CH10_007

ANSWER

TORQUE MATH #1 POINTS

1 Calculate Angle Between Vectors

Finished Code:

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

// 1. Include this directory to get ALL math features
#include "math/mMath.h"

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

   // 2. Copy point data into point[0] and point[1] using dSscanf()
   dSscanf(argv[1], "%f %f %f", &point[0].x, &point[0].y, &point[0].z);
   dSscanf(argv[2], "%f %f %f", &point[1].x, &point[1].y, &point[1].z);

   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. Normalize the vectors
   point[0].normalize();
   point[1].normalize();

   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 );
```

TORQUE MATH #1 POINTS

```
   // 4. Check for good vectors and abort if this is not true.
   F32 vectorLen = 0.0f;

   vectorLen = point[0].len();
   if( (1.0f - vectorLen) > POINT_EPSILON )
   {
      return;
   }

   vectorLen = point[1].len();
   if( (1.0f - vectorLen) > POINT_EPSILON )
   {
      return;
   }

   // 5. Calculate the dot product
   F32 dotProduct = 0.0f;
   dotProduct = mDot( point[0], point[1] );

   // 6. Determine the angle between the vectors.
   F32 theta = 0.0f;
   if( 1.0f == dotProduct ) // Same direction
      theta = 0.0f;
   else if( -1.0f == dotProduct ) // Exactly opposing directions
      theta = 180.0f;
   else if( 0.0f == dotProduct ) // @ Right angles
      theta = 90.0f;
   else // Let's calculate the angle
      theta = mRadToDeg( mAcos( dotProduct ) );

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

Answers:

1. 45 degrees
2. 135 degrees