

Entwicklung einer immersiven Mixed-Reality-Anwendung zur virtuellen Tierbeobachtung und -fotografie

Silja-Marie Fischer, Julia Krauß

Abstract—In den letzten Jahren gewinnt Mixed Reality (MR) zunehmend an Bedeutung, besonders für Bildungs- und Unterhaltungsanwendungen. In dieser Arbeit wird ein Mixed-Reality-Prototyp zur virtuellen Tierbeobachtung auf der Meta Quest 3 entwickelt. Die Anwendung ermöglicht es, virtuelle Tiere in der realen Umgebung zu beobachten und zu fotografieren. Die Architektur folgt dem MVVM-3D-Muster, das Daten, Logik und Benutzeroberfläche klar voneinander trennt und gleichzeitig in der Library komplexere Berechnungen durchführt. Ein zentraler Service sorgt für die Verbindung zwischen Anwendung und Datenbasis. Der Prototyp zeigt, wie MR-Anwendungen mit realistischen Animationen, intuitiver Bedienung und raumbezogener Interaktion umgesetzt werden können.

Index Terms— Mixed Reality, Virtual Wildlife Observation, Virtual Interaction, Interactive Learning, MVVM-3D, Meta Quest.

I. EINLEITUNG

Die wachsende Digitalisierung und häufige Nutzung mobiler Geräte haben das Lernen und die Wahrnehmung von Informationen stark beeinflusst. Untersuchungen zeigen, dass die Aufmerksamkeitsspanne in der Gesellschaft zunehmend kürzer wird [1]. Interaktive Lernmethoden, insbesondere Gamification und Extended Reality (XR), können allerdings die Motivation der Nutzer steigern und fördern langfristiges Lernen. Gamifizierte Elemente wie Herausforderungen, Belohnungen und interaktive Elemente sorgen für eine spielerische und gleichzeitig effektive Wissensvermittlung [2]. Mixed-Reality-Anwendungen (MR) können zudem das psychische Wohlbefinden fördern, indem sie virtuelle Naturerfahrungen schaffen. Ähnlich wie echte Naturaufenthalte können so Stress reduziert und kognitive Leistungen gesteigert werden [3].

Im Bereich Tierbeobachtung und Naturerfahrung existieren bislang nur wenige Anwendungen, die edukativen Mehrwert, spielerisches Erleben und Interaktion verbinden. VR-Lösungen sind häufig in ihren Bewegungs- und Interaktionsmöglichkeiten eingeschränkt, während Mixed-Reality-Ansätze mit virtuellen Tieren in realen Umgebungen bisher selten und meist nur rudimentär umgesetzt werden.

Diese Arbeit wurde am 30. August 2025 im Rahmen des Forschungsprojekts zu Mixed-Reality-Anwendungen für Tierbeobachtung und -fotografie unter der Betreuung von Prof. Dr.-Ing. Sascha Seifert und M.Sc. Max Barchet eingereicht. S.-M. Fischer und J. Krauß sind Studierende des M.Sc.

Dieses Projekt zielt darauf ab, eine prototypische MR-Anwendung zu entwickeln, die es Nutzern ermöglicht, Tiere zu beobachten, mit ihnen zu interagieren und Fotos zu sammeln. Ein digitales Entdeckertagebuch dient der Bereitstellung tierbezogener Informationen. Die Anwendung soll Unterhaltung mit Bildung kombinieren, das Interesse an biologischer Vielfalt wecken und das Bewusstsein für Naturschutz stärken.

II. STAND DER TECHNIK

A. Extended Reality

Extended Reality (XR), oder auch erweiterte Realität, beschreibt die Erweiterung der realen Welt um digitale Elemente [4]. Dies umfasst Augmented Reality (AR), Mixed Reality (MR) und Virtual Reality (VR) [5]. Ziel ist es, eine immersive Umgebung zu schaffen, in der das Bewusstsein von der physischen Welt entkoppelt ist [6]. VR ersetzt die Realität vollständig durch computergenerierte Welten, AR ergänzt die reale Umgebung um virtuelle Objekte, und MR erlaubt zudem Interaktionen zwischen realen und virtuellen Elementen [5], [7].

XR wird in Werbung, Bildung und Unterhaltung eingesetzt, insbesondere aufgrund von Interaktivität, Multisensorik und erhöhter Emotionalität [7], [8], [9].

B. Head-Mounted Displays

Head-Mounted Displays (HMDs) sind visuelle Ausgabegeräte, die am Kopf getragen werden und virtuelle Projektionen direkt vor den Augen erzeugen, wodurch eine 360°-Ansicht ermöglicht wird. Sie kommen in Gaming, Ausbildung, Industrie und Marketing zum Einsatz. HMDs lassen sich in drei Typen unterteilen: See-Through-HMDs erweitern die reale Umgebung mit digitalen Inhalten, Look-Around-HMDs beanspruchen nur einen Teil des Sichtfeldes, sodass die reale Welt weiterhin sichtbar bleibt, und Non-See-Through-HMDs isolieren den Nutzer vollständig [10]. HMDs nutzen stereoskopische Darstellungen, bei denen für jedes Auge ein leicht versetztes Bild angezeigt wird, wodurch ein Eindruck von Tiefe und Raum entsteht [11]. Technisch bestehen sie aus Displays, Linsen, Kameras, Sensoren, Prozessor, Speicher,

Studiengangs Advanced Information Technology (MIT) an der Fakultät für Technik der Hochschule Pforzheim, Tiefenbronner Str. 66, 75175 Pforzheim, Deutschland.
(E-Mail: fische2l@hs-pforzheim.de, kraussju@hs-pforzheim.de)

Batterie sowie Controllern [12]. Das Tracking erfolgt über Inside-Out- oder Outside-In-Systeme [13], Hand- und Fingerbewegungen werden durch Controller, Handschuhe oder optische Systeme erfasst, Eye- und Gesichtstracking übertragen Blick und Mimik [14]. Zusätzliche Technologien wie Passthrough [15], Audiotechnik und haptisches Feedback erhöhen die Immersion und die Interaktivität [14].

