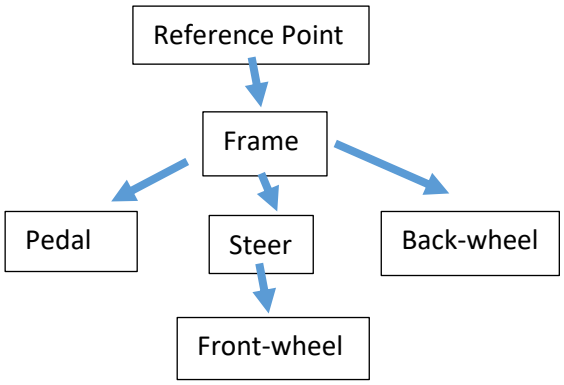
The Framebuffer is a portion of memory (usually in RAM) that contains the complete bit-mapped image that is sent to the monitor, in other words the destination for rendering. OpenGl has two kind of framebuffers, one default framebuffer window or display device, and one user-created framebuffer for textures or rendering. As mentioned above it is the final image, hence usually being last in the OpenGl pipeline.

The ViewPort is a area defined screen coordinates in which the objects of interest are going to be rendered. Clipping is usually done before the view-port transformation, thus view-port in openGl being in the post-vertex processing part of the pipeline.

Rendering the Scene Graph

4 pictures have been added at the end of the document. 2 at a slightly tilted axis and 2 directly above of the Sun.

Scene Graph for bike:

