

# **Abschlusspräsentation Immersives VR-Basketballtraining**

Modul: Augmented and virtual Reality

Prof.: Prof. Dr. Thies Pfeiffer

// Gruppe Basketball: Samantha, Mirjam, Maylin

## **Agenda:**

1. Projektidee & Zielsetzung
2. Theoretische Grundlagen
3. Didaktisches & Interaktionskonzept
4. Technisches Umsetzung
5. Skripts
6. Video
7. Herausforderungen
8. Fazit

# Projektidee und Zielsetzung

**Projekt:** Immersives VR-Basketballtraining

**Projektvorstellung:** Das Projekt ist eine interaktive VR-Anwendung, in der Nutzer in einer virtuellen Basketballhalle Körbe werfen und durch sich verändernde Korbpositionen ihre Tiefenwahrnehmung und Koordination trainieren.

**Zielsetzung:** Die Anwendung soll Nutzer:innen ermöglichen, grundlegende Prinzipien der visuellen Wahrnehmung, insbesondere der Tiefenwahrnehmung, in VR aktiv zu erleben. Gleichzeitig sollen Aspekte wie Hand-Auge-Koordination, motorisches Lernen und kognitive Anpassung durch wiederholte Interaktion gestärkt werden.

## Konkrete Zielsetzungen:

- Tiefenwahrnehmung erfahrbar machen durch variierende Korbpositionen (Nah-/Fernbereich)
- Motorisches Lernen fördern durch Wiederholung und Feedback
- Multisensorisches Feedback einbauen
- Gamifizierte Motivation durch Punktesystem und adaptives Zielverhalten
- Räumliches Orientierungsvermögen in der virtuellen Umgebung

**Technologie Stack:** Blender, Unity, Meta Quest Link, Meta Quest 2 u. 3

# Theoretische Grundlagen

## **Fokussierung auf Tiefenwahrnehmung als Kernthema:**

- Fokus auf die visuelle Tiefenwahrnehmung in VR
- Räumliche Einschätzung bei verschiedenen Entfernungen und Größen

## **Aufbau:**

- Korb wechselt seine Positionen im Nah- und Fernbereich
- Verschiedene Entfernungen zum Nutzer:innen der Anwendung
- Im Fernbereich wird der Korb größer skaliert

# Didaktisches- und Interaktionskonzept

## Didaktisches Konzept

- Erlebnisorientierter Lernansatz -> direkte Interaktion mit der virtuellen Umgebung
- Tiefenwahrnehmung und Raumorientierung: Durch das Verschieben des Basketballkorbs wird die visuelle Einschätzung von Distanzen geschult (Unterschied zwischen Nah- und Fernbereich)
- Kognitive Anpassung: Wiederholte Wurfversuche fördern das motorische Lernen
- Feedbacklernen: Echtzeit-Rückmeldungen durch Sound und Vibration
- Prinzip „Learning by Doing“: aktives Erleben und Erproben
- Gamification-Elemente: Punktetafel und bewegende Ziele

## Interaktionskonzept:

- Steuerung über VR-Controller
- Werfen und Fangen durch natürliche Bewegungen
- Feedbacksystem:
  - Auditiv: Soundeffekte bei Aufprall des Balls
  - Haptisch: Vibration beim aufheben des Balls
  - Visuell: Punktanzeige und Timer
  - Adaptiv: Ziel verschwindet nach Treffer und taucht an neuer Stelle auf

# Technische Umsetzung

## Entwicklungsumgebung

- **Game Engine:** Unity
- **3D-Modellierung:** Blender (z. B. für Korb, Netz, Arena)
- **Versionskontrolle:** Git + GitHub
- **VR-Plattform:** Meta Quest 3

## Projektstruktur & Szenenaufbau (Unity)

Szenenaufbau in Main.unity

Die Szene ist logisch in einzelne GameObjects gegliedert:

### 1. Main (Root GameObject)

- Übergeordneter Container für alle Hauptelemente der Szene

### 2. Arena

- Bodenplatte (Basketball court) als Spielfeld

### 3. Korb (Basketball Hoop)

- Importiertes Modell aus Blender/FBX
- Besteht aus Ring, Backboard und Netz
- Wird im Spiel über Skript zu wechselnden Positionen bewegt
- Enthält Trigger zur Treffererkennung

