MTRandom

Generated by Doxygen 1.8.7

Thu May 15 2014 12:25:53

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Namespace Documentation

4.1 Package UMT

Classes

- class **ExponentialDistribution**

  Exponential distribution. FROM http://stackoverflow.com/questions/2106503/pseudorandom-number-genera
- class **GammaDistribution**

  Gamma distribution. Returns a deviate distributed as a gamma distribution of integer of order ia, i.e. the waiting time for the iath event in a Poisson process of unit mean Use _rand as a source of uniform deviates http://www.↩ nrbook.com/a/bookcpdf.php
- class MersenneTwister

  Generates pseudo-random numbers using the Mersenne Twister algorithm.
- class **NormalDistribution**

  Convert a Uniform Distribution to a Normal Distribution
- class **PoissonDistribution**

  Poisson distribution.
- class **PowerLaw**

  Power law.
- class **RandomCube**

  Random cube.
- class **RandomDisk**

  Random disk.
- class **RandomSphere**

  Random sphere.
- class **RandomSquare**

  Random square.
- class WaveToRgb

  Wave to rgb converter.

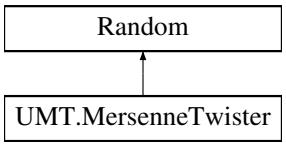
Chapter 5

Class Documentation

5.1 UMT.MersenneTwister Class Reference

Generates pseudo-random numbers using the Mersenne Twister algorithm.

Inheritance diagram for UMT.MersenneTwister:



Public Member Functions

- MersenneTwister (Int32 seed)

  Creates a new pseudo-random number generator with a given seed.
- MersenneTwister ()

  Creates a new pseudo-random number generator with a default seed.
- MersenneTwister (Int32[] initKey)

  Creates a pseudo-random number generator initialized with the given array.
- virtual UInt32 NextUInt32 ()

  Returns the next pseudo-random UInt32.
- virtual UInt32 NextUInt32 (UInt32 maxValue)

  Returns the next pseudo-random UInt32 up to maxValue .
- virtual UInt32 NextUInt32 (UInt32 minValue, UInt32 maxValue)

  Returns the next pseudo-random UInt32 at least minValue and up to maxValue .
- override Int32 Next ()

  Returns the next pseudo-random Int32.
- override Int32 Next (Int32 maxValue)

  Returns the next pseudo-random Int32 up to maxValue .
- override Int32 Next (Int32 minValue, Int32 maxValue)

  Returns the next pseudo-random Int32 at least minValue and up to maxValue .
- override void NextBytes (Byte[] buffer)

  Fills a buffer with pseudo-random bytes.
- override Double NextDouble ()

  Returns the next pseudo-random Double value.
- Double NextDouble (Boolean includeOne)

Returns a pseudo-random number greater than or equal to zero, and either strictly less than one, or less than or equal to one, depending on the value of the given parameter.

- Double NextDoublePositive ()

  Returns a pseudo-random number greater than 0.0 and less than 1.0.

- Single NextSingle ()

  Returns a pseudo-random number between 0.0 and 1.0.

- Single NextSingle (Boolean includeOne)

  Returns a pseudo-random number greater than or equal to zero, and either strictly less than one, or less than or equal to one, depending on the value of the given boolean parameter.

- Single NextSinglePositive ()

  Returns a pseudo-random number greater than 0.0 and less than 1.0.

Protected Member Functions

- UInt32 GenerateUInt32 ()

  Generates a new pseudo-random UInt32.

5.1.1 Detailed Description

Generates pseudo-random numbers using the Mersenne Twister algorithm.

See http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html for details on the algorithm.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 UMT.MersenneTwister.MersenneTwister (Int32 *seed*) [inline]

Creates a new pseudo-random number generator with a given seed.

Parameters

seed	A value to use as a seed.

5.1.2.2 UMT.MersenneTwister.MersenneTwister ( ) [inline]

Creates a new pseudo-random number generator with a default seed.

`new System.Random().Random.Next()` is used for the seed.

5.1.2.3 UMT.MersenneTwister.MersenneTwister (Int32[] *initKey*) [inline]

Creates a pseudo-random number generator initialized with the given array.

Parameters

initKey	The array for initializing keys.

5.1.3 Member Function Documentation

5.1.3.1 UInt32 UMT.MersenneTwister.GenerateUInt32 ( ) [inline], [protected]

Generates a new pseudo-random UInt32.

Returns

A pseudo-random UInt32.

