TORQUE MATH #1: POINTS

Exercise Files

Starter – "engine/exercises/chapter10/exer_007.cc"
Answers – "engine/answers/chapter10/exer_007.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 Point3F class and start learning how to solve concrete problems with it.

Prerequisites

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

Exercises

1. *Calculate Angle Between Vectors (pg 2)*

TORQUE MATH #1: POINTS

1 Calculate Angle Between Vectors

Goal: Calculate the angle between two 3D vectors, using Point3F.

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

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

// 1

ConsoleFunction(ch10_exer_007, void, 3, 3,
                "ch10_exer_007( point0 , point1 )")
{
   Point3F point[2];


   // 2
   // Add code here
   // Add code here
   Con::printf("Point0 orig => %f %f %f",
               point[0].x, point[0].y, point[0].z );
   Con::printf("Point1 orig => %f %f %f\n",
               point[1].x, point[1].y, point[1].z );


   // 3
   // Add code here
   // Add code here
   Con::printf("Point0 norm => %f %f %f",
               point[0].x, point[0].y, point[0].z );
   Con::printf("Point1 norm => %f %f %f\n",
               point[1].x, point[1].y, point[1].z );

   // 4
   F32 vectorLen = 0.0f;

   // Add code here
   if( (1.0f - vectorLen) > POINT_EPSILON ) // point[0]
   {
      return;
   }

   // Add code here
   if( (1.0f - vectorLen) > POINT_EPSILON ) // point[1]
   {
      return;
   }
```


TORQUE MATH #1: POINTS

```
   // 5
   F32 dotProduct = 0.0f;

   // Add code here

   // 6
   F32 theta = 0.0f;

   if( 1.0f == dotProduct ) // Same direction
      theta = 0.0f; // Modify this code.
   else if( -1.0f == dotProduct ) // Exactly opposing directions
      theta = 0.0f; // Modify this code.
   else if( 0.0f == dotProduct ) // @ Right angles
      theta = 0.0f; // Modify this code.
   else // Let's calculate the angle
      theta = 0.0f; // Modify this code.

   Con::printf("The angle between the two vectors is %f degrees.",
               theta );
}
```

Steps:

1. Include the proper header file(s) to incorporate the math support required to solve this exercise.
2. Copy the point vector data passed into this console function into the supplied point[0] and point[0] vectors.
3. Normalize the vectors. (This is not necessarily required, but it makes the subsequent operations simpler.)
4. Check that the vectors both normalized well by checking that their new lengths are within POINT_EPSILON of 1.0.
5. Calculate the dot product of the two vectors.
6. Convert that dot product into an angle theta (between the vectors).

Output Goal:

When you successfully complete this exercise and rebuild Torque, you will we 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_007( "0 0 1" , "0 0 1" ) ;
Point0 orig => 0.000000 0.000000 1.000000
Point1 orig => 0.000000 0.000000 1.000000

Point0 norm => 0.000000 0.000000 1.000000
Point1 norm => 0.000000 0.000000 1.000000

The angle between the two vectors is 0.000000 degrees.
```


TORQUE MATH #1: POINTS

```
==>ch10_exer_007( "0 0 1" , "0 0 -1" ) ;
Point0 orig => 0.000000 0.000000 1.000000
Point1 orig => 0.000000 0.000000 -1.000000

Point0 norm => 0.000000 0.000000 1.000000
Point1 norm => 0.000000 0.000000 -1.000000

The angle between the two vectors is 180.000000 degrees.
```

Hints:

1. The two points are passed into the console function as the const char * strings in the arrays argv[1] and argv[2]. You'll need to extract the <x y z> values from these strings and get them into the point[0] and point[1] vectors some how.
   - For extra help on this see the (C++) implementation of the console function "SimpleMeshLoader_createVert" in the file "engine\constructor\constructorSimpleMesh.cc".
2. The angle between two vectors can be calculated using the dot product.
   - A dot product of 1.0 is equal to a 0-degree angle between the vectors.
   - A dot product of -1.0 is equal to a 180-degree angle between the vectors.
   - A dot product of 0.0 is equal to a 90-degree angle between the vectors.
   - Any other value requires us to calculate the angle theta on our own.
3. The following dot product relationship is true: a . b == |a| x |b| x cosine(theta)
   - a . b means "the dot product between vectors a and b"
   - |a| means "the magnitude (length) of vector a."
4. Remember, that the ANSI-C function acos() returns a value in radians, so you may need to do some conversion to get that to degrees.
5. Remember to check appendix B.6.3 for useful math utilities.

Questions:

1. What is the angle between the two vectors: "0 0 1" and "0 0.5 0.5"?
2. What is the angle between the two vectors: "0 0 1" and "0 0.5 -0.5"?