### 4. Ball

- Interaktiv über XR-Grabbing aufnehmbar
- Rigidbody + Collider für realistische Physik
- Haptisches Feedback beim aufheben
- Sound beim Aufprall des Balls

### 5. TriggerZone

- Enthält die Zielpositionen für den Korb
- GameObject mit einem Skript, das Transform[] für Zufallsplatzierungen verwaltet
- Child-Objekte wie Position\_1, Position\_2, ... markieren die Stellen im Raum

### 6. StartButton

- UI-Canvas in der Szene
- Startet das Spiel und startet einen Timer
- Interaktion über XR-Ray Interactor

### 7. Punktetafel (Anzeigetafel)

- Anzeige von Treffern und Punkten
- Mit TextMesh Pro realisiert
- Aktualisierung per Skript bei Treffer

### 8. XR-System

- XR Rig + Kamerasetup für VR-Nutzer
- Handcontroller & Interaktionssystem über XR Interaction Toolkit integriert

# Scripts

## 1. Spielstart

- **ScoreboardManager.cs**
  - Startet das Spiel durch einen Button
  - Zählt den Timer
  - Aktualisiert die Zeitanzeige (Canvas, TextMeshPro)

## 2. Ballinteraktion

- **BallHaptics.cs**
  - Implementiert haptisches Feedback beim Greifen oder Werfen des Balls
  - Nutzt wahrscheinlich XR-Toolkit-Komponenten (z. B. XRGrabInteractable, Haptics)
- **BallBounceSound.cs**
  - Spielt Soundeffekte beim Aufprall ab
  - Trigger über OnCollisionEnter() oder ähnliche Unity-Events

