

DHG Signal Generator

Dark Horde Games

Jeffrey Cooper

Statement of Rights

This software is covered by the
GNU General Public License

Critical Files

(anything not listed below can be removed for your convenience)

DHGSIGNAL
DHGSIGNALGenerator
DHGSIGNALGeneratorEditor
Signals/DHGBetaSignal
Signals/DHGChiSqSignal
Signals/DHGConstantSignal
Signals/DHGExponentialSignal
Signals/DHGGammaSignal
Signals/DHGGaussianSignal
Signals/DHGHemisphericalSignal
Signals/DHGLogNormalSignal
Signals/DHGSineSignal
Signals/DHGSquareSignal
Signals/DHGTangentSignal
Signals/DHGTriangleSignal
Signals/DHGWeibullSignal

Description

This signal generator allows for users to programmatically generate complex digital wave signals during program execution. These signals are attached to game objects allowing for users to apply them as necessary for any signal based effect. These include, but are not limited too, statistical frequency based events (object spawning), geometry deformation (ie. wave or explosion ripples), dynamic sound generation (sirens etc), and more.

NOTE:

The associated statistical calculations are not scientifically accurate and should not be used for publishing level research. These approximations should be accurate enough for use in videos, games, and proof of concept operations.

Usage and Operation

- 1) Import DHG Signal Generator Script Files
- 2) Create your GameObject
- 3) Drag the DHGSignalGenerator Script onto your GameObject
- 4) Choose Limitless or Limited configuration.
- 4a) Limited configuration requires you configure how the signal is pre-generated
- 5) Add/Remove signals to get the desired number of signals.

This can be done any time during construction.

New signals are added to the end.

Removed signals are removed from the end.

If you wish to remove a signal in the middle, simply disable it instead of removing it.

- 6) Configure each signal

The unique properties of each signal require that you know what you are trying to achieve. Geometric Signals are defined by their Period

and Magnitude, while statistical signals each are defined by their own properties.

Google the signal to understand how the inputs affect the shape and domain.

It is not recommended to mix statistical and geometric signals as they can operate in drastically different time and magnitude domains.

That said, experiment and see what you can create.

- 7) Update your game object script to use the DHG Signal Generator using the following:

- a) Get the handle to the signal generator

```
DHGSignalGenerator sigGen = GetComponent<DHGSignalGenerator>();
```

- b) Query values

The format of the output query function is:

```
public float DHGSignalGenerator::GetSignal(float timeT)
```

```
float mySignalValue = sigGen.GetValue(timeT);
```

Design Documentation

Class DHGSignalGenerator

Control class for Signal operations. This allows for obfuscation of individual signal classes and uniform interaction with those signals. This also provides a single point of contact for the Unity Editor.

Inherits from: *MonoBehaviour (UnityEngine)*

Enumerations

None

Variables

public bool limitless - Flag indicating if value should be pre-calculated, and thus limited for where they can be queried.

public int numberOfSteps - Number of values to pre-calculate. Only used if [limitless] is set to false.

public float startTimeT - First time value to start all pre-calculations from. Only used if [limitless] is set to false.

public float timeStep - Signal time elapsed between samples for pre-calculated values. Only used if [limitless] is set to false.

public bool genSignalOnValidate - Flag indicating if the visualization and pre-calculation outputs should be regenerated every time a variable is updated. Only used if [limitless] is set to false.

public float visualizationScale - Not used in game execution, only for Editor operation. Allows for dynamic adjustment of x axis on the editor signal visualization bitmap.

public List<DHGSignal> signals - List of all signals used by this generator. Only public to allow DHGSignalGeneratorEditor access. Do not use.

public float[] signalOutput - Array used to store the pre-calculated signal values. Only used if [limitless] is set to false. Only public to allow DHGSignalGeneratorEditor access. Do not use.

private int numberOfSignals - Number of signals used by this signal generator. Internal use only to allow for dynamic addition and removal of signals through editor.

private bool editInitRequired - Flag indicating if the editor has run for the first time.

Functions

1) public float GetSignal(float timet)

This function queries the signals for the combined signal strength at the provided time.

THIS IS THE PRIMARY OUTPUT FUNCTION FOR THIS TOOL

Inputs	Outputs
Float timet - time value at which to query for the signal strength	float- value of the signal strength at the provided time

2) public void OnValidate()

This function checks the variables specific to signal generator for validation. Used by the editor to ensure controls do not exceed or violate signal rules. This also will adjust and allocate signal objects and, if enabled, generate the pre-calculation values.

