VR Screencheat

Dai-Rong Han
Stanford University
drhan@stanford.edu

Abstract

This project explores an adaptation of the game Screencheat into a single-player virtual reality experience, developed using the Unity game engine. In this version, the player must locate and shoot an invisible enemy by interpreting a fixed camera view from the enemy's perspective, visible in a small window overlay. The game focuses on spatial reasoning and perspective-taking rather than reflex-based shooting. The results highlight the potential for innovative mechanics in VR that subvert traditional shooter paradigms while emphasizing usability considerations for immersive gameplay.

1. Introduction

This project is inspired by Screencheat [4], a first-person shooter multiplayer game where all players are invisible, and victory depends on interpreting each other's perspectives. The game flips the idea of screen cheating, viewing other people's screens, which is usually considered unfair, into a core mechanic. This project brings this concept into VR, emphasizing spatial perception and head-tracking rather than fast-paced shooting. Developed in Unity, the project simplifies gameplay for solo play by using a computer-controlled enemy.

2. Related Work

While many first-person shooters and VR games exist in Unity, few have explored spatial deduction from another player's viewpoint, particularly in VR contexts. Prior work has focused on standard FPS mechanics in Unity [5], FPS VR maze navigation and spatial immersion [2], and comparative studies of movement and view control across VR and PC platforms [1].

3. Game Design

The player is placed in a 3D environment (Figure 1) where a target enemy is invisible. The challenge is to locate the target using a fixed camera view from the target opponent's perspective, which will be displayed in the top left corner of the player's screen. There are two gaming modes:

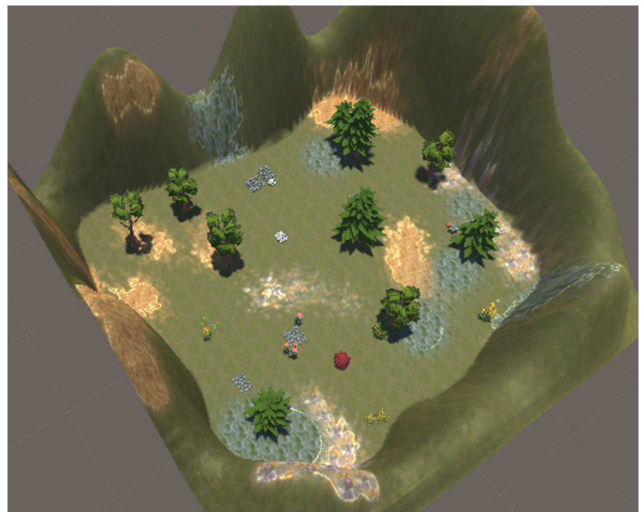

Figure 1. Game environment.

- Easy mode: The enemy stays in a fixed location.
- Hard mode: The enemy moves around randomly.

The game view is shown in Figure 2. A countdown timer of 60 seconds is shown in the top right corner of the player view. A weapon cooldown time is set on the bottom right of the screen for 1 second each time the player fires a bullet to prevent the player from shooting randomly everywhere. If the player fails to shoot the target in the given time, the player fails.

3.1. User Interface

Once the player enters the game, a main menu (Figure 3) shows the different levels of play to enter the game scene and a quit option for users to exit the application. Once the game ends, there is panel indicating the success or failure of the game, and allows the player to return to the main menu (Figure 4 & 5).

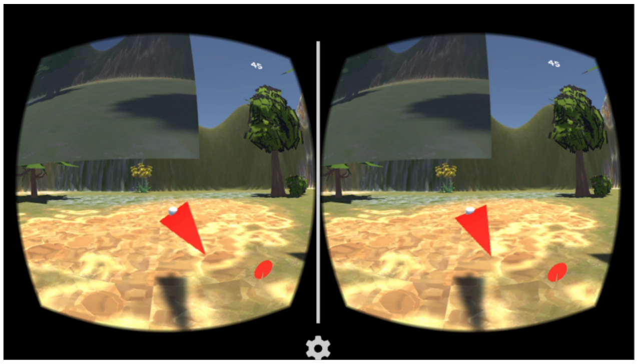

Figure 2. Player view and a fired bullet.

