Vector 2 Class – Vec2.h

Vec2()

Default constructor, creates a vector of (0.f, 0.f)

Vec2(float *a_fpV2)

Constructor that will create a vector of (a_fpV2[1], a_fpV2[2])

Vec2(const float a_fX, const float a_fY)

Constructor that will create a vector of (a_fX, a_fY)

operator float *()

Casts the vector to a float array

operator const float *() const

Casts the vector to a float array constant

void **operator=**(const**Vec2**&a_Vec1)

Sets the vector to the vector of a_Vec1

bool **operator==**(const**Vec2**&a_Vec1)

Compares the vector to a vector of a_Vec1 and returns true if the vectors are the same

Vec2 **operator+**(const**Vec2**&a_Vec1)

Returns the result of the vector components added to the components of a_Vec1

Vec2 **operator+**(const float a_fAdd)

Returns the result of the float a_fAdd added to the components of the vector

void **operator+=**(const**Vec2**&a_Vec1)

Adds the components of a_Vec1 to the components of the original vector

void **operator+=**(const float a_fAdd)

Adds the float a_fAdd to the components of the original vector

Vec2 **operator-**(const**Vec2**&a_Vec1)

Returns the result of the vector components subtracted from the components of a_Vec1

Vec2 **operator-**(const float a_fSub)

Returns the result of the float a_fSub subtracted from the components of the vector

void **operator-=**(const**Vec2**&a_Vec1)

Subtracts the components of a_Vec1 from the components of the original vector

void **operator-=**(const float a_fSub)

Subtracts the float a_fSub from the components of the original vector

Vec2 **operator/**(const float a_fDiv)

Returns the result of the components of the vector divided by a_fDiv

Vec2 **operator/**(const**Vec2**&a_Vec1)

Returns the result of the vector components divided by the components of a_Vec1

void **operator/=**(const float a_fDiv)

Divides the components of the original vector by a_fDiv

void **operator/=**(const**Vec2**&a_Vec1)

Divides the components of the original vector by the components of a_Vec1

Vec2 **operator***(const float a_fMul)

Returns the result of the components of the original vector multiplied by a_fMul

Vec2 **operator***(const**Vec2**&a_Vec1)

Returns the result of the components of the original vector multiplied by the components of a_Vec1

void **operator*=**(const float a_fMul)

Multiplies the components of the original vector by a_fMul

void **operator*=**(const**Vec2**&a_Vec1)

Multiplies the components of the original vector by the components of a_Vec1

float **GetMagnitude**()

Returns the distance between a point and the origin (0.f, 0.f)

Vec2 **NormaliseVec2**()

Normalises the original vector returning values between 0 and 1

Vec2 **LinearInterpolation**(const**Vec2**&a_pVec1, float a_fDest)

Vec2 **RotateVec2**(const**Vec2**&a_pVec1, float a_fRotationInDegrees)

Rotates the point of the original vector around a_pVec1 by a_fRotationInDegrees

float **DotProduct**(const**Vec2**&a_pVec1)

Returns a float of the product of two vectors

float **GetAngle**(const**Vec2**&a_pVec1)

Gets the angle between the original vector as a point and the point of a_pVec1

Vector 3 Class – Vec3.h

Vec3()

Default constructor, creates a vector of (0.f, 0.f, 0.f)

Vec3(float a_fX, float a_fY, float a_fZ)

Constructor that will create a vector of (a_fX, a_fY, a_fZ)

operator float *()

Casts the vector to a float array

operator const float *() const

Casts the vector to a float array consant

float **GetMagnitude**()

Returns the distance between a point and the origin (0.f, 0.f, 0.f)

Vec3 **NormaliseVec3**()

Normalises the original vector components returning values between 0 and 1

Vec3 **LinearInterpolation**(const**Vec3**&a_pVec1, float a_fDest)

float **DotProduct**(const**Vec3**&a_pVec1)

Returns a float of the product of two vectors

float **GetAngle**(const**Vec3**&a_pVec1)

Gets the angle between the original vector as a point and the point of a_pVec1

void **operator=**(const**Vec3**&a_Vec1)

Sets the vector to the vector of a_Vec1

bool **operator==**(const**Vec3**&a_Vec1)

Compares the vector to a vector of a_Vec1 and returns true if the vectors are the same

Vec3 **operator+**(const**Vec3**&a_Vec1)

Returns the result of the vector components added to the components of a_Vec1

Vec3 **operator+**(const float a_fAdd)

Returns the result of the float a_fAdd added to the components of the vector

void **operator+=**(const**Vec3**&a_Vec1)

Adds the components of a_Vec1 to the components of the original vector

void **operator+=**(const float a_fAdd)

Adds the float a_fAdd to the components of the original vector

Vec3 **operator-**(const**Vec3**&a_Vec1)

Returns the result of the vector components subtracted from the components of a_Vec1

Vec3 **operator-**(const float a_fSub)

Returns the result of the float a_fSub subtracted from the components of the vector

void **operator-=**(const**Vec3**&a_Vec1)

Subtracts the components of a_Vec1 from the components of the original vector

void **operator-=**(const float a_fSub)

Subtracts the float a_fSub from the components of the original vector

Vec3 **operator/**(const float a_fDiv)

Returns the result of the components of the vector divided by a_fDiv

Vec3 **operator/**(const**Vec3**&a_Vec1)

