

Sage Code Reference

1.6.10

Generated by Doxygen 1.7.6.1

Mon Oct 28 2013 21:34:34

Contents

1	Class Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	SageBaseController Class Reference	5
3.1.1	Detailed Description	8
3.1.2	Constructor & Destructor Documentation	8
3.1.2.1	SageBaseController	8
3.1.2.2	SageBaseController	8
3.1.3	Member Function Documentation	8
3.1.3.1	AddSageController< T >	9
3.1.3.2	Awake	9
3.1.3.3	DisplayRuntimeInfoGUI	9
3.1.3.4	ForceState	9
3.1.3.5	ForceState	10
3.1.3.6	ForceState	10
3.1.3.7	GetFloat	10
3.1.3.8	GetGlobalSpeed	11
3.1.3.9	GetStateMachineCurrentState	11
3.1.3.10	IsStateMachineActive	11
3.1.3.11	IsSuspended	12
3.1.3.12	OnDestroy	12
3.1.3.13	RecacheAnimations	12

3.1.3.14	RecreateLibrary	12
3.1.3.15	Reset	12
3.1.3.16	Resume	12
3.1.3.17	SetApplyMovementDelegate	12
3.1.3.18	SetFloat	13
3.1.3.19	SetFloat	13
3.1.3.20	SetGlobalSpeed	13
3.1.3.21	SetStateEnteredDelegate	14
3.1.3.22	SetStateExitedDelegate	14
3.1.3.23	SetStateUpdatedDelegate	14
3.1.3.24	StartStateMachine	15
3.1.3.25	StartStateMachine	15
3.1.3.26	StartStateMachine	15
3.1.3.27	StopAllStateMachines	16
3.1.3.28	StopAllStateMachines	16
3.1.3.29	StopStateMachine	16
3.1.3.30	StopStateMachine	17
3.1.3.31	Suspend	17
3.1.3.32	SuspendAndPlayAnimation	17
3.1.3.33	SuspendAndPlayAnimations	18
3.1.3.34	TransisionToState	18
3.1.3.35	TransisionToState	18
3.1.3.36	Update	19
3.1.4	Member Data Documentation	19
3.1.4.1	animationTarget	19
3.1.4.2	displayRuntimeInfo	19
3.1.4.3	libraryAsset	19
3.1.4.4	movementTarget	19
3.1.5	Property Documentation	19
3.1.5.1	LibraryRuntime	19
3.2	SageController Class Reference	20
3.2.1	Detailed Description	20
3.2.2	Member Function Documentation	20
3.2.2.1	OnGUI	20

3.3	SageTestingGUI Class Reference	20
3.3.1	Detailed Description	21
3.3.2	Member Data Documentation	21
3.3.2.1	depthOffset	21
3.3.2.2	horizontalOffset	21
3.3.2.3	rotation	21
3.3.2.4	targetController	21
3.3.2.5	verticalOffset	21
3.3.3	Property Documentation	22
3.3.3.1	Instance	22

Chapter 1

Class Index

1.1 Class Hierarchy

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

SageBaseController	5
SageController	20
SageTestingGUI	20

Chapter 2

Class Index

2.1 Class List

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

SageBaseController	This is the base Sage Controller class, where most of the functionality of the Sage Controller resides	5
SageController	This is the main class that controls a Sage Library. This is attached to any animated GameObject, and using the Sage Library, it will control all of the animations on the GameObject	20
SageTestingGUI	This class is used to test a Sage Library in game. Only one instance of this class can be active at a time	20

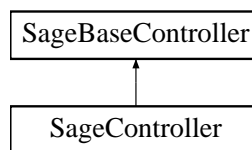
Chapter 3

Class Documentation

3.1 SageBaseController Class Reference

This is the base Sage Controller class, where most of the functionality of the Sage Controller resides.

Inheritance diagram for SageBaseController:



Public Member Functions

- [SageBaseController \(\)](#)
Default constructor.
- [SageBaseController \(ISageLibrary library\)](#)
This is the main constructor for a Sage Controller.
- virtual void [Reset \(\)](#)
This function is called when this object is reset in the editor.
- void [Awake \(\)](#)
This function is called when this script is awoken.
- void [OnDestroy \(\)](#)
This function is called when this script is destroyed.
- void [RecreateLibrary \(\)](#)
This will stop all current state machines, remove the current library, and then recreate everything.
- void [RecacheAnimations \(\)](#)

This will recache all animations that have been cached. This needs to be called if and when the Animation object has its animation clips changed dynamically.