5.1.3.2 override Int32 UMT.MersenneTwister.Next ( ) [inline]

Returns the next pseudo-random Int32.

Returns

A pseudo-random Int32 value.

5.1.3.3 override Int32 UMT.MersenneTwister.Next (Int32 *maxValue*) [inline]

Returns the next pseudo-random Int32 up to *maxValue* .

Parameters

maxValue	The maximum value of the pseudo-random number to create.

Returns

A pseudo-random Int32 value which is at most *maxValue* .

Exceptions

ArgumentOutOfRange↩ Exception	When *maxValue* < 0.

5.1.3.4 override Int32 UMT.MersenneTwister.Next (Int32 *minValue,* Int32 *maxValue*) [inline]

Returns the next pseudo-random Int32 at least *minValue* and up to *maxValue* .

Parameters

minValue	The minimum value of the pseudo-random number to create.
maxValue	The maximum value of the pseudo-random number to create.

Returns

A pseudo-random Int32 value which is at least *minValue* and at most *maxValue* .

Exceptions

ArgumentOutOfRange↩ Exception	If `minValue >= maxValue` .

5.1.3.5 override void UMT.MersenneTwister.NextBytes (Byte[] *buffer*) [inline]

Fills a buffer with pseudo-random bytes.

Parameters

buffer	The buffer to fill.

Exceptions

ArgumentNullException	If `buffer == .`

5.1.3.6 override Double UMT.MersenneTwister.NextDouble ( ) [inline]

Returns the next pseudo-random Double value.

Returns

A pseudo-random double floating point value.

There are two common ways to create a double floating point using MT19937: using GenerateUInt32 and dividing by 0xFFFFFFFF + 1, or else generating two double words and shifting the first by 26 bits and adding the second.

In a newer measurement of the randomness of MT19937 published in the journal "Monte Carlo Methods and Applications, Vol. 12, No. 5-6, pp. 385 393 (2006)" entitled "A Repetition Test for Pseudo-Random Number Generators", it was found that the 32-bit version of generating a double fails at the 95% confidence level when measuring for expected repetitions of a particular number in a sequence of numbers generated by the algorithm.

Due to this, the 53-bit method is implemented here and the 32-bit method of generating a double is not. If, for some reason, the 32-bit method is needed, it can be generated by the following:

```
(Double)NextUInt32() / ((UInt64)UInt32.MaxValue + 1);
```

5.1.3.7 Double UMT.MersenneTwister.NextDouble (Boolean *includeOne*) [inline]

Returns a pseudo-random number greater than or equal to zero, and either strictly less than one, or less than or equal to one, depending on the value of the given parameter.

Parameters

includeOne	If , the pseudo-random number returned will be less than or equal to one; otherwise, the pseudo-random number returned will be strictly less than one.

Returns

If *includeOne* is , this method returns a double-precision pseudo-random number greater than or equal to zero, and less than or equal to one. If *includeOne* is , this method returns a double-precision pseudo-random number greater than or equal to zero and strictly less than one.

5.1.3.8 Double UMT.MersenneTwister.NextDoublePositive ( ) [inline]

Returns a pseudo-random number greater than 0.0 and less than 1.0.

Returns

A pseudo-random number greater than 0.0 and less than 1.0.

5.1.3.9 Single UMT.MersenneTwister.NextSingle ( ) [inline]

Returns a pseudo-random number between 0.0 and 1.0.

Returns

A single-precision floating point number greater than or equal to 0.0, and less than 1.0.

5.1.3.10 Single UMT.MersenneTwister.NextSingle (Boolean *includeOne*) [inline]

Returns a pseudo-random number greater than or equal to zero, and either strictly less than one, or less than or equal to one, depending on the value of the given boolean parameter.

Parameters

includeOne	If , the pseudo-random number returned will be less than or equal to one; otherwise, the pseudo-random number returned will be strictly less than one.

Returns

If *includeOne* is , this method returns a single-precision pseudo-random number greater than or equal to zero, and less than or equal to one. If *includeOne* is , this method returns a single-precision pseudo-random number greater than or equal to zero and strictly less than one.

5.1.3.11 Single UMT.MersenneTwister.NextSinglePositive ( ) [inline]

Returns a pseudo-random number greater than 0.0 and less than 1.0.

Returns

A pseudo-random number greater than 0.0 and less than 1.0.

5.1.3.12 virtual UInt32 UMT.MersenneTwister.NextUInt32 ( ) [inline], [virtual]

Returns the next pseudo-random UInt32.

