vectormath Documentation

Release 0.1.1

3point Science

November 08, 2016

Contents

Vector math utilities for Python built on NumPy

CHAPTER 1

Why

The `vectormath` package provides a fast, simple library of vector math utilities by leveraging NumPy. This allows explicit geometric constructs to be created (for example, `Vector3` and `Plane`) without redefining the underlying array math.

CHAPTER 2

Scope

The `vectormath` package includes `Vector3`/`Vector2` and `Vector3Array`/`Vector2Array`.

CHAPTER 3

Goals

- Speed: All low-level operations rely on NumPy arrays. These are densely packed, typed, and partially implemented in C. The `VectorArray` classes in particular take advantage of this speed by performing vector operations on all Vectors at once, rather than in a loop.
- Simplicty: High-level operations are explicit and straight-forward. This library should be usable by Programmers, Mathematicians, and Geologists.

CHAPTER 4

Alternatives

- NumPy can be used for any array operations
- Many small libraries on PyPI (e.g. vectors) implement vector math operations but are are only built with single vectors in mind.

CHAPTER 5

Connections

- properties uses `vectormath` as the underlying framework for Vector properties.

CHAPTER 6

Installation

To install the repository, ensure that you have pip installed and run:

```
pip install vectormath
```

For the development version:

```
git clone https://github.com/3ptscience/vectormath.git
cd vectormath
pip install -e .
```

CHAPTER 7

Examples

This example gives a brief demonstration of some of the notable features of `Vector3` and `Vector3Array`

```
import numpy as np
import vectormath as vmath

# Single Vectors
v = vmath.Vector3(5, 0, 0)
v.normalize()
print(v)                            # >> [1, 0, 0]
print(v.x)                          # >> 1.0

# VectorArrays are much faster than a for loop over Vectors
v_array = vmath.Vector3Array([[4, 0, 0], [0, 2, 0], [0, 0, 3]])
print(v_array.x)                    # >> [4, 0, 0]
print(v_array.length)               # >> [4, 2, 3]
print(v_array.normalize())          # >> [[1, 0, 0], [0, 1, 0], [0, 0, 1]]

# Vectors can be accessed individually or in slices
print(type(v_array[1:]))            # >> vectormath.Vector3Array
print(type(v_array[2]))             # >> vectormath.Vector3

# All these classes are just numpy arrays
print(isinstance(v, np.ndarray))    # >> True
print(type(v_array[1:, 1:]))        # >> numpy.ndarray
```

Current version: v0.1.1

Contents:

7.1 Basic Vector Math

7.1.1 BaseVector

class `vectormath.vector.`**`BaseVector`**

Class to contain basic operations used by all Vector classes

`as_length`(*value*)

Return a new vector scaled to given length

`as_percent`(*value*)

Return a new vector scaled by given decimal percent

as_unit()
Return a new vector scaled to length 1

cross(*vec*)
Cross product with another vector

dot(*vec*)
Dot product with another vector

length
Length of vector

normalize()
Scale the length of a vector to 1 in place

x
x-component of vector

y
y-component of vector

7.1.2 BaseVectorArray

class vectormath.vector.**BaseVectorArray**
Class to contain basic operations used by all VectorArray classes

dims
Tuple of different dimension names for Vector type

dot(*vec*)
Dot product with another vector

length
Array of vector lengths

nV
Number of vectors

normalize()
Scale the length of all vectors to 1 in place

x
Array of x-component of vectors

y
Array of y-component of vectors

7.1.3 Vector3

class vectormath.vector.**Vector3**
Primitive 3D vector defined from the origin

New Vector3 can be created with:

- another Vector3
- length-3 array
- x, y, and y values
- no input (returns [0., 0., 0.])

z
z-component of vector

7.1.4 Vector2

class vectormath.vector.**Vector2**
Primitive 2D vector defined from the origin

New Vector2 can be created with:

- another Vector2
- length-2 array
- x and y values
- no input (returns [0., 0.])

7.1.5 Vector3Array

class vectormath.vector.**Vector3Array**
List of Vector3

A new Vector3Array can be created with:

- another Vector3Array
- x/y/z lists of equal length
- n x 3 array
- nothing (returns [[0., 0., 0.]])

cross(*vec*)
Cross product with another Vector3Array

dims

z
Array of z-component of vectors

7.1.6 Vector2Array

class vectormath.vector.**Vector2Array**
List of Vector2

A new Vector2Array can be created with:

- another Vector2Array
- x/y lists of equal length
- n x 2 array
- nothing (returns [[0., 0.]])

dims

CHAPTER 8

Indices and tables

- genindex
- modindex
- search