Returns the result of the vector components divided by the components of a_Vec1

void **operator/=**(const float a_fDiv)

Divides the components of the original vector by a_fDiv

void **operator/=**(const**Vec3**&a_Vec1)

Divides the components of the original vector by the components of a_Vec1

Vec3 **operator***(const float a_fMul)

Returns the result of the components of the original vector multiplied by a_fMul

Vec3 **operator***(const**Vec3**&a_Vec1)

Returns the result of the components of the original vector multiplied by the components of a_Vec1

void **operator*=**(const float a_fMul)

Multiplies the components of the original vector by a_fMul void **operator*=**(const**Vec3**&a_Vec1)

Multiplies the components of the original vector by the components of a_Vec1

Vector 4 Class – Vec4.h

Vec4()

Default constructor, creates a vector of (0.f, 0.f, 0.f, 0.f)

Vec4(float a_fX, float a_fY, float a_fZ, float a_fW)

Constructor that will create a vector of (a_fX, a_fY, a_fZ, a_fW)

Vec4(unsigned int a_uiHex)

Creates a vector based on a hexadecimal value

float **GetMagnitude**()

Returns the distance between a point and the origin (0.f, 0.f, 0.f, 0.f)

Vec4 **NormaliseVec4**()

Normalises the original vector returning values between 0 and 1

void **operator=**(const**Vec4**&a_Vec1)

Sets the vector to the vector of a_Vec1

bool **operator==**(const**Vec4**&a_Vec1)

Compares the vector to a vector of a_Vec1 and returns true if the vectors are the same

Vec4 **operator+**(const**Vec4**&a_Vec1)

Returns the result of the vector components added to the components of a_Vec1

Vec4& **operator+=**(const**Vec4**&a_Vec1)

Adds the components of a_Vec1 to the components of the original vector

Vec4 **operator-**(const**Vec4**&a_Vec1)

Returns the result of the vector components subtracted from the components of a_Vec1

Vec4& **operator-=**(const**Vec4**&a_Vec1)

Subtracts the components of a_Vec1 from the components of the original vector

Vec4 operator/(const float a_fDiv)

Returns the result of the components of the vector divided by a_fDiv

Vec4 operator/(const**Vec4**&a_Vec1)

Returns the result of the vector components divided by the components of a_Vec1

Vec4& **operator/=**(const float a_fDiv)

Divides the components of the original vector by a_fDiv

Vec4& **operator/=**(const**Vec4**&a_Vec1)

Divides the components of the original vector by the components of a_Vec1

Vec4 operator*(const float a_fMul)

Returns the result of the components of the original vector multiplied by a_fMul

Vec4 operator*(const**Vec4**&a_Vec1)

Returns the result of the components of the original vector multiplied by the components of a_Vec1

Vec4& **operator*=**(const float a_fMul)

Multiplies the components of the original vector by a_fMul

Vec4& **operator*=**(const**Vec4**&a_Vec1)

Multiplies the components of the original vector by the components of a_Vec1

Matrix 3 Class – Matrix3.h

Matrix3()

Creates an empty Matrix with 9 elements of 0

Matrix3(float a_fMatrixArray[3][3])

Creates a Matrix that has the elements of a_fMatrixArray

operator float *()

Casts the Matrix to a float pointer

operator const float *() const

Casts the matrix to a constant float pointer

Matrix3 IdentityMatrix()

Returns an Identity Matrix

Matrix3 **OrthoView**()

Returns an Orthogonal View Matrix

Matrix3 **ZeroMatrix**()

Returns a Matrix with all elements set to 0.

Matrix3 **RotateMatrixX**(float a_fRotation)

Returns a Matrix rotated by a_fRotation on the X axis

Matrix3 **RotateMatrixY**(float a_fRotation)

Returns a Matrix rotated by a_fRotation on the Y axis

Matrix3 **RotateMatrixZ**(float a_fRotation)

Returns a Matrix rotated by a_fRotation on the Z axis

Matrix3 **RotateMatrix**(**Vec3**&a_Vec3)

Returns a Matrix rotated by the values in a_Vec3 for each axis

Vec3 **TransformPoint**(**Vec3**a_pVec3)

Transforms the Matrix by a_pVec3

void **operator=**(const**Matrix3**&a_Mat1)

bool **operator==**(const**Matrix3**&a_Mat1)

bool **operator!=**(const**Matrix3**&a_Mat1)

Matrix3 **operator+**(const**Matrix3**&a_Mat1)

Matrix3 **operator+**(const**Vec3**&a_Vec3)

Matrix3 **operator+=**(const**Matrix3**&a_Mat1)

Matrix3 **operator+=**(const**Vec3**&a_Vec3)

Matrix3 **operator-**(const**Matrix3**&a_Mat1)

Matrix3 **operator-**(const**Vec3**&a_Vec3)

Matrix3 **operator-=**(const**Matrix3**&a_Mat1)

```
  Matrix3  operator-=(constVec3&a_Vec3)

  Matrix3  operator*(constMatrix3&a_Mat1)

Matrix3&  operator*=(constMatrix3&a_Mat1)

  Matrix3  operator*(const float &a_Float)

Matrix3&  operator*=(const float &a_Float)

     Vec3  operator*(constVec3&a_Vec3)
```