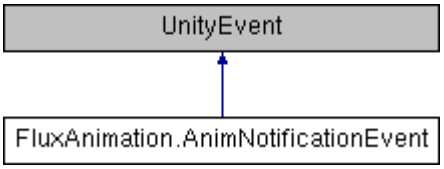
Flux UI Documentation

Class Documentation

FluxAnimation.AnimNotificationEvent Class Reference

Inheritance diagram for FluxAnimation.AnimNotificationEvent:



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

- **FluxAnimation.cs**

FluxAnimation.EquationFunctionData Struct Reference

Public Attributes

- **EquationFunction XOpenFunction**
- **EquationFunction YOpenFunction**
- **EquationFunction ZOpenFunction**
- **EquationFunction XCloseFunction**
- **EquationFunction YCloseFunction**
- **EquationFunction ZCloseFunction**

Member Data Documentation

EquationFunction FluxAnimation.EquationFunctionData.XCloseFunction

EquationFunction FluxAnimation.EquationFunctionData.XOpenFunction

EquationFunction FluxAnimation.EquationFunctionData.YCloseFunction

EquationFunction FluxAnimation.EquationFunctionData.YOpenFunction

EquationFunction FluxAnimation.EquationFunctionData.ZCloseFunction

EquationFunction FluxAnimation.EquationFunctionData.ZOpenFunction

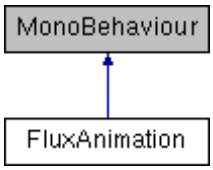
---

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

- **FluxAnimation.cs**

FluxAnimation Class Reference

Inheritance diagram for FluxAnimation:



Classes

- class **AnimNotificationEvent**
- struct **EquationFunctionData**
- class **FluxAnimationData**
- class **FluxColorData**
- class **VCAlphaData**
- class **VCAlphaInfo**
- class **VCAnimationInfo**
- class **VCColorInfo**

Public Types

- enum **VCTimeMode** { **VCTimeMode.ScaledTime**, **VCTimeMode.UnscaledTime** }
- enum **VCAnimationDirection** { **VCAnimationDirection.Forward**, **VCAnimationDirection.Reverse** }
- enum **VCAnimationState** { **VCAnimationState.AnimStarted**, **VCAnimationState.AnimFinished** }
- enum **VCAnimationMethod** { **VCAnimationMethod.AnimFunction**, **VCAnimationMethod.AnimCurve** }
- enum **VCColorComponentType** { **VCColorComponentType.None**, **VCColorComponentType.SpriteRenderer**, **VCColorComponentType.Renderer**, **VCColorComponentType.Image**, **VCColorComponentType.Text** }
- enum **Equations** { **Equations.None**, **Equations.Linear**, **Equations.QuadEaseOut**, **Equations.QuadEaseIn**, **Equations.QuadEaseInOut**, **Equations.QuadEaseOutIn**, **Equations.ExpoEaseOut**, **Equations.ExpoEaseIn**, **Equations.ExpoEaseInOut**, **Equations.ExpoEaseOutIn**, **Equations.CubicEaseOut**, **Equations.CubicEaseIn**, **Equations.CubicEaseInOut**, **Equations.CubicEaseOutIn**, **Equations.QuartEaseOut**, **Equations.QuartEaseIn**, **Equations.QuartEaseInOut**, **Equations.QuartEaseOutIn**, **Equations.QuintEaseOut**, **Equations.QuintEaseIn**, **Equations.QuintEaseInOut**, **Equations.QuintEaseOutIn**, **Equations.CircEaseOut**, **Equations.CircEaseIn**, **Equations.CircEaseInOut**, **Equations.CircEaseOutIn**, **Equations.SineEaseOut**, **Equations.SineEaseIn**, **Equations.SineEaseInOut**, **Equations.SineEaseOutIn**, **Equations.ElasticEaseOut**, **Equations.ElasticEaseIn**, **Equations.ElasticEaseInOut**, **Equations.ElasticEaseOutIn**, **Equations.BounceEaseOut**, **Equations.BounceEaseIn**, **Equations.BounceEaseInOut**, **Equations.BounceEaseOutIn**, **Equations.BackEaseOut**, **Equations.BackEaseIn**, **Equations.BackEaseInOut**, **Equations.BackEaseOutIn** }*Enumeration of all easing equations.*

