# Music Cube VR: A Music Interaction Game

Essie Cao
Institute of Computational and Mathematical Engineering
Stanford Univeristy
essiecao@stanford.edu

## 1. Introduction

Music Cube VR is a virtual reality rhythm game designed to investigate how different input modalities affect user immersion. Traditional rhythm games use button-based inputs and flat screens, lacking the the ability to provide immersive experience. Even in VR, most rhythm games rely on controllers, missing the opportunity to explore natural hand gestures. This project implements a dual interaction system where players slice cubes using either VR controllers or hand gestures. I conducted an user study with 10 users, which showed that most participants found the controller mode more immersive.

## 2. Related Work

Rhythm-based games such as Dance Dance Revolution (DDR) and Guitar Hero have traditionally relied on button inputs and screen-based visuals to engage players. While effective in gameplay, these methods offer limited immersion due to their reliance on discrete inputs and two-dimensional interfaces [1].

Beat Saber introduced a VR-native rhythm experience using motion-controlled swords, which involes full-body engagement in music-driven interactions. However, many VR rhythm games, including Beat Saber, depend only on hand-held controllers.

Prior research on VR interaction has explored the role of hand tracking in enhancing presence and reducing user fatigue [3] [4]. Studies suggest that gesture-based interaction can lead to higher user satisfaction, a stronger sense of embodiment, and improved physical comfort over extended use [2].

## 3. Methods

To explore how different input modalities influence immersion in virtual rhythm games, I developed Music Cube VR, a Unity-based game that enables users to slice incoming musical cubes either using VR controllers or hand gestures. The game was built using Unity's XR Interaction Toolkit.

### 3.1. Game Mechanics

The core mechanic involves slicing flying cubes in rhythm. Every few seconds, a cube is instantiated at a random point, rotated by a multiple of 90°, and moves forward along the Z-axis with synchronized musical beat.

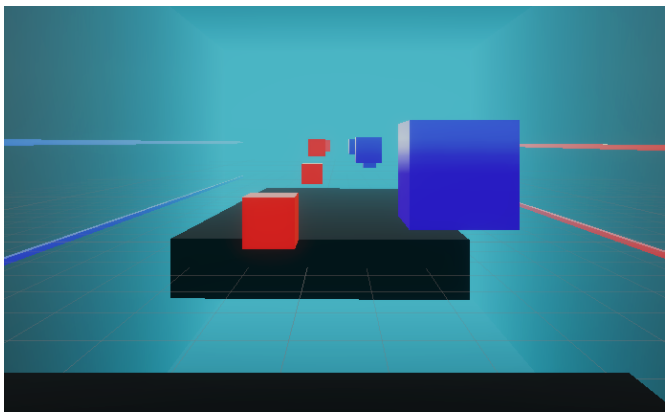

Figure 1. Cubes flying towards the user

### 3.2. Slicing Mechanism

Slicing is handled by the SliceObject script, which uses EzySlice, a real-time mesh slicing library. The slicing plane is defined by two Transform points (startSlicePoint and endSlicePoint), representing the path of the user's slicing motion. A Physics.Linecast is used to detect whether the gesture intersects with a cube:

```
hasHit = Physics.Linecast(start, end, hit, layer)
```

When a hit is detected, the script creates upper and lower hulls of the cube, and destroys the original cube. The Cube will then drop to the ground.