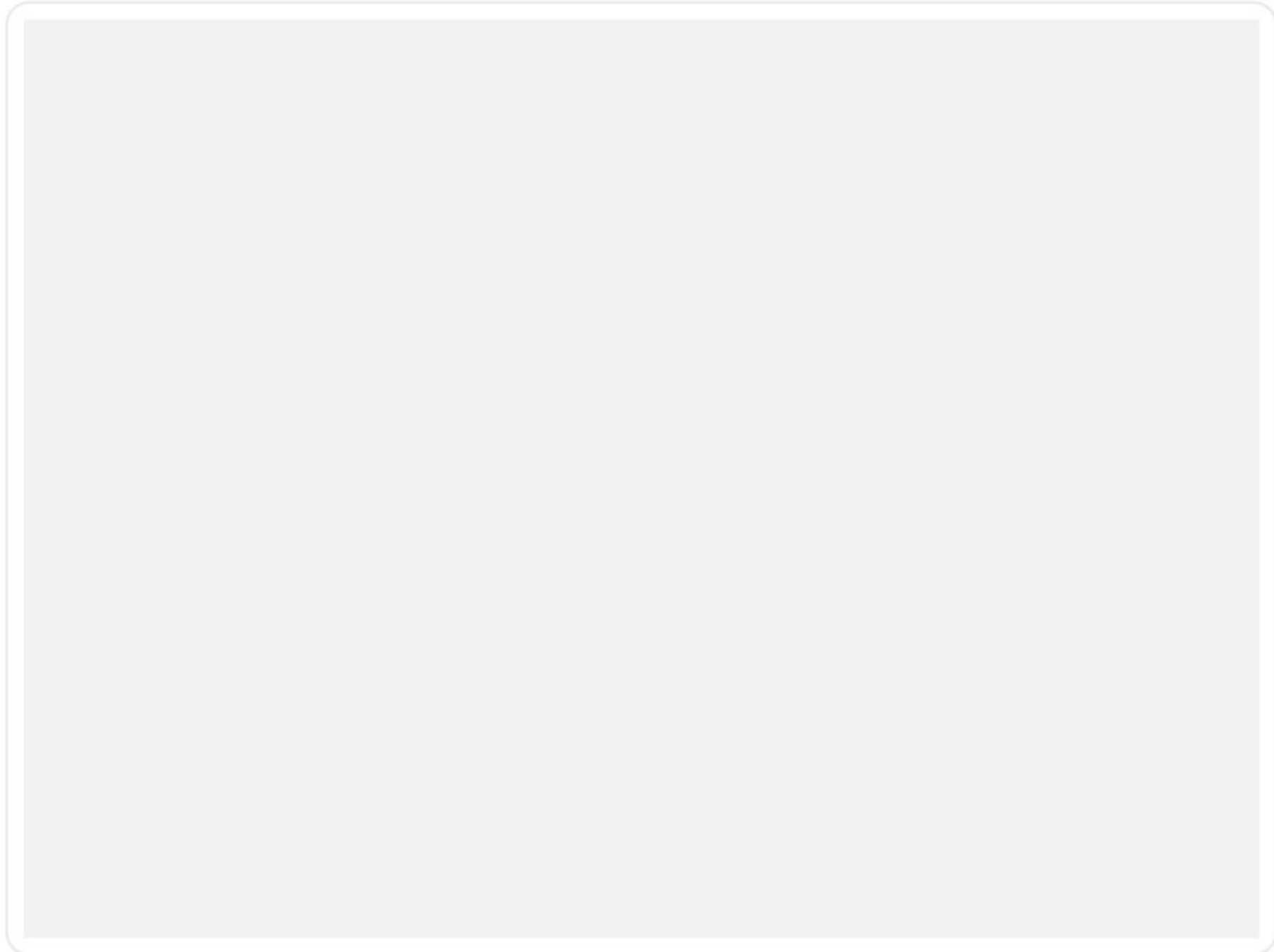
## 3. Treffererkennung & Scoring

- **BasketTrigger.cs**
  - Triggerzone (am Ring oder Netz) erkennt, ob der Ball durch den Korb geht
  - Verknüpft die Treffererkennung mit dem Punktesystem other)
  - Bei erfolgreichem Treffer → Score erhöhen

## 4. Punkteanzeige

- **ScoreManager.cs**
  - Zentrale Verwaltung der Punktzahl und Punkteanzeige
  - Methode zum Erhöhen des Scores
  - Aktualisiert die Punktetafel (Canvas, TextMeshPro)

## Video





# Herausforderungen

## **Probleme mit GitHub und Unity (insbesondere Main.unity)**

- Immer wieder überschriebene und verloren gegangene Daten
- Merge-Konflikte durch überschriebene Daten

## **Probleme mit .gitignore-Datei**

- Immer wieder neue untracked files
- Konflikte beim Comitten, pullen und pushen durch untracked files

## **Probleme bei der Projekteinrichtung**

- Meta Quest Brille mit PC verbinden (lange Ladezeiten)
- Fehlende Rechenleistung für Unity (lange Ladezeiten und Abbrüche)

## **Einarbeitung in die Programme**

- Unity erlernen und einarbeiten
- Mit Blender und Unity parallel arbeiten ging nicht (zu hohe Rechenleistung)
- Zeitverlust durch Ladezeiten der Programme

## **Nicht geschaffte Tasks:**

- **Netzanimation**
  - Umsetzung mit Bones in Blender, Cloth in Unity
- **Sound bei Netz-Berührungen**
- **Reset-Button nach durchlaufen der Spielzeit**
- **Mit Tiefwahrnehmung experimentieren**
  - Mono-Modus simulieren
  - Perspektivische Verzerrung
  - Depth-Cue-Störung

## Fazit

- Die Entwicklung unseres VR-Basketballtrainings in Unity war eine intensive, aber äußerst lehrreiche Erfahrung
- Kein Vorerfahrungen im Team
- Herausforderungen durch GitHub-Zusammenarbeit
- Herausforderungen bei der Projekteinrichtung

Trotz dessen konnten wir diese zentralen Funktionen umsetzen:

- Ballphysik inkl. realistischer Wurfbewegungen und Trigger-Erkennung am Korb
- Bewegliche Korb-Positionen, die das Ziel nach jedem Treffer anpassen
- Anzeigetafel mit Punktestand, die in Echtzeit reagiert
- Audiovisuelle Rückmeldungen beim Ballaufprall

Durch die Arbeit mit Unity und Blender konnten wir erste praxisnahe Einblicke in die VR-Entwicklung gewinnen und dabei wertvolle technische und gestalterische Erfahrungen sammeln.



Vielen Dank für die Aufmerksamkeit!