Returns

A pseudo-random UInt32 value.

5.1.3.13 virtual UInt32 UMT.MersenneTwister.NextUInt32 (UInt32 *maxValue*) [inline], [virtual]

Returns the next pseudo-random UInt32 up to *maxValue* .

Parameters

maxValue	The maximum value of the pseudo-random number to create.

Returns

A pseudo-random UInt32 value which is at most *maxValue* .

5.1.3.14 virtual UInt32 UMT.MersenneTwister.NextUInt32 (UInt32 *minValue,* UInt32 *maxValue*) [inline], [virtual]

Returns the next pseudo-random UInt32 at least *minValue* and up to *maxValue* .

Parameters

minValue	The minimum value of the pseudo-random number to create.
maxValue	The maximum value of the pseudo-random number to create.

Returns

A pseudo-random UInt32 value which is at least *minValue* and at most *maxValue* .

Exceptions

ArgumentOutOfRange↩ Exception	If `minValue >= maxValue` .

The documentation for this class was generated from the following file:

- /Users/tucano/Documents/Devel/Unity/UnityProjects/MTRandom/Assets/MTRandom/Scripts/lib/Mersenne↩ Twister.cs

5.2 MTRandom Class Reference

MT random main class.

Public Types

- enum **Normalization** { **STDNORMAL** = 0, **POWERLAW** = 1 }

Public Member Functions

- MTRandom ()

  Initializes a new instance of the MTRandom class.
- MTRandom (int seed)

  Initializes a new instance of the MTRandom class.
- MTRandom (string phrase)

  Initializes a new instance of the MTRandom class.
- float value ()

  Returns a pseudo-random number between 0.0 [inclusive] and 1.0 [inclusive] (Read Only).
- float value (bool includeOne)

  Returns a pseudo-random number greater than or equal to zero, and either strictly less than one, or less than or equal to one, depending on the value of the given boolean parameter.
- float valueNorm (float temperature)

  Returns a normalized pseudo-random number between 0.0 [inclusive] and 1.0 [inclusive] (Read Only).
- float valuePower (float temperature)

  Returns a power-law pseudo-random number between 0.0 [inclusive] and 1.0 [inclusive] (Read Only).
- float valuePoisson (float lambda)

  Returns a pseudo-random number in Poisson distribution between 0.0 [inclusive] and 1.0 [inclusive] (Read Only).
- float valueExponential (float lambda)

  Returns a pseudo-random number in Exponential distribution between 0.0 [inclusive] and 1.0 [inclusive] (Read Only).
- float valueGamma (int order)

  Returns a pseudo-random number on the gamma distribution.
- int Range (int min, int max)

  Returns the next pseudo-random number integer between min [inclusive] and max [inclusive].

- int Range (int min, int max, bool includeMax)

  Returns the next pseudo-random number integer between min [inclusive] and max [depend on includeMax].
- float Range (float min, float max)

  Returns the next pseudo-random number integer between min [inclusive] and max [inclusive].
- float RangeNorm (float min, float max, float temperature)

  Returns the next pseudo-random number integer between min [inclusive] and max [inclusive].
- float RangePower (float min, float max, float temperature)

  Returns the next pseudo-random number integer between min [inclusive] and max [inclusive]. in Power Law distribution
- Color color ()

  Return the next pseudo-random number as a Color
- Color color (float min, float max)

  Return the next pseudo-random number as a Color with a linear scale [0.0-1.0]. Use a range between min [inclusive] and max [inclusive] to reduce the color range.
- Vector2 PointInASquare ()

  pseudo-random number as a point in a square.
- Vector2 PointInASquare (Normalization n, float t)

  pseudo-random number as a point in a square.
- Vector2 PointInACircle ()

  pseudo-random number as a point in a circle.
- Vector2 PointInACircle (Normalization n, float t)

  pseudo-random number as a point in a circle.
- Vector2 PointInADisk ()

  pseudo-random number as a point in a disk.
- Vector2 PointInADisk (Normalization n, float t)

  pseudo-random number as a point in a disk.
- Vector3 PointInACube ()

  pseudo-random number as a point in a cube.
- Vector3 PointInACube (Normalization n, float t)

  pseudo-random number as a point in a cube.
- Vector3 PointOnACube ()

  pseudo-random number as a point on a cube surface.
- Vector3 PointOnACube (Normalization n, float t)

  pseudo-random number as a point on a cube surface.
- Vector3 PointOnASphere ()

  pseudo-random number as a point on a sphere surface.
- Vector3 PointInASphere ()

  pseudo-random number as a point in a sphere.
- Vector3 PointOnCap (float spotAngle)

  pseudo-random number as a point on a cap surface.
- Vector3 PointOnRing (float innerAngle, float outerAngle)

  pseudo-random number as a point on a ring surface.

