



Blade Master: AI-Driven Melee Combat in Unreal Engine 5

Abdul Quadir^a, Abhishek Kumar^b, Rahul kumar^c, R.K. Pandey^d

^{a, b, c} Computer Science & Technology, KIPM College of engineering & Technology, Gida, Gorakhpur, IN

^d Associate Professor, Computer Science & Technology, KIPM College of engineering & Technology, Gida, Gorakhpur, IN

Gamerabdulup456@gmail.com, Abhishek817502@gmail.com, Rajanrajankumar094@gmail.com, rakeshpandey1976@gmail.com

KEYWORD

Blade Master;
Unreal Engine 5;
Sword Combat;
AI Behavior;
Blueprints;
Game Development;
Niagara Effects;
Combat Mechanics;

ABSTRACT

Blade Master is a third-person sword combat game developed using Unreal Engine 5, designed to deliver realistic one-on-one melee combat with accessible controls and strategic gameplay depth. The project utilizes Unreal Engine's Blueprint visual scripting system to implement core mechanics such as weapon handling, attack combos, enemy AI behavior, and real-time collision detection without extensive manual coding. The enemy AI is developed using Behavior Trees and Environment Query System (EQS) to enable adaptive combat behavior, including chasing, attacking, and defensive responses. Visual feedback is enhanced using the Niagara particle system, while physics-based interactions are handled through Unreal Engine's Chaos Physics system. The game is optimized to run at approximately 60 FPS on mid-range hardware. This paper presents the system architecture, development methodology, performance evaluation, and future scope, including multiplayer expansion and virtual reality integration. The project demonstrates the effectiveness of modern game engines in building high-quality interactive combat systems within a student development environment.

1. Introduction

Blade Master is a third-person sword combat game designed to provide an immersive and skill-based melee fighting experience. The primary objective of the game is to create a combat system where success depends on timing, precision, and strategy rather than repetitive input. Many existing combat games either focus on overly complex mechanics that are difficult for beginners or overly simplified systems that reduce long-term engagement. Blade Master addresses this gap by offering a balanced system that is both accessible and skill-oriented. The game is developed using Unreal Engine 5, which provides advanced tools such as Blueprints, Behavior Trees, Niagara VFX, and Chaos Physics. These systems enable efficient development of gameplay mechanics, artificial intelligence, and visual effects without requiring extensive low-level programming. The gameplay focuses on one-on-one duels against AI-controlled enemies. Each encounter emphasizes reaction time, positioning, and behavioral prediction, supported by real-time animation-driven combat and accurate collision detection. The objective of this work, *Blade Master*, is to develop a real-time third-person sword combat game using Unreal Engine 5 that provides an immersive and skill-based fighting experience. The project focuses

Corresponding Author: Abdul Quadir, Scholar, Department of Computer Science & Technology, KIPM College of Engineering and Technology, U.P., India

Email: Gamerabdulup456@gmail.com

on creating a responsive melee combat system where player performance depends on timing, precision, and strategic decision-making. The game is designed to simulate one-on-one sword duels against AI-controlled enemies, emphasizing realistic combat interactions through animation-driven mechanics and accurate collision detection. The system integrates core gameplay elements such as movement, attack combinations, blocking, and dodging, supported by AI behavior systems for dynamic enemy responses. This section provides a general background of the developed system and its core objectives without including a detailed literature survey or summarizing the final results.

1.1. Material and Methods

The development of *Blade Master* was carried out using Unreal Engine 5 as the primary game development platform due to its advanced real-time rendering, physics simulation, and built-in gameplay systems. The project utilizes Blueprint visual scripting for implementing core gameplay logic, reducing the need for extensive low-level coding and enabling rapid development and testing.

The player control system was designed using Unreal Engine input mapping to handle character movement, camera control, and combat actions such as attacking, blocking, and dodging. Combat mechanics were implemented using animation montages combined with animation notifies to trigger precise hit detection events during sword attacks.