Inputs	Outputs
none	none

3) public void Start()

This function validates all input values and then, if enabled, generates the pre-calculated signal table at program execution start.

Inputs	Outputs
none	none

4) public void AddSignal()

This function increments the internal count of number of desired signals, and then forces validation of all input values.

Inputs	Outputs
none	none

5) public void RemoveSignal()

This function decrements the internal count of number of desired signals, and then forces validation of all input values.

Inputs	Outputs
none	none

6) public void EditorInit()

This function forces validation of all provided values upon editor startup. In essence this forces calculation of the sample signal table every time the user switches between signal generator objects.

Inputs	Outputs
none	none

5) private float GetLocalSignalValue(float timet)

This function queries all enabled signals for their relative values at the provided time, and sums the outputs for the cumulative value.

Inputs	Outputs
Float timet - time value to query for signal strength.	Float - cumulative signal strength at provided time

5) public void GenerateSignal()

This function creates the pre-calculated table of signal values when enabled. [Limitless] signal generators will do nothing.

Inputs	Outputs
none	none

Class DHGSignal

Abstract base class for all Signals used by the DHGSignalGenerator. This class stores ALL variables used by the child classes to allow for proper serialization and Unity Editor access to those variables.

Inherits from: *ScriptableObject (UnityEngine)*

Enumerations

Public SignalMethod - List of plain text names for all supported signal types. This list should be updated to add any user created signals. All instances where this list is used in code should be updated to support instances for new user created signals. Associated numerical values are not relied upon, and should not be considered fixed.

CONSTANT
SINE
TANGENT
TRIANGLE
SQUARE
HEMISPHERE
GAUSSIAN
EXPONENTIAL
LOG_NORMAL
CHI_SQ
GAMMA
BETA
WEIBULL

Variables

public bool enabled - flag indicating if this signal is to be used in computation. Used by all signals.

public double magnitude - multiplicative numerical value for a signal to output. This value is multiplied by the normal output of a given signal to boost or decrease the resulting value. Used by all signals.

public SignalMethod signalMethod - enumerated value indicating what method this signal has been assigned to be. If the contained value does not match the class, then the signal is reassigned upon variable Validation. Used by all signals. Refer to DHGConstantSignal for example of how to properly use this variable in a user custom signal.

public double constantVal - Constant output value. Used by DHGConstantSignal.

public double period - Length of time, from 0 to this value (0,+), for a single instance of a periodic signal to occur. Used by DHGHemisphericalSignal, DHGSineSignal, DHGSquareSignal, DHGTangentSignal, and DHGTriangleSignal.

public double alpha - Shape descriptor for beta distribution [.5,+). Used by DHGBetaSignal.

public double beta - Shape descriptor for beta distribution [.5,+). Used by DHGBetaSignal.

public double degreesOfFreedom - Number of degrees of freedom [.5,+) for Chi Squared distribution. Used by DHGChiSqSignal.

public double lambda -Rate of Occurance [1,+) for Exponential Distribution. Used by DHGExponentialSignal.

public double shape -Shape Descriptor for Gamma and Weibull distributions [.5,+). Used by DHGGammaSignal, and DHGWeibullSignal.

public double scale -Scale parameter for Gamma and Weibull distributions [.5,+). Used by DHGGammaSignal, and DHGWeibullSignal.

public double mean -Stastical Mean (-,+). Used by DHGGaussianSignal and DHGLogNormalSignal.

public double standardDeviation - Statistical Standard Deviation (0,+). Used by DHGGaussianSignal and DHGLogNormalSignal.

Functions

1) abstract public double GetValue(float timet)

This function queries the signal for the signal strength at the provided time.

Inputs	Outputs
Float timet - time value to query the signal for	Double - value of the signal at the provided time

2) abstract public bool OnValidate()

This function checks the variables specific to this signal for validation. Used by the editor to ensure signal controls do not exceed or violate signal rules.

Inputs	Outputs
none	Bool - true if signalMethod is equal to expected value for this class, false if they do not match. This indicates the signal object must be replaced with the proper signal type.

Class DHGBetaSignal

Child class implementation for Beta Distribution.

Inherits from: **DHGSIGNAL**

Enumerations

none

Variables

none

Functions

1) public double GetValue(float timet)

This function queries the signal for the signal strength at the provided time.