C. Virtuelle Tierbeobachtung

In der virtuellen Tierbeobachtung kommen XR-Technologien in Form von AR-Einblendungen (WWF) [16], VR-Erlebnissen (Immotion) [17] oder vollständig virtuellen Zoos zum Einsatz [18], [19]. Vorteile sind Gefahrenfreiheit, Ortsunabhängigkeit und die Möglichkeit, ausgestorbene Arten realitätsnah darzustellen [17], [18], [20], [21]. Aktuelle XR-Spiele für die Meta Quest zeigen deutliche Defizite: National Geographic Explore VR erlaubt Beobachtung und Fotografie von Tieren, bleibt jedoch interaktiv begrenzt [22], [23]. BRINK Traveler und Nature Treks VR fokussieren sich auf Landschaftserkundung bzw. Entspannung, bieten aber kaum Interaktion [24], [25], [26]. Ocean Rift und WildXR liefern Unterwasserwelten oder 360°-Videos, ohne dass Nutzer aktiv eingreifen können [27], [28], [29]. AR- und MR-Anwendungen wie Living Room oder Animalz VR platzieren Tiere in der eigenen Umgebung, liefern aber meist keinen edukativen Mehrwert [30], [31], [32]. Nur ZOSU Zoo kombiniert virtuelle Umgebungen mit Mixed-Reality, wird jedoch hauptsächlich im Zoo-Kontext eingesetzt und adressiert Lerninhalte nur begrenzt [33].

Die Analyse zeigt, dass VR-Anwendungen im Bereich Natur- und Tiererfahrung am weitesten entwickelt sind, während es AR- und MR-Anwendungen oft an realistischen Darstellungen, Interaktivität oder edukativem Nutzen fehlt. Daraus ergibt sich das Potenzial für die geplante Mixed-Reality-Anwendung, die virtuelle Tiere realitätsnah in die persönliche Umgebung integriert, aktive Beobachtung und Fotografie ermöglicht und die Erlebnisse in einem digitalen Entdeckertagebuch dokumentiert. Dadurch wird neben Unterhaltung auch ein klarer edukativer Mehrwert geschaffen, der Wissen über Artenvielfalt vermittelt und für Naturschutz sensibilisiert.

III. ANFORDERUNGEN

Tabelle 1 gibt einen Überblick über die Anforderungen der MR-Anwendung und deren Priorisierung. Sie bildet die Grundlage für die Planung und Entwicklung des Prototyps.

IV. TECHNOLOGIEAUSWAHL

Die Wahl der passenden Technologien ist entscheidend für die erfolgreiche Umsetzung der Anwendung. Da die Hardware bereits durch die Verfügbarkeit an der Hochschule festgelegt war, lag der Fokus auf der Auswahl der geeigneten Entwicklungsumgebung, Programmiersprache und des Architekturmodells.

A. Hardware

Für die Entwicklung wurde die Meta Quest 3 genutzt, die

TABELLE I
ANFORDERUNGEN

Kategorie	Anforderung	Priorität
Umgebung	Nutzer bewegt sich frei auf begehbare Strecke	Muss
Umgebung	Verschiedene Umgebungen (z. B. Wald, Savanne)	Kann
Umgebung	Jede Umgebung enthält einzigartige Tiere	Muss
Umgebung	Unterschiedliche Tageszeiten und Lichtstimmungen	Kann
Tiere	Realistisches Verhalten (z. B. flüssige Bewegungen, Flucht)	Sollte
Tiere	Tiere nur zu bestimmten Tageszeiten aktiv	Sollte
Tiere	Tiere erscheinen zufällig	Sollte
Tiere	Auftreten hängt von Tierart ab	Kann
Kamera	Tiere mit virtueller Kamera fotografieren	Muss
Kamera	Kamera mit Zoom, Fokus, Belichtung	Kann
Kamera	Blitz für Nachtaufnahmen	Kann
Tagebuch	Fotos werden gespeichert	Muss
Tagebuch	Steckbrief mit Infos zu jedem Foto	Muss
Tagebuch	Kombination aus echten und animierten Bildern	Sollte
Tagebuch	„Seltene Entdeckungen“ markieren	Kann
Tagebuch	Personalisierung mit Nutzernamen	Kann
Plattform	App für Meta Quest 3	Muss
Benutzeroberfläche	Intuitive Steuerung von Kamera und Tagebuch	Sollte
Benutzeroberfläche	Wischgesten im Tagebuch	Kann
Account	Account erstellen	Muss
Account	Login	Muss
Account	Eigenes Entdeckertagebuch	Muss
Mixed Reality	Virtuelle Tiere in echter Umgebung	Muss
Mixed Reality	Virtuelle Umweltelemente in echter Umgebung	Sollte
Mixed Reality	Interaktion mit Umgebung	Kann
Belohnungssystem	Fotos werden bewertet	Kann
Belohnungssystem	Seltene Tiere bringen mehr Punkte	Kann
Belohnungssystem	Neue Kamerafunktionen/Umgebungen freischalten	Kann
Missionssystem	Fotografie-Herausforderungen	Kann
Missionssystem	Fortschritt im Tagebuch dokumentieren	Kann
Erweiterungen	Regelmäßige Updates	Kann
Multiplayer	Kooperativer Multiplayer-Modus	Kann

