E10: Axis Re-aligned Bounding Box

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: RigidBody.cpp file (single file, unzipped)

**Objective:**

The objective of this exercise is to help students understand how to create an **Axis Realigned Bounding Box (ARBB)** around an **Oriented Bounding Box (OBB)** that has been rotated in an arbitrary orientation. This ARBB will allow students to apply the same collision detection logic used in **Axis-Aligned Bounding Boxes (AABB)** vs. **AABB** tests, providing a more efficient but less precise method for detecting collisions between **OBBs**. The ARBB will serve as a quick pre-test for collision detection before investing computational resources into more expensive methods like **Separation Axis Theorem (SAT)**.

Through this exercise, students will:

* Learn how to generate an ARBB that tightly fits an OBB, regardless of its orientation, by recalculating its **Minimum**, **Maximum**, **Center**, and **Halfwidth** vectors based on the rotated geometry.
* Understand how the ARBB acts as a bounding box that encapsulates an OBB, which simplifies collision detection and makes it computationally cheaper compared to SAT-based methods.
* Explore the tradeoff between performance and precision, where ARBBs offer a fast, less precise collision detection method and are used as a first-pass check before using SAT for more accurate collision testing.
* Learn how to apply efficient collision detection techniques between two ARBBs, similar to how AABBs are used for simple collision checks.
* Develop an understanding of how **OBB** and **ARBB** relationships work, and why recalculating the ARBB is necessary to handle rotated objects in a 3D environment.
* Recognize that ARBBs are typically used in real-time applications for broad-phase collision detection, where efficiency is prioritized over precise, per-frame collision testing.

This exercise will prepare students for more advanced collision detection techniques like SAT, giving them a foundational understanding of how simplified bounding box methods can be used to quickly eliminate false positives in the early stages of a collision detection pipeline.

**Important note**

While the term ARBB is not common in the industry, it is used in this course for simplicity’s sake. A more accepted term would be “Transformed Bounding Box” or “Axis-Aligned Bounding Box that encloses an Oriented Bounding Box”. As you can see, ARBB is easier to talk about when brought over and over.

**Instructions:**

This exercise follows lecture D10

1. Under \_Binary look for the example solution. It will look like this when the creeper is rotated:

A screenshot of a computer

Description automatically generated

1. Out of the box the yellow box does not work, you need to make a box that surrounds the AABB (now called OBB as is no longer Axis Aligned but Oriented)

A screenshot of a computer

Description automatically generated

1. For this you will need to follow the comments found in the method SetModelMatrix under RigidBody.cpp:

//Calculate the 8 corners of the oriented box

//translate them to global space

//from those points calculate a box that surround them

1. You are only modifying the RibigBody.cpp so this is the only file you need to submit