- virtual void [Update](#) ()
This function is called when this script is updated.
- bool [SetGlobalSpeed](#) (float speed)
This function sets the global speed of this [SageController](#).
- float [GetGlobalSpeed](#) ()
This function gets the global speed of this [SageController](#).
- bool [SetFloat](#) (string variableName, float value)
This function sets a float variable on this [SageController](#).
- bool [SetFloat](#) (string variableName, float value, bool instant)
This function sets a float variable on this [SageController](#).
- float [GetFloat](#) (string variableName)
This function gets a float variable on this [SageController](#).
- bool [StartStateMachine](#) (string stateMachine)
This function starts a state machine on this [SageController](#).
- bool [StartStateMachine](#) (string stateMachine, float transitionTime)
This function starts a state machine on this [SageController](#).
- bool [StartStateMachine](#) (string stateMachine, float transitionTime, string initialState)
This function starts a state machine on this [SageController](#).
- bool [StopStateMachine](#) (string stateMachine)
This function stops a state machine on this [SageController](#).
- bool [StopStateMachine](#) (string stateMachine, float transitionTime)
This function stops a state machine on this [SageController](#).
- bool [StopAllStateMachines](#) ()
This function stops all state machines on this [SageController](#).
- bool [StopAllStateMachines](#) (float transitionTime)
This function stops all state machines on this [SageController](#).
- bool [IsSuspended](#) ()
This function checks if this [SageController](#) is currently suspended.
- bool [Suspend](#) (float outTime)
This function suspends this [SageController](#), which will pause the [SageController](#) and preserve all current states and active state machines.
- bool [SuspendAndPlayAnimation](#) (string animation, float speed, float outTime, float inTime)
This function suspends this [SageController](#), plays an animation, and then will automatically resume the [SageController](#).
- bool [SuspendAndPlayAnimations](#) (List< string > animations, float speed, float outTime, float crossBlendTime, float inTime)
This function suspends this [SageController](#), plays a list of animations, and then will automatically resume the [SageController](#).
- bool [Resume](#) (float inTime)
This function resumes this [SageController](#), if suspended, reactivating all previous activate states and state machines.

- bool [IsStateMachineActive](#) (string stateMachine)
This function checks if a state machine is currently activate on this [SageController](#).
- string [GetStateMachineCurrentState](#) (string stateMachine)
This function gets the current state of an activate state machine.
- bool [ForceState](#) (string stateMachine, string newState)
This functions forces a state on a state machine, ignoring normal transistion paths.
- bool [ForceState](#) (string stateMachine, string newState, float transitionTime)
This functions forces a state on a state machine, ignoring normal transistion paths.
- bool [ForceState](#) (string stateMachine, string newState, float transitionTime, bool allowStateRestart)
This functions forces a state on a state machine, ignoring normal transistion paths.
- bool [TransistionToState](#) (string stateMachine, string newState)
This function causes a state machine to transition from it's current state to a new state, using only normal transition paths.
- bool [TransistionToState](#) (string stateMachine, string newState, bool allowStateRestart)
This function causes a state machine to transition from it's current state to a new state, using only normal transition paths.
- bool [SetStateEnteredDelegate](#) (string stateMachine, string state, SageStateEnteredDelegate enteredDelegate)
This function sets the callback for entering a state in a specific state machine.
- bool [SetStateUpdatedDelegate](#) (string stateMachine, string state, SageStateUpdatedDelegate updatedDelegate)
This function sets the callback for updating a state in a specific state machine.
- bool [SetStateExitedDelegate](#) (string stateMachine, string state, SageStateExitedDelegate exitedDelegate)
This function sets the callback for exiting a state in a specific state machine.
- bool [SetApplyMovementDelegate](#) (SageApplyMovementDelegate applyMovementDelegate)
This sets a delegate function that should be called when movement should be applied.
- void [DisplayRuntimeInfoGUI](#) ()
This is the GUI function that display the runtime info of this Sage Controller if needed.