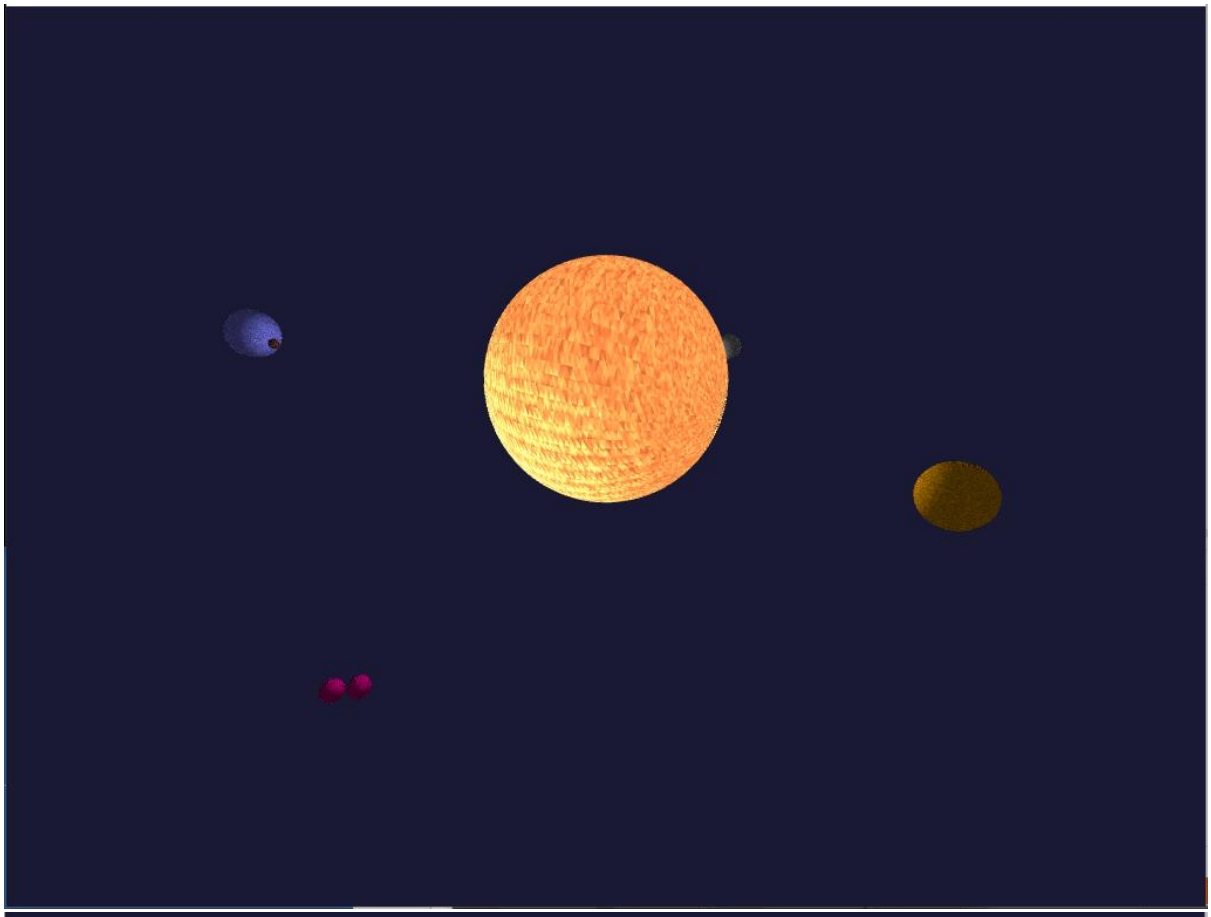


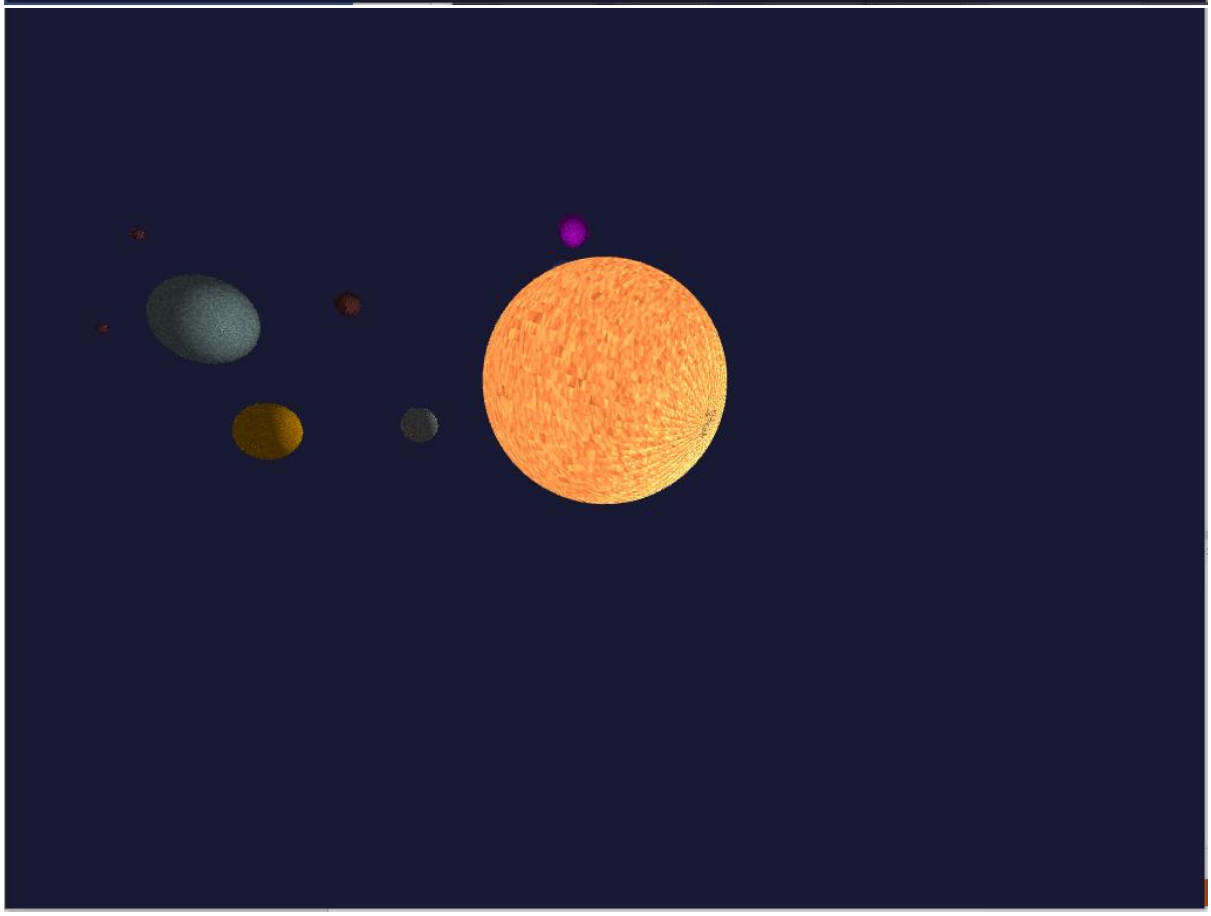
Affine Transformation for rotation Back-wheel:
[Translate Back to Position]*[Translate Back to Reference]*[Rotate 30 degrees around Z axis]*]*[Translate to Origo] *[Translate to Reference] *Backwheel
→
Translate(-5,3)* Translate(100,27)*Rotate(30 degrees, Z-axis)*
Translate(-100,-27)*Translate(5,-3) * Backwheel

Optional Feedback

Did not enjoy the way image processing part of this course, as the gradings of the deliveries where way too harsh. In my opinion being punished for not writing optimal code is unnecessary in a course that don't empathizes on it, as long as it works of course. In addition, the exam questions where a true pain for me, as I only received full score on one of them even though giving it my best. The graphics part was entertaining, though a little rough as we are fresh to c++ AND openGl.

Graphics – Assignment 3
Mikkel Svagård





Graphics – Assignment 3
Mikkel Svagård