Muss: Unverzichtbare Funktionen für den Prototypen

Soll: Wichtige, aber nicht zwingend erforderliche Funktionen

Kann: Optionale Funktionen zur Erweiterung des Erlebnisses

über einen Snapdragon XR2 Gen 2-Prozessor, hochauflösendes Passthrough durch Farbkameras und Tiefensensoren sowie ein 4K-Display mit bis zu 120 Hz verfügt. Zusätzlich ermöglichen Pancake-Linsen, integrierte Lautsprecher und Handtracking eine hohe Immersion. Damit ist die Brille sowohl für VR- als auch MR-Anwendungen geeignet und bildet die technische Grundlage des Projekts [34].

B. Entwicklungsumgebung

Für die Entwicklung des Prototyps wird eine geeignete MR-Entwicklungsumgebung benötigt.

Unity bietet eine Entwicklung mit C#, wird als einsteigerfreundlich beschrieben und ermöglicht ein schnelles Prototyping. Die Einarbeitung wird durch eine große Community und gute Dokumentation unterstützt. Dank Asset Store, Plugins und guter Integration für die Meta Quest 3 eignet sich Unity besonders für MR-Anwendungen. Nachteilig ist eine geringere Grafikqualität im Vergleich zu Unreal [34], [35], [36].

Die Unreal Engine ermöglicht eine Entwicklung mit C++ und bietet eine hohe Leistung für grafisch hochwertige Spiele. Allerdings ist sie komplexer, schwieriger im Einstieg und mit höheren Kosten verbunden [34], [35], [36].

Da Unity geeigneter für die Entwicklung von Prototypen ist und guten Support für die Meta Quest bietet, wurde diese Entwicklungsumgebung gewählt.

C. Programmiersprache

Die Wahl der Programmiersprache hängt von Umgebung und Zielplattform ab. In Unity wird vor allem C# genutzt, das durch einfache Syntax, automatische Speicherverwaltung und eine große Community die Entwicklung von Prototypen erleichtert. In der Unreal Engine dominiert C++, das durch manuelle Speicherverwaltung hohe Leistung ermöglicht, aber mehr Erfahrung erfordert. Mit Blueprints sind auch Anwendungen ohne Code per Drag-and-Drop umsetzbar [37]. Für dieses Projekt ist die Kombination aus Unity und C# optimal, während C++ eher für komplexe, leistungsorientierte Anwendungen geeignet ist.

D. Architekturmodell

Zur Reduktion der Komplexität und zur klaren Trennung von Zuständigkeiten wird der Code in Frontend und Backend aufgeteilt. Das Frontend auf der Brille steuert Funktionen wie Szenen, animierte Tiere und Interaktionen, während das Backend für die Datenverarbeitung und -bereitstellung zuständig ist. Diese Trennung verbessert Lesbarkeit, Wartbarkeit und Testbarkeit des Codes.

Für das Backend wird das MVC-Modell eingesetzt, da die Anzahl der Schnittstellen überschaubar ist und eine einfache Entkopplung von UI und Logik ausreicht. Das Frontend hingegen ist stark UI- und interaktionslastig. Hier kommt das MVVM-Pattern zum Einsatz, da Data Binding eine automatische Synchronisation zwischen View und Model ermöglicht und so Interaktionen zwischen realen und virtuellen Elementen effizient umgesetzt werden können [38]. Das klassische MVVM wird um eine Library-Schicht erweitert, die komplexe Berechnungen sowie XR-Funktionen wie Tracking, Physikberechnungen und Gestenerkennung kapselt. Zusätzlich enthält die View Komponenten für unterschiedliche Eingaben, etwa über Controller oder Handgesten. Dieses erweiterte MVVM-Modell bildet die Basisarchitektur für das Frontend und wird in dieser Arbeit als MVVM-3D bezeichnet. Dafür bildet das MVC-3DC-Modell nach [39] die Grundlage. Die Trennung von Frontend und Backend unterstützt zudem kollaborative Anwendungen und ermöglicht eine strukturierte, wartbare Implementierung.

V. KONZEPT

A. Aufbau der Anwendung

Die Benutzeroberfläche startet mit der Login-Szene, in der sich Nutzer anmelden oder registrieren können. Nach erfolgreichem Login erscheint das Hauptmenü mit den Optionen Strecke wählen, Konto, Tagebuch und Abmelden. Über die Kontoszene können persönliche Daten bearbeitet, gespeichert oder das Konto gelöscht werden; eine Rückleitung zur Login-Szene ist möglich.

In der Strecke-Szene wählt der Nutzer verschiedene Umgebungen wie Wald, Regenwald oder Antarktis aus. Jede Strecke enthält eigene Tiere und Umgebungselemente, die

zufällig. Die Tiere verhalten sich realistisch. Ein Reh flieht bei Annäherung, eine Schnecke bewegt sich unbeirrt. Im Mixed-Reality-Kontext wird die Interaktion mit der realen Umgebung berücksichtigt, sodass virtuelle Tiere korrekt von realen Elementen verdeckt werden und auf echtem Untergrund stehen.

