

Übungsblatt 1 – Bericht

Entwicklung eines begehbaren 3D-Labyrinths in Unity mit Ray Tracing

Jonathan Stengl

Konzept

Das Spielkonzept sieht vor, dass der Spieler in einem labyrinthartigen Raum startet und den Ausgang finden muss. Um die Orientierung in den teilweise dunklen Gängen zu erleichtern, ist im Labyrinth eine Taschenlampe versteckt, die der Spieler aufsammeln kann. Mit ihr lassen sich besonders dunkle Bereiche gezielt ausleuchten.

Materialien

Für die visuelle Gestaltung der Szene wurden verschiedene Materialien eingesetzt, um die Anforderungen an Absorption, Reflexion und Transparenz gemäß der Aufgabenstellung zu erfüllen:

- **Boden:** Der Boden besteht aus einem absorbierenden Teppichmaterial mit detailreicher Textur und einer *Normal Map*, die die Struktur des Teppichs plastisch wirken lässt.
- **Wände:** Die Labyrinthwände bestehen größtenteils aus einem Ziegelmaterial, das ebenfalls über eine Bump Map verfügt. An bestimmten Stellen wurde ein stark reflektierendes Material eingesetzt, das nahezu spiegelähnliche Oberflächen simuliert und Ray-Tracing-Reflexionen besonders deutlich sichtbar macht.
- **Decke:** Die Decke verwendet ein Material mit Emissionseigenschaften. Über eine *Emission Map* werden leuchtende Neonröhren simuliert, die mithilfe von Ray Tracing reale Lichtquellen imitieren.
- **Fenster:** Im Startbereich kommen transparente Materialien zum Einsatz, um Fenster realistisch darzustellen.

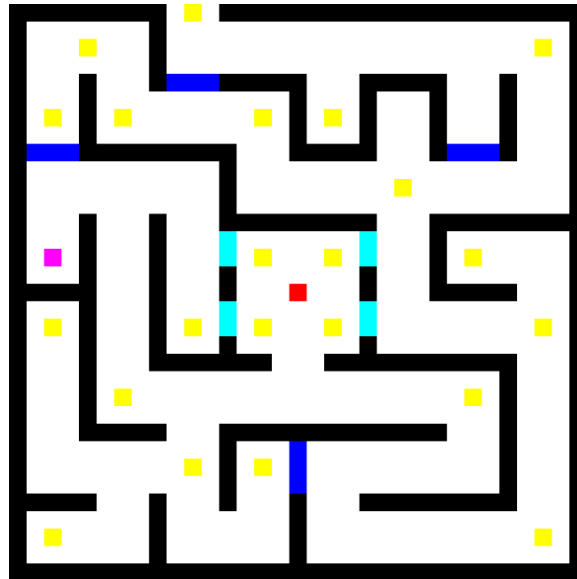
Beleuchtung

Zur Ausleuchtung der Szene wurden mehrere Punktlichter (Point Lights) im Labyrinth platziert, um sowohl direktes als auch indirektes Lichtverhalten sichtbar zu machen. Die Beleuchtung wurde so gesetzt, dass Schattenwurf, Spiegelungen und Materialreaktionen im Ray-Tracing-Modus deutlich erkennbar sind.

Grundriss

Der Plan des Labyrinths wurde mit dem Online-Tool *Piskel* erstellt. Dort konnte der Entwurf nicht nur als PNG-Datei, sondern auch als C-Array exportiert werden. Dieses C-Array wird in die Klasse *MazeGenerator* eingebunden und dient als Grundlage für die Generierung des Levels.

Im C-Array bzw. in der PNG-Datei repräsentieren bestimmte Farben jeweils vordefinierte Objekte. Der *MazeGenerator* liest das Array aus und platziert anhand der Farbwerte die entsprechenden Objekte im Spiel, wodurch das Labyrinth aufgebaut wird. Boden und Decke sind standardmäßig immer vorhanden:



Legende der Farben:

- **Schwarz:** Ziegelsteinwand
- **Blau:** Spiegelnde Wand
- **Gelb:** Punktlicht (Point Light)
- **Cyan:** Fenster
- **Rot:** Spieler
- **Magenta:** Versteckte Taschenlampe

Spieler

Der Spieler stellt die zentrale Figur der Steuerung im Spiel dar und verfügt über eine Reihe grundlegender Fähigkeiten zur Bewegung und Interaktion innerhalb der Spielwelt. Die Steuerung basiert auf Unitys „Character Controller“-Komponente (um dem Spielerobjekt grundlegende Physik zu verleihen) in Kombination mit der Klasse *PlayerMovement*. Das Player-Objekt beinhaltet die Spielerkamera (Egoperspektive) sowie eine Taschenlampe (Spotlight), deren Ausrichtung an die, der Spielerkamera gekoppelt ist.