Static Public Member Functions

- static bool [AddSageController< T >](#) (GameObject gameObject, SageLibraryAsset libraryAsset, Animation animationTarget, GameObject movementTarget)
This helper function makes it possible to add a [SageController](#) (or a base class of it) to any game object at runtime, and to pass in itilization values. It makes it possible to set the libraryAsset, animationTarget, and movementTarget that should be used in that controller upon waking up.

Public Attributes

- SageLibraryAsset [libraryAsset](#) = null
This is the Sage Library Asset that this Sage Controller is using.
- Animation [animationTarget](#) = null
This is the Animation component that is the target for animation for this Sage - Controller. If this is null, it will default to looking for an Animation component on the same game object as the [SageController](#) component.
- GameObject [movementTarget](#) = null
This is the GameObject that is the target for world movement for this Sage Controller. If this is null, it will default to use the GameObject that this [SageController](#) component is attached to.
- bool [displayRuntimeInfo](#) = false
This will display the runtime info of this Sage Controller. Such as what graphs are active and animations are playing etc.

Properties

- ISageLibrary [LibraryRuntime](#) [get]
This gets the library runtime that is assigned to this Sage Controller. NOTE: USING THIS DIRECTLY IS AN ADVANCED OPTION, AND NOT FULLY SUPPORTED!

3.1.1 Detailed Description

This is the base Sage Controller class, where most of the functionality of the Sage Controller resides.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 SageBaseController.SageBaseController ()

Default constructor.

3.1.2.2 SageBaseController.SageBaseController (ISageLibrary library)

This is the main constructor for a Sage Controller.

Parameters

<i>library</i>	The library that this Sage Controller is using.
----------------	---

3.1.3 Member Function Documentation

3.1.3.1 `static bool SageBaseController.AddSageController< T > (GameObject gameObject, SageLibraryAsset libraryAsset, Animation animationTarget, GameObject movementTarget) [static]`

This helper function makes it possible to add a [SageController](#) (or a base class of it) to any game object at runtime, and to pass in initialization values. It makes it possible to set the `libraryAsset`, `animationTarget`, and `movementTarget` that should be used in that controller upon waking up.

Template Parameters

<i>T</i>	The type of SageController class to add as a component.
----------	---

Parameters

<i>gameObject</i>	The game object to add the component to.
<i>libraryAsset</i>	The library asset to assign to the controller.
<i>animation-Target</i>	The animation target to assign to the controller.
<i>movement-Target</i>	The movement target to assign to the controller.

Returns

Whether or not the controller was successfully added.

Type Constraints

T: [SageBaseController](#)

3.1.3.2 `void SageBaseController.Awake ()`

This function is called when this script is awoken.

3.1.3.3 `void SageBaseController.DisplayRuntimeInfoGUI ()`

This is the GUI function that display the runtime info of this Sage Controller if needed.

3.1.3.4 `bool SageBaseController.ForceState (string stateMachine, string newState)`

This functions forces a state on a state machine, ignoring normal transistion paths.

Parameters

<i>state-Machine</i>	The state machine to access.
<i>newState</i>	The new state to force on the state machine.

Returns

Returns whether or not this function was executed successfully.

3.1.3.5 bool SageBaseController.ForceState (string *stateMachine*, string *newState*, float *transitionTime*)

This functions forces a state on a state machine, ignoring normal transistion paths.

Parameters

<i>state-Machine</i>	The state machine to access.
<i>newState</i>	The new state to force on the state machine.
<i>transition-Time</i>	This is the amount of time (in seconds) it should take to transistion into this state.

Returns

Returns whether or not this function was executed successfully.

3.1.3.6 bool SageBaseController.ForceState (string *stateMachine*, string *newState*, float *transitionTime*, bool *allowStateRestart*)

This functions forces a state on a state machine, ignoring normal transistion paths.

Parameters

<i>state-Machine</i>	The state machine to access.
<i>newState</i>	The new state to force on the state machine.
<i>transition-Time</i>	This is the amount of time (in seconds) it should take to transistion into this state.
<i>allowState-Restart</i>	This indicates if when trying to transition into the same state, that the state should be allowed to start over from the beginning.

Returns

Returns whether or not this function was executed successfully.

3.1.3.7 float SageBaseController.GetFloat (string *variableName*)

This function gets a float variable on this [SageController](#).

