TORQUE MATH #2: BOXES

Exercise Files

Starter – "engine/exercises/chapter10/exer_008.cc"
Answers – "engine/answers/chapter10/exer_008.cc"

Exercise Mission

n/a

Special Steps

Please remember, when you modify the engine and compile, you must copy the new executable over to your Kit/ directory before you can run it and see the changes in the Kit (as instructed below).

Synopsis

In this exercise, we will explore some features of the Box3F class and start learning how to solve concrete problems with it.

Prerequisites

1. *ch1_001.pdf "Using The Kit"*

Exercises

1. *Ray Collision Against A Box (pg 2)*

TORQUE MATH #2: BOXES

1 Ray Collision Against A Box

Goal: Calculate a ray (given a start and end point for the ray) and see if it collides with a box whose min and max points we specify.

Starter Code: You are provided with the following partially written source file.

```
#include "console/simBase.h"

// 1

ConsoleFunction(ch10_exer_008, bool, 5, 5,
                "ch10_exer_008( start , end, min, max )")
{
   Point3F rayStart;
   Point3F rayEnd;

   Point3F boxMin;
   Point3F boxMax;

   // 2
   dSscanf(argv[1], "%f %f %f", &rayStart.x, &rayStart.y, &rayStart.z);
   dSscanf(argv[2], "%f %f %f", &rayEnd.x, &rayEnd.y, &rayEnd.z);

   Con::printf("Ray start => %f %f %f",
               rayStart.x, rayStart.y, rayStart.z );
   Con::printf("Ray end   => %f %f %f",
               rayEnd.x, rayEnd.y, rayEnd.z );

   // 3
   dSscanf(argv[3], "%f %f %f", &boxMin.x, &boxMin.y, &boxMin.z);
   dSscanf(argv[4], "%f %f %f", &boxMax.x, &boxMax.y, &boxMax.z);

   Con::printf("Box min => %f %f %f",
               boxMin.x, boxMin.y, boxMin.z );
   Con::printf("Box max => %f %f %f",
               boxMax.x, boxMax.y, boxMax.z );

   // 4
   // Add Code here

   // 5
   if( 0 )  // Modify me
   {
      return false;
   }
```


TORQUE MATH #2: BOXES

```
   // 6
   // Add Code here  (several lines; see the book)


   // 7
   Con::printf("No collision occurred between our box and the ray");

   return false;
}
```

Steps:

1. Include the proper header file(s) to incorporate the math support required to solve this exercise.
2. Copy the ray data into rayStart and rayEnd. (Done for you.)
3. Copy the box data into boxMin and boxMax. (Done for you.)
4. Add the necessary code to create a new box variable named aBox.
5. Modify the if-statement to test for a poorly defined box.
6. Write all of the code necessary to test for a ray (line) collision and to print the location of that collision to the console. (Remember to return true if a collision is detected).
7. Print a message and return false if no collision occurs. (Done for you.)

Output Goal:

When you successfully complete this exercise and rebuild Torque, you will be able to start the kit and to open the console(~). Then, you can try the following calls to the new console function and should get the given results.

```
==>ch10_exer_008( "5 20 5", "5 0 5", "0 0 0", "10 10 10" );
Ray start => 5.000000 20.000000 5.000000
Ray end   => 5.000000 0.000000 5.000000
Box min => 0.000000 0.000000 0.000000
Box max => 10.000000 10.000000 10.000000
Our ray collided with the box at point location <5.000000 10.000000 5.000000>


==>ch10_exer_008( "5 20 5", "5 25 5", "0 0 0", "10 10 10" );
Ray start => 5.000000 20.000000 5.000000
Ray end   => 5.000000 25.000000 5.000000
Box min => 0.000000 0.000000 0.000000
Box max => 10.000000 10.000000 10.000000
No collision occurred between our box and the ray
```


TORQUE MATH #2: BOXES

```
==>ch10_exer_008( "5 20 5", "5 0 5", "10 10 10", "0 0 0" );
Ray start => 5.000000 20.000000 5.000000
Ray end   => 5.000000 0.000000 5.000000
Box min => 0.000000 0.000000 0.000000
Box max => 10.000000 10.000000 10.000000
Our ray collided with the box at point location <5.000000 10.000000 5.000000>

==>ch10_exer_008( "5 20 5", "5 0 5", "5 10 5", "5 10 5" );
Ray start => 5.000000 20.000000 5.000000
Ray end   => 5.000000 0.000000 5.000000
Box min => 0.000000 0.000000 0.000000
Box max => 0.000000 0.000000 0.000000
Our ray collided with the box at point location <5.000000 10.000000 5.000000>
```

Questions:

1. In the third output sample above, we specified the min values "10 10 10" and "0 0 0" respectively, which are bad bounds for a box (they should be reversed). However, as can be seen, the console function still worked, and we still got a collision. Why did this happen?
2. In the fourth example, we specified a box with zero volume, and yet the ray still collided with it. Why did this happen?

Hints:

1. Box3F has a constructor that takes Point3F values to initialize the box.

