Dokumentation

I. Wie man spielt

Steuerung des Spielers (First Person)

W – vorwärts gehen
A – links gehen
D - rechts gehen
S - rückwärts gehen
Maus – Blickrichtung verändern

Die Bewegung mittels WADS ist nicht möglich, wenn der Geist gerade den Weg vorzeigt. Eingaben werden blockiert. Eine Änderung der Blickrichtung mit der Maus ist aber jederzeit möglich.

L – drücken, damit der Geist den Weg vorzeigt
Es ist nur möglich L zu drücken, wenn man einen Checkpoint erreicht hat. Bzw. zu Beginn des Spiels damit der Geist zum ersten Checkpoint fliegt.

Betritt man ein falsches Quadrat, wird man zum letzten erreichten Checkpoint zurückgesetzt. Im Spiel hat man drei Leben wobei man eins verliert, wenn man den falschen Weg wählt.

II. Beschreibung der Implementierung

Gameplay

- Playable

- 3D Geometry
  Die Kristalle, der Baum und die kleinen Geister sind Blender Modelle, die wir selbst erstellt haben. Das Modell für den Leitgeist haben wir aus dem Internet, diese war bereits geriggt und wurde von uns noch animiert.

- Win/Loose Condition
  Der Spieler/die Spielerin hat im Spiel drei Versuche den richtigen Weg zu finden. Tritt man auf ein falsches Feld, verliert man ein Leben und wird zum letzten Checkpoint zurückgesetzt. Hat man alle Leben verloren erscheint ein „Game Over“ Text.
  Hat man es bis zum letzten Checkpoint geschafft, gewinnt man das Spiel.

- Intuitive Controls
- Intuitive Camera
- Textures
- Moving Objects (Leitgeist)
- Documentation
- Adjustable Parameters

optional:

- Heads-Up Display

Der HUD befindet sich am oberen Bildschirmrand. Links sieht der Spieler/die Spielerin wie viele Checkpoints er/sie bereits erreicht hat. Rechts sieht man die Anzahl der verbliebenen Leben als Herzen dargestellt.

Darunter befindet sich ein Hinweistext. Dieser sagt aus, dass man die Taste L drücken soll, wenn der Geist einem den Weg vorzeigen soll. Dieser Text verschwindet anschließend und erscheint wieder, wenn man den nächsten Checkpoint erreicht hat.

- View-Frustum Culling

Für das Frustum Culling haben wir Bounding Boxen verwendet. Die Punkte der Boxen werden dann gegen das Frustum getestet und nur dann, wenn alle Bezugspunkte außerhalb des Frustums sind, wird das Objekt nicht gezeichnet, sonst schon. Mit Drücken der Taste F8 kann man das Culling ein-und ausschalten.

Eine entsprechende Ausgabe, welche Objekte gezeichnet werden und welche nicht kann man sich ausgeben lassen, wenn man die Taste C drückt (falls das Frustum Culling aktiv ist).

Effekte

- Glow
  Den Glow Effekt verwenden wir bei den pinken Kristallen (Checkpoints), bei den zwei kleinen Geistern (weiß und grün) auf dem Baum und bei dem Geist der den Spieler/die Spielerin führt. Ebenfalls verwenden wir den Effekt, um ein Leuchten der Particles zu erzielen.
  Um zu bestimmen, welche Modelle leuchten sollen verwenden wir im Fragmentshader eine uniform Variable (*uniform int glowSample*). Ist diese auf 1 gesetzt, dann wird die Farbe des Modells in der Helligkeitstextur gesetzt ansonsten wird die Farbe mit 0 multipliziert und ist in Folge schwarz.

- Procedural Texturing
  Wir haben hier eine Holz-Textur erzeugt und auf den Baum angewandt, den man links neben dem Spielfeld sieht. Das verwendete Tutorial, um die Berechnungen aufzustellen, haben wir unter dem Punkt „Verwendete Ressourcen“ angeführt.
  Wir haben das Tutorial geringfügig unseren Vorstellungen angepasst und beispielsweise zwei verschiedene Farben verwendet, die sich abwechseln.

