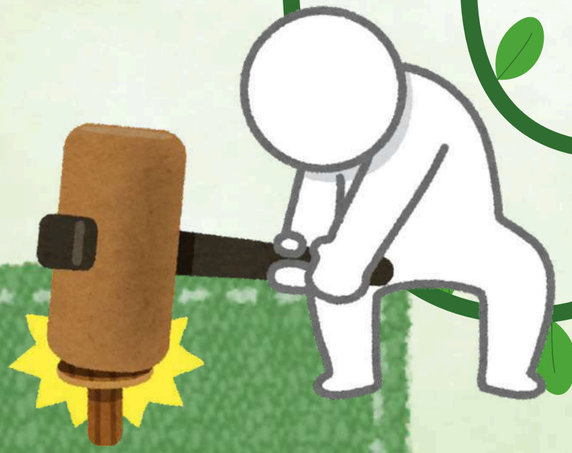
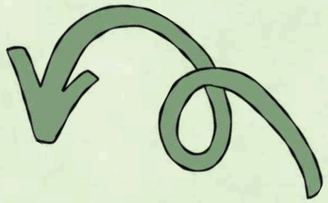


Extended Reality in Industry 4.0

MOLE—XR—MAYHEM

An XR Whack-a-Mole Game for AR & VR

PROJECT VISION



To transform the classic 'Whack-a-Mole' arcade game into a highly immersive and interactive Extended Reality (XR) experience that seamlessly bridges the physical and virtual worlds.

THE PROBLEM & OPPORTUNITY

- The Problem: Classic arcade games are nostalgic but lack modern immersion. Mobile AR can feel gimmicky, and VR can be isolating.
- The Opportunity: New XR technology allows us to create flexible, fun, "pick-up-and-play" games that adapt to the user's device.
- The Goal: Create one game that demonstrates a mastery of both AR and VR, showing a clear development path from one to the other.

OUR SOLUTION: A DUAL-MODE GAME

- A single Unity application that can be built for two different platforms, reusing the same core logic and assets.
  - 1. AR Mode (Mobile): Uses a phone's camera to bring the game to life on any flat surface.
  - 2. VR/MR Mode (Headset): A fully immersive 3D world, or a Mixed Reality (MR) experience that places moles in your real room.

FEATURE DEEP—DIVE: VR/MR MODE

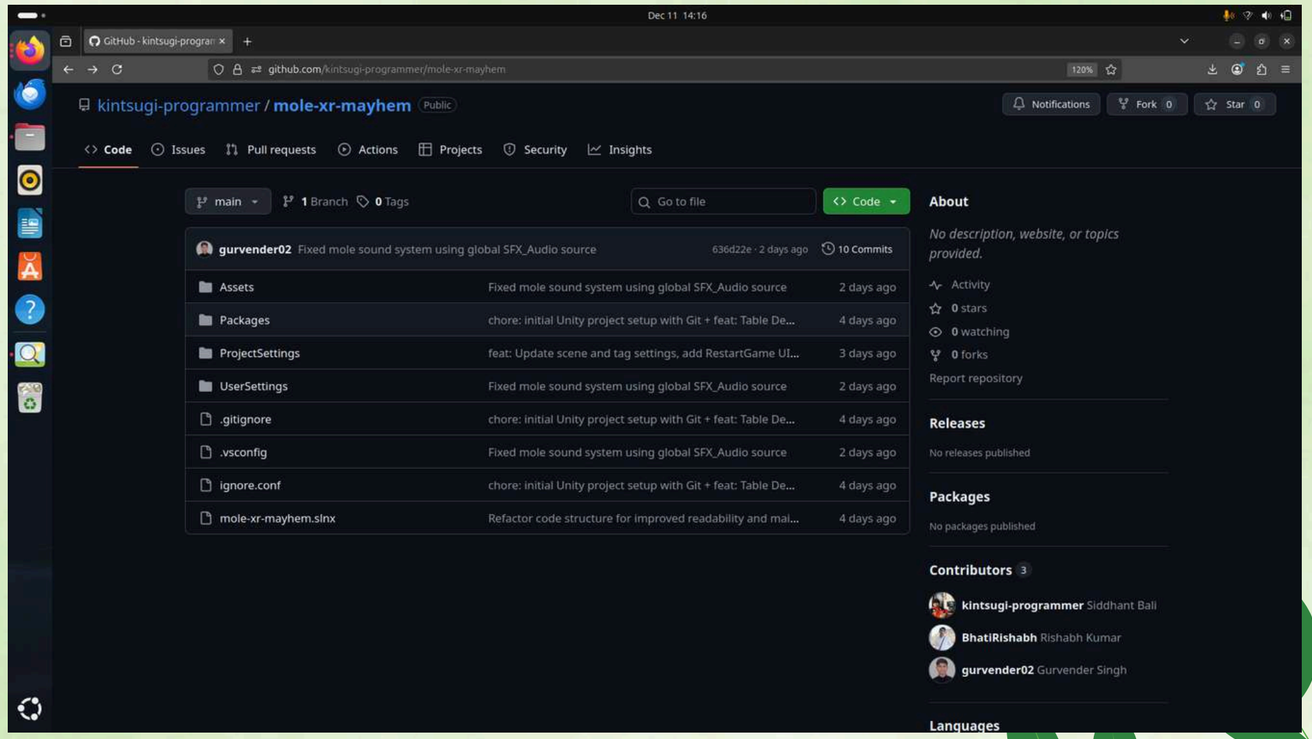
- Platform: Meta Quest 2/3, PC VR
- Tracking: Full 6-DOF (Room Scale). The player can move around the game.
- Interaction: Motion-Controlled Whacking. The player uses the VR controllers as virtual hammers.