Parameters

<i>variable-Name</i>	The name of the variable to get.
----------------------	----------------------------------

Returns

Returns the float value of the variable.

3.1.3.8 float SageBaseController.GetGlobalSpeed ()

This function gets the global speed of this [SageController](#).

Returns

Returns the global speed.

3.1.3.9 string SageBaseController.GetStateMachineCurrentState (string *stateMachine*)

This function gets the current state of an activate state machine.

Parameters

<i>state-Machine</i>	The state machine to get the state of.
----------------------	--

Returns

Returns the current state of the state machine.

3.1.3.10 bool SageBaseController.IsStateMachineActive (string *stateMachine*)

This function checks if a state machine is currently activate on this [SageController](#).

Parameters

<i>state-Machine</i>	The state machine to check.
----------------------	-----------------------------

Returns

Returns if the requested state machine is currently active.

3.1.3.11 bool SageBaseController.IsSuspended ()

This function checks if this [SageController](#) is currently suspended.

Returns

Returns if this [SageController](#) is suspended.

3.1.3.12 void SageBaseController.OnDestroy ()

This function is called when this script is destroyed.

3.1.3.13 void SageBaseController.RecacheAnimations ()

This will recache all animations that have been cached. This needs to be called if and when the Animation object has its animation clips changed dynamically.

3.1.3.14 void SageBaseController.RecreateLibrary ()

This will stop all current state machines, remove the current library, and then recreate everything.

3.1.3.15 virtual void SageBaseController.Reset () [virtual]

This function is called when this object is reset in the editor.

3.1.3.16 bool SageBaseController.Resume (float *inTime*)

This function resumes this [SageController](#), if suspended, reactivating all previous activate states and state machines.

Parameters

<i>inTime</i>	This is how long (in seconds) it should take to resume this SageController .
---------------	--

Returns

Returns whether or not this function was executed successfully.

3.1.3.17 bool SageBaseController.SetApplyMovementDelegate (SageApplyMovementDelegate *applyMovementDelegate*)

This sets a delegate function that should be called when movement should be applied.

Parameters

<i>apply-Movement-Delegate</i>	The function to call.
--------------------------------	-----------------------

Returns

Whether or not the delegate function was set properly.

3.1.3.18 `bool SageBaseController.SetFloat (string variableName, float value)`

This function sets a float variable on this [SageController](#).

Parameters

<i>variable-Name</i>	The name of the variable to be set.
<i>value</i>	The value to set the variable to.

Returns

Returns whether or not this function was executed successfully.

3.1.3.19 `bool SageBaseController.SetFloat (string variableName, float value, bool instant)`

This function sets a float variable on this [SageController](#).

Parameters

<i>variable-Name</i>	The name of the variable to be set.
<i>value</i>	The value to set the variable to.
<i>instant</i>	This indicates if this value should be set instantly, or if it should enforce any change speed restrictions present on the variable.

Returns

Returns whether or not this function was executed successfully.

3.1.3.20 `bool SageBaseController.SetGlobalSpeed (float speed)`

This function sets the global speed of this [SageController](#).

Parameters

<i>speed</i>	The speed to set it to.
--------------	-------------------------

Returns

Returns whether or not this function was executed successfully.

3.1.3.21 `bool SageBaseController.SetStateEnteredDelegate (string stateMachine, string state, SageStateEnteredDelegate enteredDelegate)`

This function sets the callback for entering a state in a specific state machine.

Parameters

<i>state-Machine</i>	The state machine to find the state in.
<i>state</i>	The state to set the entered delegate on.
<i>entered-Delegate</i>	The entered delegate to set. Passing in null clears any existing delegate.

Returns

Whether or not the entered delegate was set successfully.

3.1.3.22 `bool SageBaseController.SetStateExitedDelegate (string stateMachine, string state, SageStateExitedDelegate exitedDelegate)`

This function sets the callback for exiting a state in a specific state machine.

Parameters

<i>state-Machine</i>	The state machine to find the state in.
<i>state</i>	The state to set the exited delegate on.
<i>exited-Delegate</i>	The exited delegate to set. Passing in null clears any existing delegate.

Returns

Whether or not the exited delegate was set successfully.

