E13: Grid Bounds

Course: IGME 309 – Real Time Simulations for Games II

Golisano College of Computing and Information Sciences

School of Interactive Games and Media

Rochester Institute of Technology

Due: Check in MyCourses

Deliverable: AppClass.cpp file (single file, unzipped)

**Objective:**

The objective of this exercise is for students to implement a method to create a **Bounding Box** that encloses all objects (referred to as **entities**) within the simulated world. This exercise introduces the concept of **bounding volumes** as a fundamental building block for spatial partitioning techniques, such as **Octrees**, and teaches students how to query information about the entities' rigid bodies to calculate a bounding box that can encompass all entities within the simulation.

By completing this exercise, students will:

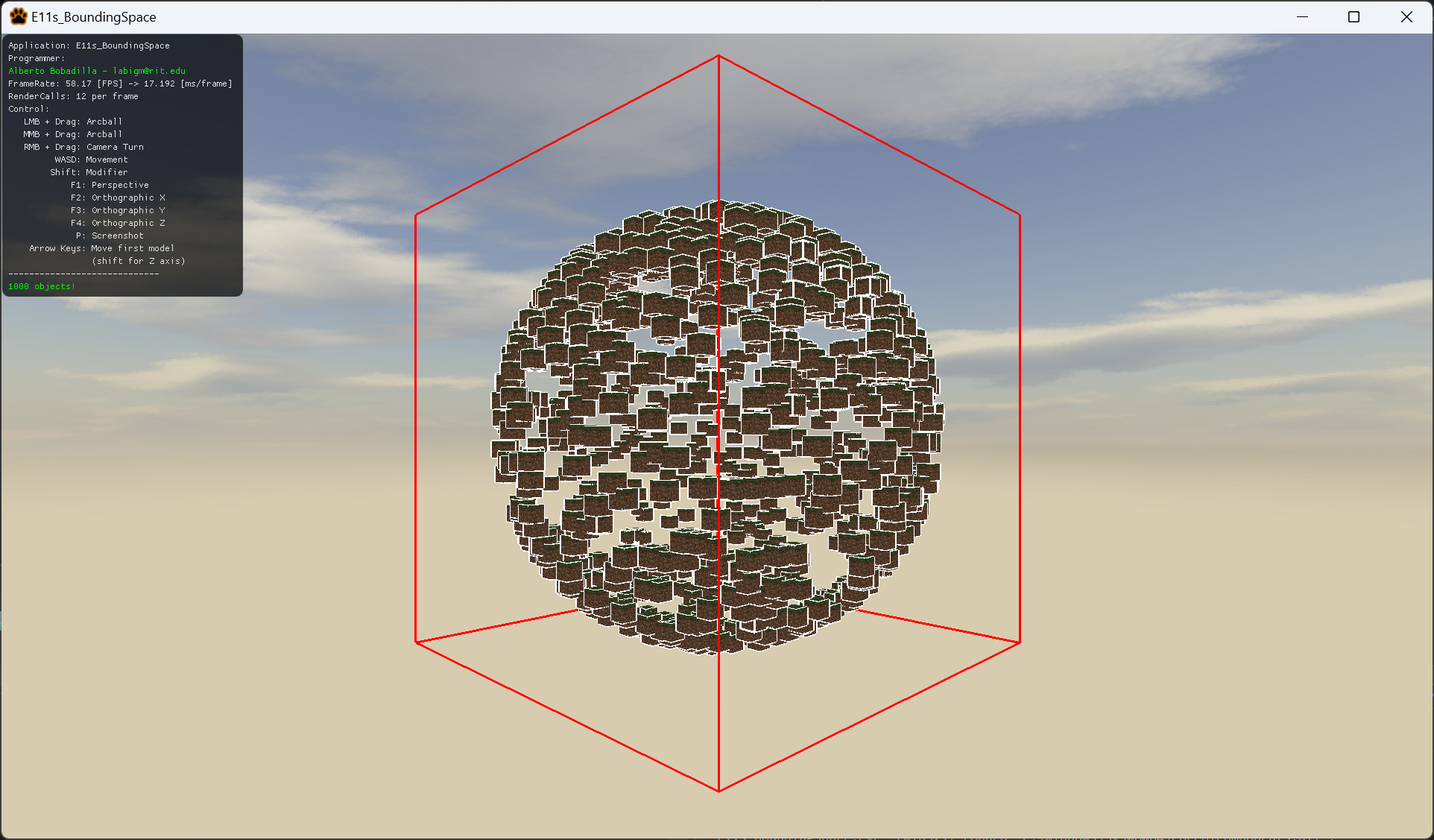
* Learn how to access and query rigid body data, specifically the **minimum** and **maximum** values of the Axis-Aligned Bounding Boxes (AABBs) of entities in **global space**.
* Understand how to compute a bounding box that contains all entities in the world by merging individual AABBs into a larger bounding volume that fully encloses the scene.
* Gain practical experience with the concept of spatial encapsulation, ensuring that a bounding box can accurately represent the overall extent of all entities in the simulation.
* Develop an understanding of how bounding boxes are used in various applications like collision detection, visibility culling, and as the first step towards implementing an **Octree** for efficient spatial partitioning.
* Explore the process of working with both **local space** and **global space** coordinates, and how to manage transformations between them when calculating bounding volumes.
* Learn how to manage dynamic scenes where entities can move, resize, or change their AABBs, and how to update the world’s bounding box accordingly.