Die Kamerafunktion blendet eine virtuelle Kamera ins Sichtfeld ein, mit der Fotos aufgenommen und automatisch im Tagebuch gespeichert werden. Jeder Eintrag zeigt Name, Lebensraum, realistische Abbildung, Aufnahmezeitpunkt und Datum. Neue Aufnahmen erweitern das Tagebuch kontinuierlich. Das Spiel kann jederzeit über das Hauptmenü beendet werden. Die Umsetzung der beschriebenen Funktionen erfordert eine durchdachte Softwarearchitektur, um animierte Tiere realitätsnah in den physischen Raum zu integrieren und Interaktionen mit der Umgebung zu ermöglichen.

B. Grundlegende Architektur

Die Architektur ist in drei Schichten gegliedert: Datenbasis, Backend und Frontend. Die Datenbasis verwaltet die Informationen unabhängig vom restlichen Programm, was parallelen Zugriff und flexible Speicherwahl ermöglicht. Das Backend übernimmt Zugriffskontrolle, Datenverarbeitung und Spiellogik und stellt die Daten über standardisierte APIs (Application Programming Interface) mit dem Hypertext Transfer Protocol (HTTP) bereit. Das Frontend ist für die Benutzeroberfläche zuständig und enthält ausschließlich Präsentationslogik.

Die Trennung reduziert Komplexität, verbessert Wartbarkeit und erleichtert Erweiterungen sowie eine mögliche Skalierung oder Multiplayer-Erweiterung.

1) Datenbank

In einer produktiven Anwendung übernimmt die Datenbank eine zentrale Rolle als „Single Source of Truth“ und muss Konsistenz, Integrität, Verfügbarkeit sowie Sicherheit gewährleisten. Für den Prototyp liegt der Fokus jedoch auf der Architektur und den interaktiven Aspekten der MR-Anwendung, nicht auf dauerhafter Datenpersistenz oder Performanz. Daher wird die Datenhaltung durch lokal gespeicherte CSV-Dateien simuliert, die leicht lesbar, versionierbar und plattformunabhängig nutzbar sind. Diese Struktur erlaubt eine spätere Migration in relationale Datenbanken. Bilder und Videos werden ebenfalls lokal gespeichert, könnten in produktiven Umgebungen jedoch in Objekt-Datenbanken ausgelagert werden.

2) Backend

Das Backend folgt dem MVC-Muster und wird durch ein Repository ergänzt, das den Datenbankzugriff kapselt. Dadurch entsteht eine modulare und skalierbare Architektur. Das Repository übernimmt das Laden, Speichern und Löschen von Datensätzen und wandelt diese in nutzbare Objekte um. Das Model definiert die Datenstrukturen wie etwa eine Klasse „User“ mit ID, Name und E-Mail. Die View wird in Form von REST-API-Endpunkten bereitgestellt, über die Clients auf Funktionen und Daten zugreifen können. Der Controller koordiniert den Datenfluss zwischen den Schichten, enthält zusätzliche Logik wie das Anlegen von Accounts und übernimmt bei Erweiterungen zentrale Aufgaben wie Validierung und

Synchronisierung. Durch die klare Trennung bleiben Anpassungen auf einzelne Schichten beschränkt, was Wartbarkeit und Erweiterbarkeit sicherstellt.

3) Frontend

Das Frontend basiert auf dem Architekturmodell MVVM-3D, das die klassischen MVVM-Komponenten Model, View, Library und ViewModel für XR erweitert. Das ViewModel übernimmt die zentrale Steuerung, empfängt Änderungen von View oder Model und synchronisiert diese über Data Binding. Properties verknüpfen UI-Elemente mit Variablen und gewährleisten eine automatische Aktualisierung zwischen Anzeige und Datenbasis.

Die klare Trennung erlaubt zudem die Einbindung von Logiken aus der Library. Nutzeraktionen werden von der View an das ViewModel übergeben, dort — wenn nötig unter Einbezug der Librarylogik — verarbeitet und an das Model sowie Backend weitergeleitet, Änderungen fließen zurück in die View. Für Mehrbenutzerszenarien wurde MVVM-3D nach [39] konzeptionell zu MVVM-3DC erweitert. Grundlage kann der sogenannte Centralized Mode bieten, bei dem das Backend als Server und mehrere Frontend-Instanzen als Clients agieren. Zudem muss eine bidirektionale Kommunikation zwischen Server und Clients hergestellt werden, sodass Nutzeraktionen in Echtzeit synchronisiert werden können.

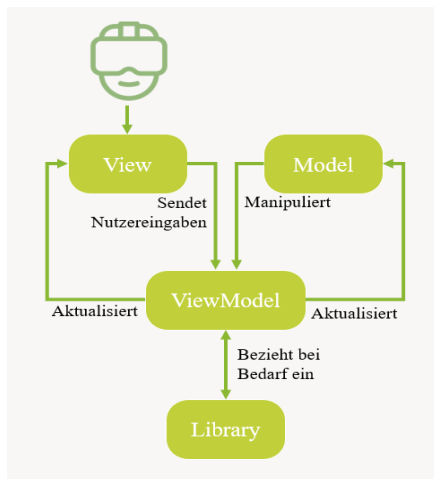


Abb. 1. MVVM-3D (eigene Darstellung)