3.1.3.23 `bool SageBaseController.SetStateUpdatedDelegate (string stateMachine, string state, SageStateUpdatedDelegate updatedDelegate)`

This function sets the callback for updating a state in a specific state machine.

Parameters

<i>state-Machine</i>	The state machine to find the state in.
<i>state</i>	The state to set the updated delegate on.
<i>updated-Delegate</i>	The updated delegate to set. Passing in null clears any existing delegate.

Returns

Whether or not the updated delegate was set successfully.

3.1.3.24 `bool SageBaseController.StartStateMachine (string stateMachine)`

This function starts a state machine on this [SageController](#).

Parameters

<i>state-Machine</i>	The name of the state machine to start.
----------------------	---

Returns

Returns whether or not this function was executed successfully.

3.1.3.25 `bool SageBaseController.StartStateMachine (string stateMachine, float transitionTime)`

This function starts a state machine on this [SageController](#).

Parameters

<i>state-Machine</i>	The name of the state machine to start.
<i>transition-Time</i>	This is how long it should take to fully start the state machine.

Returns

Returns whether or not this function was executed successfully.

3.1.3.26 `bool SageBaseController.StartStateMachine (string stateMachine, float transitionTime, string initialState)`

This function starts a state machine on this [SageController](#).

Parameters

<i>state-Machine</i>	The name of the state machine to start.
<i>transition-Time</i>	This is how long (in seconds) it should take to fully start the state machine.
<i>initialState</i>	The name of the initial state to start this state machine in.

Returns

Returns whether or not this function was executed successfully.

3.1.3.27 `bool SageBaseController.StopAllStateMachines ()`

This function stops all state machines on this [SageController](#).

Returns

Returns whether or not this function was executed successfully.

3.1.3.28 `bool SageBaseController.StopAllStateMachines (float transitionTime)`

This function stops all state machines on this [SageController](#).

Parameters

<i>transition-Time</i>	This is how long (in seconds) it should take to fully stop all of the state machine.
------------------------	--

Returns

Returns whether or not this function was executed successfully.

3.1.3.29 `bool SageBaseController.StopStateMachine (string stateMachine)`

This function stops a state machine on this [SageController](#).

Parameters

<i>state-Machine</i>	The name of the state machine to stop.
----------------------	--

Returns

Returns whether or not this function was executed successfully.

3.1.3.30 `bool SageBaseController.StopStateMachine (string stateMachine, float transitionTime)`

This function stops a state machine on this [SageController](#).

Parameters

<i>state-Machine</i>	The name of the state machine to stop.
<i>transition-Time</i>	This is how long (in seconds) it should take to fully stop the state machine.

Returns

Returns whether or not this function was executed successfully.

3.1.3.31 `bool SageBaseController.Suspend (float outTime)`

This function suspends this [SageController](#), which will pause the [SageController](#) and preserve all current states and active state machines.

Parameters

<i>outTime</i>	This is how long (in seconds) it should take to suspend this SageController .
----------------	---

Returns

Returns whether or not this function was executed successfully.

3.1.3.32 `bool SageBaseController.SuspendAndPlayAnimation (string animation, float speed, float outTime, float inTime)`

This function suspends this [SageController](#), plays an animation, and then will automatically resume the [SageController](#).

Parameters

<i>animation</i>	The animation to play.
<i>speed</i>	The speed to playback the animation at.
<i>outTime</i>	This is how long (in seconds) it should take to suspend and blend into this animation.
<i>inTime</i>	This is how long (in seconds) it should take to resume and blend out of this animation.

Returns

Returns whether or not this function was executed successfully.

3.1.3.33 bool SageBaseController.SuspendAndPlayAnimations (List< string > animations, float speed, float outTime, float crossBlendTime, float inTime)

This function suspends this [SageController](#), plays a list of animations, and then will automatically resume the [SageController](#).

Parameters

<i>animations</i>	The list of animations to play, in order.
<i>speed</i>	The speed to playback the animations at.
<i>outTime</i>	This is how long (in seconds) it should take to suspend and blend into these animations.
<i>crossBlend-Time</i>	This is how long (in seconds) it should take to blend between animations in the list.
<i>inTime</i>	This is how long (in seconds) it should take to resume and blend out of these animations.

Returns

Returns whether or not this function was executed successfully.

