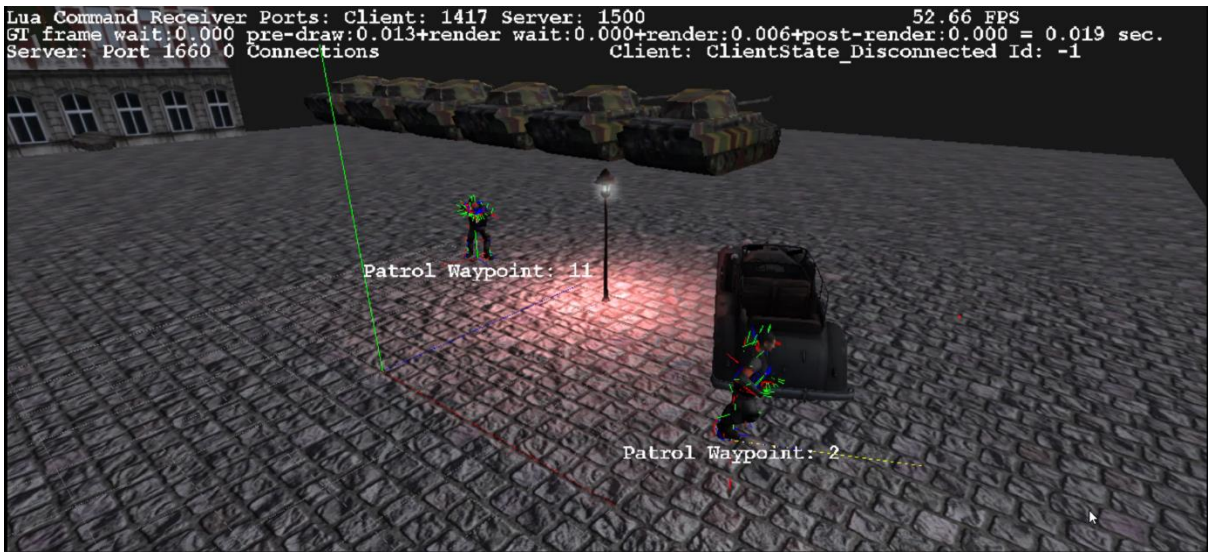
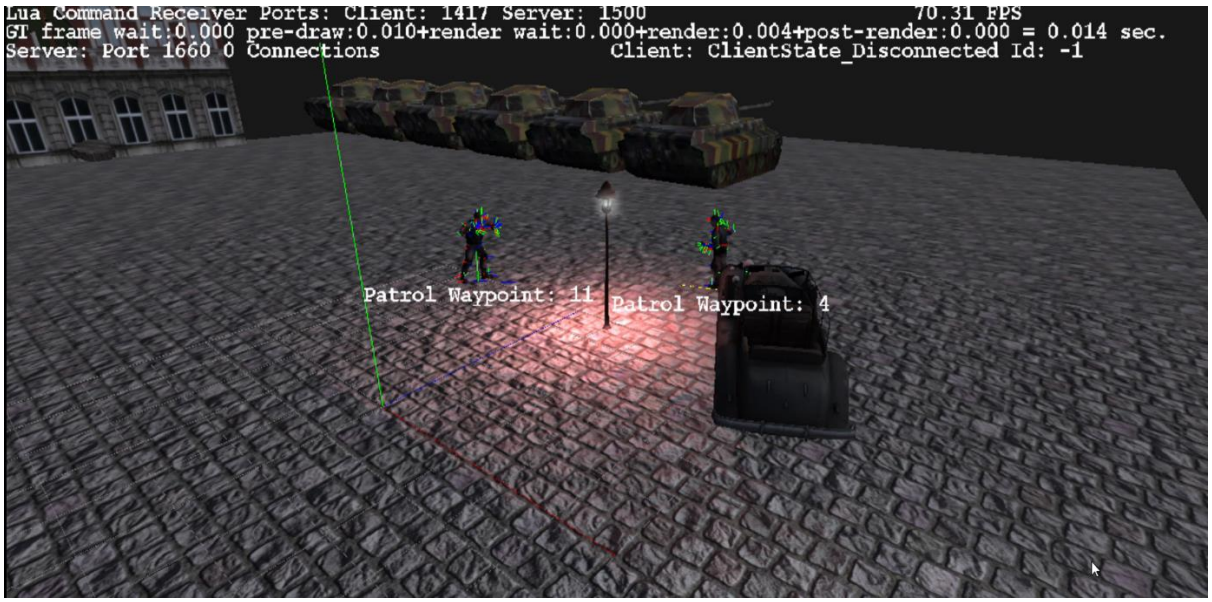
Game Project Contributions

1. NPC Pathfinding and Navigation Mesh Implementation

- Designed and developed a new level to thoroughly test NPC behaviour.
- Implemented a Navigation Mesh system with the following features:
    - **Collision-Free Shortest Path Calculation**: The NPC determines the shortest path while avoiding collisions. Waypoints are set along the path, and if an obstacle is detected, the algorithm dynamically adjusts movement until a clear route is found.
    - **Dynamic Path Recalculation**: The path updates in real-time based on camera position changes, ensuring smooth and adaptive navigation.