C. Interaktion

Die Interaktion erfolgt über die Meta Quest 3 mit Controllern, Handtracking und Texteingaben. Controller dienen zur Navigation, Aktionen und Fotografie, Handtracking ermöglicht natürliche Gesten, und Texteingaben unterstützen Anmeldung und Personalisierung. Ein zentrales Element ist die vordefinierte, kollisionsfähige Umgebung, die über ein Mesh (3D-Gitter aus Polygonen) abgebildet wird [40]. Das Mesh simuliert Boden, Wände, Möbel und Hindernisse und ermöglicht Tieren realistische Flucht- und Interaktionsreaktionen. Wird der Raum nicht vollständig erfasst, können unrealistische Effekte wie Durchlaufen realer Objekte auftreten. Für Szenarien wie einen Waldspaziergang wäre kontinuierliches Scannen nötig, was technisch

anspruchsvoll ist, hohen Speicher- und Rechenaufwand erfordert und gegebenenfalls eine Cloud-Auslagerung sowie eine stabile Internetverbindung nötig machen würde. Zudem empfiehlt Meta die Nutzung der Quest-Headsets nur in Innenräumen [41]. Im Prototyp wird daher ein einzelner, statisch gescannter Raum verwendet, der als Grundlage für Interaktion und Kollisionserkennung dient. Dies gewährleistet stabile Performance und ermöglicht spätere Integration dynamischer Scanverfahren.

Bewegungen der Tiere erfolgen in Abhängigkeit zur Nutzerposition. Die Library berechnet den Abstand zwischen Nutzer und Tier. Ein definierter Fluchtradius löst dann Zustandswechsel der Animationen aus, wenn er unterschritten wird. Fotos werden über den Controller aufgenommen. Das System greift auf das aktive Tier zu, erzeugt einen eindeutigen Dateinamen und speichert das Bild im zugehörigen Verzeichnis. Das ViewModel legt die Verbindung zwischen Tagebuch und Eintrag an und übergibt sie an den Service, der über einen API-Endpunkt den Eintrag dauerhaft speichert. Nach erfolgreicher Speicherung wird er sofort in der Benutzeroberfläche angezeigt.

Das Konzept kombiniert also verschiedene Eingabeformen mit Controllern und Händen, eine begehbare Umgebung, tierisches Verhalten und Fotografie. Es ist zugleich für Erweiterungen wie Raumscanning oder Multiplayer vorbereitet.

VI. PROTOTYPENTWICKLUNG

In den folgenden Unterkapiteln werden zentrale Aspekte der Prototypimplementierung erläutert, insbesondere die Umsetzung von MVVM-3D und Interaktionen.

A. Datenbank

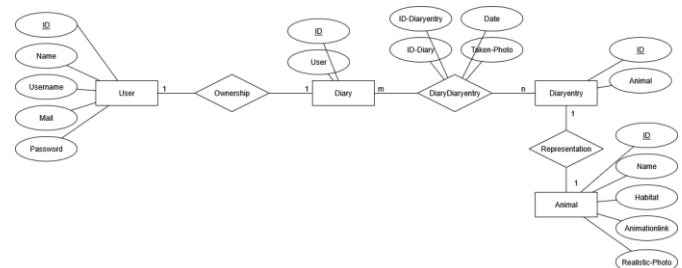


Abb. 2. ER-Diagramm Datenbank (eigene Darstellung)

Für die Datenbasis wurden fünf CSV-Dateien lokal gespeichert und über GitHub versioniert, sodass ein Austausch möglich war, ohne ein produktives System zu nutzen. Alle Dateien liegen in Textform vor und können später in eine relationale Datenbank überführt werden. Abbildung 2 zeigt das ER-Diagramm mit den Klassen *Animal*, *User*, *Diary*, *Diaryentry* sowie der Verbindungsklasse *DiaryDiaryentry*. Die Klasse *Animal* enthält Informationen zu den virtuellen Tieren (ID, Name, Lebensraum, Pfade zu 3D-Objekten und Fotos). Medien werden nur über Pfade referenziert. *User* speichert ID, Name, Username, Mail und Passwort. Für ein produktives System ist hier ein externer Dienst sinnvoll, da die Authentifizierung so sicherer abgewickelt werden kann. *Diaryentry* verknüpft Tagebücher mit Tieren. Da ein Tagebuch

mehrere Einträge enthalten und ein Eintrag mehreren Tagebüchern zugeordnet werden kann, wurde die Verbindungsklasse *DiaryDiaryentry* eingeführt. Diese enthält zusätzlich Datum und Pfad zu aufgenommenen Fotos, wodurch personalisierte Einträge ohne Redundanz möglich sind. Die Struktur trennt generische Inhalte von nutzerspezifischen Informationen, bleibt erweiterbar und erleichtert die spätere Migration in eine relationale Datenbank.

B. Backend

Das Backend wurde in C# mit LINQ-Ausdrücken und dem Framework ASP.NET Core implementiert. Alle Repositories folgen demselben Aufbau. Sie erstellen eine Verbindung zu den CSV-Dateien, lesen deren Inhalte zeilenweise ein und mappen sie auf Attribute von C#-Objekten. Änderungen erfolgen durch Rückschreiben der Attributwerte in die entsprechenden Zellen. Welche Objekte geladen werden, wird über LINQ-Ausdrücke gesteuert, die SQL-ähnliche Abfragen im Code ermöglichen.