3.1.3.34 bool SageBaseController.TransistionToState (string stateMachine, string newState)

This function causes a state machine to transition from it's current state to a new state, using only normal transition paths.

Parameters

<i>state-Machine</i>	The state machine to access.
<i>newState</i>	The new state to transistion to on the state machine.

Returns

Returns whether or not this function was executed successfully.

3.1.3.35 bool SageBaseController.TransistionToState (string stateMachine, string newState, bool allowStateRestart)

This function causes a state machine to transition from it's current state to a new state, using only normal transition paths.

Parameters

<i>state-Machine</i>	The state machine to access.
<i>newState</i>	The new state to transition to on the state machine.
<i>allowState-Restart</i>	This indicates if when trying to transition into the same state, that the state should be allowed to start over from the beginning.

Returns

Returns whether or not this function was executed successfully.

3.1.3.36 virtual void SageBaseController.Update () [virtual]

This function is called when this script is updated.

3.1.4 Member Data Documentation

3.1.4.1 Animation SageBaseController.animationTarget = null

This is the Animation component that is the target for animation for this Sage Controller. If this is null, it will default to looking for an Animation component on the same game object as the SageController component.

3.1.4.2 bool SageBaseController.displayRuntimeInfo = false

This will display the runtime info of this Sage Controller. Such as what graphs are active and animations are playing etc.

3.1.4.3 SageLibraryAsset SageBaseController.libraryAsset = null

This is the Sage Library Asset that this Sage Controller is using.

3.1.4.4 GameObject SageBaseController.movementTarget = null

This is the GameObject that is the target for world movement for this Sage Controller. If this is null, it will default to use the GameObject that this SageController component is attached to.

3.1.5 Property Documentation

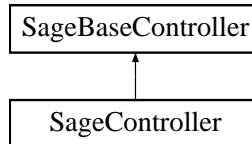
3.1.5.1 ISageLibrary SageBaseController.LibraryRuntime [get]

This gets the library runtime that is assigned to this Sage Controller. NOTE: USING THIS DIRECTLY IS AN ADVANCED OPTION, AND NOT FULLY SUPPORTED!

3.2 SageController Class Reference

This is the main class that controls a Sage Library. This is attached to any animated `GameObject`, and using the Sage Library, it will control all of the animations on the `GameObject`.

Inheritance diagram for SageController:



Public Member Functions

- virtual void [OnGUI](#) ()

This is the GUI function that display the runtime info of this Sage Controller if needed. This function can be commented out if you do not wish the [OnGUI\(\)](#) function to ever be called on your game controller object.

3.2.1 Detailed Description

This is the main class that controls a Sage Library. This is attached to any animated `GameObject`, and using the Sage Library, it will control all of the animations on the `GameObject`.

3.2.2 Member Function Documentation

3.2.2.1 virtual void SageController.OnGUI() [virtual]

This is the GUI function that display the runtime info of this Sage Controller if needed. This function can be commented out if you do not wish the [OnGUI\(\)](#) function to ever be called on your game controller object.

3.3 SageTestingGUI Class Reference

This class is used to test a Sage Library in game. Only one instance of this class can be active at a time.

Public Attributes

- [SageBaseController](#) `targetController` = null

This is the target controller that this Sage Testing GUI should be testing.

- float `depthOffset` = 2.0f

This is how much to offset the target controller away from the main camera.

- float `horizontalOffset` = 0.0f

This is how much to offset the target controller to the right of the main camera.

- float `verticalOffset` = 0.75f

This is how much to offset the target controller vertically from the main camera.

- float `rotation` = 270.0f

This is the number of degrees to rotate the main camera around the target controller.

Properties

- static `SageTestingGUI Instance` [get]

This returns the current instance of the Sage Testing GUI.

3.3.1 Detailed Description

This class is used to test a Sage Library in game. Only one instance of this class can be active at a time.

3.3.2 Member Data Documentation

3.3.2.1 float SageTestingGUI.depthOffset = 2.0f

This is how much to offset the target controller away from the main camera.

3.3.2.2 float SageTestingGUI.horizontalOffset = 0.0f

This is how much to offset the target controller to the right of the main camera.

3.3.2.3 float SageTestingGUI.rotation = 270.0f

This is the number of degrees to rotate the main camera around the target controller.