Enemy artificial intelligence was developed using Behavior Trees and the Environment Query System (EQS). These systems allow AI-controlled characters to make decisions based on player position, distance, and combat state, enabling behaviors such as chasing, attacking, defending, and retreating.

Real-time collision detection was implemented using Unreal Engine's Chaos Physics system. Weapon hit detection is achieved through capsule and trace-based collision checks synchronized with animation frames to ensure accurate and responsive combat interactions.

Visual effects were created using the Niagara particle system for elements such as sword impact sparks, hit effects, and environmental feedback. Audio feedback was integrated using MetaSounds to enhance immersion during combat interactions.

The methods used in this project are based on standard Unreal Engine development practices, with modifications applied specifically for implementing a one-on-one sword combat system optimized for performance and responsiveness.

2. Theory/Calculation

The development of *Blade Master* is based on the concept of real-time interactive systems where gameplay is driven by player input, animation systems, and AI-based decision-making. The theoretical foundation of the project lies in real-time game loop processing, where continuous updates are performed for rendering, physics simulation, input handling, and AI behavior execution.

The combat system follows an animation-driven interaction model in which attack actions are triggered through input events and executed using animation montages. Collision detection is synchronized with animation frames using animation notifies, ensuring that damage calculations occur only at precise interaction points between weapon and target. This approach improves both realism and responsiveness in melee combat.

From an AI perspective, the system is based on Behavior Tree logic, which provides a structured decision-making framework. The AI evaluates conditions such as player distance, visibility, and combat state to determine actions like attacking, blocking, or pursuing. This creates dynamic and adaptive enemy behavior rather than fixed scripted responses.

The calculation of damage is based on predefined gameplay parameters such as attack type, weapon state, and hit confirmation events. When a valid collision is detected, health values are reduced accordingly, and appropriate visual and audio feedback is triggered.

Thus, the theoretical model combines real-time game engine architecture with AI decision systems and animation-based combat logic to create a responsive and interactive sword fighting experience

2.1. Results

The developed game *Blade Master* demonstrates a functional and responsive real-time sword combat system using Unreal Engine 5. The gameplay provides smooth character movement, accurate combat interactions, and stable AI behavior during one-on-one duels.

Performance testing shows that the game maintains approximately 55–60 FPS on mid-range hardware and remains playable on lower-end systems with optimized settings. The combat system responds accurately to player inputs, with minimal delay between action and execution.

2.2. System Overview

This appendix summarizes the overall system structure of *Blade Master*. The game is built using Unreal Engine 5 and consists of interconnected modules including player control, combat system, artificial intelligence, animation system, and visual/audio feedback systems. Each module operates independently but communicates through Blueprint-based event handling to ensure smooth gameplay integration.

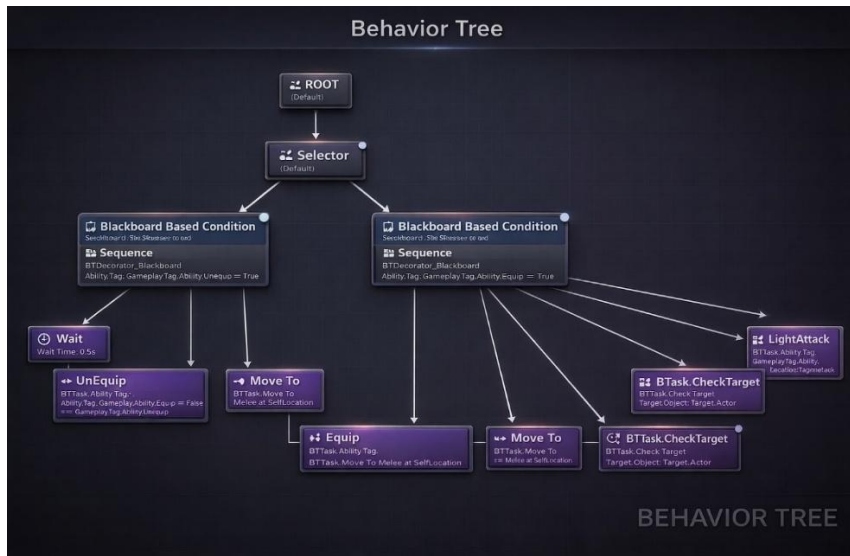
2.3. Combat Logic Representation

The combat system is based on animation-driven execution where each attack is triggered through input events and linked with animation montages. Damage application occurs through animation notifies combined with collision traces. This ensures that hit detection is synchronized with weapon movement during combat.



