Artificial Intelligence Behaviour System for Minecraft

STUDENTS: ALAA SLEEK, SAMUEL KIM, BATINA SHIKHALIEVA

Motivation

- Minecraft has over 131 million active users each running more than 300 mob entities per game
- Mob entities compete for resources and produce high load on the Minecraft server
- Optimizing mob behavior improves game performance and reduces the server cost

System Overview

- AI Behaviour System enhances the Minecraft mobs ecosystem
- AI Behaviour System improves mob performance and system scalability
- AI Behaviour System allows for efficient use of resources

The system spans in three main components:

Biome AI
Al communication system that optimizes the mob behavior.

ECS Behavior System
System to imitate various mobs attack and run away behavior.

Unreal Engine Environment
A standalone application to demonstrate AI capabilities.

System Implementation

- AI System was built to act as a communication tool that stores information about Minecraft world in a single accessible source.
- The AI System acts as an optimized representation of the Minecraft entities and allows faster, more efficient approach of detecting the mob target.
- The Minecraft mobs behaviors were replicated with Entity Component System. The system part of ECS represents the behaviors that act upon entities (search for food, attack other entities, run away from enemy, etc) to simulate active Minecraft environment.

Use Cases

Implementation has significant improvements as compared to the existing Minecraft environment. It offers:

- O(1) lookup for mob types in a given world location,
- O(N) mob/player relocation from one place to another,
- O(N) lookup for locations with given mobs type.

Results

- Solution scales better than N^2 with increasing mob counts.
- Higher performance and flexibility for modding.
- Independent and self-sufficient system. Implemented as ENTT components separate from the current environment.
- The behaviour of mobs is recreated through data-driven mechanisms with the new AI system

System Future

The future applications of the system:
Integration within MC existing coodebase:
• Adaptation to the active Minecraft environment
• Estimation of the deployment resources

Minecraft codebase AI assessment:
• Evaluation of the proposed AI algorithms
• Estimation of proposed AI mechanisms use for Minecraft environment

Acknowledgments

We would like to say a special thank you to:
• Minecraft Principal Lead SDE John Seghers for his devotion to our project and the endless guidance
• Minecraft Software Engineer Adrian Orszulak for continuous help and support throughout the journey