

Overview of Design and Development of a Third-Person Shooter Maze Escape Game Using Unreal Engine

Ishan Gupta*, Reyaansh Dhammi, Prabhjot Kaur

Department of Computer Application, Shri Guru Tegh Bahadur Institute of Management and Information Technology

ABSTRACT

implemented through the use of Behavior Trees and Unreal's Perception System to create simulated intelligent enemy AI. The mission of the game is for the player to find a way out of a maze without being discovered by patrolling enemies that will detect the presence of the player through sight, sound, and damage indicators. The player engages with the game via intuitive third-person controls, maximizing spatial awareness and immersion. The AI of the enemies responds dynamically to input, changing states depending on sensory feedback to generate a challenging but immersive experience. This one-level demo showcases the employment of visual scripting and level design in creating a responsive and immersive gameplay experience. The project illustrates the applicability of key concepts in AI, interactive design, and 3D game development within a small-game setup. For further work, this system may be scaled for large maps, level generation based on procedural models, and adaptive behavior in AI.

Keywords: Artificial Intelligence, Behavior Trees, Unreal Engine, Third-Person Shooter, Game Development

INTRODUCTION

Game development has seen a huge shift with the advent of sophisticated engines such as Unreal Engine, which allows developers to design engaging, high-performance games. Of the many genres, third-person shooter (TPS) games are especially favored because they offer a mix of action, strategy, and cinematic view, providing players with control over character movement and spatial awareness. This study emphasizes the creation of a one-level TPS maze escape game where the player is required to get to the end of a maze while evading smart enemy AI patrols. The AI system, developed through the use of Unreal Engine's Behavior Trees and Perception System, enables enemies to locate the player via vision, hearing, and damage detection. The behavior of the enemy adaptively shifts as a function of these inputs from passive patrol through alert or aggressive states, based on what the player does. The main objective of the project is to investigate the functionality of modular AI systems in limited gameplay conditions and the influence these systems have on player interaction and difficulty. The project focuses on stealth, strategy, and direction instead of speeded-up

combat, centering on smart decision-making from the player as well as from AI enemies.

LITERATURE REVIEW

Game development research has thoroughly investigated many aspects of artificial intelligence (AI), player interaction, and level design. AI-based gameplay has emerged as a key field in contemporary games, with developers attempting to design more realistic and responsive non-playable characters (NPCs). This research capitalizes on the use of Behavior Trees, a standard framework for representing intricate decision-making in AI agents. Following Isla (2005), behaviour trees are a more modular, easier-to-read design than finite state machines that can be applied to in-game logic of considerable complexity. The Unreal Engine offers this as a visually oriented tool for constructing AI behaviour without necessarily writing code directly, which makes it useful for prototyping in a hurry or teaching. In the area of stealth games, the value of player immersion and credible AI perception systems has been explored by Champandard (2010), who discusses how sensory inputs such as vision and hearing play a role in interactive gameplay. Such

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.



systems have been used in games such as Metal Gear Solid and Splinter Cell, where enemy AI responds to the presence of the player via environmental feedback. Pathfinding and maze-based games have also been researched extensively. Utilization of A* and NavMesh-based pathing systems enables intelligent movement of enemy characters within enclosed areas. Navigation Mesh in Unreal Engine makes it easier by providing dynamic pathing within complex level designs. Third-person perspective in game design has been found to enhance spatial awareness and player interaction (Sweetser & Wyeth, 2005). It enables players to comprehend their environment more effectively, which is essential in maze games where exploration and avoidance are central mechanics. Lastly, research on game difficulty and balance (Hunicke, LeBlanc & Zubek, 2004) highlights the need to provide a balanced challenge. The player should experience tension and risk without feeling overwhelmed, which this project solves by creating a one-level maze with evenly distributed AI patrols and detection areas. Literature gives good groundwork for this project's creation through the union of established AI practices, sensory detection systems, and player-centered design to develop a targeted and engaging gameplay experience.

METHODOLOGY

This section outlines the approach followed in the design and implementation of a third-person shooter (TPS) maze escape game in Unreal Engine. The process of development is modular, step-by-step, and architecture-driven game, player mechanics, AI behavior, and environment design. Unreal Engine's Blueprint Visual Scripting, Behavior Trees, and Perception Systems were the technologies used mostly for implementation.

a. Technologies Used

- Game Engine: Unreal Engine (UE4/UE5)
- Scripting: Blueprint Visual Scripting (Node-based)
- AI Structure: Blackboards, Behavior Trees, and AI Controllers
- AI Perception: Vision module, Hearing module, and Damage detection modul
- Design Tools: NavMesh System, Unreal Engine Level Editor

- Hardware: 8GB RAM PC, GPU, Windows OS

b. Game Design and Architecture

The game has one level of 3D maze constructed using the Unreal Engine level editor. The player must navigate this maze by third-person controls, avoiding being detected by the patrolling enemy AI units. The maze features a few paths and obstacles for the addition of exploration and strategic decision-making.

- Level Design: The maze is constructed with static meshes and obstacles. The paths are narrowly made to enhance tension.
- Navigation Mesh: Included a NavMesh bounds volume to enable AI pathing in the maze level.

c. Player Mechanics

Player character is entered through common third-person movement keys:

- Movement: WASD keys or joystick
- Camera Control: Free camera rotation with the mouse or right joystick
- Actions: Added shooting and interaction mechanics through input binding

Player character has an animation blueprint for smooth transitions and minimalist HUD for observing health and detection warnings.

d. AI Behavior Design

AI enemies are done with the following parts:

- AI Controller: Manages logic and interaction with world
- Blackboard: Serves as the data repository including player location, patrol state, and AI status
- Behavior Tree: Defines the enemy behavior (e.g., patrol, detection, chase, attack)

i. Patrol State

Enemies traverse along a preset route with Target Points. Without detection of the player, AI remains in the patrol state.

ii. Detection System



AI Perception component is implemented with:

- Sight: Detects line-of-sight in a field-of-view cone
- Hearing: Detects player sounds (player footsteps, shooting)
- Damage: Player attacks, and enemy strikes back

Detection causes a switch to Chase Mode where the AI travels to the player's last known location.

iii. Attack Mode

Once the player reaches close range, the AI transitions to attack mode and attempts to kill the player. If the player escapes, the AI reverts to patrol.

e. Game Logic and Conditions

- Win Condition: The player escapes to the exit without being noticed
- Lose Condition: The AI notices and attacks the player, and the game ends

The gameplay is controlled through level blueprints and custom logic controlling transitions between game states.

