Augmented Reality (AR) & Virtual Reality Project Report (2240721)

Submitted for the partial fulfilment of the degree of

Bachelor of Technology

In

Artificial Intelligence & Robotics

Submitted By

Almas Babar (0901AD221005)

Kshitij Jain (0901AD221046)

Vinay Pratap Singh (0901AD221074)

UNDER THE SUPERVISION AND GUIDANCE OF

Dr. Vibha Tiwari

(ASSISTANT PROFESSOR)

Centre for Artificial Intelligence

माधव प्रौद्योगिकी एवं विज्ञान संस्थान, ग्वालियर (म.प्र.), भारत

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR (M.P.), INDIA

Deemed University

(Declared under Distinct Category by Ministry of Education, Government of India)

NAAC ACCREDITED WITH A++ GRADE

July- Dec 2025

CERTIFICATE

This is to certify that the work contained in these projects have been carried out by students mentioned below from the Centre for Artificial Intelligence. These Projects were done on partial fulfillment of B.Tech laboratory **"Augmented Reality(AR) & Virtual Reality(VR)".** It has been found to be satisfactory and hereby approved for submission.

Name of Students	Enrollment
Almas Babar	0901AD221005
Kshitij Jain	0901AD221046
Vinay Pratap Singh	0901AD221074

Guided By:

Dr. Vibha Tiwari
Dr. Sunil Kumar Shukla
[ASSISTANT PROFESSOR]
Centre for Artificial Intelligence
MITS Deemed University, Gwalior

Approved by HoD

Dr. R. R. Singh
Associate Professor and Head
Centre for Artificial Intelligence

ACKNOWLEDGEMENT

We would like to express our greatest appreciation to all the individuals who have helped and supported us throughout this lab Project Report. We are thankful to the whole Centre for Artificial Intelligence department for their ongoing support during the experiments, from initial advice and provision of contact in the first stages through ongoing advice and encouragement, which led to the final report of this **Augmented Reality(AR) & Virtual Reality(VR)** Report.

A special acknowledgement goes to our colleagues who helped us in completing the file and by exchanging interesting ideas to deal with problems and sharing the experience.

We wish to thank our professor **Dr. Vibha Tiwari and Dr. Sunil Kumar Shukla** as well for her undivided support and interests.

Almas Babar

Kshitij Jain

Vinay Pratap Singh

DECLARATION BY THE CANDIDATES

Our team AAKASH KUSHWAH, BHUMI SHIVHARE AND SHRUTI MISHRA declare that the PROJECT REPORTS for the course **Augmented Reality (AR)** entitled **"AR Smart Textbook"** & **Virtual Reality(VR)** entitled **"VR Shooting Game"** are being submitted in the partial fulfillment of the requirement for the award of Bachelor of Technology in Artificial Intelligence and Robotics. All the information in this document has been obtained and presented in accordance with academic rule and ethical conduct. During the session July-Dec 2025.

Almas Babar (0901AD221005)
Kshitij Jain (0901AD221046)
Vinay Pratap Singh (0901AD221074)

Date: 13th November, 2025
Place: Gwalior

Augmented Reality (AR) Report

Description

This project focuses on developing **Augmented Reality (AR) flashcards** to make language learning more interactive, engaging, and effective. The system uses AR technology to display **3D animations and text** when a learner scans a printed flashcard using a smartphone or tablet camera.

The **Virtual Zoo Tour** project is an **Augmented Reality (AR)-based educational experience** that allows users to explore a zoo environment virtually. The project integrates **3D animal models**—such as the **elephant, tiger, and horse**—into a real-world setting through a mobile device's camera.

When the AR application is launched, users can scan a marker or image target, and the selected animal appears as a lifelike 3D model on the screen. Each animal is placed on the Unity Zoo Map, which represents different sections of a zoo. The user can move their camera around to view the animals from different angles, zoom in for details, and observe their movements and behaviors.

The main goal of this project is to create an interactive and immersive learning environment where students and visitors can explore and understand wildlife without needing to visit an actual zoo. The AR-based zoo provides an exciting, safe, and educational experience suitable for children, schools, and learning institutions.

By merging real-world visuals with virtual 3D content, this project demonstrates the power of AR technology in education, tourism, and entertainment.

Objectives

The main aim of the **AR Virtual Zoo Tour** project is to create an **interactive Augmented Reality (AR)** educational application that allows users to explore and learn about dinosaurs in an immersive and engaging way. The project combines technology, education, and creativity to enhance the learning experience through visualization and interactivity.

