

Documentation for 2nd submission

The incomplete Maze: Bernd Prantner (0726446), Florian Burmann (1128132)

Implementation der Anforderungen:

1. Gameplay

Zu Beginn ist alles weiß. Schießt man einen Ball weg, so fliegt dieser in die Blickrichtung der Kamera (inkl. vertikaler Winkel). Wird eine Wand oder der Boden getroffen, so entsteht dort ein Farbkleck. Der Spieler befindet sich in der First-Person Perspektive und kann sich frei durch den Raum bewegen. Es gibt drei verschiedene Bonus Items, welche mit einem Spotlight markiert sind um sie schon in der Ferne erkennen zu können. Wird ein Bonus Item getroffen, so erhält man mehr Bälle, mehr Zeit oder die Möglichkeit drei Bälle gleichzeitig zu schießen. Man hat gewonnen, sobald man durch das gefundene Zieltor geht, bevor die Zeit zu Ende ist. Manche Positionen sind von der Höhe aus günstig, um das Ziel aus der Ferne lokalisieren zu können.

Als Bonus Item gibt es die Ente (erhöht die Anzahl der Bälle um 50 - blaues Spotlight), das Auto (erhöht die Zeit um 30 Sekunden - rotes Spotlight) und den Croctopus (aktiviert den Triple-Shot auf der rechten Maustaste - grünes Spotlight). Im linken oberen Rand ist immer die Zeit, die übrigen Bälle und Entfernung zum Ziel (Luftlinie) sichtbar.

Man befindet sich immer auf der Höhe des Bodens. Mit der Taste F6 ist es möglich den Fly-Mode zu aktivieren, mit F7 wird alles sichtbar (Wände+Boden für Debug-Zwecke), so ist man unabhängig von der Heightmap und man kann außerdem durch Wände gehen.

2. Effects

Alle Bälle und Bonus Objekte besitzen Schatten welche auf dem Boden und den Wänden sichtbar sind (Shadow Mapping mit PCF).

Farbkleckse werden als Projected Textures an die Wände und den Boden projiziert.

Bonus Items, sowie das Ziel, werden mit einem Cel Shader mit Konturen (Backface-Technik) gezeichnet.

Bei jedem Treffer wird ein Partikelsystem erzeugt, welche das Wegspritzen von Farbe simuliert.

3. Complex Objects

Komplexe Objekte sind Bonus Items und das Ziel, welche als .obj-Dateien über einen Model-Loader mit oder ohne Textur ins Spiel geladen werden. Auch eine beliebig große Heightmap wird mit einem eigenen Loader integriert.

4. Animated Objects

Alle Bonus Items sind animiert und drehen sich sobald sie getroffen sind. Außerdem drehen sich (wie Elektronen um einen Atomkern) drei kleine Kugeln wiederum um diese Bonus Items. Die Drehungen sind hierarchisch voneinander abhängig.

5. Transparency

Über jedem Bonus Item befindet sich ein Spotlight-Cone, damit man die Position der Items bereits aus der Ferne ausmachen kann. Diese Cones werden immer transparenter in Richtung des unteren Endes. Auch das Zieltor hat eine transparente Ebene. Mit F9 kann man den Transparenzmodus ein- und ausschalten.

6. Controls

Die Tasteneingaben und Mausbewegungen werden mit Callbacks erkannt.

- Mausbewegung: Blickrichtung ändern
- Linke Maustaste: Ball schießen
- W, A, S, D: Bewegen
- Rechte Maustaste: drei Bälle gleichzeitig schießen (Croctopus Bonus Item notwendig)

7. Experimenting with OpenGL

VBOs, VAOs und FBOs (Shadow Map, Particle System,...) werden verwendet.

