

Room 13

Sebastian Adam – 1525543

Heiko Mayer – 1525814



1. Steuerung

In unserem Spiel bewegt sich der Spieler in dem so genannten First-Person-Mode. Die Steuerung ergibt sich deswegen wie folgt:

1.1 Bewegung

Mit den Tasten **W**, **A**, **S**, **D** & **SPACE** kann sich der Spieler nach vorne (**W**), nach hinten (**S**), zur Seite nach links (**A**) und nach rechts (**D**) bewegen. Um zu springen, drückt der Spieler **SPACE**. Diese Angaben stehen jeweils in Relation zur aktuellen Blickrichtung der Kamera. Diese kann wiederum durch das bewegen der **Maus** verändert werden.

Wird die Maus nach oben bewegt, so blickt auch die Spielfigur nach oben. Selbiges gilt für alle anderen Richtungen.

1.2 Objektinteraktion

In unserem Spiel gibt es drei Arten von Objekten. Diese wären:

- bewegliche Objekte
- unbewegliche nicht interagierbare Objekte
- unbewegliche interagierbare Objekte

Die beweglichen Objekte können positioniert werden, in dem der Spieler mit dem Cursor (Zauberstab) auf das Objekt zielt und dann die **linke Maustaste** drückt (und **gedrückt** hält). Dies hat zur Folge das der Spieler das Objekt aufnimmt und es an einer beliebigen anderen Stelle wieder ablegen kann (**linke Maustaste** wieder **loslassen**).

Um mit unbeweglichen Objekten zu interagieren (sofern möglich) muss der Spieler mit dem Cursor (Zauberstab) auf dieses zeigen und die **linke Maustaste** drücken (z.B. Portal im Bilderrahmen).

1.3 Weitere Tastenbelegungen

ESC - Fenster schließen

G - Gravitation ein-/ausschalten

C – (Cheat) Portale ein-/ausschalten

Einstellungen:

F2 - Frametime einblenden/ausblenden (in Konsole)

F3 - Wireframe ein-/ausschalten

F4 - Texture-Sampling-Quality: GL_LINEAR / GL_NEAREST

F5 - Mip-Mapping-Quality: GL_LINEAR / GL_NEAREST /
GL_NEAREST_MIPMAP_NEAREST / GL_LINEAR_MIPMAP_NEAREST /
GL_NEAREST_MIPMAP_LINEAR / GL_LINEAR_MIPMAP_LINEAR

F8 - Frustum-Culling ein-/ausschalten

F9 - Blending ein-/ausschalten

WICHTIG: Die momentanen Einstellungen können jederzeit der Konsole entnommen werden.

2. Objekte

Alle in dem Spiel vorkommenden Objekte haben eine Textur, die von dem Modelloader per UV-Map auf die Meshes gemappt wird.

Die Meisten Models wurden von uns selbst erstellt. Ungefähr ein Drittel der Modelle haben wir aus dem Internet (genaue Quelle können dem jeweiligen Source.txt entnommen werden).

2.1 Beleuchtung

Als Lichtquelle jedes Raumes dient das Objekt Kerze (Shadowmapping von der Kerze als Mittelpunkt aus, genaueres im Punkt 5.1).

3. Libraries

3.1 FreeImage

Um Texturen einfach in unser Programm zu laden haben wir die FreeImage-Library verwendet.

Quelle: <http://freeimage.sourceforge.net/> (zuletzt aufgerufen am 19.06.2017)

3.2 Assimp

Um Models einfach in unser Programm zu laden haben wir die Assimp-Library verwendet.

Quelle: <http://assimp.sourceforge.net/> (zuletzt aufgerufen am 19.06.2017)

3.3 Bullet

Um Physik & Collision Detection in unser Spiel zu bekommen haben wir die Bullet-Physik-Engine verwendet.

Quelle: <http://bulletphysics.org/wordpress/> (zuletzt aufgerufen am 19.06.2017)

3.4 Weitere Librarys

Alle weiteren Bibliotheken haben wir wie im CGUE-Tutorial erwähnt hinzugefügt.