Specific Objectives:

- **To create an immersive virtual zoo experience** that allows users to interact with 3D animal models through AR.
- **To provide an educational and engaging platform** for learning about animals, their appearance, and habitats.
- **To utilize Unity and AR SDKs** (like Vuforia or AR Foundation) for rendering realistic and interactive 3D animal animations.
- **To demonstrate the potential of AR** in transforming traditional educational tools into interactive digital experiences.
- **To offer an accessible solution** for virtual field trips where physical visits to zoos are not possible.
- **To design a user-friendly interface** that seamlessly blends real-world and digital environments.

Development Tools

The **AR Virtual Zoo Tour** project integrates multiple software tools and platforms to design, develop, and deploy an interactive Augmented Reality (AR) educational application. Each tool plays a crucial role in creating a seamless and immersive learning experience.

- **Unity 3D:**
  Used as the main development platform for creating the AR environment, designing the zoo map, and placing 3D models of animals.
- **Vuforia Engine / AR Foundation:**
  The AR SDK integrated into Unity to enable image tracking, plane detection, and display of 3D models in real-world views.
- **Blender / Sketchfab (3D Assets):**
  Used for creating or importing 3D models of animals like elephant, tiger, and horse, along with basic animations.
- **Android Studio (optional):**
  Used for building and testing the final APK file for Android mobile devices.
- **Hardware Requirements:**
  - Android smartphone with AR support
  - Webcam (for PC testing)
  - Sufficient RAM and GPU for 3D rendering

The development of **AR Virtual Zoo Tour** relies on **Unity + Vuforia** as the core AR framework, supported by **Sketchfab, Blender, and Visual Studio** for 3D content creation and interactivity. Supporting tools like **Photoshop** and **Audacity** enhance the visual and audio aspects, while **Android Studio/Xcode** manage deployment and testing.

Working Flow of the Project

The **AR Virtual Zoo Tour** application was developed using **Unity 3D** and **Vuforia Engine**, following a structured and step-by-step workflow to ensure smooth development, interactivity, and performance. The working flow describes how the project was designed, implemented, and tested from start to finish.

1. Project Planning and Setup

- The development began by defining the scope and features of the project, including the number of dinosaur species, types of AR interactions (animations, information overlays), and educational content.
- Installed Unity Hub and created a new 3D project using the latest LTS version of Unity.
- In Project Settings, enabled Vuforia Augmented Reality Support to integrate AR functionalities into the Unity environment.

2. Vuforia Configuration

- Registered on the Vuforia Developer Portal to generate a **license key** for the project.
- Imported the Vuforia Engine package into Unity via the Package Manager.
- Configured the ARCamera prefab in the scene by replacing Unity's default camera to handle marker tracking and rendering.

3. Importing 3D Models

- Collected high-quality Animals 3D models from Sketchfab in .fbx or .obj format.
- Imported all 3D assets into Unity's **Assets** folder and organized them by dinosaur name or category.
- Created a layout map of the zoo environment in Unity (elephant, tiger, and horse zones).

4. Creating Image Targets

- Used Vuforia Target Manager to upload images that serve as markers or flashcards (for example, pictures of dinosaurs or educational cards).
- Downloaded the generated target database and imported it into Unity.

- Placed each animal model in its respective zone (e.g., tiger in jungle area, elephant near the pond, horse in open field).

5. AR Implementation:

- Configured the Vuforia AR Camera in Unity.
- Set up image targets or ground planes so that when the user scans them, the animals appear in 3D.
- Linked each target to the respective 3D animal prefab

5. Testing and Optimization

- Tested the project on AR-supported mobile devices to ensure proper scaling, alignment, and tracking.
- Exported the final project as an Android APK for installation and demonstration.

6. Build and Deployment

- Configured **Player Settings** (package name, orientation, and icon) for Android platform.
- Built and deployed the APK file to mobile devices for final testing.
- Verified that all markers and Aimal function correctly in real-time AR mode.

Conclusion

The **Virtual Zoo Tour** project is a remarkable demonstration of how **Augmented Reality (AR)** can revolutionize the way we experience and learn about the world around us. By blending the **virtual and real environments**, this project offers users an engaging and interactive way to explore wildlife from the comfort of their homes or classrooms. The integration of 3D animal models such as the **elephant, tiger, and horse**, combined with realistic animations and the immersive Unity zoo environment, provides an experience that is both **educational and entertaining**.