Inputs	Outputs
Float timet - time value to query the signal for	Double - value of the signal at the provided time

2) public bool OnValidate()

This function checks the variables specific to this signal for validation. Used by the editor to ensure signal controls do not exceed or violate signal rules.

Inputs	Outputs
none	Bool - true if signalMethod is equal to expected value for this class, false if they do not match. This indicates the signal object must be replaced with the proper signal type.

Class DHGChiSqSignal

Child class implementation for Chi Squared Distribution.

Inherits from: **DHGSignal**

Enumerations

none

Variables

none

Functions

1) public double GetValue(float timet)

This function queries the signal for the signal strength at the provided time.

Inputs	Outputs
Float timet - time value to query the signal for	Double - value of the signal at the provided time

2) public bool OnValidate()

This function checks the variables specific to this signal for validation. Used by the editor to ensure signal controls do not exceed or violate signal rules.

Inputs	Outputs
none	Bool - true if signalMethod is equal to expected value for this class, false if they do not match. This indicates the signal object must be replaced with the proper signal type.

Class DHGConstantSignal

Child class implementation for a geometric Constant wave.

Inherits from: **DHGSignal**

Enumerations

none

Variables

none

Functions

1) public double GetValue(float timet)

This function queries the signal for the signal strength at the provided time.

Inputs	Outputs
Float timet - time value to query the signal for	Double - value of the signal at the provided time

2) public bool OnValidate()

This function checks the variables specific to this signal for validation. Used by the editor to ensure signal controls do not exceed or violate signal rules.

Inputs	Outputs
none	Bool - true if signalMethod is equal to expected value for this class, false if they do not match. This indicates the signal object must be replaced with the proper signal type.

Class DHGExponentialSignal

Child class implementation for Exponential (Lambda) Distribution.

Inherits from: **DHGSignal**

Enumerations

none

Variables

none

Functions

1) public double GetValue(float timet)

This function queries the signal for the signal strength at the provided time.

Inputs	Outputs
Float timet - time value to query the signal for	Double - value of the signal at the provided time

2) public bool OnValidate()

This function checks the variables specific to this signal for validation. Used by the editor to ensure signal controls do not exceed or violate signal rules.

Inputs	Outputs
none	Bool - true if signalMethod is equal to expected value for this class, false if they do not match. This indicates the signal object must be replaced with the proper signal type.

Class DHGGammaSignal

Child class implementation for Gamma Distribution.

Inherits from: **DHGSIGNAL**

Enumerations

none

Variables

none

Functions

1) public double GetValue(float timet)

This function queries the signal for the signal strength at the provided time.

Inputs	Outputs
Float timet - time value to query the signal for	Double - value of the signal at the provided time

2) public bool OnValidate()

This function checks the variables specific to this signal for validation. Used by the editor to ensure signal controls do not exceed or violate signal rules.

Inputs	Outputs
none	Bool - true if signalMethod is equal to expected value for this class, false if they do not match. This indicates the signal object must be replaced with the proper signal type.

Class DHGGaussianSignal

Child class implementation for Gaussian (normal) Distribution.

Inherits from: **DHGSignal**

Enumerations

none

Variables

none

Functions

1) public double GetValue(float timet)

This function queries the signal for the signal strength at the provided time.

Inputs	Outputs
Float timet - time value to query the signal for	Double - value of the signal at the provided time

2) public bool OnValidate()

This function checks the variables specific to this signal for validation. Used by the editor to ensure signal controls do not exceed or violate signal rules.

Inputs	Outputs
none	Bool - true if signalMethod is equal to expected value for this class, false if they do not match. This indicates the signal object must be replaced with the proper signal type.

Class DHGHemisphericalSignal

Child class implementation for a geometric Hemispherical Wave.

Inherits from: **DHGSignal**

Enumerations

none

Variables

none

Functions

1) public double GetValue(float timet)

This function queries the signal for the signal strength at the provided time.

Inputs	Outputs
Float timet - time value to query the signal for	Double - value of the signal at the provided time

2) public bool OnValidate()

This function checks the variables specific to this signal for validation. Used by the editor to ensure signal controls do not exceed or violate signal rules.

Inputs	Outputs
none	Bool - true if signalMethod is equal to expected value for this class, false if they do not match. This indicates the signal object must be replaced with the proper signal type.

Class DHGLogNormalSignal

Child class implementation for Log-Normal Distribution.

Inherits from: **DHGSignal**

Enumerations

none

Variables

none

Functions

1) public double GetValue(float timet)

This function queries the signal for the signal strength at the provided time.