Attachments:

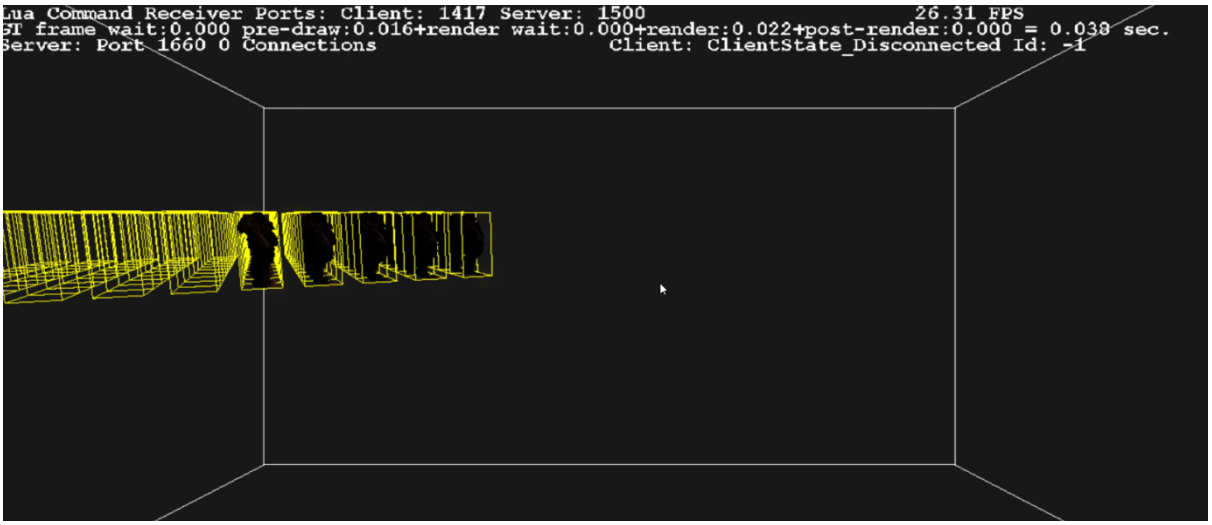


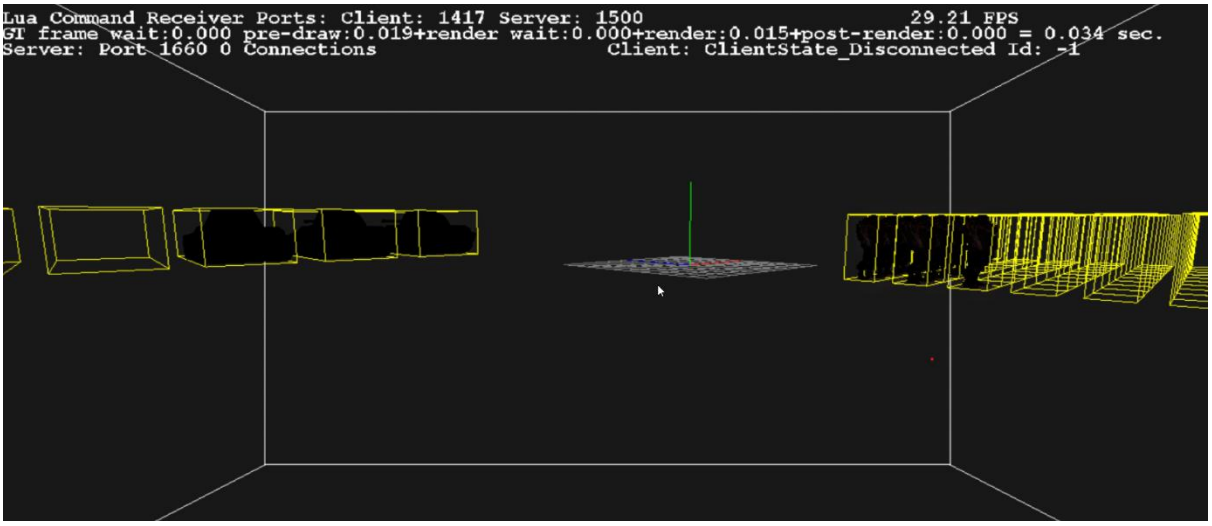


2. Bounding Volumes and Frustum Culling

- Implemented object culling based on the camera frustum and Axis-Aligned Bounding Boxes (AABB) to optimize rendering.
- Calculated AABB using the position buffer (m_hPositionBufferCPU) in the MeshCPU struct and stored it in the Mesh struct.
- In CameraSceneNode::do_CALCULATE_TRANSFORMATIONS, computed the six camera frustum planes and adjusted plane equations for debugging.
- Updated the DebugRenderer to visually render bounding boxes and frustum for easier validation.
- Added checks in SH_DRAW.cpp to determine if AABB points intersect the frustum, culling objects outside the camera's view to improve performance.