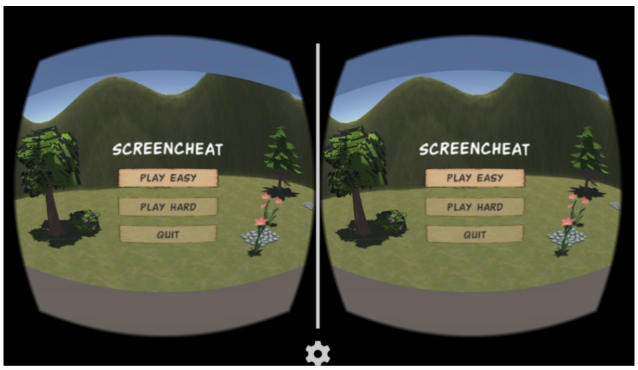

Figure 3. Main menu UI.

Figure 4. Victory message after enemy is hit.

3.2. Game Control

The player must explore the environment using head movement to look around and WASD keys for movement. Rotation is with regard to the orientation of the head. As shown in Figure 2 , once the player aligns the center of their view with the target, they can press spacebar to shoot the target. For UI scenes, players can navigate through UI elements using W and S keys, and the spacebar for selection.

Figure 5. Timeout message.

3.3. Technical Implementation

- Player: The player is created using the Google Cardboard model. Its translation is controlled by keyboard input while the rotation is determined by the headset.
- Enemy: The enemy game object initializes by randomly placing the enemy on the terrain, ensuring it's within bounds and positioned correctly above the ground based on terrain height and collider size. In hard mode, the enemy uses raycasting to detect obstacles and rotates in a new random direction if one is encountered.
- Bullet: The bullet is a physical projectile instantiated at the weapon's tip and propelled forward using a force-based Rigidbody. Collision with the invisible target is detected using Unity's physics system.

4. Results and Discussion

This project successfully replicates the core concepts from Screencheat [4] into a VR 3D environment. The reverse-perspective mechanic is preserved and provides a novel form of gameplay that emphasizes spatial deduction. Playtests revealed that the game is best experienced while seated in a swivel chair, allowing players to rotate physically in sync with head movement while holding the computer connected to the headset for keyboard control.

However, when playing for long periods of time, the game is prone to inducing dizziness and motion sickness. This situation is less severe when playing in the easy mode, where the enemy does not move. Overall, the project shows promise but requires usability improvements to be more comfortable and engaging in VR.

5. Challenges and Future Work

- Cybersickness: To reduce discomfort, different speeds of movement and scenes can be tested to examine cybersickness affected by navigation speed and scene complexity [3].

- Control: Supporting hand controllers for both navigation and shooting would create a more embodied, intuitive experience and reduce reliance on keyboard input.
- Computer enemy: The current random-walk behavior lacks realism. Incorporating Unity's NavMesh system or simple pathfinding algorithms could produce more believable motion patterns and enhance player engagement.
- Multiplayer: A multiplayer version could increase the game's engagement, making it a worthwhile avenue for future exploration.

6. Conclusion

This project demonstrates the feasibility of adapting Screencheat into a VR setting using Unity. The game preserves the reverse-perspective mechanic in a novel form while presenting new challenges in the virtual environment. The results suggest that spatial deduction and perspective-taking can be compelling gameplay elements in VR. Further development, especially toward multiplayer support and improved embodiment, could help create a fully immersive and innovative VR shooter.

References

[1] M. Eylenbosch. *Defining Differences in Movement and View in Tactical First-Person Shooter Games between Virtual Reality and Personal Computer*. PhD thesis, University of Applied Sciences, 2022.

[2] D. Herumurti, A. A. Yunanto, G. A. Senna, I. Kuswardayan, and S. Arifiani. Development of first-person shooter game with survival maze based on virtual reality. In *2020 6th Information Technology International Seminar (ITIS)*, pages 81–86. IEEE, 2020.

[3] L. Rebenitsch and C. Owen. Review on cybersickness in applications and visual displays. *Virtual reality*, 20:101–125, 2016.

[4] Samurai Punk. Screencheat. `https://store.steampowered.com/app/301970/Screencheat/`, 2014. First-person shooter video game.

[5] L. Soni and A. Kaur. Getting started: First person shooter game using unity. In *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*, volume 6, pages 610–613. IEEE, 2023.