- Experience:
- VR: A fully immersive 3D environment (e.g., a cartoon garden).
- MR: Uses the headset's passthrough cameras to place moles on the actual floor and tables in the player's room.

CORE TECHNOLOGY STACK

This project will be built on a modern, cross-platform framework.

- Game Engine: Unity 3D (2022 LTS or newer)
- XR Framework: OpenXR (The industry standard for VR/MR compatibility)
- Interaction: Unity's XR Interaction Toolkit (XRI) (Manages all VR/MR controller inputs, gestures, and raycasting)
- AR/MR Layer: Unity's AR Foundation (Manages passthrough cameras, plane detection, and image tracking)
- Code: C# in Visual Studio

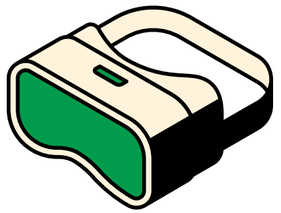


Git Version Control

https://github.com/kintsugi-programmer/mole-xr-mayhem

proposed :

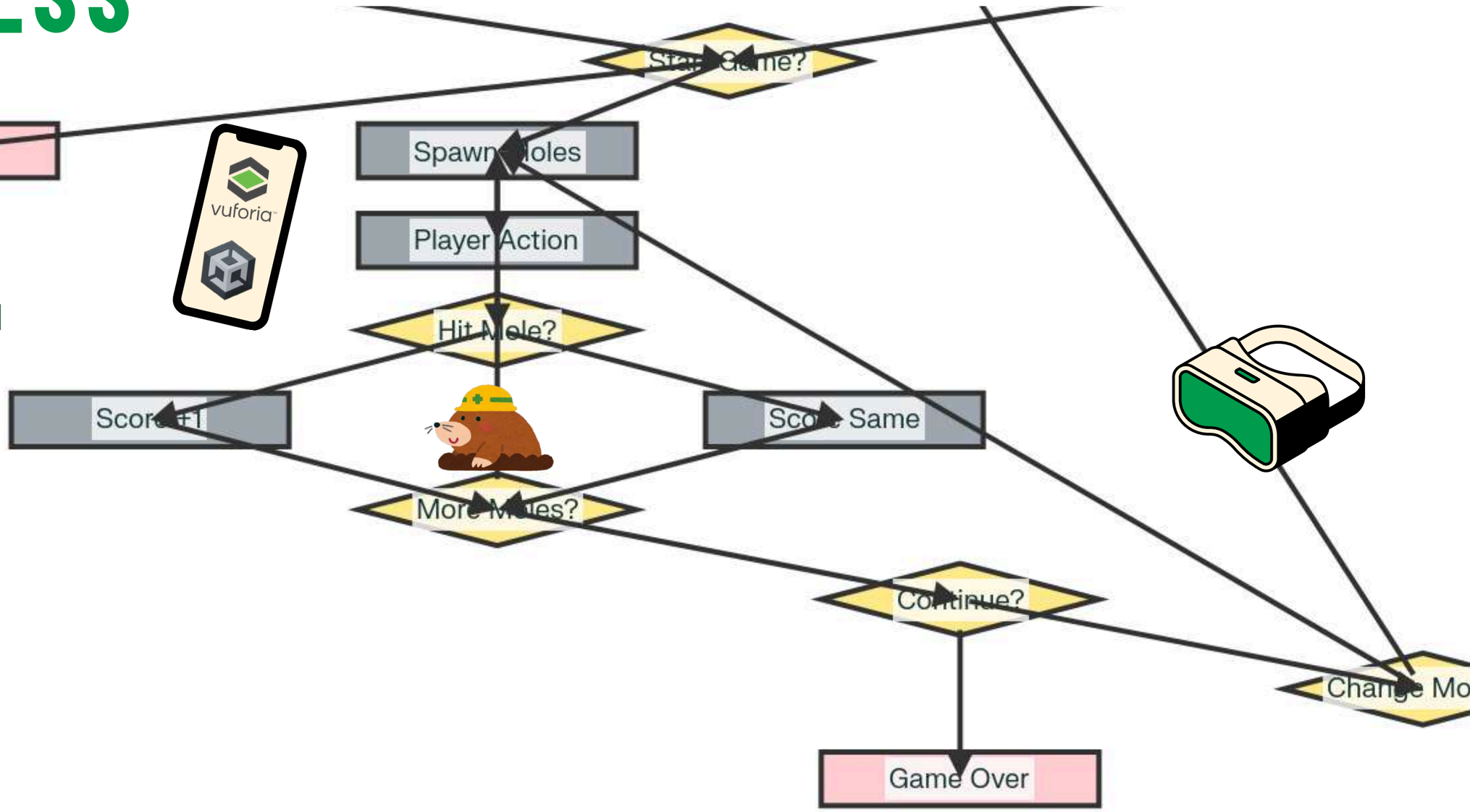
PRODUCT WORKFLOW



final :