Public Member Functions

- delegate float **EquationFunction** (float t, float b, float c, float d)
- void **StartOpeningAnimation** ()
  Starts All the Opening Animations
- void **StartClosingAnimation** ()
  Starts all the closing animations
- void **StopAnimation** ()
  Stops All the Open Animations
- void **OnAnimationFinished** ()

Calls Automatically when each Animation is finished

- void **StartMovementOpenAnimation** ()
  Starts Movement Open Animation
- void **StartScaleOpenAnimation** ()
  Starts Scale Open Animation
- void **StartRotationOpenAnimation** ()
  Starts Rotation Open Animation
- void **StartAlphaOpenAnimation** ()
  Starts Alpha Open Animation
- void **StartColorOpenAnimation** ()
  Starts Color Open Animation
- void **StartMovementCloseAnimation** ()
  Starts Movement Close Animation
- void **StartScaleCloseAnimation** ()
  Starts Scale Close Animation
- void **StartRotationCloseAnimation** ()
  Starts Rotation Close Animation
- void **StartAlphaCloseAnimation** ()
  Starts Alpha Close Animation
- void **StartColorCloseAnimation** ()
  Starts Color Close Animation
- void **StopMovementAnimation** ()
  Stop Movement Animation
- void **StopScaleAnimation** ()
  Stop Scale Animation
- void **StopRotationAnimation** ()
  Stop Rotation Animation
- void **StopAlphaAnimation** ()
  Stop Alpha Animation
- void **StopColorAnimation** ()
  Stop Color Animation
- IEnumerator **MoveFunc** (bool forward)
  Movement Enumerator Function
- IEnumerator **ScaleFunc** (bool forward)
  Scale Enumerator Function
- IEnumerator **RotationFunc** (bool forward)
  Rotation Enumerator Function
- IEnumerator **AlphaFunc** (bool forward)
  Alpha Enumerator Function
- IEnumerator **ColorFunc** (bool forward)
  Color Enumerator Function
- IEnumerator **WaitForRealSeconds** (float time)
  Wait for Seconds Based on the Time Mode
- float **GetDeltaTime** ()
  Returns the delta time based on Current Time Mode

Static Public Member Functions

- static float **Linear** (float t, float b, float c, float d)
  Easing equation function for a simple linear tweening, with no easing.
- static float **ExpoEaseOut** (float t, float b, float c, float d)
  Easing equation function for an exponential (2^t) easing out: decelerating from zero velocity.
- static float **ExpoEaseIn** (float t, float b, float c, float d)
  Easing equation function for an exponential (2^t) easing in: accelerating from zero velocity.
- static float **ExpoEaseInOut** (float t, float b, float c, float d)
  Easing equation function for an exponential (2^t) easing in/out: acceleration until halfway, then deceleration.
- static float **ExpoEaseOutIn** (float t, float b, float c, float d)
  Easing equation function for an exponential (2^t) easing out/in: deceleration until halfway, then acceleration.
- static float **CircEaseOut** (float t, float b, float c, float d)
  Easing equation function for a circular (sqrt(1-t^2)) easing out: decelerating from zero velocity.
- static float **CircEaseIn** (float t, float b, float c, float d)
  Easing equation function for a circular (sqrt(1-t^2)) easing in: accelerating from zero velocity.
- static float **CircEaseInOut** (float t, float b, float c, float d)
  Easing equation function for a circular (sqrt(1-t^2)) easing in/out: acceleration until halfway, then deceleration.
- static float **CircEaseOutIn** (float t, float b, float c, float d)
  Easing equation function for a circular (sqrt(1-t^2)) easing in/out: acceleration until halfway, then deceleration.
- static float **QuadEaseOut** (float t, float b, float c, float d)
  Easing equation function for a quadratic (t^2) easing out: decelerating from zero velocity.
- static float **QuadEaseIn** (float t, float b, float c, float d)
  Easing equation function for a quadratic (t^2) easing in: accelerating from zero velocity.
- static float **QuadEaseInOut** (float t, float b, float c, float d)
  Easing equation function for a quadratic (t^2) easing in/out: acceleration until halfway, then deceleration.
- static float **QuadEaseOutIn** (float t, float b, float c, float d)
  Easing equation function for a quadratic (t^2) easing out/in: deceleration until halfway, then acceleration.
- static float **SineEaseOut** (float t, float b, float c, float d)
  Easing equation function for a sinusoidal (sin(t)) easing out: decelerating from zero velocity.
- static float **SineEaseIn** (float t, float b, float c, float d)
  Easing equation function for a sinusoidal (sin(t)) easing in: accelerating from zero velocity.
- static float **SineEaseInOut** (float t, float b, float c, float d)
  Easing equation function for a sinusoidal (sin(t)) easing in/out: acceleration until halfway, then deceleration.
- static float **SineEaseOutIn** (float t, float b, float c, float d)
  Easing equation function for a sinusoidal (sin(t)) easing in/out: deceleration until halfway, then acceleration.
- static float **CubicEaseOut** (float t, float b, float c, float d)
  Easing equation function for a cubic (t^3) easing out: decelerating from zero velocity.
- static float **CubicEaseIn** (float t, float b, float c, float d)
  Easing equation function for a cubic (t^3) easing in: accelerating from zero velocity.
- static float **CubicEaseInOut** (float t, float b, float c, float d)
  Easing equation function for a cubic (t^3) easing in/out: acceleration until halfway, then deceleration.
- static float **CubicEaseOutIn** (float t, float b, float c, float d)
  Easing equation function for a cubic (t^3) easing out/in: deceleration until halfway, then acceleration.
- static float **QuartEaseOut** (float t, float b, float c, float d)