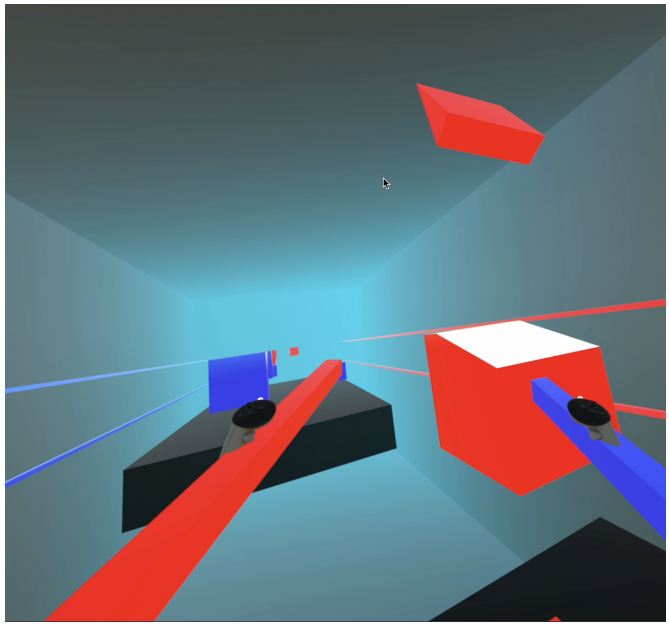

Figure 2. User slices the cube

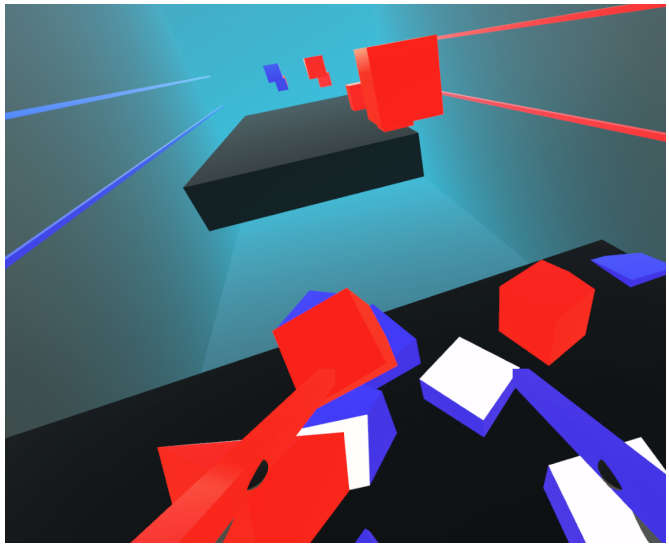

Figure 3. The sliced cubes fall onto the ground

### 3.3. Gesture & Velocity Estimation

To determine slice direction, I integrated a VelocityEstimator component to estimate both linear and angular velocity based on the movement of the slicing object across frames.

### 3.4. Input Modalities

- **Controller Mode:** Use the left and right hand Meta Quest Pro controllers as the swords to slice the cubes.
- **Hand Gesture Mode:** Unity's hand tracking detects finger or palm motion and maps it to slicing gestures. Users perform slicing motions with hands.

## 4. Evaluation and Results

I conducted a user study with 10 participants to evaluate the impact of input modality on user experience in a VR rhythm game. Each participant played the game using both hand gestures and VR controllers, then rated their experience on a scale from 0 to 5 for each mode. Several participants preferred hand gestures, citing enhanced immersion, a more cyberpunk-like experience, and the potential for more expressive and flexible input. They loved the intuitiveness of using bare hands. However, others preferred controllers, noting that they felt more accurate, less likely for tracking errors, and more satisfying for simulating sword-like actions. Some also pointed out that hand gesture recognition was occasionally unreliable, leading to frustration. Overall, controller input received slightly higher average ratings, suggesting better usability in its current implementation.

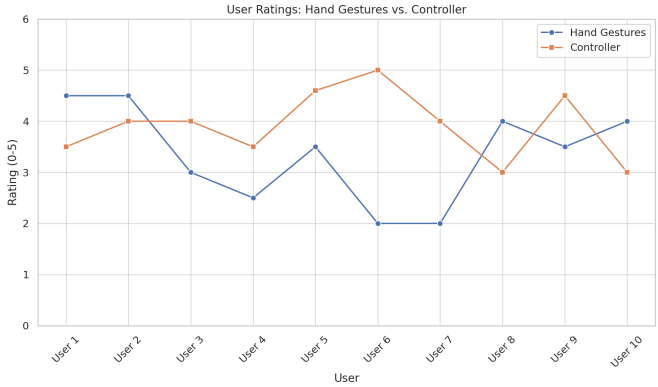


Figure 4. User ratings comparing hand gesture and controller input methods.

## 5. Future Work

I plan to enhance the game by implementing a scoring system that provides performance tracking for players. Additionally, I want to explore a wider range of hand gestures beyond the current slicing motion, evaluating how different types of gestures influence user experience. Finally, I'm interested in extending to other genres of VR games where hand gestures play a more central role in interaction. Since our current rhythm game primarily relies on a simple slicing motion, studying more gesture-intensive games may reveal deeper insights into the strengths and limitations of gesture-based input in VR.

## References


[1] M. Alghofaili, M. Alibrahim, T. Abualait, M. A. Alzahrani, and M. A. Aldaihan. Active video games using virtual reality: A systematic review, 2020. Published in IEEE Access, Vol. 8, pp. 216915–216933. DOI: 10.1109/ACCESS.2020.3040243.

[2] D. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev. 3d user interfaces: Theory and practice, 2004. Published by Addison-Wesley.

[3] E. D'mello and H. Franklin. Designing for locomotion with hand tracking in vr, 2021. CHI 2021 Proceedings. DOI: 10.1145/3411764.3445358.

[4] M. Smith, L. Johnson, and R. Kumar. Touchless input in vr: A usability study, 2020. IEEE VR Workshop on Emerging Technologies, pp. 45–52.