- F1: Help an/aus (= Pause)
- F2: Frame Time an/aus
- F3: Wireframe an/aus
- F4: Texture-Sampling-Quality nearest/bilinear
(nicht wirklich sichtbar, da Textur highdef)
- F5: Mip Mapping-Quality none/nearest/linear
(nur für Farbkleckse und Bonus Items)
- F6: Fly-Mode an/aus
(deaktiviert auch Collision Detection und Erkennung des Erreichens des Ziel)
- F7: Heightmap/alle Wände sichtbar machen
(deaktiviert aber Schatten und Kleckse)
- F8: Frustum Culling an/aus
- F9: Transparency an/aus
(betrifft nur Spotlight Cones von Bonus Items und Ziel)
- F10: Debug Werte im Spiel anzeigen/ausblenden

8. Frustum Culling

Wird mit der Radar Methode durchgeführt¹ und ist besonders wichtig, damit nicht zu viele Projected Textures in den Shader kommen. Natürlich werden so auch nur sichtbare Wände, Bälle und Bonus Items gezeichnet. Auch beim rendern der Shadow Map kommt Frustum Culling teilweise zum Einsatz. Mit F8 kann man Frustum Culling ein und ausschalten, was unmittelbare Auswirkung auf die Performance hat. Für die Heightmap wird keine Culling verwendet, da diese ein einziges Objekt ist und beim Laden bereits geteilt werden müsste.

¹ <http://www.lighthouse3d.com/tutorials/view-frustum-culling/radar-approach-testing-points/>

Features des Spiels:

- Model Loader (obj. Dateien mit oder ohne Textur)
- Heightmap (mit Loader)
- eigene Collision Detection (für Bälle/Wände/Heightmap und Kamera/Wände/Ziel)
- mathematisch korrekte Wurfparabel für Bälle
- dynamisch hinzugefügte Projektionen für Farbkleckse
- verschiedene Bonus Items

Weitere Features:

- Phong Shading der Bälle
- Strafing: Läuft man gegen eine Wand so gleitet man an dieser entlang
- Baryzentrische Heightmap Interpolation der 16x16 Nachbarschaft (für smoothes bergauf- und abgehen)
- 3D Sounds²
(gehen, Ball trifft, je Bonus Item, Zielnähe, win/lose, Background, Treffen auf Wand)
- Head-up Display (HUD)³

Beleuchtete und texturierte Objekte:

Es werden alle Bälle über eine fixe Lichtquelle mit Phong Shading beleuchtet. Bonus Items und das Ziel werden mit Toon Shading mit der selben Quelle beleuchtet und besitzen auch Texturen. Wände und Boden werden (zwecks Gameplay) absichtlich nicht beleuchtet, sondern bleiben weiß.

Zusätzliche Libraries:

- FreeImage Library 3.15.4
Zum Laden der Heightmap
<http://freeimage.sourceforge.net>
- irrKlang 1.4.0b
für Sounds
<http://www.ambiera.com/irrklang/>

² <http://www.ambiera.com/irrklang/>

³ <http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-11-2d-text/>

Implementierte Effekte:

- **Shadow Maps** (with PCF) 1.5 P
Quellen: Folien Repetitorium,
<http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-16-shadow-mapping>
- **Projected Textures** 0.5 P
Quellen: Folien Repetitorium,
http://http.developer.nvidia.com/CgTutorial/cg_tutorial_chapter09.html,
http://www.ozone3d.net/tutorials/glsl_texturing_p08.php#part_8
- **Cel Shading** 0.5 P
Quelle: <http://prideout.net/blog/?tag=toon-shader>
- **+ Contoures (backfaces)** 0.5 P
Quelle: keine
- **Particle System** 1.0 P
inkl. Transform Feedback
Quellen: Folien Repetitorium,
<http://ogldev.atspace.co.uk/www/tutorial28/tutorial28.html>

Model Tools:

Es wurde hauptsächlich Blender verwendet. Die Modelle für die 3 Bonus Items (Duck, Car, Croctopus) stammen samt Texturen aus der LVA "UE Einführung in die Computergrafik". Wände werden erzeugt, indem die XZ-Koordinaten aus einem Textfile geladen werden und dann diese dynamisch dorthin rotiert, verschoben und geschert werden. Die korrekte Y-Koordinate wird aus der Heightmap ausgelesen.