Easing equation function for a quartic (t^4) easing out: decelerating from zero velocity.

- static float **QuartEaseIn** (float t, float b, float c, float d)
  Easing equation function for a quartic (t^4) easing in: accelerating from zero velocity.
- static float **QuartEaseInOut** (float t, float b, float c, float d)
  Easing equation function for a quartic (t^4) easing in/out: acceleration until halfway, then deceleration.
- static float **QuartEaseOutIn** (float t, float b, float c, float d)
  Easing equation function for a quartic (t^4) easing out/in: deceleration until halfway, then acceleration.
- static float **QuintEaseOut** (float t, float b, float c, float d)
  Easing equation function for a quintic (t^5) easing out: decelerating from zero velocity.
- static float **QuintEaseIn** (float t, float b, float c, float d)
  Easing equation function for a quintic (t^5) easing in: accelerating from zero velocity.
- static float **QuintEaseInOut** (float t, float b, float c, float d)
  Easing equation function for a quintic (t^5) easing in/out: acceleration until halfway, then deceleration.
- static float **QuintEaseOutIn** (float t, float b, float c, float d)
  Easing equation function for a quintic (t^5) easing in/out: acceleration until halfway, then deceleration.
- static float **ElasticEaseOut** (float t, float b, float c, float d)
  Easing equation function for an elastic (exponentially decaying sine wave) easing out: decelerating from zero velocity.
- static float **ElasticEaseIn** (float t, float b, float c, float d)
  Easing equation function for an elastic (exponentially decaying sine wave) easing in: accelerating from zero velocity.
- static float **ElasticEaseInOut** (float t, float b, float c, float d)
  Easing equation function for an elastic (exponentially decaying sine wave) easing in/out: acceleration until halfway, then deceleration.
- static float **ElasticEaseOutIn** (float t, float b, float c, float d)
  Easing equation function for an elastic (exponentially decaying sine wave) easing out/in: deceleration until halfway, then acceleration.
- static float **BounceEaseOut** (float t, float b, float c, float d)
  Easing equation function for a bounce (exponentially decaying parabolic bounce) easing out: decelerating from zero velocity.
- static float **BounceEaseIn** (float t, float b, float c, float d)
  Easing equation function for a bounce (exponentially decaying parabolic bounce) easing in: accelerating from zero velocity.
- static float **BounceEaseInOut** (float t, float b, float c, float d)
  Easing equation function for a bounce (exponentially decaying parabolic bounce) easing in/out: acceleration until halfway, then deceleration.
- static float **BounceEaseOutIn** (float t, float b, float c, float d)
  Easing equation function for a bounce (exponentially decaying parabolic bounce) easing out/in: deceleration until halfway, then acceleration.
- static float **BackEaseOut** (float t, float b, float c, float d)
  *Easing equation function for a back (overshooting cubic easing: (s+1)*t^3 - s*t^2) easing out: decelerating from zero velocity.*
- static float **BackEaseIn** (float t, float b, float c, float d)
  *Easing equation function for a back (overshooting cubic easing: (s+1)*t^3 - s*t^2) easing in: accelerating from zero velocity.*
- static float **BackEaseInOut** (float t, float b, float c, float d)
  *Easing equation function for a back (overshooting cubic easing: (s+1)*t^3 - s*t^2) easing in/out: acceleration until halfway, then deceleration.*