This project successfully shows how **AR** technology can bridge the gap between digital innovation and experiential learning. Instead of simply reading about animals or viewing static pictures, users can now visualize them in three dimensions, observe their size and movement, and feel a sense of presence as if visiting a real zoo. Such an approach not only enhances visual understanding and curiosity among learners but also fosters interactive engagement and knowledge retention. The Virtual Zoo Tour can serve as a foundation for smart learning tools, virtual field trips, and immersive educational applications in schools, museums, and wildlife organizations. It also opens opportunities in eco-tourism, wildlife awareness campaigns, and digital exhibitions.

From a technological perspective, the project showcases the synergy between 3D modeling, real-time rendering, and AR integration, proving how these components can work together to create impactful digital experiences. The simplicity of interaction—just scanning and exploring—makes the application user-friendly and accessible to learners of all ages.

In conclusion, the **Virtual Zoo Tour** project not only fulfills its objective of offering a realistic zoo experience through AR but also lays the groundwork for future advancements in immersive education. It encourages creative thinking, supports sustainable learning by reducing the need for physical infrastructure, and demonstrates how technology can be used responsibly to **educate, engage, and inspire**.

Target Image

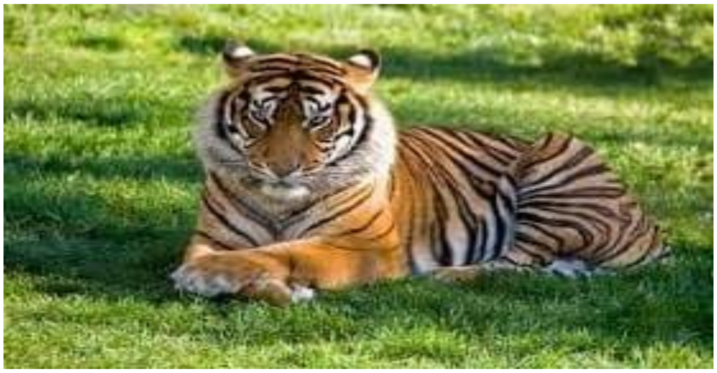


Fig: 01

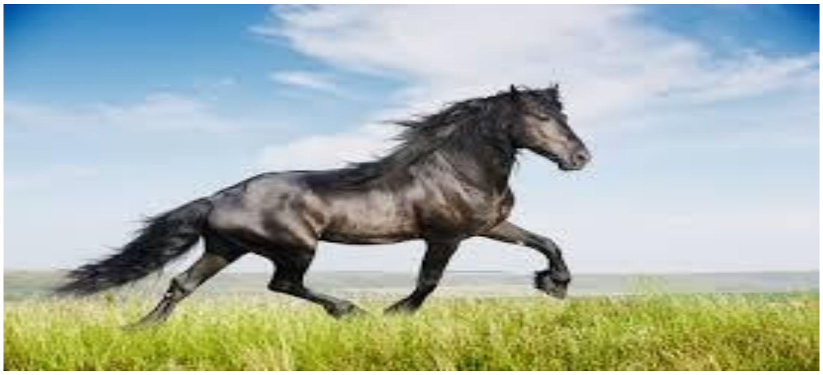


Fig: 02

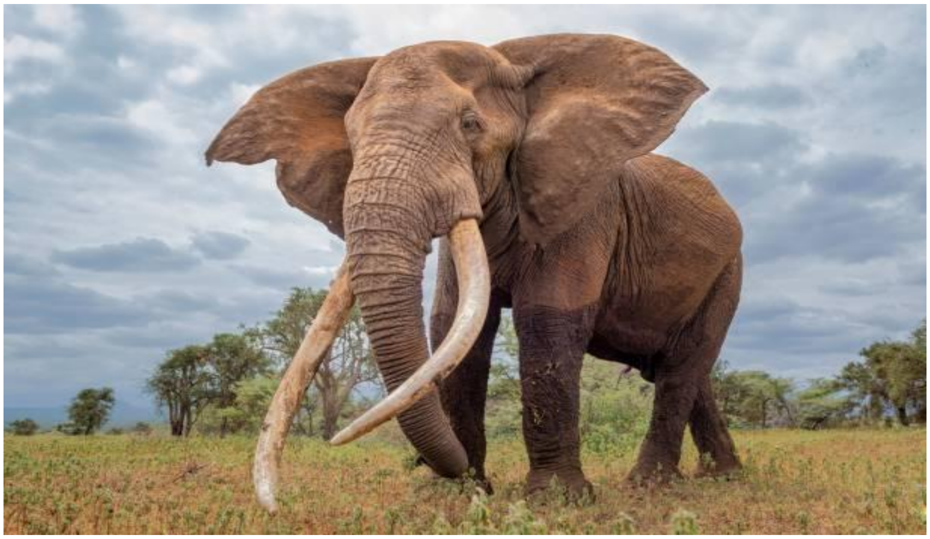


Fig: 03

Visual Output

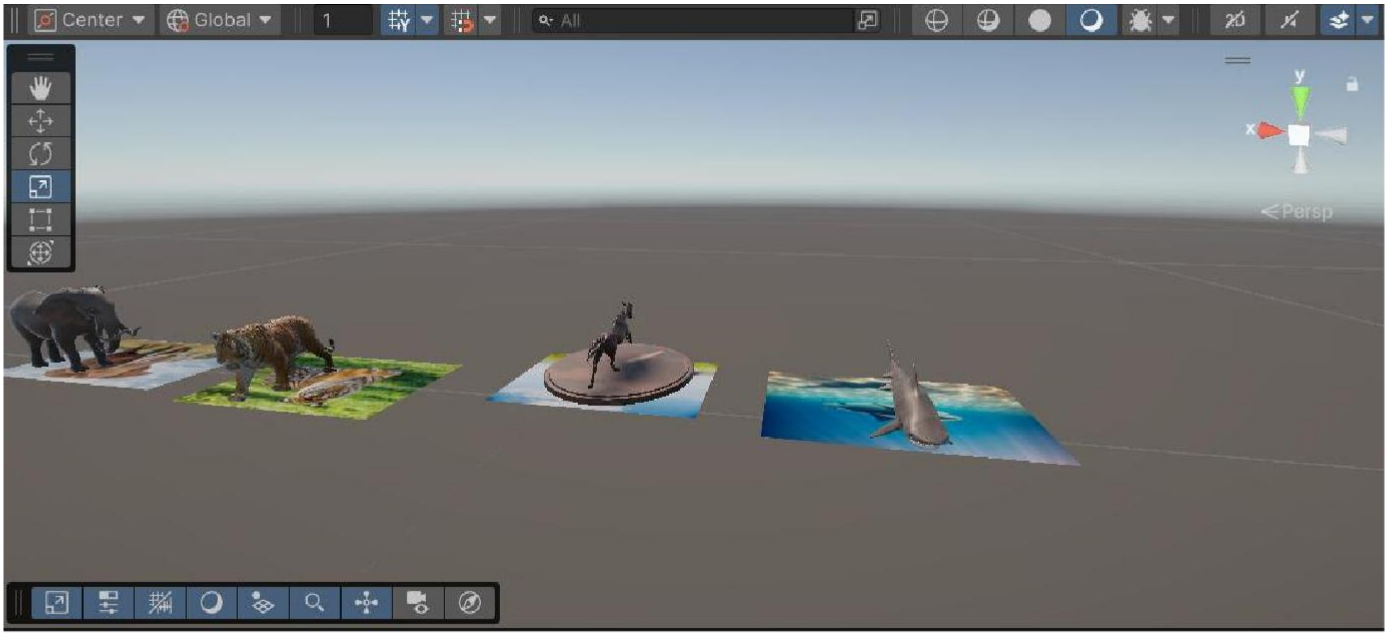


Fig: 04

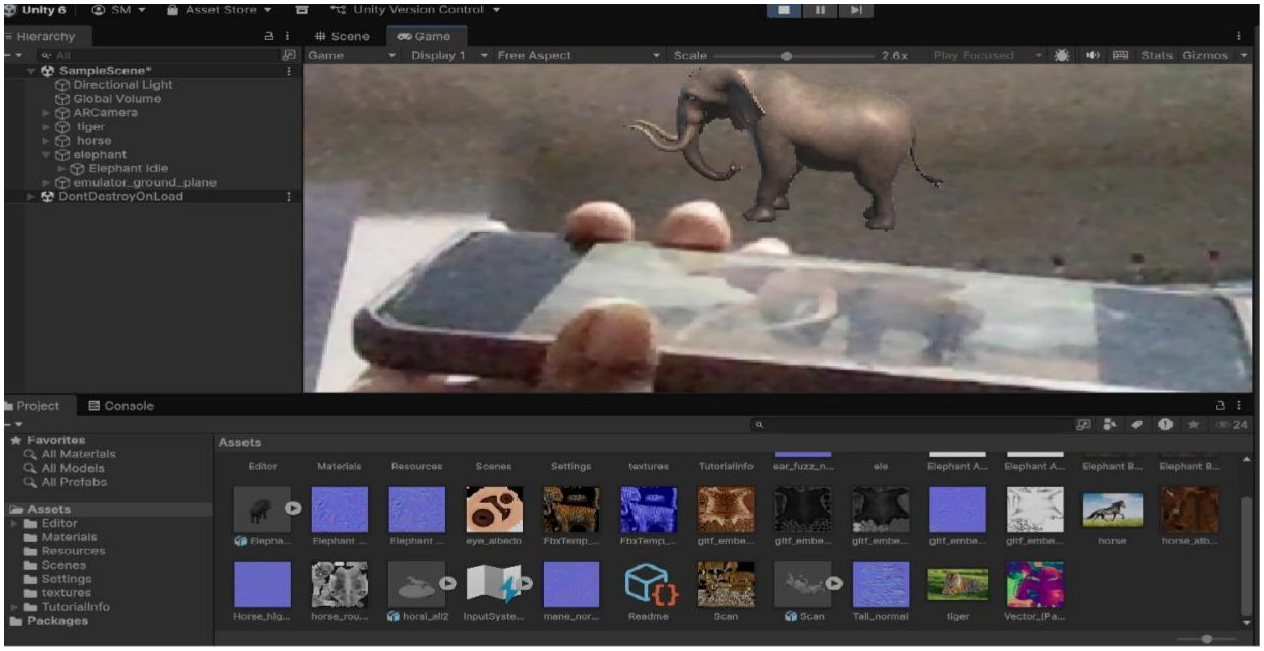


Fig: 05

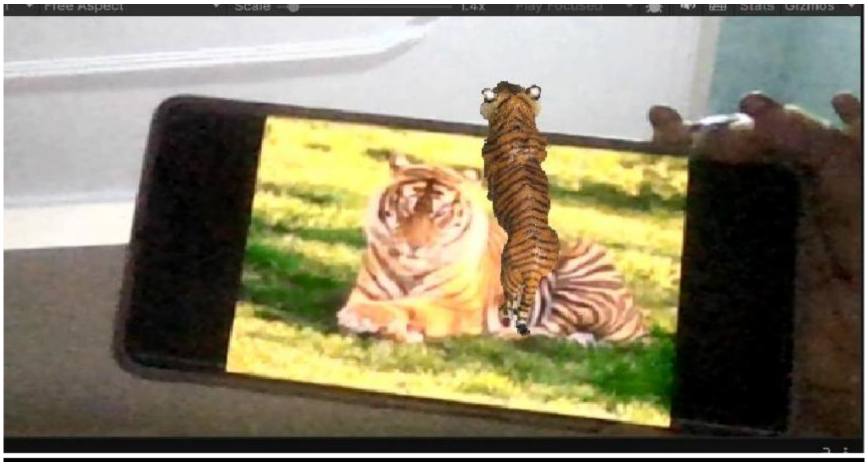


Fig: 06

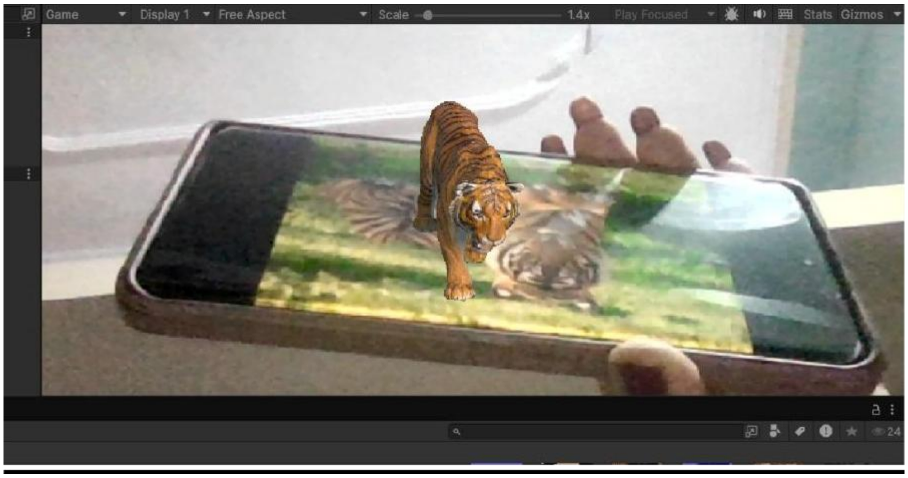


Fig: 07

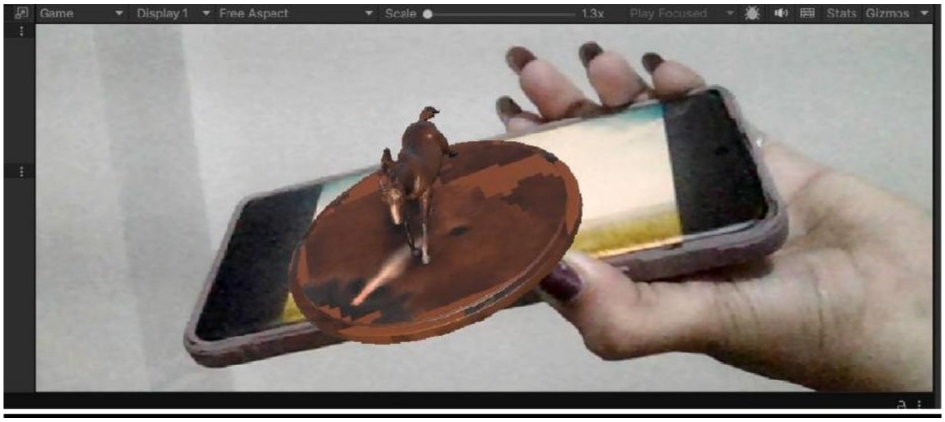


Fig: 08

Virtual Reality (VR) Report

Description

This project focuses on developing the **VR Cube Shooter** project is an immersive and interactive **Virtual Reality (VR) game** designed to demonstrate the core principles of VR development — including real-time tracking, physics-based interactions, precise input handling, collision detection, and dynamic scoring systems. In this game, the player is placed within a fully interactive 3D virtual environment, standing or sitting on a flat plane filled with floating or scattered cubes. The player is equipped with VR controllers mapped as guns, which can fire virtual bullets or raycasts toward the cubes. When a projectile successfully hits a cube, the cube is destroyed with visual and audio effects, and the player's score increases accordingly.

This project emphasizes a highly responsive and intuitive gameplay experience**.** It integrates haptic feedback (controller vibration), explosive particle effects, and real-time sound feedback to enhance immersion. A dynamic scoring system continuously updates in the player's field of view, creating a satisfying feedback loop of aim → shoot → destroy → score.