Quelle:

<https://tuwel.tuwien.ac.at/mod/book/view.php?id=345049&chapterid=196>

(zuletzt aufgerufen am 19.06.2017)

4. Features

4.1 Moveable Camera

Wie bereits im Punkt "1.1 Bewegung" erwähnt kann man unsere Kamera im Rahmen der Physikgesetze frei steuern.

Für die Implementierung der Kamera haben wir uns an folgendes Tutorial gehalten:

<http://www.opengl-tutorial.org/beginners-tutorials/tutorial-6-keyboard-and-mouse/> (zuletzt aufgerufen am 19.06.2017)

4.2 Movable objects

Einige unserer Objekte lassen sich wie im Punkt "1.2 Objektinteraktion" beschrieben bewegen. Um dies zu ermöglichen haben wir uns dazu entschieden die s.g. Color-Picking Methode in unser Programm zu implementieren. Diese funktioniert grob beschrieben wie folgt:

1. Jedes bewegliche Objekt wird eine zweite Farbe zugeordnet (jede Farbe darf dabei nur 1-mal vergeben werden!).
2. Wird die linke Maustaste gedrückt wird die ganze Szenen mit diesen zweiten Farben neu gezeichnet (allerdings wird dies nie dem Spieler offensichtlich angezeigt).
3. Aus der Szene mit den zweiten Farben wird der Pixel an der Position des Mauszeigers ausgelesen.

4. Jetzt wird in einer Look-Up-Table (bei uns einem Array aus SceneObjects) nachgesehen ob diese Farbe einem Objekt zugeordnet werden kann.

5. Ist dies der Fall haben wir das gewünschte Objekt identifiziert und können es an die gewünschte Stelle verschieben.

Diese Idee haben wir von folgender Seite:

<http://www.codeguru.com/cpp/g-m/opengl/article.php/c5579/A-3D-Object-Picking-Technique-that-Uses-OpenGL.htm> (zuletzt aufgerufen am 19.06.2017)

4.3 Non-moveable interactive objects

Dies sind in unserem Spiel die „Portale“ (im Bilderrahmen). Hat ein Spieler ein Level (Raum) geschafft, öffnet sich in diesem ein Portal in das nächste Level. Klickt der Spieler dieses an, so wird er in den nächsten Raum teleportiert und das Portal schließt sich wieder. Zur Identifizierung des Portals, wird wie bereits bei den „moveable objects“ die Color-Picking-Methode verwendet.

5. Effekte

5.1 Shadow Maps (with PCF) + Omni-Directional

Dieser Effekt kann in jedem beleuchteten Raum beobachtet werden. Für die Implementierung haben wir uns an folgendes Tutorial gehalten:

<https://learnopengl.com/#!Advanced-Lighting/Shadows/Point-Shadows> (zuletzt besucht am 19.06.2017)

5.2 GPU-Particle System (+Transform Feedback, Instancing)

Dieser Effekt kann in jedem beleuchteten Raum beobachtet werden (die Partikel dienen als „Feuereffekt“ auf dem Kerzendocht). Für die Implementierung haben wir uns an folgendes Tutorial gehalten:

<http://www.mbsoftworks.sk/index.php?page=tutorials&series=1&tutorial=26> (zuletzt besucht am 19.06.2017)

5.3 Portal Rendering

Dieser Effekt kann beobachtet werden, wenn ein Rätsel gelöst wurde. Dann öffnet sich in dem Raum ein Portal, welches in das nächste Level führt. Für die Implementierung haben wir uns an folgendes Tutorial gehalten:

<https://torinmr.github.io/cs148/> (zuletzt besucht am 19.06.2017)

6. Gameplay

In unserem Spiel geht es darum, sich aus einem Raum zu befreien und somit die Welt zu retten (natürlich nur im übertragenen Sinne).

Der Spieler sollte sich im Raum gut umsehen, denn um in den nächsten Raum zu gelangen, muss ein Rätsel gelöst werden.

Wurde das Rätsel gelöst, geht das Kerzenlicht aus und es öffnet sich ein Portal, durch das der Spieler in den nächsten Raum gelangt.

THE WORLD IS YOURS!