2.4. AI Behavior Structure

Enemy AI is implemented using Behavior Trees and EQS. The decision-making structure includes states such as idle, chase, attack, and defense. Transitions between states are based on player distance, visibility, and combat conditions, allowing adaptive enemy behavior during gameplay.



2.5. Performance Configuration

The game was tested on multiple hardware configurations to evaluate performance. Optimization techniques include level-of-detail adjustments, controlled particle effects, and efficient Blueprint execution to maintain stable frame rates across different systems.



The AI system successfully performs core behaviors such as chasing the player, initiating attacks, and switching to defensive actions based on combat conditions. Collision detection using animation-synced traces provides consistent hit registration during gameplay.

User testing indicates that the control system is easy to understand, with most users adapting to basic mechanics within a few minutes. Overall, the results confirm that the system meets its primary objective of delivering a smooth and engaging sword combat experience.

2.6. Discussion

The results of *Blade Master* demonstrate that a real-time sword combat system can be effectively implemented using Unreal Engine 5 while maintaining both performance and playability. The combination of animation-driven combat and Behavior Tree-based AI contributes to a responsive and engaging gameplay experience.

The stability of frame rates across different hardware configurations indicates that the system is reasonably optimized for real-time execution. This shows that advanced engine features such as Niagara and Chaos Physics can be used without significantly degrading performance when properly managed. The AI behavior adds meaningful challenge to gameplay by enabling adaptive enemy responses rather than static patterns. However, the observed repetition in certain attack sequences suggests that further improvement in AI decision diversity could enhance long-term engagement.

Overall, the system achieves a balanced integration of combat mechanics, AI behavior, and visual feedback, making the gameplay experience both interactive and immersive while leaving scope for future enhancement in complexity and variability.

3. Conclusions

The Blade Master project successfully demonstrates the development of a real-time third-person sword combat game using Unreal Engine 5. The system effectively integrates combat mechanics, artificial intelligence, animation systems, and visual effects to deliver an interactive and responsive gameplay experience. The study concludes that animation-driven combat combined with Behavior Tree-based AI provides a reliable foundation for creating immersive melee combat systems. The game achieves smooth performance on different hardware configurations while maintaining accurate collision detection and responsive controls. Overall, the project meets its primary objective of developing a skill-based sword combat system and establishes a strong base for further improvements in gameplay depth, AI complexity, and feature expansion.

4. References

- [1] Epic Games, "Unreal Engine 5 Documentation," 2025. [Online]. Available: <https://docs.unrealengine.com>
- [2] Epic Games, "Behavior Trees in Unreal Engine," 2025. [Online]. Available: <https://docs.unrealengine.com>
- [3] Epic Games, "Niagara Visual Effects System Documentation," 2025. [Online]. Available: <https://docs.unrealengine.com>
- [4] Epic Games, "Chaos Physics System Overview," 2025. [Online]. Available: <https://docs.unrealengine.com>
- [5] Epic Games, "Animation Blueprints Documentation," 2025. [Online]. Available: <https://docs.unrealengine.com>
- [6] Epic Games, "MetaSounds Audio System Documentation," 2025. [Online]. Available: <https://docs.unrealengine.com>
- [7] HEMA Alliance, "Historical European Martial Arts Resources," 2023. [Online]. Available: <https://www.hemaalliance.com>
- [8] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," arXiv preprint arXiv:1312.6114, 2013.
- [9] J. Schell, *The Art of Game Design: A Book of Lenses*. Boca Raton, FL, USA: CRC Press, 2014.
- [10] Unreal Engine Forums, "Melee Combat System Discussions," 2024. [Online]. Available: <https://forums.unrealengine.com>