**VR HORROR GAME**

|  |  |  |
| --- | --- | --- |
| Prof. Punam Bagul  Guide  KCCEMSR, Thane East  [punam.bagul@kccemsr.edu.in](mailto:punam.bagul@kccemsr.edu.in) | Ritvik Babre  Student  KCCEMSR, Thane East  [ritvikbabre@kccemsr.edu.in](mailto:ritvikbabre@kccemsr.edu.in) | Hitesh Behera  Student  KCCEMSR, Thane East  [hiteshbehera@kccemsr.edu.in](mailto:hiteshbehera@kccemsr.edu.in) |
| Shruti Sabbani  Student  KCCEMSR, Thane East  [shrutisabbani@kccemsr.edu.in](mailto:shrutisabbani@kccemsr.edu.in) | Swapnil Yadav  Student  KCCEMSR, Thane East  [swapnilyadav@kccemsr.edu.in](mailto:swapnilyadav@kccemsr.edu.in) |  |

**ABSTRACT**

The study proposed an analysis framework examining participants' immersive experiences in VR gaming across three dimensions: enjoyment, and spatial integration. The immersive VR experience pushes players to confront fears and overcome obstacles, testing their survival instincts in a journey through darkness.

**OBJECTIVE**

VR horror games capitalize on immersion and emotional responses, showcasing VR effectively and attracting attention in India. With a cultural interest in ghost stories, they appeal to younger demographics and offer unique experiences, differentiating VR from traditional mediums. Leveraging their immersive power can drive adoption and interest in VR among Indian consumers.

**FEATURES**

**GAME MANAGER:**

In this game, the Game Manager serves as the central controller for managing game state, user interface, resource handling, input, events, scene transitions, game logic, and potentially save/load functionality. It orchestrates these elements to ensure a cohesive and enjoyable player experience.

It manages states of the game; includes Win Condition, Lose Condition and Restart

**HEAD TRACKING**:

Head tracking in VR horror games on Android involves using the device's built-in sensors such as gyroscope to track head movements. Unity's Android VR integration enables developers to update the virtual camera's position and orientation based on this tracking data, ensuring alignment with real-world movements.

**ITEM SPAWNER:**

It is strategically placing items with animations and audio cues aids puzzle-solving and progression. Continuous evaluation of player actions ensures a dynamic experience, contributing to atmosphere, challenge, and narrative progression.

**AI ENEMY:**

In a Unity VR horror game, enemy movement is controlled by states: idle, patrol, alerted, chase, attack, and return to patrol. Enemies shift between these states based on player interaction, creating dynamic and suspenseful encounters, enhancing immersion, and offering challenging gameplay.

**GAME MAP:**

A VR horror game map includes specific locations where players navigate and encounter challenges, often featuring eerie settings and strategically placed elements to enhance atmosphere and gameplay.

**METHODOLOGY:**

**Initialization and Setup:** Initialize game engine, VR hardware, and load assets.

**Main Game Loop:** Continuously update the game world and player interactions.

**Player Movement and Interaction:** Track head and hand movements, implement movement controls, and enable object interaction.

**Game Environment:** Design eerie settings, implement lighting and sound effects, and trigger horror events.

**AI Entities:** Create AI-controlled entities with behavior patterns and interactions.

**Game Logic and Puzzles:** Introduce logic puzzles, define conditions for progression, and control game flow.

**Audio Management:** Play background music, trigger audio cues, and utilize 3D audio.

**User Interface (UI):** Create VR-friendly UI elements.

**End Game:** Handle game completion options.

**Cleanup and Resource Release:** Release resources and close connections.

**RESULTS AND CONCLUTION:**

A VR horror game is an immersive and terrifying experience that challenges players to confront their fears in a virtual environment, typically featuring intense gameplay, atmospheric settings, and narrative-driven horror elements.

* **Abstract**
* **Keyword**
* **Objective**
* **Architecture**
* **Features**
  + **Item Spawner**
  + **Item Picker**
  + **Enemy AI**
  + **Game Manager**
* **Future scope**
* **Conclusion**