Static Public Member Functions

- static float ScaleFloatToRange (float x, float newMin, float newMax, float oldMin, float oldMax)

  Scales the float to any range.

5.2.1 Detailed Description

MT random main class.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 MTRandom.MTRandom ( ) [inline]

Initializes a new instance of the MTRandom class.

5.2.2.2 MTRandom.MTRandom (int *seed*) [inline]

Initializes a new instance of the MTRandom class.

Parameters

seed	Seed (integer).

5.2.2.3 MTRandom.MTRandom (string *phrase*) [inline]

Initializes a new instance of the MTRandom class.

Parameters

phrase	Phrase (seed string).

5.2.3 Member Function Documentation

5.2.3.1 Color MTRandom.color ( ) [inline]

Return the next pseudo-random number as a Color

5.2.3.2 Color MTRandom.color (float *min*, float *max*) [inline]

Return the next pseudo-random number as a Color with a linear scale [0.0-1.0]. Use a range between *min* [inclusive] and *max* [inclusive] to reduce the color range.

Parameters

min	Minimum.
max	Max.

5.2.3.3 Vector2 MTRandom.PointInACircle ( ) [inline]

pseudo-random number as a point in a circle.

Returns

The in point as Vector2.

5.2.3.4 Vector2 MTRandom.PointInACircle (Normalization *n*, float *t*) [inline]

pseudo-random number as a point in a circle.

Returns

The in point as Vector2.

Parameters

n	Normalization type (STDNORMAL or POWERLAW).
t	Temperature.

5.2.3.5 Vector3 MTRandom.PointInACube ( ) [inline]

pseudo-random number as a point in a cube.

Returns

The in point as Vector3.

5.2.3.6 Vector3 MTRandom.PointInACube (Normalization *n*, float *t*) [inline]

pseudo-random number as a point in a cube.

Returns

The in point as Vector3.

Parameters

n	Normalization type (STDNORMAL or POWERLAW).
t	Temperature.

5.2.3.7 Vector2 MTRandom.PointInADisk ( ) [inline]

pseudo-random number as a point in a disk.

Returns

The in point as Vector2.

5.2.3.8 Vector2 MTRandom.PointInADisk (Normalization *n*, float *t*) [inline]

pseudo-random number as a point in a disk.

Returns

The in point as Vector2.

Parameters

n	Normalization type (STDNORMAL or POWERLAW).
t	Temperature.

5.2.3.9 Vector3 MTRandom.PointInASphere ( ) [inline]

pseudo-random number as a point in a sphere.

Returns

The in point as Vector3.

5.2.3.10 Vector2 MTRandom.PointInASquare ( ) [inline]

pseudo-random number as a point in a square.

Returns

The in point as Vector2.

5.2.3.11 Vector2 MTRandom.PointInASquare (Normalization *n*, float *t*) [inline]

pseudo-random number as a point in a square.

Returns

The in point as Vector2.

Parameters

n	Normalization type (STDNORMAL or POWERLAW).
t	Temperature.

5.2.3.12 Vector3 MTRandom.PointOnACube ( ) [inline]

pseudo-random number as a point on a cube surface.

Returns

The in point as Vector3.

5.2.3.13 Vector3 MTRandom.PointOnACube (Normalization *n*, float *t*) [inline]

pseudo-random number as a point on a cube surface.

Returns

The in point as Vector3.

Parameters

n	Normalization type (STDNORMAL or POWERLAW).
t	Temperature.

5.2.3.14 Vector3 MTRandom.PointOnASphere ( ) [inline]

pseudo-random number as a point on a sphere surface.

Returns

The in point as Vector3.

5.2.3.15 Vector3 MTRandom.PointOnCap (float *spotAngle*) [inline]

pseudo-random number as a point on a cap surface.

Returns

The in point as Vector3.

5.2.3.16 Vector3 MTRandom.PointOnRing (float *innerAngle,* float *outerAngle*) [inline]

pseudo-random number as a point on a ring surface.

Returns

The in point as Vector3.

5.2.3.17 int MTRandom.Range (int *min,* int *max*) [inline]

Returns the next pseudo-random number integer between *min* [inclusive] and *max* [inclusive].

Parameters

min	Minimum.
max	Maximum.