The game also demonstrates the use of **object pooling** to optimize performance, ensuring smooth frame rates even when multiple projectiles and cubes are rendered simultaneously. By utilizing **Unity's XR Interaction Toolkit** (or similar SDK), the project ensures accurate hand tracking, realistic controller physics, and comfortable interaction design to minimize motion sickness and enhance playability.

Beyond entertainment, **VR Cube Shooter** serves as a **technical learning project** that combines the foundational elements of VR programming, 3D spatial awareness, and user-centered interface design.

Overall, **VR Cube Shooter** delivers both an educational and entertaining experience while providing hands-on exposure to the immersive potential of virtual reality technology.

Objectives

The main objective of the VR Shooting Game is to design and develop an immersive virtual reality environment where players can interact naturally using VR controllers to aim and shoot targets. The project focuses on creating an engaging gameplay experience where each trigger pull destroys cubes placed on a plane, and players earn points for every cube destroyed. It aims to combine entertainment with hands-on learning in VR development—covering concepts like input handling, object physics, hit detection, particle effects, and real-time scoring. Additionally, the project seeks to demonstrate the potential of VR technology in developing responsive, interactive, and performance-optimized 3D applications that provide both fun and technical insight into modern immersive systems.

Specific Objectives:

- **Implement core VR interaction:** map VR controller inputs (trigger/button) to a firearm that can fire projectiles or raycasts and respond to hits.

- **Provide reliable hit detection:** destroy cube targets upon impact using physics collisions or precise raycasting.

- **Maintain high performance:** use object pooling for bullets/projectiles and optimize cube geometry/textures so the game runs smoothly on target VR hardware.

- **Create immediate feedback:** add particle effects, sounds, and haptics to reinforce each successful hit.

- **Implement a scoring and UI system:** show a live score HUD that updates on each cube destruction and persists across sessions if desired.

- **Follow modular code structure:** write reusable, well-documented C# scripts for shooting, scoring, spawn management, and UI to support future expansion.

- **Test across VR platforms:** verify input and rendering across at least one headset (e.g., Meta Quest or SteamVR) and iterate based on playtests.

Development Tools

The **VR Shooting Game** project integrates multiple software tools and platforms to design, develop, and deploy an interactive Augmented Reality (AR) educational application. Each tool plays a crucial role in creating a seamless and immersive learning experience.

