

CGUE 2015

Pedal to the Metal

Submission 2

Christoph Rothhammer
Christoph Bruckner

0828235
0825176

Implementierung:

Camera:

Camera kann mit "C" vom Auto gelöst werden. Dann kann mit der Maus die Blickrichtung, und mit WASD geflogen werden, um die Effekte besser begutachten zu können.

Moving Objects:

Das Auto kann vom Spieler mit den Pfeiltasten gesteuert werden. Das Auto bewegt sich mit PhysX.

Texturemapping:

Zum Laden der Texturen verwenden wir SOIL. Assimp kann die im 3D Modelling-Programm festgelegten UV-Koordinaten, sowie 3 gespeicherte Texturen einlesen, die wir dann im Shader weiterverarbeiten. So können Specularmap, Reflectionmap und Diffusemap aus einem obj + mtl file gelesen werden. Alle Objekte verwenden Texturemapping.

Controls:

Zum Steuern des Autos werden die Pfeiltasten verwendet. Durch Key-Polling wird ein PhysX dynamic cube Objekt mit 4 Spheres und fixed Joints in verschiedene Richtungen beschleunigt. Mit der Taste F3 kann die Wireframe-Ansicht umgeschaltet werden, beim Spielstart kann ausgewählt werden, ob das Spiel im Fullscreen-Modus gestartet werden soll. Auch kann die Fenstergröße geändert werden und der OpenGL Inhalt wird angepasst.

Gameplay:

Das Auto kann auf der Rennstrecke bewegt werden. Ziel des Spiels ist es im Moment, ohne von der Fahrbahn abzukommen die Strecke zu fahren. Die Kisten auf der Strecke und die Neigung des Autos sehr leicht zu driften, erschweren das Vorhaben.

Modelloader:

Als Modelloader haben wir Assimp verwendet. Wir können damit Szenen mit mehreren Meshes z.B. aus .obj Dateien laden. Das Auto, die Rennstrecke und andere Objekte werden so geladen.

Skybox:

Für die Skybox wurde ein zusätzlicher, einfacherer CubeMap-Shader Implementiert.

Environmental Mapping:

Am Auto und auf der Straße haben wir dynamic Environmental Mapping implementiert, das die Skybox, sowie die dynamischen Würfel und die Strecke reflektiert. Die Cubemap wird jeden Frame mittels eines Geometryshaders vom Blickpunkt des Autos aus neu generiert. Mit Reflectionmaps wird festgelegt, welche Teile der Strecke (grüne Pfützen) oder des Autos (Lack, Chrom) die Cubemap reflektieren.

Omnidirectional Shadow Mapping mit PCF:

Die Punktlichtquelle am Auto erzeugt Schatten hinter den Würfeln auf der Strecke und anderen Würfeln. Dazu wird die Szene bzw. die Depth-Werte mittels Geometryshader in eine Cubemap gerendert.

Verwendete Libraries:

Assimp: <http://assimp.sourceforge.net/>

GLM: <http://glm.g-truc.net/0.9.5/index.html>

GLFW: <http://www.glfw.org/>

PhysX: <https://developer.nvidia.com>

Tutorials:

<http://ogldev.atspace.co.uk/>

http://en.wikibooks.org/wiki/GLSL_Programming

<http://www.lighthouse3d.com/>

<http://learnopengl.com/#!Advanced-OpenGL/Geometry-Shader>