Custom Math Library

Navarre Luce

Introduction:

This is a custom math library built for first year Game Programming class at AIE.

The library contains 3 vector classes; Vector2, Vector3, and Vector4. These can hold 2, 3, and 4 dimensional vectors.

The library contains 2 matrix classes; Matrix3x3 and Matrix4x4.

The library also contains a few utility functions.

Vector2 class

Constructors

Vector2()

> Creates a zero vector.

Vector2(float x,float y)

> Creates a vector initialized to x and y.

Vector3 class

Constructors

Vector3()

> Creates a zero vector.

Vector3(float x,float y, float z)

> Creates a vector initialized to x, y, and z.

Vector3(float x,float y)

> Creates a vector initialized to x, y, and 0 (for z).

Function unique to Vector3

CrossProduct(Vector3 other)

> Returns a vector that is the cross product of this and other.

Vector4 class

Constructors

Vector4()

Creates a zero vector.

Vector4(float x,float y, float z, float w)

Creates a vector initialized to x, y,z, and w.

Vector4(float x,float y)

Creates a vector initialized to x, y, and 0 (for z and w).

Vector4(int colorValue)

Creates a vector initialized to a color value. Input color value can be a HTML hex color value such as 0xff0000 (red) or 0x808080 (grey). Resulting vector color is normalized to the range 0.0 to 1.0.

Functions shared by all vector classes

Operator –

Subtracts a vector from another

Operator +

Adds a vector to another

Operator =

Sets a vector to the value of another

Operator * (scalar)

Multiplies a vector my a scalar

DotProduct(Vector other)

Calculates the dot product of this vector and another vector

LinerInterpolate(Vector other, float factor)

Calculates a vector that is the linear interpolation of this vector and another vector based on a scalar factor.

getMagnitude()

Calculates the magnitude of this vector.

normalize()

Modifies this vector to be unit length.

getNormalized()

Returns a vector that is a normalized version of this, without modifying this.

Matrix3x3

Constructors

Matrix3x3()

Creates the identity matrix

Matrix3x3(float radians)

Creates a rotation matrix. The matrix represents a rotation around the z axis.

Matrix4x4

Constructors

Matrix4x4()

Creates the identity matrix

Matrix4x4(Vector3 axis, float radians)

Creates a rotation matrix. The matrix represents a rotation around the given axis. The axis must be a unit vector.

Functions (shared by both matrix classes)

Transpose()

Transposes this matrix in place and returns it.

GetTranspose()

Creates a new matrix that is the transpose of this. Does not modify this.

SetScale(float scale)

Modifies this matrix to be a matrix that will scale by the given factor.

SetTranslate(float x, float y) – Matrix3x3 only

SetTranslate(float x, float y, float z) – Matrix4x4 only

Modifies this matrix to be a matrix that will translate by the given values.

Operator+

Adds a matrix to another

Operator-

Subtracts a matrix (right hand side) from this (left hand side).

Operator* (matrix * matrix)

Performs standard matrix multiplication on 2 matrixes. Return the resulting matrix.

Operator * (matrix * vector)

Transforms the given vector by the given matrix. Returns the transformed vector. Matrix3x3 can transform a Vecto2 or Vector3. Matrix4x4 can transform a Vector3 or Vector4.

Operator +=

Adds a matrix (right hand side) to this (left hand side). Return this.

Operator -=

Subtracts a matrix (right hand side) from this (left hand side). Return this.

Operator *=

Performs standard matrix multiply between the left hand matrix and the right hand matrix. The left hand matrix is modified and returned.

Nonclass Functions

LinearInterpolate(float a, float b, float f)

Returns a float that is the linear interpolation between a and b by the factor f.

DegreesToRadians(float degrees)

Converts the given value in degrees to radians.

RadiansToDegrees(float radians)

Converts the given value in radians to degrees.

ConvertToNextPowerOfTwo(unsigned int input)

Converts the given input to the next higher power of 2. For example an input of 18 will return 32.

ConvertToNearestPowerOfTwo(unsigned int input)

Converts the given input to the nearest power of 2. For example an input of 18 will return 16.