RPS Tanks

Fabian Eichhorner (1227328)
Florian Steinschorn (1227109)

Implementation

We are using the game from CGUE made by Johannes Wawerda and Florian Steinschorn (Link → "Achievement Award"). It uses a Scene Graph with objects imported as .obj files and textures. A special case is the heightmap, which is generated from a grayscale image. We have ambient light, illumination from the sun in the form of a directional light, as well as spotlights on the front of the tanks. Effects already implemented contain shadow maps for the sun, depth of field and a particle system. Previously cell shading and edge detection were implemented, but they have been taken out to make place for new effects.

It is possible to activate or deactivate with the F keys a frame timer as well as toggle a few options and effects. A full list is shown when pressing the F1 key.

For this course we added Bloom, Screen Space Ambient Occlusion as well as Normal Mapping. To convert the game into a demo, we have the ability to let the tanks execute a fixed order of commands, as well as predetermine the movement of the camera.

Demo

For the demo system, we extended the existing tanks and cameras with remote-control versions. Those versions additionally take a list of functions which are applied at a specific time. For the camera it is possible to set the distance, change the angle and set the inclination. The tank is possible to emulate all defined actions.

In the demo program it is possible to switch between different cameras. We defined cameras for the root object as well as the two tanks. At certain events automatic movement and a switch is applied to create an exciting demo. The general controls beside the F keys are deactivated for the demo.

Effects

Bloom

Bloom was implemented similar to the description on slide 7 of the lecture slides on Screen Space Effects. The texture used to store the bloom values is half scale of the full resolution. Additionally to adding bloom to any bright enough point, it is possible to define bloom areas via a bloom map. These areas don't get lighting applied to them, so they will be over the bloom threshold if the original color is bright enough. This is used on the glowing parts of the tanks showing the team colors.

Normal Mapping

Normal mapping was done again according to the Screen Space Effects slides (32ff). The parallax corrected position is calculated and used to read the color and normal textures and the adjusted normals are then passed to the lighting algorithm.

Screen Space Ambient Occlusion

SSAO is implemented according to the slides on Screen Space Effects from the lecture Algorithms for Real-Time Rendering. We use a hemisphere aligned to the surface normal with samples being clustered more densely near the center.

Tools used

To edit the 3d models Blender was used. Textures are created using GIMP and Paint. The ground texture as well as the normal map for the tanks was taken from the Total Textures Repository.

Controls

How to build an start the demo can be found on our github readme.
Additionally it is possible to switch the implemented effects of and on to compare the results:
F1 … list of all settings
F7 … normal mapping on/off
F9 … SSAO on/off
F11 ... Bloom on/off

Additional Libraries

Assimp: http://assimp.sourceforge.net/
Bullet Physics: http://bulletphysics.org/wordpress/
Freeimage: http://freeimage.sourceforge.net/
glew: http://glew.sourceforge.net/
glfw: http://www.glfw.org/
Irrklang: https://www.ambiera.com/irrklang/

Tested on

Geforce GTX 1050ti
Geforce GTX 970