Die Models bilden ausschließlich die Datenstruktur ab und stellen Getter- und Setter-Methoden bereit. Die Controller übernehmen die Bereitstellung der Daten ans Frontend über REST-Endpunkte in ASP.NET Core. Dabei enthalten die Controller zusätzliche Logik, wie das automatische Erzeugen einer ID, das Anlegen eines verknüpften Tagebucheintrags oder das Löschen aller zugehörigen Einträge vor der Entfernung eines Tieres, um Konsistenz und Speicheroptimierung zu gewährleisten.

C. Frontend

Für die Einrichtung der Unity-Umgebung in Verbindung mit der Meta Quest 3 wird Unity über den Hub installiert und das Build-Target auf Android gesetzt. Im Entwicklermodus können Anwendungen per USB-C mit „Build and Run“ direkt als APK auf die Meta Quest übertragen und gestartet werden, wodurch ein schnelles Testen und Anpassen möglich ist.

Die Implementierung des Frontends erfolgt in C# mit Unity-spezifischen Befehlen. Inhalte werden in „Scenes“ organisiert, UI-Elemente im Inspector angepasst und mit Variablen oder Events verknüpft. Models spiegeln die Backend-Daten wider und werden per *SerializeField* serialisierbar gemacht. Services kapseln die Kommunikation mit dem Backend über einen zentralen HTTP-Client und geben Objekte an die ViewModels weiter. Die ViewModels implementieren die Logik zwischen View und Model, verwalten Daten in *ObservableCollections* und sorgen über *PropertyChanged* für die automatische UI-Aktualisierung. Komplexere Berechnungen werden in die Library ausgelagert, die zudem eine Brücke zu Unity-spezifischen Funktionen (*MonoBehaviour*) bereitstellt. Die View-Schicht bindet UI-Elemente an ViewModel-Funktionen und ist damit vollständig Unity-spezifisch. Szenenwechsel oder einfache Aktionen werden direkt in der View umgesetzt, während komplexe Logiken über das ViewModel angestoßen werden. Um Daten wie den aktuellen User über Szenenwechsel hinweg verfügbar zu machen, wird eine *SessionData*-Klasse genutzt, die sich auch für Erweiterungen wie Spielstände eignet.

D. Interaktion

Die Interaktion im Prototypen wurde mit dem Meta All-in-One Software Development Kit (SDK) umgesetzt. Dieses stellt unter anderem Building Blocks für Controllersteuerung, Handtracking und Passthrough bereit [42]. Für die Umgebung wird das Mixed Reality Utility Kit (MRUK) in Kombination mit Effect Mesh genutzt. Dadurch entsteht ein statisches, kollisionsfähiges Mesh, das Wände, Boden und Hindernisse abbildet. Virtuelle Tiere reagieren realistisch auf die Umgebung, und die Performance bleibt stabil. Passthrough sorgt dafür, dass reale Elemente weiterhin sichtbar sind. Controller erlauben präzise Eingaben. Handtracking erkennt Gesten wie Zeigen, Greifen oder Wischen.

VII. ZUSAMMENFASSUNG UND AUSBLICK

Die vorliegende Arbeit demonstriert die Umsetzung eines Mixed-Reality-Prototyps für die virtuelle Tierbeobachtung auf der Meta Quest 3. Durch die Verwendung der MVVM-3D-Architektur konnte eine klare Trennung zwischen Daten, Logik und Darstellung erreicht werden. Controller, Handtracking und kollisionsfähige Meshes ermöglichen eine interaktive, immersive Nutzung. Data Binding sorgt dafür, dass Änderungen in Echtzeit in der Ansicht sichtbar werden, was besonders für Fotografie und Echtzeitinteraktionen relevant ist. Trotz bestehender Einschränkungen bei der Bedienung von UI-Elementen auf der Brille zeigt der Prototyp das Potenzial von Mixed-Reality-Anwendungen für Lern- und Spielerlebnisse. Die Nutzung von Building Blocks des Meta SDK beschleunigt die Implementierung und bietet eine stabile Grundlage für zukünftige Erweiterungen.

Zukünftige Entwicklungen könnten qualitativ hochwertigere Tieranimationen, eine virtuelle Tastatur zur direkten Texteingabe, dynamisches Raumschwenken oder Multiplayer-Funktionen umfassen. So könnten mehrere Nutzer gleichzeitig dieselbe Szene erleben und gemeinsam mit den virtuellen Tieren interagieren. Auch die Verbesserung der Reaktionsfähigkeit durch optimiertes Data Binding eröffnet neue Möglichkeiten für Interaktion und Fotografie.

Insgesamt liefert der Prototyp einen soliden Ausgangspunkt für die Weiterentwicklung immersiver, interaktiver Tierbeobachtungsanwendungen.

