

## Dokumentation Hyperspace

### 1. Free movable camera

Es wurde eine 3rd Person Kamera implementiert, die den Avatar im Focus hat.

### 2. Moving objects

Der Avatar lässt sich frei in der Welt bewegen.

### 3. Texture Mapping

Das Level Mesh wird mit einer Textur gezeichnet.

### 4. Simple lighting and materials

Als Beleuchtung wird ein Blinn-Phong Shader verwendet. Im Level befinden sich zwei selbstgesetzte Lichtquellen, die zurzeit als Spotlights verwendet werden.

### 5. CPU Partikel

Bei Aktivierung der Drehfunktion werden bunte Partikel generiert, die in zufällige Richtungen fliegen. Quelle:

<http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-instancing/>

### 6. Animated Textures

Der Level wird mit einer animierten Textur verziert. Diese kann man auf den TV Bildschirmen und dem Portalen beobachten. Quellen:

[http://www.swiftless.com/tutorials/opengl/texture\\_animation.html](http://www.swiftless.com/tutorials/opengl/texture_animation.html)

<https://rdmilligan.wordpress.com/2016/04/15/display-video-using-opengl>

### 7. Normalmapping

Kann auf den Levelblöcken und auf den TV Geräten beobachtet werden. Kann mit F6 aktiviert oder deaktiviert werden. Quelle:

<https://learnopengl.com/#!Advanced-Lighting/Normal-Mapping>

## 8. Shadowmapping

Der Schatten ist auf der Sonnenseite sichtbar. Die Shadowmap wird wiederholt im Fragmentshader abgefragt um Artefakte zu verhindern. Hier werden FBOs verwendet.

Quelle:

<https://learnopengl.com/#!Advanced-Lighting/Shadows/Shadow-Mapping>

<http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-16-shadow-mapping/>

## 9. Controls

W - Vorwärts

A - Links

S - Rückwärts

D - Rechts

SPACE – Jetpack

F2 – zeige FPS

F3 – Wireframemode

F6 – Normalmapping

F9 – Blending

G – Godmode

R -Reset Level

T – Shadowmap Debugmode on

Z - Shadowmap Debugmode off

Y/Z (Je nach Tastaturlayout) - Welt um 180° drehen.

## 10. Basic Gameplay

Die grundlegende Spielmechanik ist mit der Funktion zum Drehen der Welt vorhanden. Kollisionserkennung und Schwerkraft sind vorhanden. Die Steuerung des Charakters wurde verbessert. Das Jetpack kann durch drücken der Leertaste aktiviert werden um kurze Distanzen zu fliegen. Falls der Charakter zu lange fällt verliert man das Spiel.

## 11. Additional Libraries

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

GLFW-Window Manager <http://www.glfw.org/>

GLEW-OpenGL Extension Wrangler <http://glew.sourceforge.net/>

FreeImage-Imageloading <http://freeimage.sourceforge.net/>

Bullet-Physics and Collisiondetection <http://bulletphysics.org/wordpress/>

GLM-Mathematics <http://glm.g-truc.net/0.9.8/index.html>

NShader- GLSL Syntax highlighting [www.horsedrawngames.com/shader-syntax-highlighting-in-visual-studio-2013/](http://www.horsedrawngames.com/shader-syntax-highlighting-in-visual-studio-2013/)

OpenCV: Videos laden und einzelne Frames abfragen.

Freetype für den Text <https://www.freetype.org/>

Modelle wurden mithilfe von Blender selbst erstellt. Die verwendete Textur wurde aus dem Internet heruntergeladen. Leider ist uns die Quelle abhandengekommen.

GLBlend Funktion wird beim Textrendering verwendet.

## 12. Texturen

Das Katzenvideo entstand aus Eigenproduktion

Die Texturen für den Fernseher und das Level kommen von folgendem Link:

<http://jylhis.deviantart.com/art/Free-Metal-Textures-pt-1-318314339>

## 13. Beleuchtung

Das Level wird mit einer Sonne (bidirektionales Licht) beleuchtet. Der Charakter verfügt über ein pulsierendes Punktlicht, welches die Farbe ändert falls die Welt rotiert wird. Charakter und Level haben BlinnPhong-Beleuchtung. Die Videos haben keine Schatten und auch keine Beleuchtung.