

SpaceGarden Documentation

Team:

Manuel Matusich (0806024)

Claudia Hoffmann (1004538)

SpaceGarden.exe starten:

Das Spiel startet mit 1024*768 um window mode. Für diverse andere Konfigurationen sind .bat Dateien vorhanden und entsprechend benannt. Das Spiel kann auch aus der Konsole gestartet werden.

SpaceGarden.exe <width> <height> <window mode> <refreshrate>

Beispiel: SpaceGarden.exe 1024 768 0 60

window mode 0 = window

window mode 1 = fullscreen

Gameplay:

Ziel ist es die Powerups(weiß, rote Würfel) einzusammeln und die roten Blumen zu zerstören. Jedoch wachsen die Blumen. Wenn man vom Planeten runter fällt, kann man den Roboter mit R in die Mitte der Plattform zurücksetzen. Wenn man R gedrückt hält kann die Kamera um den Roboter rotiert werden. Das Spiel ist zu Ende wenn alle 16 Slots mit Pflanzen befüllt sind, wenn weniger als 2 rote Pflanzen über sind gewinnt man, sonst verliert man.

Controls

R	Reset
W,A,S,D	Steuer
F	Zerstören der Blumen
Leertaste	Springen
F2	Frame Time
F3	Wireframe
F7	Config display on/off
F8	View Frustum Culling
F9	Transparency
ESC	Beenden des Spieles

Effects:

1. Spotlight:
An der Position des Roboters wird die Umgebung mit einem Spotlight beleuchtet.
2. Shadow Map with PCF:
Für Shadow Map verwenden wir ein „directional light“ für alle Objekte. Zuerst wird aus Sicht des Lichtes gerendert (1.Pass). Mithilfe der Depthmap wird im Fragment Shader festgestellt was im Schatten und was im Licht ist. Mithilfe der Folien des Repetitorium und dem Tutorial
<http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-16-shadow-mapping/>
wurde der Effekt implimentiert.

3. Glow :

Der Glow Effekt wird auf das Laserschwert („lightsaber“) und auf die Powerups angewendet. Um den Effekt auf dem Laserschwert zu sehen muss ein Powerup eingesammelt werden. Um das Schwert aus einer anderen Sicht zu sehen kann man R gedrückt halten und die Kamera rotieren. Das Laserschwert wird in seinen Farben allein gerendert, alles andere ist schwarz. Darauf wird zuerst der horizontale und dann der vertikale Gaußfilter angewendet. Dann wird das normale Bild mit dem gefilterten Bild zusammengeführt.

Hilfreiche Tutorials:

http://www.ozone3d.net/tutorials/image_filtering_p2.php

http://http.developer.nvidia.com/GPUGems/gpugems_ch21.html

4. Normal Mapping:

Normal Mapping wurde auf alle Steine, die Plattform(Boden) und den Zylinder angewendet. Der Tangentvektor wurde mit der Assimp library berechnet. Die Normalmap sind vom Total Textures Repository oder wurden durch Gimp2 oder durch cpetry.github.io/NormalMap-Online/ erstellt.

Hilfreiche Tutorials:

<http://ogldev.atspace.co.uk/www/tutorial26/tutorial26.html>

<https://www.youtube.com/watch?v=dF5rOveGOJc&index=2&list=PLEETnX-uPtBVG1ao7GCESh2vOayJXDbAl>

Complex Object:

Fast alle Objekte sind von der Website 3D Wharehouse. <https://3dwarehouse.sketchup.com/>
Die Blumen wachsen. Die Models laden wir mit der Assimp library und sind im Format OBJ - Wavefront Objekt. Wir verwenden die Library Glew.

Rote Blume und Roboter: Keine Textur, Specular Effekt

grün/gelbe Blume: Textur, nur directional light

powerUp Cubes (weiß/rot): Keine Textur, glow

Restlichen Objekte: Textur + Normalmapping

Animated Objects:

Beim Gehen bewegt der Roboter beide Armen separat. Beim Springen hebt er beide Arme hoch. Eine Schlaganimation mit dem rechten Arm wenn man F drückt.

View Frustum Culling:

Jedes Objekt hat eine BoundingSphere, sobald diese sich mit einer der sechs Clipplanes des Frustums überschneidet oder innerhalb liegt, wird es gezeichnet.

Tutorial:

<http://www.crownandcutlass.com/features/technicaldetails/frustum.html>

Transparency:

Das transparente Object ist ein Eiswürfel und wird mit glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_DST_COLOR) gezeichnet.

Features:

- Text render:
Mithilfe der library freetype erstellen wir aus der ttf-Fonts Texturen
Freetype : <http://www.freetype.org/>
Für Text render wurde die Methode rendertext() (im Code nochmal angeführt) von diesem Tutorial verwendet
<http://www.learnopengl.com/#!In-Practice/Text-Rendering>
- Collision detection :
wurde implementiert mithilfe der Library Bullet <http://bulletphysics.org/wordpress/>
Wir verwenden den btKinematicCharacterController zur Steuerung des Roboters. Die Kamera folgt dem Roboter.
- Sound :
Wir haben eine Hintergrundmusik und einen Soundeffekt für das Einsammeln und Zerstören der Blumen implementiert. Fmodex : <http://www.fmod.org/>

Tools:

Zum Bearbeiten der .obj Files wurde Blender, SketchUp und der AssimpViewer verwendet.