LITERATURVERZEICHNIS

- [1] Max-Planck-Institut für Bildungsforschung, „Mit der Informationsflut sinkt die Aufmerksamkeitsspanne der Gesellschaft“. Zugegriffen: 16. August 2025. [Online]. Verfügbar unter: <https://www.mpib-berlin.mpg.de/pressemitteilungen/informationsflut-senkt-aufmerksamkeitsspanne>
- [2] E. Ratinho und C. Martins, „The role of gamified learning strategies in student’s motivation in high school and higher education: A systematic review“, *Heliyon*, Bd. 9, Nr. 8, S. e19033, Aug. 2023, doi: 10.1016/j.heliyon.2023.e19033.
- [3] M. G. Berman u. a., „Interacting with nature improves cognition and affect for individuals with depression“, *J. Affect. Disord.*, Bd. 140, Nr. 3, S. 300–305, Nov. 2012, doi: 10.1016/j.jad.2012.03.012.
- [4] B. Zobel, S. Werning, D. Metzger, und O. Thomas, „Augmented und Virtual Reality: Stand der Technik, Nutzenpotenziale und Einsatzgebiete“, in

- Handbuch Mobile Learning*, C. de Witt und C. Gloerfeld, Hrsg., Wiesbaden: Springer Fachmedien, 2018, S. 123–140. doi: 10.1007/978-3-658-19123-8_7.
- [5] L. Tremosa, „Beyond AR vs. VR: What is the Difference between AR vs. MR vs. VR vs. XR?“, The Interaction Design Foundation. Zugriffen: 25. März 2025. [Online]. Verfügbar unter: <https://www.interaction-design.org/literature/article/beyond-ar-vs-vr-what-is-the-difference-between-ar-vs-mr-vs-vr-vs-xr>
- [6] S. Agrawal, A. Simon, S. Bech, K. Bærentsen, und S. Forchhammer, „Defining Immersion: Literature Review and Implications for Research on Immersive Audiovisual Experiences: 147th AES Pro Audio International Convention“, *J. Audio Eng. Soc.*, Bd. 68, Nr. 6, S. 404–417, 2019, doi: 10.17743/jaes.2020.0039.
- [7] A. Mehler-Bicher und L. Steiger, „Augmentierte und Virtuelle Realität“, in *CSR und Digitalisierung: Der digitale Wandel als Chance und Herausforderung für Wirtschaft und Gesellschaft*, A. Hildebrandt und W. Landhäußer, Hrsg., Berlin, Heidelberg: Springer, 2021, S. 243–258. doi: 10.1007/978-3-662-61836-3_16.
- [8] S. Barta, R. Gurra, und C. Flavián, „Augmented reality experiences: Consumer-centered augmented reality framework and research agenda“, *Psychol. Mark.*, Bd. 42, Nr. 2, S. 634–650, 2025, doi: 10.1002/mar.22143.
- [9] A. Mehler-Bicher und L. Steiger, *Augmented Reality: Theorie und Praxis*. De Gruyter Oldenbourg, 2014. doi: 10.1524/9783110353853.
- [10] Seydel, „Head Mounted Display - 5 Einsatzgebiete garantiert“. Zugriffen: 28. August 2025. [Online]. Verfügbar unter: <https://triboot.de/head-mounted-displays/>
- [11] J. Erl und B. Danneberg, „VR-Brillen 2025: Vergleich & Kaufberatung – Das müsst ihr wissen“. Zugriffen: 27. August 2025. [Online]. Verfügbar unter: <https://mixed.de/vr-brillen-vergleich/>
- [12] IMARC Group, „Head-Mounted Displays (HMDs): A Look into Immersive Technologies“. Zugriffen: 29. August 2025. [Online]. Verfügbar unter: <https://www.imarcgroup.com/insight/head-mounted-displays-hmds-a-look-into-immersive-technologies>
- [13] R. William, „Inside-Out vs Outside-In VR Tracking: The Ultimate Guide“. Zugriffen: 29. August 2025. [Online]. Verfügbar unter: <https://arvrtips.com/inside-out-outside-in-vr-tracking/>
- [14] M. Wölfel, *Immersive Virtuelle Realität: Grundlagen, Technologien, Anwendungen*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2023. doi: 10.1007/978-3-662-66908-2.
- [15] R. Carter, „What is VR Passthrough and How is it Shaping the Future of XR?“, *XR Today*. Zugriffen: 30. August 2025. [Online]. Verfügbar unter: <https://www.xrtoday.com/virtual-reality/what-is-vr-passthrough-and-how-is-it-shaping-the-future-of-xr/>
- [16] WWF, „Heimische Wildtiere per Augmented Reality erleben“. Zugriffen: 25. März 2025. [Online]. Verfügbar unter: <https://www.wwf.de/aktiv-werden/augmented-reality>
- [17] Immotion, „IMMOTION | The Global Leader in Immersive Edutainment“, IMMOTION | The Global Leader in Immersive Edutainment. Zugriffen: 18. März 2025. [Online]. Verfügbar unter: <https://edu.immotion.co>
- [18] Gaia, „Zomertour 2023 | GAIA“. Zugriffen: 18. März 2025. [Online]. Verfügbar unter: <https://www.gaia.be/nl/campagnes/zomertour-2023>
- [19] Nathalie, „China VR Zoo“, Schweizer Virtual Reality News. Zugriffen: 18. März 2025. [Online]. Verfügbar unter: <https://vr-room.ch/2018/01/08/erster-vr-zoo-in-china-eroeffnet/>
- [20] Zoo Leipzig, „Gorilla Trek“, Zoo Leipzig. Zugriffen: 18. März 2025. [Online]. Verfügbar unter: <https://www.zoo-leipzig.de/tiere-erlebniswelten/erlebniswelten/pongoland/gorilla-trek/>
- [21] F. Sukmawati, E. B. Santosa, und T. Rejekiningsih, „Design of Virtual Reality Zoos Through Internet of Things (IoT) for Student Learning about Wild Animals. | EBSCOhost“. Zugriffen: 18. März 2025. [Online]. Verfügbar unter: <https://openurl.ebsco.com/contentitem/doi:10.18280%2Fria.370225?sid=ebsco:plink:crawler&id=ebsco:doi:10.18280%2Fria.370225>
- [22] Meta, „National Geographic Explore VR für Meta Quest | Quest VR-Games | Meta Store“. Zugriffen: 26. August 2025. [Online]. Verfügbar unter: <https://www.meta.com/de-de/experiences/national-geographic-explore-vr/2046607608728563/?srsltid=AfmBOorM2xf3Rcf918a1jHmO7uqH9ncTd5dBgzzV7yaCc8soghS5ln>
- [23] T. Bezmalinovic, „National Geographic VR im Test: VR-Reisen für Quest (2)“, MIXED. Zugriffen: 26. August 2025. [Online]. Verfügbar unter: <https://mixed.de/national-geographic-vr-test/>
- [24] Meta, „BRINK Traveler auf Meta Quest“, Oculus. Zugriffen: 26. August 2025. [Online]. Verfügbar unter: <https://www.meta.com/de-de/experiences/brink-traveler/3635172946605196/>
- [25] „BRINK XR“. Zugriffen: 26. August 2025. [Online]. Verfügbar unter: <https://www.BrinkXR.com/>
- [26] Meta, „Nature Treks VR für Meta Quest | Quest VR-Games | Meta Store“. Zugriffen: 26. August 2025. [Online]. Verfügbar unter: <https://www.meta.com/de-de/experiences/nature-treks-vr/2616537008386430/>
- [27] Meta, „Ocean Rift auf Meta Quest“, Oculus. Zugriffen: 26. August 2025. [Online]. Verfügbar unter: <https://www.meta.com/de-de/experiences/ocean-rift/2134272053250863/>
- [28] Picselica, „Ocean Rift“, Picselica. Zugriffen: 26. August 2025. [Online]. Verfügbar unter: <https://www.picselica.com/oceanrift>
- [29] Meta, „WildXR für Meta Quest | Quest VR-Games | Meta Store“. Zugriffen: 26. August 2025. [Online]. Verfügbar unter: <https://www.meta.com/de-de/experiences/wildxr/3634799699973926/>
- [30] Meta, „Living Room auf Meta Quest“, Oculus. Zugriffen: 26. August 2025. [Online]. Verfügbar unter: <https://www.meta.com/de-de/experiences/living-room/7778145568911617/>
- [31] N. Streule, „Living Room“, Thoughtfish. Zugriffen: 26. August 2025. [Online]. Verfügbar unter: <https://www.thoughtfish.de/projects/living-room/>
- [32] Meta, „Animalz Mixed Reality auf Meta Quest“, Oculus. Zugriffen: 26. August 2025. [Online]. Verfügbar unter: <https://www.meta.com/de-de/experiences/animalz-mixed-reality/5583019775107608/>
- [33] Meta, „ZOSU Zoo auf Meta Quest“, Oculus. Zugriffen: 26. August 2025. [Online]. Verfügbar unter: <https://www.meta.com/de-de/experiences/zosu-zoo/3623396967784471/>
- [34] C. Dadson, „Unreal vs. Unity“, Design4Real. Zugriffen: 12. April 2025. [Online]. Verfügbar unter: <https://design4real.de/unreal-vs-unity/>
- [35] M. Frank, „Unity vs. Unreal - Welche Engine ist besser?“, nobreakpoints. Zugriffen: 12. April 2025. [Online]. Verfügbar unter: <https://blog.nobreakpoints.com/unity-vs-unreal-engine/>
- [36] Rocketbrush, „Unity vs Unreal: What to Choose in 2025?“, Zugriffen: 12. April 2025. [Online]. Verfügbar unter: <https://rocketbrush.com/blog/unity-vs-unreal-engine-which-one-should-you-choose-in-2024>
- [37] Circuit Stream, „Circuit Stream · C# vs C++: Complete Comparison Between Unity and Unreal Programming Language“. Zugriffen: 27. August