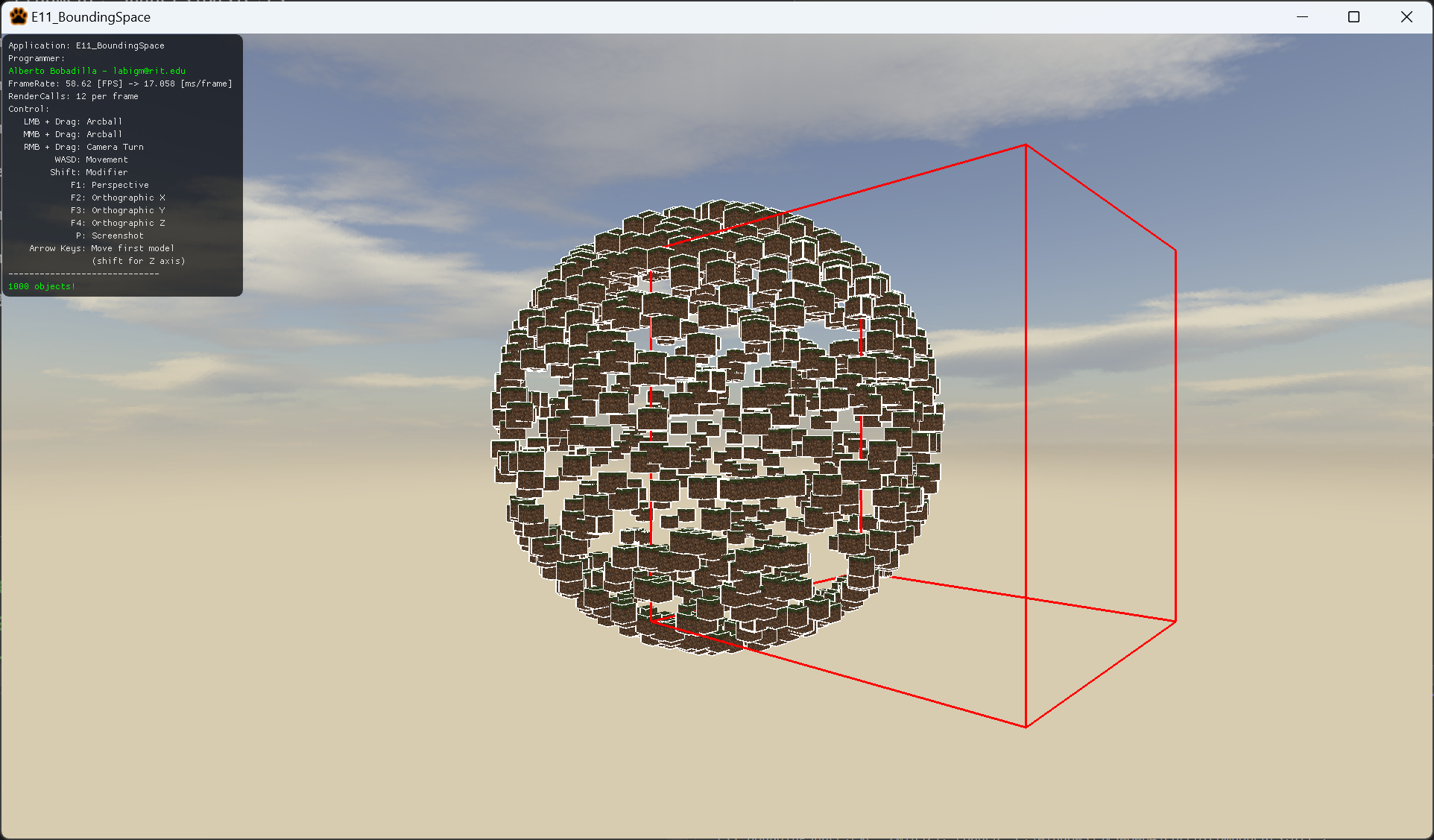
This exercise will serve as the first step towards building a more complex spatial partitioning structure, such as an **Octree**, that will eventually improve performance in simulations by dividing the world into smaller, manageable sections. Understanding how to create a bounding box that encloses all entities is critical for later stages of optimization and for ensuring that the spatial partitioning structure can properly account for all entities in the simulation.

**Instructions:**

This exercise follows lecture D11

1. Under \_Binary look for the example solution. It will look like this:



1. Out of the box some values are added for v3Center and v3Size to show something like this

The center is not in the right place neither is the size.

1. For this you will need to initialize the variables taking in account all objects in the world and make a box that surrounds the object exactly.
2. You are only modifying the AppClass.cpp so this is the only file you need to submit