- GPU Particle System using Compute Shader
  Unser Particle System besteht aus Glühwürmchen, die durch die Szene fliegen. Es ist definiert wie viele Particles pro Sekunde gespawnt werden und sobald sie die Szene verlassen (x,y,z Wert unter- bzw überschreiten) hören sie auf zu existieren. Die Glühwürmchen bewegen sich in einem Schwarm weiter, wobei die individuellen Bewegungen auf Zufallswerten basieren.

- **Vertex Skinning**
  Das Vertex Skinning haben wir für den Leitgeist verwendet der dem Spieler/der Spielerin den Weg vorzeigen soll. Wir haben dabei ein Modell aus dem Internet verwendet, welches bereits geriggt war, und habe zwei Animationen hinzugefügt. Die erste Animation (Winken) wird ausgeführt, wenn der Geist auf den Spieler/die Spielerin wartet. Die Laufanimation wird ausgeführt, wenn der Spieler/die Spielerin L drückt und der Geist den Weg vorzeigt.

III. Beleuchtung und Texturen

Die Objekte in unserem Spiel werden derzeit mit drei Lichtquellen ausgeleuchtet. Zwei Directional Lights und ein Point Light.

Das **erste Directional Light** schaut in die -z Richtung und soll die Szene generell ausleuchten.

Das **zweite Directional Light** geht vom Player aus.

Das **Point Light** geht in unserem Spiel vom Geist aus. Wenn sich der Geist bewegt, um dem richtigen Weg auszuleuchten, dann bewegt sich auch das Licht stetig weiter. Das Licht zeigt senkrecht nach unten.

Betreffend die Texturierung befinden sich zwei Arten von Objekten im Spiel (kleine Geister, Kristalle) die in Blender modelliert wurden und dort eine Textur zugewiesen bekommen haben, mittels UV Mapping. Der große Baum wurde in Blender modelliert, weißt aber eine prozedural erstellte Textur auf.

Der Boden besteht aus einfachen geometrischen Objekten (Quadraten) die mittels eigenen Setzens der Vertex und UV Koordinaten sowie der Normalen erzeugt wurden.

IV. Verwendete Bibliotheken

- DevIL um Texturen zu laden (*http://openil.sourceforge.net/)*
- ASSIMP um unsere Blender Modelle zu laden (*http://www.assimp.org/)*
- IRRKLANG zum Abspielen der Hintergrundmusik (*https://www.ambiera.com/irrklang/*)
- GLM um Matrizen und Vektoren verwenden zu können
- GLFW
- GLEW

Die drei letzteren wurden aus dem bereitgestellten ECG-Framework übernommen.

V. Verwendete Tutorials/Ressourcen

Camera
https://learnopengl.com/Getting-started/Camera

Assimp, Model und Mesh
https://learnopengl.com/Model-Loading/Assimp
https://learnopengl.com/Model-Loading/Model
https://learnopengl.com/Model-Loading/Mesh

DevIL
http://www.lighthouse3d.com/cg-topics/code-samples/loading-an-image-and-creating-a-texture/
http://openil.sourceforge.net/docs/DevIL%20Manual.pdf

Glow/Bloom
https://learnopengl.com/Advanced-Lighting/Bloom
http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-14-render-to-texture/
http://dev.theomader.com/gaussian-kernel-calculator/

HUD
Alpha blending: https://learnopengl.com/Advanced-OpenGL/Blending
Textur Atlas: https://www.youtube.com/watch?v=6T182r4F6J8

Skybox
https://learnopengl.com/Advanced-OpenGL/Cubemaps

Vertex Skinning
http://ogldev.atspace.co.uk/www/tutorial38/tutorial38.html
https://www.youtube.com/watch?v=f3Cr8Yx3GGA

Procedural Texturing
https://thebookofshaders.com/11/?lan=de

Particle System
Computergrafik Tutorial Slides “Computeshader” und „GPU Particle Systems“
http://www.science-and-fiction.org/rendering/noise.html

Frustum Culling
http://cgvr.cs.uni-bremen.de/teaching/cg_literatur/lighthouse3d_view_frustum_culling/ (clip space approach)