Attachments:





3. Collision and Physics Component

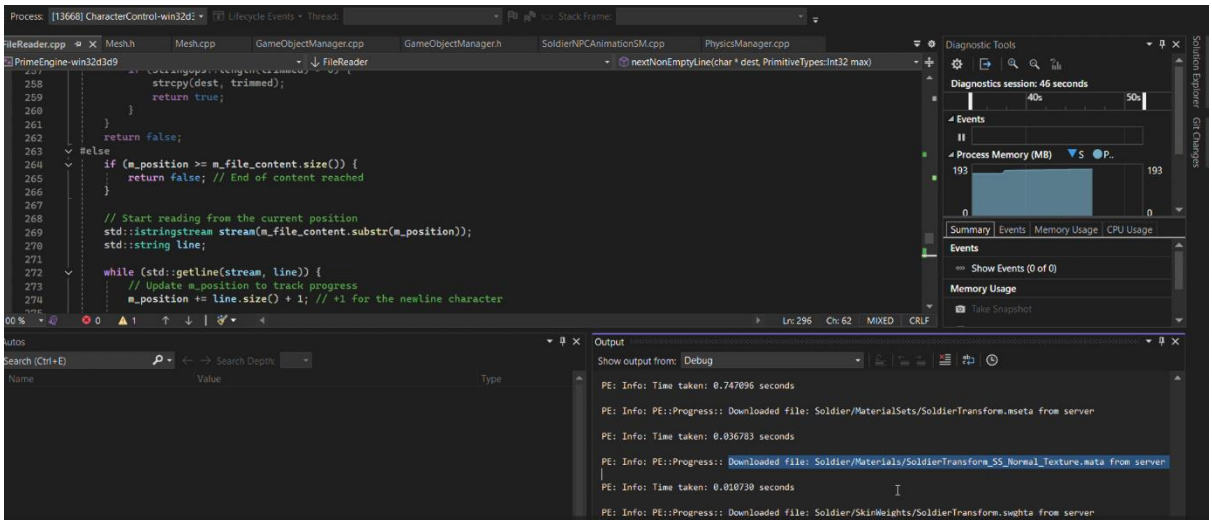
- Developed a **Physics Manager** to store bounding volumes for each mesh.
- Assigned **sphere bounding boxes** to soldiers and **AABB** to other objects.
- Implemented **collision detection** where soldiers slide along obstacles upon impact.
- Ensured gravity simulation by checking whether a soldier is above a surface, gradually adjusting their Y-coordinate if not.

Attachments: Attached with the e-mail (Video named as Assignment-4).

4. Server-Based Asset Loading

- Transitioned asset loading from local storage to a **server-based retrieval system** for enhanced scalability and centralized management.
- Set up an **HTTP server** to serve assets from the AssetsOut folder.
- Modified FileReader.cpp to perform HTTP GET requests, dynamically fetching assets instead of reading from disk.
- Integrated network-based asset management while maintaining compatibility with the existing file I/O system.

Attachments:



5. NPC Pathfinding and Navigation Mesh Implementation

Developed a new level to thoroughly test NPC behaviour. Designed and implemented the Navigation Mesh with the following features:

Collision-Free Shortest Path Calculation: Before moving an NPC, the shortest path without collisions is determined. Intermediate waypoints are evenly spaced along this path. If a collision is detected at any waypoint, the pathfinding algorithm adjusts, moving in the intended direction until a clear route to the destination is found. This ensures optimal collision-free navigation.

Dynamic Path Recalculation: The path is recalculated in real-time whenever the camera position changes, ensuring continuous adaptability.

Attachments: Attached with the e-mail (Video named as Assignment-4).

Unity Prototypes – Recent Work

1. Vehicle Simulation Prototypes

Developed two interactive vehicle simulation prototypes using Unity's physics engine:

Car Prototype

Implemented **realistic car controls**, including acceleration, braking, and steering mechanics.

Designed and tested handling dynamics with physics-based interactions.

Added basic environmental elements for improved user experience.

Play Online: Car Prototype

2. Plane Prototype

Developed a **basic flight simulation** with lift, drag, and thrust mechanics.

Implemented smooth flight controls for an intuitive experience.

Integrated a basic environment to test flight physics.

Play Online: Plane Prototype

3. Unity Essentials Training Projects

Completed Unity Essentials learning projects to refine **core game development skills**.

Built interactive gameplay mechanics and UI elements.

Play Online: Unity Essentials Project