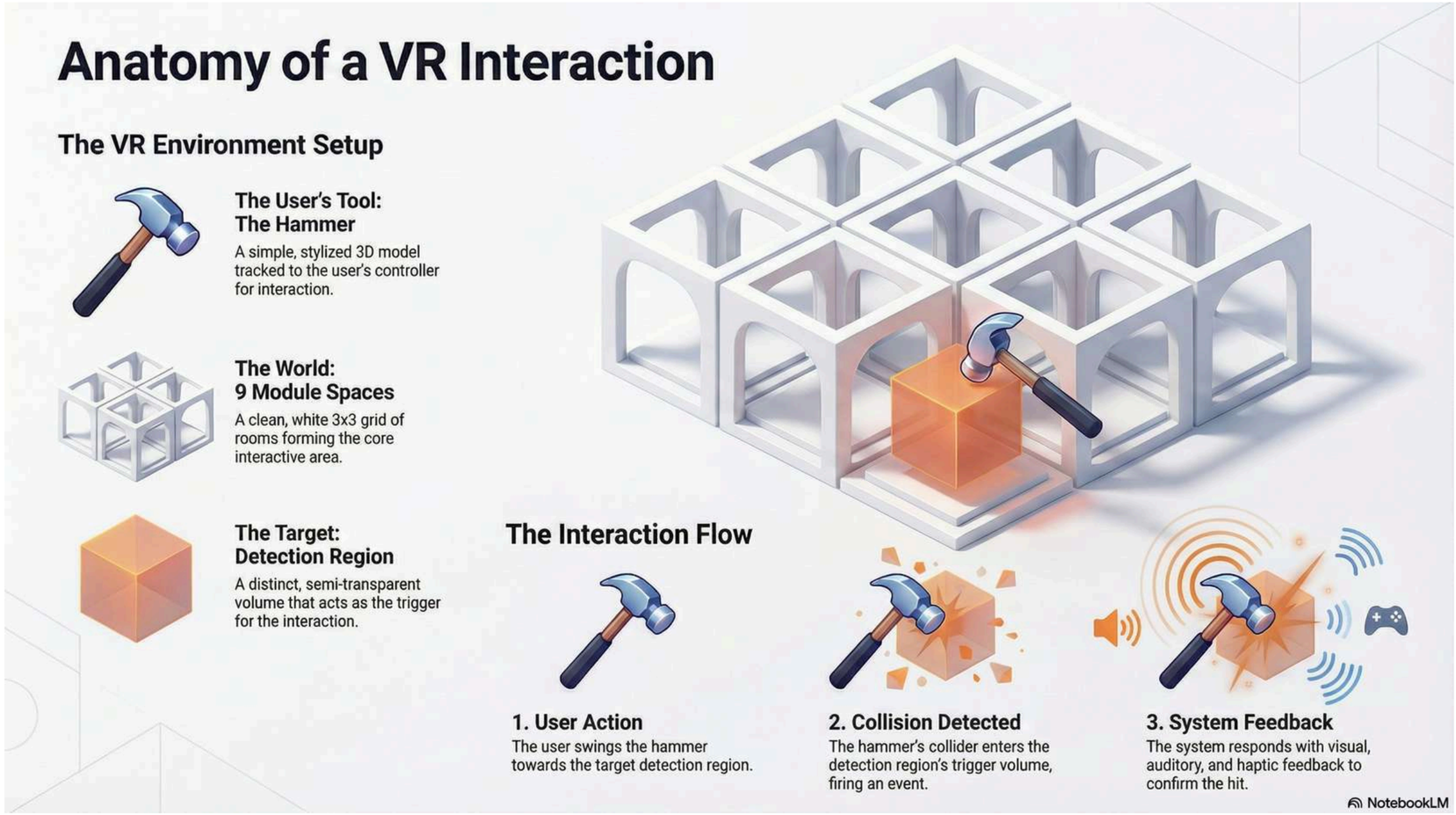
PROGRESS DONE

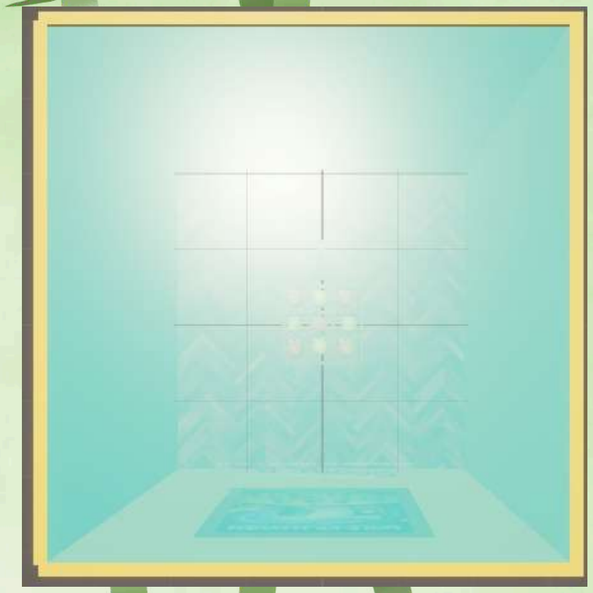
GAME MECHANICS OF INTERACTION B/W USER INPUT & VIRTUAL OBJECTS



Game Mechanics of Interaction Between User Input & Virtual Objects

- User input is captured and converted into an interaction event.
- A hit volume is generated to check for collisions in the scene.
- Virtual objects expose collision bodies that define their interactive regions.
- If the interaction vector intersects a collision body, the object triggers its Hit state.
- Missed or no-collision events trigger default or Escape behaviour.
- Game feedback and difficulty progression depend entirely on these collision outcomes.