- **Unity 3D (LTS recommended)** — primary game engine for scene creation, physics, scripting, and building the VR app.
- **XR Plugin / OpenXR** — unified runtime for VR input and tracking. Use Unity's XR Management + OpenXR plugin for cross-platform compatibility.
- **XR Interaction Toolkit (Unity)** — ease building VR interactions (grab, teleport, direct/teleport interactions) and controller bindings.
- **Oculus Integration / SteamVR (optional)** — platform-specific SDKs if you target Oculus/Meta or SteamVR features.
- **Visual Studio / Visual Studio Code** — C# development, debugging, and editing scripts.
- **Blender** — model creation/optimization for custom guns, cubes, or environment props (optional).
- **Audacity / any audio editor** — create/trim sound effects (gunshot, cube explosion, UI sounds).
- **Photoshop / GIMP** — design UI textures and HUD graphics.
- **Unity Profiler** — performance profiling for CPU/GPU/memory and to find bottlenecks.
- **Git / GitHub** — version control for collaborative development and backing up iterations.
- **Android SDK / ADB (for Quest builds)** or **PC build tools (for SteamVR)** — platform-specific build & deployment utilities.
- **Controller hardware:** development headset (e.g., Meta Quest 2) and tracked controllers for user testing.

Working Flow of the Project

1. Project initialization

- Created a new Unity 3D project using the latest LTS release.
- Installed XR Management, OpenXR plugin, and XR Interaction Toolkit via Unity Package Manager.
- Configured XR settings (enabled OpenXR runtime, set the target platform, and validated controller profiles).

2. Scene setup and player rig

- Added an XR Origin/XR Rig prefab to the scene to represent the player's head and controller transforms.
- Positioned a flat plane to serve as the arena floor. Set proper collision layers and physics material (slightly bouncy or default) depending on desired cube physics.
- Created a static environment boundary (invisible colliders) to prevent objects from drifting out of play area.

3. Gun / Input mapping

- Designed a simple gun prefab (3D model or primitive shapes) parented to the right-hand controller transform.
- Mapped controller trigger input to firing logic using the XR Interaction Toolkit action mappings (or via direct OpenXR input actions).
- Chose firing method:
    - **Raycast-based**: Fire a ray from the gun muzzle; instant hit detection (no projectile physics). Simpler, precise, minimal cost.
    - **Projectile-based**: Spawn bullet prefab with forward velocity and let physics handle collisions (more visual and satisfying but costlier).
- Implemented a muzzle flash particle effect and gunshot audio triggered on fire input.

- Added haptic feedback on trigger pull and on hit (via XR controller haptics API).

4. Cube targets

- Created a Cube prefab with:
  - Collider and Rigidbody (if using physics-based interaction).
  - Health or hit flag (for single-hit destruction set to 1).
  - Destruction visual (particle explosion) and sound.
- Placed cubes across the plane manually for a simple level, and implemented a procedural spawner to populate cubes at random positions for replayability.
- Implemented object pooling for cubes (and projectiles if used) to reduce runtime allocations and GC spikes.

5. Hit detection and destruction

- **Raycast method:** On firing, cast a ray; if it hits a cube tagged Target, call Target.Destroy() to trigger VFX/SFX and update score.
- **Projectile method:** Bullet prefab checks collisions in OnCollisionEnter(Collider other) — on hitting a cube, call the cube's destroy routine and recycle the projectile.
- Added a small delay before removing the cube to play explosion effects and optionally spawn debris.

6. Scoring & UI

- Implemented a ScoreManager singleton to maintain player score and expose an API AddScore(int points).
- Built a world-space UI canvas attached near the player (or a HUD anchored to the camera) showing current score, best score, and time left (if timed mode).
- Animated the score increment (scale-up or pop animation) to emphasize each point.

7. Game states & UX

- Created simple game state manager: Menu, Playing, Paused, GameOver.
- Implemented start menu where player can choose difficulty (cube count, spawn rate), reset score, or change firing mode.
- Added clear start/stop controls and a respawn/reset mechanism for quick testing.

8. Build & deployment

- Configured player settings (package identifier, VR support enabled, graphics API).
- Built APK/APP and installed on headset (or ran via Unity Editor with headset connected) for final playtests.
- Collected feedback and adjusted difficulty and UI readability.

Conclusion

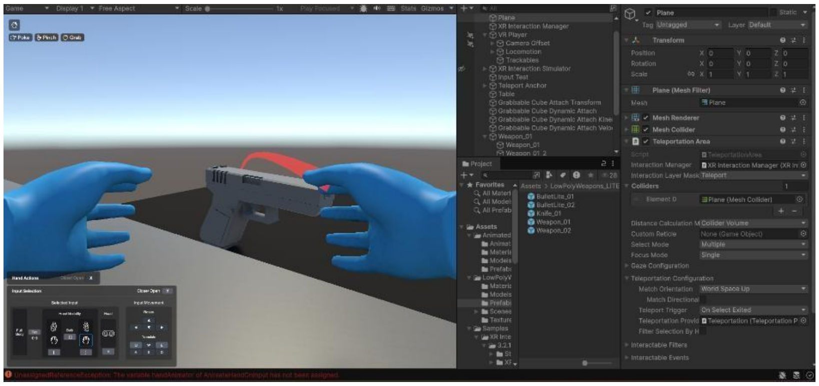
The VR Cube Shooter project successfully demonstrates the application of Virtual Reality (VR) in creating an immersive, interactive, and skill-based gaming experience. By combining real-time input tracking, dynamic physics, and responsive visual feedback, the project showcases how VR can transform traditional gaming into a deeply engaging and realistic environment.

