# Abgabe 1

## **Einleitung**

Das Ziel dieser Übung war es, eine Umgebung zu erschaffen, die visuelle Eigenschaften wie bei einem mit Raytracing ausgeleuchteten Raum zeigt. Dabei sollten verschiedene Materialien simuliert werden, darunter reflektierende, transparente und lichtabsorbierende Flächen. Zusätzlich musste der Spieler sich durch das Labyrinth bewegen können. Aufgrund der technischen Einschränkungen von macOS wurde echtes Raytracing durch Screen Space Reflections (SSR), Reflection Probes und Adaptive Probe Volumes (APV) ersetzt.

## **Technische Grundlagen**

Die Entwicklung erfolgte mit Unity in der LTS-Version 2022.3.x unter macOS. Als Render Pipeline wurde die High Definition Render Pipeline (HDRP) verwendet. Die Wahl fiel auf HDRP, da diese Pipeline realistische Lichtsimulation, SSR, Volumetrics und Refraction unterstützt. Aufgrund fehlender DirectX12-Unterstützung auf macOS wurde auf DXR-basiertes Raytracing verzichtet.

## **Labyrinthaufbau**

Zunächst gab es die Idee, das Labyrinth mithilfe eine Skripts automatisch zu erzeugen. Diese Idee hatte zunächst funktioniert, allerdings stieß ich dabei auf andere Probleme, bezüglich der Performance, weshalb das Labyrinth im späteren Verlauf statisch erzeugt wurde. Gründe dafür waren Bildraten von unter 1 FPS, wobei die Ursache zunächst nicht gefunden werden konnte. Daraufhin wurde ein Musterraum erzeugt, dieser enthält

* vier Wände,
* einen Boden,
* eine Decke,
* und eine Lichtquelle.

Dieser wurde kopiert vervielfacht, bis sich eine 20x20 Matrix ergab. Anschließend wurden Lichtquellen an nicht benötigten Stellen gelöscht und Wände für Durchgänge des Labyrinthes entfernt. Ein Aufbau des Labyrinths ist im folgenden zu sehen:

Ein Bild, das Muster, Quadrat, Rechteck, Farbigkeit enthält.

KI-generierte Inhalte können fehlerhaft sein.

* Weiß = Gang
* Schwarz = Wand (leerer Raum)
* Blau = Absorbierendes Material
* Grün = Durchlässiges Material
* Gelb = Reflektierendes Material
* Maßstab: Ein Pixel entspricht einem Feld

## **Spielerbewegung**

Zur Navigation wurde ein FirstPersonController aus den Unity Starter Assets eingebunden. Die Steuerung erfolgt per Tastatur (WASD) und Maus. Ein Problem ergab sich durch die Verwendung des alten Input-Systems im Controller, während das Projekt bereits auf das neue Input System umgestellt war. Dieser Konflikt wurde durch das Aktivieren der Einstellung „Both“ im Input-Handling („Active Input Handling“) in den Player Settings gelöst.

## **Materialien und Shader**

### **Reflektierende Oberflächen**

Für spiegelartige Wände kam ein HDRP/Lit Shader mit folgenden Eigenschaften zum Einsatz:

* **Surface Type:** Opaque
* **Metallic:** 1.0
* **Smoothness:** 1.0
* **Reflections:** Hierfür wurde eine Reflection Probe verwendet und SSR

### **Transparente Materialien**

Für Glaswände wurde der HDRP/Lit Shader mit transparentem Rendering verwendet:

* **Surface Type:** Transparent
* **Blend Mode:** Alpha Blend
* **Refraction Model:** Box
* **Farbe: Leichtes Blau, Alpha auf 40**
* **Reflections:** Verwendung eines Reflection Probe

Trotz fehlendem Raytracing konnte mithilfe der Refraction-Option und Lichtsimulation ein glaubwürdiger Glaseffekt erzeugt werden.

### **Absorptive Materialien**

Für Lichtschlucker wie schwarze Wände wurde ein Material ohne Glanz und Reflexion verwendet:

* **Base Color:** #000000
* **Metallic:** 0
* **Smoothness:** 0
* **Surface Type:** Opaque
* Keine Emission, keine SSR

Diese Oberflächen wirken tief und lichtschluckend – vergleichbar mit Vantablack oder Samt.

## **Beleuchtung und Lichtsimulation**

### **Adaptive Probe Volumes**

Ein einziges großes APV wurde verwendet, das den gesamten begehbaren Bereich abdeckt. Ursprünglich war pro Raum ein APV vorgesehen, was zu Shaderfehlern wie „Thread group size must be above zero“ führte. Die Lösung bestand darin, ein zentrales, großes Volumen zu verwenden und dieses korrekt über das Lighting-Panel zu backen.

## **Shader-Probleme & Rosa Materialien**

Nach Neustart oder Import kam es bei manchen Materialien zu einem rosa Erscheinungsbild. Ursache war meist, dass Shader-Zuweisungen nicht gefunden oder Materialien nicht mehr zugewiesen waren. Es wurde leider keine Lösung gefunden, weshalb manuell Shader zugeteilt, sowie die Materialien erneut zugewiesen werden mussten.

## **Fazit**

Trotz technischer Einschränkungen durch macOS (kein echtes Raytracing) wurde das Ziel, ein optisch glaubwürdiges, materialbasiertes 3D-Labyrinth zu realisieren, vollständig erreicht. Durch Kombination aus HDRP, SSR, Reflection Probes und APV konnten realistische Lichtverhältnisse geschaffen werden. Der modulare Aufbau ermöglichte ein dynamisches Leveldesign, und durch gezielte Optimierungen war eine flüssige Spielerfahrung gewährleistet. Die Herausforderungen wie Shader-Probleme, rosa Materialien und Performanceeinbrüche wurden während der Entwicklung systematisch analysiert und teilweise gelöst.