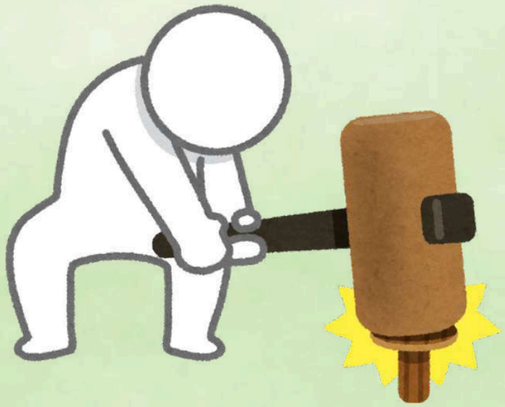


- Collision-based hit detection triggers instant mole response.
- Real-time score updates occur the moment a hit is registered.
- Hit events play synchronized sound effects for impact feedback.
- Interaction is responsive, with immediate animations and state changes.
- Gameplay adapts to free room space, ensuring unobstructed movement.
- Overall experience remains fast, reactive, and tightly looped around user actions.

TARGET AUDIENCE & PLATFORMS

- Primary Platform (VR/MR):
  - Users: Meta Quest 2/3 owners looking for fun, active, arcade-style games.
  - Device: Standalone VR Headsets.

TEAM MEMBERS



Siddhant Bali
2022496

Rishabh Kumar
2022402

Gurvender Singh
2022192

THANK YOU