

Implementation:

3D Geometry

Die Spiel Umgebung als auch die Waffe des Spielers werden als .obj-Datei bereitgestellt. Die Modelle werden in den Klassen Mesh und Model geladen und erstellt.

Intuitive Controls

Die Steuerung funktioniert allgemein wie bei herkömmlichen Shooter-Games. Die Bewegungsrichtung des Spielers funktioniert mittels WASD, Blickrichtung per Mausbewegung.

Stetiger Input (Steuerung) wurde mittels polling implementiert.

Schießen und Nachladen mittels callbacks.

Intuitive Camera

Der normalen Kamera ist es zwar möglich sich im Raum frei umzuschauen, sie ist jedoch an die y-Achse gebunden (um freies fliegen zu vermeiden).

Mittels F1 kann in den Debug-Modus der Kamera gewechselt werden (freies fliegen).

Textures

Um Texturen korrekt auf Objekten darzustellen werden Textur-Koordinaten verwendet. Sowohl Textur-Daten, als auch deren Koordinaten werden zusammen mit den Objekt-Geometriedaten geladen.

Illumination Model

Zurzeit gibt es eine vier Lichtquelle über der Map (Pointlight). Objekte besitzen object normal vectors um festzustellen, wie hell diese dargestellt werden sollen.

Zusätzlich wird der Roboter (falls aktiviert) von einem weiteren Pointlight beleuchtet (rot) und ein hellblaues Pointlight ist an den Laserstrahl der Lasergun gebunden.

Moving Objects

Manche Ziele welche getroffen werden müssen verändern ihren Standort in definierten Bewegungsmustern. Der Roboter verfolgt die Position des Spielers.

Playable

Die Steuerung, Abfeuern der Waffe, Treffen der Ziele funktionieren. Ebenso ist das Spiel via .exe file ausführbar.

Min. 60 FPS and Framerate Independence

Das Spiel läuft bei den Tests durchgehend mit 60fps. Die Bewegung des Spielers sowie andere Objekte sind framerate-unabhängig.

Adjustable Parameters

Die Auflösung (Breite und Höhe), Fullscreen-Mode, Refresh-Rate sowie der Schwierigkeitsleves des Endbosses sind mittels config file einstellbar.

Advanced Gameplay

Der Spieler muss alle gegnerischen (roten) Objekte treffen, um die Tür zum Endgegner zu öffnen. Wird ein nicht-gegnerisches (grünes) Objekt getroffen hat man verloren und es ist nicht mehr möglich im Spiel weiterzukommen. Hat man den Endgegner oft genug getroffen und wurde dabei selbst nicht abgeschossen hat man gewonnen.

Framerate Independence

Alle Bewegten Objekte sowie die Steuerung sind framerate-independent. Dies wird durch den Einfluss der errechneten Frame-Berechnungszeit auf die Bewegungsgeschwindigkeit erreicht.

Collision Detection

Dem Spieler ist es nicht möglich durch Wände, oder Hindernisse zu laufen. Ebenso kann er nicht durch Wände oder Hindernisse schießen. Zusätzlich ist es auch dem feindlichen Roboter nicht möglich durch Wände zu schießen.

Heads-Up Display

Mithilfe eines 2D Overlays werden dem Spieler zusätzliche Informationen am Bildschirm angezeigt. Insgesamt sind pro Ladung 10 Bullets zu Verfügung, die aktuelle Anzahl wird am linken oberen Bildschirmrand ausgegeben.

Ist das Spiel zu Ende wird angezeigt ob der Spieler gewonnen oder verloren hat. Mit der Taste F2 ist es möglich das HUD aus- und wieder einzublenden.

Effekte:

Lightmap

Für die Berechnung des Umgebungslichtes für statische Objekte (Wände usw.) wird eine zusätzliche Textur importiert welche bereits vorerrechnete Lichtwerte besitzt. Diese wird zur Laufzeit im Shader mit der normalen Textur kombiniert.

Hierarchical Animation

Die Laserpistole ist an den Spieler gebunden und bewegt sich durchgehend mit diesem mit. Wird die Waffe nachgeladen (Taste R) bewegt sich diese mit einer Animation.

Video Texture

Das Objekt „Flatscreen“ besitzt 72 verschiedene Texturen welche beim Erstellen des Modells geladen werden. Die Texturen werden je nach vergangener Zeit dem Objekt zugewiesen (24 Bilder pro Sekunde). So kann ein 3 sekündiges Video, bestehend aus 72 Bildern angezeigt werden.

Bloom/Glow

Lichtquellen und hellbeleuchtete Regionen werden "zum Leuchten gebracht". Dafür werden die hellen Bereiche aus dem Bild extrahiert und mit einem Blur-Filter gerendert. Danach wird das Ergebnis wieder zur ursprünglichen Szene hinzugefügt.

Zusätzliche Features:

Soundeffekte

Beim Abfeuern sowie Nachladen der Laserpistole, beim Treffen eines Ziels als auch beim Feuern mit leerem Magazin werden passende Soundeffekte abgespielt.

Tür öffnen

Um die anfangs geschlossene Tür zu öffnen müssen alle feindlichen Ziele (rot) getroffen werden. Falls ein grünes Ziel getroffen wird (nicht feindlich) ist es nicht mehr möglich die Tür zu öffnen und das Spiel muss neu gestartet werden.

Roboter besiegen

Um den Roboter zu besiegen welcher sich hinter der Tür befindet muss dieser 5x getroffen werden. Bei einem Treffer ertönt ein dumpfes Explosionsgeräusch. Hat man diesen oft genug getroffen, schaltet er sich ab und der Spieler hat gewonnen.

Schwierigkeitsgrad

Über die Einstellung „difficulty“ im config File kann zwischen 4 verschiedenen Schwierigkeitsgraden gewählt werden. Der Schwierigkeitsgrad bezieht sich hierbei auf den Roboter welcher bei höherem Grad genauer und schneller schießt (der Roboter verfolgt jedoch schon beim leichtesten Schwierigkeitsgrad die Position des Spielers)

Laser-Pistole

An das Geschoss der Laser-Pistole ist eine Lichtquelle gebunden, wodurch der Effekt entsteht, als würde die Kugel selbst leuchten. Dementsprechend wurde die Umgebung vergleichsweise dunkel konzipiert um den Laser-Effekt zu verstärken. Ebenso trägt der Effekt Bloom/Glow dazu bei.

Verwendete Websites und Tutorials:

<https://learnopengl.com/>

<https://www.youtube.com/watch?v=n4k7ANAFsIQ&t=371s>

https://www.youtube.com/watch?v=yQx_pMsYqzU&t=2621s

<https://www.youtube.com/watch?v=ZbnEMM7vwmU>

<https://learnopengl.com/Advanced-Lighting/Bloom>

<https://learnopengl.com/In-Practice/Text-Rendering>

Zusätzliche Libraries:

- | | |
|----------------------------------|---|
| (1) Assimp 5.0.1 (object-loader) | https://www.assimp.org/index.php/downloads |
| (2) stb_image.h (image-loader) | https://github.com/nothings/stb/blob/master/stb_image.h |
| (3) irrKlang (sound engine) | https://www.ambiera.com/irrclang/downloads.html |
| (4) Freetype | https://github.com/ubawurinna/freetype-windows-binaries |