EVALUATION AND RESULTS

To analyze the success of the TPS Maze Escape Game, both functional testing and casual user feedback were employed. The intentions were to gauge the AI behavior, challenge during gameplay, user experience, and stability of game mechanics under various situations.

a. Functional Testing

The game was tested on the following aspects:

- AI Patrol Accuracy: AI properly navigated pre-defined patrol routes through Unreal's navigation system without collisions or logical faults.

Aspect	Result
AI Patrol Logic	Passed - No logical errors or deadlocks
Sight-based Detection	Passed - Line-of-sight working as intended
Sound-based Detection	Passed - Triggered AI response correctly
Damage Reaction	Passed - Instant reaction upon taking hit
Player Experience	Positive - Engaging and immersive gameplay
Bugs/Crashes	None reported during testing sessions

d. Observations

- Perception System:
- Vision: AI perfectly recognized the player when approaching vision range.
- Heard: AI reacted properly upon player gunfire and steps when in the given hearing distance range.
- Damage: When fired on by the player, the AI instantaneously defaulted to chase/attack mode like expected.
- Behaviour Tree Transitions: AI easily moved through transitions from patrolling, detecting, chasing, back to patrolling without crash/bug issues.
- Player Controls and Movement: The third-person movement system was stable and responsive even with prolonged play sessions.

b. Playtesting Feedback

The game was playtested by a small group of users (peers and classmates), and the following observations were made:

- Immersion and Tension: Players reported feeling truly tense while moving through the maze, particularly when there were enemies around but not yet aware of their presence.
- Stealth Elements: Avoiding detection by sound and movement made players more cautious in their planning, enhancing involvement.
- Enemy Realism: The response of the AI to sound and visibility made them seem more "alive" and unpredictable.
- Learning Curve: Players grasped the purpose quickly but learned to master evading enemies over time, which was a good balance of challenge.

c. Results Summary

- AI behavior was predictable, and state transitions were seamless.
- Maze layout had a positive impact on tension and immersion.
- Players offered suggestions for future content such as hiding mechanics, blind spots for enemies, and more levels to enhance replayability.

CONCLUSION

This research work showcases successful game design and development of a third-person shooter (TPS) maze escape game on Unreal Engine. The main goal—to develop a stealth-centered gameplay experience powered by smart enemy AI—was accomplished through the use of Behavior Trees, AI Perception Systems, and level design modularity.

The project successfully demonstrated the capability of enemy AI to simulate realistic behavior including patrolling, detecting players via multiple senses (sight, hearing, damage), and reacting dynamically to player actions. Playtesting affirmed that the system caused a feeling of tension and strategic choice, important elements of immersion gameplay.

Technically, the utilization of Blueprint Visual Scripting facilitated quick prototyping and iteration without the need for heavy programming. The game also showed how level design and AI behavior can be combined to drive player experience, difficulty balance, and replayability. Although the current build is confined to one level, the foundation established here is a solid one for future development. Some of these might include more maze levels, procedural generation of levels, adaptive difficulty of AI, and more extensive stealth capabilities. In summary, not only does this project meet its academic needs but also provides real-world insight into how AI can be used within interactive 3D game worlds

REFERENCE

1. Unreal Engine Documentation. [Online]. Available: <https://docs.unrealengine.com/>
2. A. Isla, "Handling Complexity in the Halo 2 AI," Game Developers Conference, 2005. [Online]. Available: https://alumni.media.mit.edu/~jorkin/gdc2005/A_I_GDC05_Ilya_Isla.ppt
3. A. Champandard, "The Role of Perception in Game AI," AiGameDev.com, 2010. [Online]. Available: <http://aigamedev.com/open/article/perception-overview/>
4. K. Sweetser and P. Wyeth, "GameFlow: A Model for Evaluating Player Enjoyment in Games," *Computers in Entertainment (CIE)*, vol. 3, no. 3, pp. 1–24, Jul. 2005.
5. R. Hunicke, M. LeBlanc, and R. Zubek, "MDA: A Formal Approach to Game Design and Game Research," *Proceedings of the AAAI Workshop on Challenges in Game AI*, 2004.
6. Unreal Engine AI Behavior Trees. [Online]. Available: <https://docs.unrealengine.com/en-US/InteractiveExperiences/ArtificialIntelligence/BehaviorTrees/>
7. Unreal Engine AI Perception System. [Online]. Available: <https://docs.unrealengine.com/en-US/InteractiveExperiences/ArtificialIntelligence/AIPerception/>
8. Brackeys (YouTube Channel), "Intro to Behavior Trees in Unreal Engine," 2021. [Online]. Available: <https://www.youtube.com>.

HOW TO CITE: Ishan Gupta*, Reyaansh Dhammi, Prabhjot Kaur, Overview of Design and Development of a Third-Person Shooter Maze Escape Game Using Unreal Engine, *Int. J. Sci. R. Tech.*, 2025, 2 (5), 01-04. <https://doi.org/10.5281/zenodo.15314298>