Throughout the development process, the project emphasized hands-on interaction — allowing players to aim, shoot, and destroy cubes in a fully three-dimensional virtual space. Every cube destroyed provides instant visual and auditory feedback, reinforcing the player's sense of accomplishment and engagement. The integration of controller-based haptic feedback further enhances the realism by simulating the tactile sensation of firing a weapon, while the scoring system adds a competitive and rewarding layer to gameplay.

From a learning perspective, this project serves as an excellent practical implementation of core VR development principles, including spatial awareness, interaction design, and user comfort optimization. It highlights the importance of frame rate stability, intuitive controls, and realistic physics in providing an enjoyable user experience. The VR Cube Shooter not only demonstrates how VR can be used for entertainment but also reveals its potential in training simulations, aim accuracy improvement, and hand-eye coordination exercises. It lays the groundwork for future enhancements, such as adding multiplayer functionality, advanced AI targets, leaderboards, or realistic weapon physics.

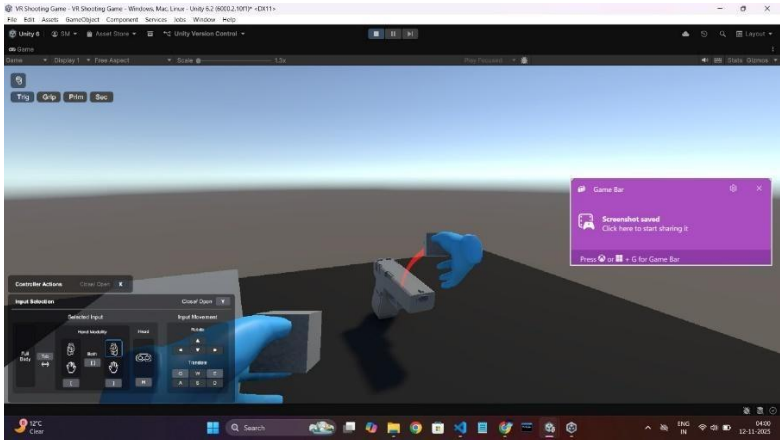
In conclusion, the project achieves its objective of creating a simple yet powerful VR experience that merges technology, creativity, and interactivity. It stands as a testament to how virtual reality can redefine player immersion, offering endless possibilities for both gaming and educational applications in the future.

Visual Output



UnassignedReferenceException: The variable handAnimator of AnimateHandOnInput has not been assigned.

Fig: 01



VR Shooting Game - VR Shooting Game - Windows, Mac, Linux - Unity 6.2 (6000.2.10f1)* <DX11>

Screenshot saved
Click here to start sharing it

Fig: 02

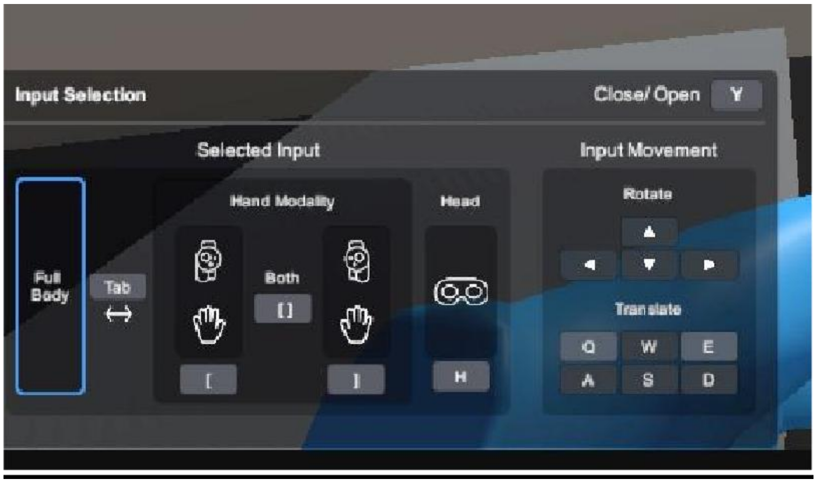


Fig: 03