3.3.2.4 SageBaseController SageTestingGUI.targetController = null

This is the target controller that this Sage Testing GUI should be testing.

3.3.2.5 float SageTestingGUI.verticalOffset = 0.75f

This is how much to offset the target controller vertically from the main camera.

3.3.3 Property Documentation

3.3.3.1 SageTestingGUI SageTestingGUI.Instance [static, get]

This returns the current instance of the Sage Testing GUI.

Index

- AddSageController< T >
 - SageBaseController, 8
- Awake
 - SageBaseController, 9
- DisplayRuntimeInfoGUI
 - SageBaseController, 9
- ForceState
 - SageBaseController, 9, 10
- GetFloat
 - SageBaseController, 10
- GetGlobalSpeed
 - SageBaseController, 11
- GetStateMachineCurrentState
 - SageBaseController, 11
- Instance
 - SageTestingGUI, 22
- IsStateMachineActive
 - SageBaseController, 11
- IsSuspended
 - SageBaseController, 11
- LibraryRuntime
 - SageBaseController, 19
- OnDestroy
 - SageBaseController, 12
- OnGUI
 - SageController, 20
- RecacheAnimations
 - SageBaseController, 12
- RecreateLibrary
 - SageBaseController, 12
- Reset
 - SageBaseController, 12
- Resume
 - SageBaseController, 12
- SageBaseController, 5
 - AddSageController< T >, 8
 - Awake, 9
 - DisplayRuntimeInfoGUI, 9
 - ForceState, 9, 10
 - GetFloat, 10
 - GetGlobalSpeed, 11
 - GetStateMachineCurrentState, 11
 - IsStateMachineActive, 11
 - IsSuspended, 11
 - LibraryRuntime, 19
 - OnDestroy, 12
 - RecacheAnimations, 12
 - RecreateLibrary, 12
 - Reset, 12
 - Resume, 12
 - SageBaseController, 8
 - SetApplyMovementDelegate, 12
 - SetFloat, 13
 - SetGlobalSpeed, 13
 - SetStateEnteredDelegate, 14
 - SetStateExitedDelegate, 14
 - SetStateUpdatedDelegate, 14
 - StartStateMachine, 15
 - StopAllStateMachines, 16
 - StopStateMachine, 16
 - Suspend, 17
 - SuspendAndPlayAnimation, 17
 - SuspendAndPlayAnimations, 18
 - TransitionToState, 18
 - Update, 19
 - animationTarget, 19
 - displayRuntimeInfo, 19
 - libraryAsset, 19
 - movementTarget, 19
 - SageBaseController, 8
- SageController, 20
 - OnGUI, 20
- SageTestingGUI, 20
 - Instance, 22
 - depthOffset, 21
 - horizontalOffset, 21
 - rotation, 21
 - targetController, 21
 - verticalOffset, 21
- SetApplyMovementDelegate
 - SageBaseController, 12
- SetFloat

- SageBaseController, [13](#)
- SetGlobalSpeed
 - SageBaseController, [13](#)
- SetStateEnteredDelegate
 - SageBaseController, [14](#)
- SetStateExitedDelegate
 - SageBaseController, [14](#)
- SetStateUpdatedDelegate
 - SageBaseController, [14](#)
- StartStateMachine
 - SageBaseController, [15](#)
- StopAllStateMachines
 - SageBaseController, [16](#)
- StopStateMachine
 - SageBaseController, [16](#)
- Suspend
 - SageBaseController, [17](#)
- SuspendAndPlayAnimation
 - SageBaseController, [17](#)
- SuspendAndPlayAnimations
 - SageBaseController, [18](#)
- TransisionToState
 - SageBaseController, [18](#)
- Update
 - SageBaseController, [19](#)

- animationTarget
 - SageBaseController, [19](#)

- depthOffset
 - SageTestingGUI, [21](#)
- displayRuntimeInfo
 - SageBaseController, [19](#)

- horizontalOffset
 - SageTestingGUI, [21](#)

- libraryAsset
 - SageBaseController, [19](#)

- movementTarget
 - SageBaseController, [19](#)

- rotation
 - SageTestingGUI, [21](#)

- targetController
 - SageTestingGUI, [21](#)

- verticalOffset
 - SageTestingGUI, [21](#)