2025. [Online]. Verfügbar unter: <https://www.circuitstream.com/en/blog/c-vs-c-complete-comparison-between-unity-and-unreal-programming-language>

[38] adegeo, „Data binding overview - WPF“. Zugegriffen: 31. Juli 2025. [Online]. Verfügbar unter: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/data/>

[39] S. Benbelkacem, D. Aouam, N. Zenati-Henda, A. Bellarbi, A. Bouhena, und S. Otmame, „MVC-3D: Adaptive Design Pattern for Virtual and Augmented Reality Systems“, 1. März 2019, *arXiv*: arXiv:1903.00185. doi: 10.48550/arXiv.1903.00185.

[40] R. Awati, „What is 3D mesh? | Definition from TechTarget“, WhatIs. Zugegriffen: 30. August 2025. [Online]. Verfügbar unter: <https://www.techtarget.com/whatis/definition/3D-mesh>

[41] Meta, „Gesundheits- und Sicherheitshinweise | Meta Quest 3 | Meta Store“. Zugegriffen: 30. August 2025. [Online]. Verfügbar unter: https://www.meta.com/de/legal/quest/health-and-safety-warnings/quest-3/?srsltid=AfmBOooPETPtFKDh3HdagR_8ydIVfx8j-LseEdJudS5afWLPXxkgjN4U.com

[42] S. Dev, „Exploring Meta’s Building Blocks“, Antaeus AR. Zugegriffen: 30. August 2025. [Online]. Verfügbar unter: <https://medium.com/antaeus-ar/exploring-metas-building-blocks-d99fdd1b3ec3>