Inputs	Outputs
Float timet - time value to query the signal for	Double - value of the signal at the provided time

2) public bool OnValidate()

This function checks the variables specific to this signal for validation. Used by the editor to ensure signal controls do not exceed or violate signal rules.

Inputs	Outputs
none	Bool - true if signalMethod is equal to expected value for this class, false if they do not match. This indicates the signal object must be replaced with the proper signal type.

Class DHGSineSignal

Child class implementation for a Trigonometric Sine calculation.

Inherits from: **DHGSignal**

Enumerations

none

Variables

none

Functions

1) public double GetValue(float timet)

This function queries the signal for the signal strength at the provided time.

Inputs	Outputs
Float timet - time value to query the signal for	Double - value of the signal at the provided time

2) public bool OnValidate()

This function checks the variables specific to this signal for validation. Used by the editor to ensure signal controls do not exceed or violate signal rules.

Inputs	Outputs
none	Bool - true if signalMethod is equal to expected value for this class, false if they do not match. This indicates the signal object must be replaced with the proper signal type.

Class DHGSquareSignal

Child class implementation for a geometric Square wave..

Inherits from: **DHGSignal**

Enumerations

none

Variables

none

Functions

1) public double GetValue(float timet)

This function queries the signal for the signal strength at the provided time.

Inputs	Outputs
Float timet - time value to query the signal for	Double - value of the signal at the provided time

2) public bool OnValidate()

This function checks the variables specific to this signal for validation. Used by the editor to ensure signal controls do not exceed or violate signal rules.

Inputs	Outputs
none	Bool - true if signalMethod is equal to expected value for this class, false if they do not match. This indicates the signal object must be replaced with the proper signal type.

Class DHGTangentSignal

Child class implementation for a Trigonometric Tangent calculation.

Inherits from: **DHGSignal**

Enumerations

none

Variables

none

Functions

1) public double GetValue(float timet)

This function queries the signal for the signal strength at the provided time.

Inputs	Outputs
Float timet - time value to query the signal for	Double - value of the signal at the provided time

2) public bool OnValidate()

This function checks the variables specific to this signal for validation. Used by the editor to ensure signal controls do not exceed or violate signal rules.

Inputs	Outputs
none	Bool - true if signalMethod is equal to expected value for this class, false if they do not match. This indicates the signal object must be replaced with the proper signal type.

Class DHGTriangleSignal

Child class implementation for a Geometric Triangle Wave.

Inherits from: **DHGSignal**

Enumerations

none

Variables

none

Functions

1) public double GetValue(float timet)

This function queries the signal for the signal strength at the provided time.

Inputs	Outputs
Float timet - time value to query the signal for	Double - value of the signal at the provided time

2) public bool OnValidate()

This function checks the variables specific to this signal for validation. Used by the editor to ensure signal controls do not exceed or violate signal rules.

Inputs	Outputs
none	Bool - true if signalMethod is equal to expected value for this class, false if they do not match. This indicates the signal object must be replaced with the proper signal type.

Class DHGWeibullSignal

Child class implementation for Weibull Distribution.

Inherits from: **DHGSignal**

Enumerations

none

Variables

none

Functions

1) public double GetValue(float timet)

This function queries the signal for the signal strength at the provided time.

Inputs	Outputs
Float timet - time value to query the signal for	Double - value of the signal at the provided time

2) public bool OnValidate()

This function checks the variables specific to this signal for validation. Used by the editor to ensure signal controls do not exceed or violate signal rules.

Inputs	Outputs
none	Bool - true if signalMethod is equal to expected value for this class, false if they do not match. This indicates the signal object must be replaced with the proper signal type.

Class DHGSignalGeneratorEditor

This class allows for custom Unity Editor window operation related to the DHGSignalGenerator script.

Inherits from: *Editor (UnityEditor)*

Variables

private Texture2D sampleSignalOutput - This texture is populated on the fly to provide the user with a graphical indication of how the signal they have generated will appear.

private List<bool> visible - Array of flags indicating if a given signal is expanded or shrunk in the editor window.

Functions

1) void OnEnable()

This function initializes the editor window object.

Inputs	Outputs
none	none

5) public void OnInspectorUpdate()

This function forces repaint when inspector changes are made. Used to support accordion operation of signal array editor display.

Inputs	Outputs
none	none

5) public override void OnInspectorGUI()

This function provides the custom behavior for the editor GUI.

Inputs	Outputs
none	none