



CGUE 20 Rolling Stones Documentation

Felix Eibl 01427688

Inhaltsverzeichnis

Settings.....	2
Steuerung.....	2
Effekte.....	3
Gameplay.....	3
Bibliotheken.....	4
Tutorials.....	4

Settings

1. Window

	Setting	Beschreibung	Standardwert
1	width	Breite des Fensters	800
2	height	Höhe des Fensters	800
3	refresh_rate	Frame Refresh Intervall	60
4	fullscreen	Spiel im Vollbild Modus Starten	false
5	title	Titel des Spiels	RollingStones
6	debug	Spiel im Debug Modus Starten	false
7	debugCam	Auswahl der Debug Kamera (0 oder 1)	0

2. Camera

	Setting	Beschreibung	Standardwert
1	fov	Field of View	70.0
2	near	Camera Near Plane	0.1
3	far	Camera Far Plane	200.0

3. Light

	Setting	Beschreibung	Standardwert
1	brightness	Multiplikator für den Alpha Wert	1
2	particles	Partikeleffekt aktivieren/deaktivieren	true

Steuerung

	Setting	Beschreibung
1	F1	Nur Wireframes Rendern aktivieren/deaktivieren
2	F2	Culling aktivieren/deaktivieren
3	ESC	Spiel beenden
4	Pfeiltaste Links/ Rechts/Rauf/Runter	Charakter in die jeweilige Richtung bewegen
5	Strg Links	Charakter Springen
6	R	Spiel zurücksetzen
7	Space	Pause
8	Shift Links	Partikeleffekt Charakter
9	W/A/S/D	Debug Kamera 0 in jeweilige Richtung bewegen (debug=true und debugCam=0)
10	Maustaste Links/Rechts	Debug Kamera 1 in jeweilige Richtung drehen (debug=true und debugCam=1)
11	Mausrad	Debug Kamera 1 nach oben/unten bewegen (debug=true und debugCam=1)

Effekte

	Effekt	Beschreibung
1	Collision Detection	Realisiert mittels PhysX, in der Klasse Physik. Kollision für alle Plattformen und Objekte. Bei Charakterkollision mit mit Untergrund wird das Springen wieder möglich.
2	Procedural Texture	Realisiert in der Klasse ProceduralTexture. Erzeugt in der Main Klasse für Wasser und Sand
3	Vertex Shader Animation	Realisiert im waterShader und waterShader2. Angewendet in der Level Klasse
4	3D Geometry	Realisiert mit Assimp in den Klassen Model und Mesh. (Totenschädel und Diamanten)
5	FPS and Framerate Independence	Realisiert in der Main und beim updaten von Objekten, Physik und Kamera
6	Intuitive Controlls	Gedrückte Pfeiltasten werden in einem Array gespeichert, Charakter wird anhand des Array bewegt
7	Intuitive Camera	Kamera Folgt dem Charakter und wird beim Springen zeitverzögert aktualisiert. Bei Aktivem Debug Mode, kann die Kamera mittels bewegen der Maus gedreht werden.
8	Illumination Model	Die Scene wird mittels 4 Pointlights beleuchtet die je nach Charakterposition, in der Level Klasse upgedatet werden. Zusätzlich wird die Attenuation der Lichter innerhalb eines Zeitintervall upgedatet. Die Beleuchtung ist im materialShader realisiert. Zusätzlich gibt es noch 4 Pointlights die zum beleuchten des Wassers verwendet werden, diese Beleuchtung wird im waterShader und waterShader2 realisiert. Zusätzlich wird noch die Entfernung zur Kamera berechnet und weiter entfernte Objekte werden dunkler als nahe Objekte gerendert. Für Specular Effekte wird eine zusätzliche Textur verwendet.
9	Textures	Alle Objekte haben Texturen.
10	Moving Objects	Charakter, Kugel, Trennwand, Diamanten
11	Shadow Map with PCF	Von jedem der 4 Pointlights aus wird eine ShadowMap erstellt und mit dem depthShader auf einen Cube gerendert. Der MaterialShader verwendet dann die 4 Cubes um Pixel welche überdeckt werden dunkler zu rendern.
12	GPU Particle System	Im particle Spawner werden Positionen und Farben und Geschwindigkeit der Partikel berechnet und dann mittels Instancing an den particleShader übergeben und gerendert.

Gameplay

Der Spieler startet auf einer Plattform am Anfang des Level. Sobald die Trennwand zwischen Charakter und Kugel heruntergefallen ist beginnt die Kugel sich auf den Charakter zuzubewegen und der Charakter kann bewegt werden. Das Ziel ist es, das Ende des Level zu erreichen und dabei alle Gegenstände einzusammeln. Wenn der Charakter von einer der Plattformen fällt wird das Spiel resettet, ebenso falls der Charakter von der Kugel erfasst wird. Wenn der Charakter es bis ans Ende des Levels geschafft hat ist das Spiel gewonnen und die Benötigte Zeit und erreichten Punkte werden in der Konsole ausgegeben.

Bibliotheken

- Assimp - <https://www.assimp.org/index.php/downloads>
- PhysX - <https://github.com/NVIDIAGameWorks/PhysX>
- GLEW - <http://glew.sourceforge.net/>
- GLFW - <https://www.glfw.org/>
- STB Image - <https://github.com/nothings/stb>

Tutorials

- Water Ripples: <https://medium.com/@joshmarinacci/water-ripples-with-vertex-shaders-6a9ecbdf091f>
- Perlin Noise: <https://flafla2.github.io/2014/08/09/perlinnoise.html>
- PhysiX: <https://documentation.help/NVIDIA-PhysX-SDK-Guide/ScenesAndActors.html>
- Charakter Kontroller: <https://www.informatik-forum.at/forum/index.php?thread/113020-physx-character-controller/>
- OpenGL: <https://learnopengl.com/Introduction>
- Partikeleffekt: <http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-instancing/>
<http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/billboards/>
- Shadow Map: <https://learnopengl.com/Advanced-Lighting/Shadows/Point-Shadows>