Fähigkeiten des Spielers:

- **Bewegung:** Der Spieler kann sich zu Fuß bewegen, wobei zwischen Gehen (Standardgeschwindigkeit) und Laufen (durch Halten der Umschalttaste) unterschieden wird.
- **Springen:** Durch Drücken der Leertaste kann der Spieler in die Luft springen.
- **Ducken:** Durch Halten der R-Taste duckt sich der Spieler. Dabei wird seine Körpergröße angepasst und die Geschwindigkeit reduziert.
- **Kamerasteuerung:** Die Mausbewegung ermöglicht eine vollständige Ansichtskontrolle. Das vertikale Schauen wird dabei begrenzt, um ein realistisches Sichtfeld zu erhalten.
- **Taschenlampe:** Der Spieler kann eine Taschenlampe benutzen, sofern er sie eingesammelt hat (Ziel-Objekt zur Erweiterung der Spielmechanik). Mit der F-Taste lässt

sich das Licht ein- und ausschalten. Die Taschenlampe ist standardmäßig deaktiviert, bis sie durch das Einsammeln aktiviert wird.

Taschenlampe einsammeln

Bevor der Spieler die Taschenlampe verwenden kann, muss er sie zunächst im Labyrinth finden und aufsammeln. Die Taschenlampe befindet sich sichtbar auf dem Boden und dient als interaktives Objekt innerhalb der Spielwelt.

Funktionsweise:

Die Taschenlampe wird im Level an einer bestimmten Position abgelegt, an der der Spieler sie entdecken kann. Das Taschenlampen-Objekt ist mit einem „Box Collider“ ausgestattet, der auf „Trigger“ eingestellt ist. Sobald der Spieler den Collider berührt, wird das Ereignis ausgelöst.

Das Taschenlampen-Objekt ist mit dem Skript *SimpleTrigger* verbunden. Beim Auslösen des Triggers führt dieses Skript folgende Aktionen aus:

1. Die Taschenlampe wird entfernt, um das Aufheben zu simulieren.
2. Die Methode *PickupFlashlight()* aus der *PlayerMovement*-Klasse wird aufgerufen.
3. **Der Spieler erhält Licht:** Die Methode *PickupFlashlight()* schaltet das Taschenlampen-Licht des Spielers ein und markiert intern, dass der Spieler nun im Besitz der Taschenlampe ist (*hasFlashlight = true*). Der Spieler kann ab diesem Moment das Licht mit der F-Taste ein- und ausschalten.

Unity & HDRP

Für das Spiel wurde die High Definition Render Pipeline (HDRP) der 3D-Engine Unity verwendet, um realistische Grafik zu ermöglichen. HDRP erlaubt realistische Beleuchtung, Materialien und Schatten und ist ideal für Szenarien mit hohem visuellem Anspruch. Sie bietet moderne Beleuchtungs- und Rendering-Techniken und ist trotz fortschrittlicher Features auch für Einsteiger gut nutzbar – viele Optionen lassen sich per Klick aktivieren. Unity ist darüber hinaus im Internet gut dokumentiert, mit zahlreichen Anleitungen und Tutorials.

Eingesetzte HDRP-Techniken:

- **Raytraced Ambient Occlusion:** Sorgt für realistische Schatten in Ecken und Spalten.
- **Raytraced Global Illumination:** Licht verteilt sich realistisch im Raum – Objekte können indirekt beleuchtet werden.
- **Raytraced Reflections:** Echte Spiegelungen auf Oberflächen, keine Nachahmungen.
- **Raytraced Shadows:** Präzise, weiche Schatten.

Aufgrund von Hardwarebeschränkungen kann es notwendig sein, die Qualität einiger dieser Grafikoptionen zu reduzieren. Dies kann entweder in der „Raytracing Performance“-Datei, die die HDRP-Einstellungen beschreibt, oder im „Global Volume“-Objekt der Szene erfolgen. Mutmaßlich haben die Anzahl und Qualität der Raytraced Shadows einen großen Einfluss auf die Performance des Spiels. Die Schattenqualität einzelner Lichter kann ebenfalls beim entsprechenden Lichtobjekt individuell eingestellt werden.

Quellen (13.05.2025)

- Pixel Editor: <https://www.piskelapp.com/>
- Ziegelstein-Textur: <https://ambientcg.com/view?id=Bricks093>
- Decken-Textur: <https://ambientcg.com/view?id=OfficeCeiling003>
- Teppich-Textur: <https://ambientcg.com/view?id=Carpet016>
- Glas-Textur: <https://3dtextures.me/2020/08/27/glass-frosted-001/>
- Taschenlampen-Objekt: <https://assetstore.unity.com/packages/package/18972>
- Unity: <https://unity.com/de>
- Unity- Tutorial: https://www.youtube.com/watch?v=ad9f_nKU0ZA