5.2.3.18 int MTRandom.Range (int *min,* int *max,* bool *includeMax*) [inline]

Returns the next pseudo-random number integer between *min* [inclusive] and *max* [depend on *includeMax*].

Parameters

min	Minimum.
max	Max.
includeMax	If set to `true` include *Max* .

5.2.3.19 float MTRandom.Range (float *min,* float *max*) [inline]

Returns the next pseudo-random number integer between *min* [inclusive] and *max* [inclusive].

Parameters

min	Minimum.
max	Maximum.

5.2.3.20 float MTRandom.RangeNorm (float *min,* float *max,* float *temperature*) [inline]

Returns the next pseudo-random number integer between *min* [inclusive] and *max* [inclusive].

Parameters

min	Minimum.
max	Maximum.
temperature	Temperature.

5.2.3.21 float MTRandom.RangePower (float *min,* float *max,* float *temperature*) [inline]

Returns the next pseudo-random number integer between *min* [inclusive] and *max* [inclusive]. in Power Law distribution

Parameters

min	Minimum.
max	Maximum.
temperature	Temperature.

5.2.3.22 static float MTRandom.ScaleFloatToRange (float *x,* float *newMin,* float *newMax,* float *oldMin,* float *oldMax*) [inline],[static]

Scales the float to any range.

Returns

The float to range.

Parameters

x	The x coordinate.
newMin	New minimum.
newMax	New max.
oldMin	Old minimum.
oldMax	Old max.

5.2.3.23 float MTRandom.value ( ) [inline]

Returns a pseudo-random number between 0.0 [inclusive] and 1.0 [inclusive] (Read Only).

Returns

This method returns a single-precision pseudo-random number greater than or equal to zero, and less than or equal to one.

5.2.3.24 float MTRandom.value (bool *includeOne*) [inline]

Returns a pseudo-random number greater than or equal to zero, and either strictly less than one, or less than or equal to one, depending on the value of the given boolean parameter.

Parameters

includeOne	If , the pseudo-random number returned will be less than or equal to one; otherwise, the pseudo-random number returned will be strictly less than one.

Returns

If *includeOne* is , this method returns a single-precision pseudo-random number greater than or equal to zero, and less than or equal to one. If *includeOne* is , this method returns a single-precision pseudo-random number greater than or equal to zero and strictly less than one.

5.2.3.25 float MTRandom.valueExponential (float *lambda*) [inline]

Returns a pseudo-random number in Exponential distribution between 0.0 [inclusive] and 1.0 [inclusive] (Read Only).

Returns

The value.

5.2.3.26 float MTRandom.valueGamma (int *order*) [inline]

Returns a pseudo-random number on the gamma distribution.

Returns

The gamma value.

Parameters

order	Order.

5.2.3.27 float MTRandom.valueNorm (float *temperature*) [inline]

Returns a normalized pseudo-random number between 0.0 [inclusive] and 1.0 [inclusive] (Read Only).

Returns

This method returns a single-precision pseudo-random number greater than or equal to zero, and less than or equal to one.

Parameters

temperature	Temperature.

5.2.3.28 float MTRandom.valuePoisson (float *lambda*) [inline]

Returns a pseudo-random number in Poisson distribution between 0.0 [inclusive] and 1.0 [inclusive] (Read Only).

Returns

The value.

5.2.3.29 float MTRandom.valuePower (float *temperature*) [inline]

Returns a power-law pseudo-random number between 0.0 [inclusive] and 1.0 [inclusive] (Read Only).

Returns

This method returns a single-precision pseudo-random number greater than or equal to zero, and less than or equal to one.

Parameters

temperature	Temperature.

The documentation for this class was generated from the following file:

- /Users/tucano/Documents/Devel/Unity/UnityProjects/MTRandom/Assets/MTRandom/Scripts/lib/MTRandom.↩ cs

5.3 UMT.WaveToRgb Class Reference

Wave to rgb converter.

Static Public Member Functions

- static Color **LinearToRgb** (float linearvalue)

Public Attributes

- const float **MinVisibleWaveLength** = 350.0f
- const float **MaxVisibleWaveLength** = 650.0f
- const float **Gamma** = 0.80f
- const int **IntesityMax** = 255

5.3.1 Detailed Description

Wave to rgb converter.

The documentation for this class was generated from the following file:

- /Users/tucano/Documents/Devel/Unity/UnityProjects/MTRandom/Assets/MTRandom/Scripts/lib/WaveTo↩ Rgb.cs