- static float **BackEaseOutIn** (float t, float b, float c, float d)
  *Easing equation function for a back (overshooting cubic easing: (s+1)*t^3 - s*t^2) easing out/in: deceleration until halfway, then acceleration.*

Public Attributes

- **FluxAnimationData Movement**
- **FluxAnimationData Rotation**
- **FluxAnimationData Scale**
- **VCAlphaData Alpha**
- **FluxColorData ColorData**
- **VCTimeMode TimeMode**
- **AnimNotificationEvent onOpenStart** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onOpenEnd** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onCloseStart** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onCloseEnd** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onMovementOpenStart** = new **AnimNotificationEvent**()
  Movement.
- **AnimNotificationEvent onMovementOpenEnd** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onMovementCloseStart** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onMovementCloseEnd** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onRotationOpenStart** = new **AnimNotificationEvent**()
  Rotation.
- **AnimNotificationEvent onRotationOpenEnd** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onRotationCloseStart** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onRotationCloseEnd** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onScaleOpenStart** = new **AnimNotificationEvent**()
  Scale.
- **AnimNotificationEvent onScaleOpenEnd** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onScaleCloseStart** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onScaleCloseEnd** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onAlphaOpenStart** = new **AnimNotificationEvent**()
  Alpha.
- **AnimNotificationEvent onAlphaOpenEnd** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onAlphaCloseStart** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onAlphaCloseEnd** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onColorOpenStart** = new **AnimNotificationEvent**()
  Color.
- **AnimNotificationEvent onColorOpenEnd** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onColorCloseStart** = new **AnimNotificationEvent**()
- **AnimNotificationEvent onColorCloseEnd** = new **AnimNotificationEvent**()
- bool **ShowAnimations** = false
- bool **ShowEvents** = false
- bool **ShowNormalEvents** = false
- bool **ShowMovementEvents** = false
- bool **ShowRotationEvents** = false
- bool **ShowScaleEvents** = false
- bool **ShowAlphaEvents** = false
- bool **ShowColorEvents** = false

Member Enumeration Documentation

enum FluxAnimation.Equations [strong]

Enumeration of all easing equations.

Enumerator

None
Linear
QuadEaseOut
QuadEaseIn
QuadEaseInOut
QuadEaseOutIn
ExpoEaseOut
ExpoEaseIn
ExpoEaseInOut
ExpoEaseOutIn
CubicEaseOut
CubicEaseIn
CubicEaseInOut
CubicEaseOutIn
QuartEaseOut
QuartEaseIn
QuartEaseInOut
QuartEaseOutIn
QuintEaseOut
QuintEaseIn
QuintEaseInOut
QuintEaseOutIn
CircEaseOut
CircEaseIn
CircEaseInOut
CircEaseOutIn
SineEaseOut
SineEaseIn
SineEaseInOut
SineEaseOutIn
ElasticEaseOut
ElasticEaseIn
ElasticEaseInOut
ElasticEaseOutIn
BounceEaseOut
BounceEaseIn
BounceEaseInOut
BounceEaseOutIn
BackEaseOut
BackEaseIn
BackEaseInOut
BackEaseOutIn

enum FluxAnimation.VCAnimationDirection [strong]

Enumerator

Forward
Reverse

enum FluxAnimation.VCAnimationMethod `[strong]`

Enumerator

AnimFunction
AnimCurve

enum FluxAnimation.VCAnimationState `[strong]`

Enumerator

AnimStarted
AnimFinished

enum FluxAnimation.VCColorComponentType `[strong]`

Enumerator

None
SpriteRenderer
Renderer
Image
Text

enum FluxAnimation.VCTimeMode `[strong]`

Enumerator

ScaledTime
UnscaledTime

Member Function Documentation

IEnumerator FluxAnimation.AlphaFunc (bool *forward*)

Alpha Enumerator Function

Parameters:

forward	True if it's an open animation, False if its a Close Animation

Returns:

The IEnumerator

static float FluxAnimation.BackEaseIn (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a back (overshooting cubic easing: (s+1)*t^3 - s*t^2) easing in: accelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.BackEaseInOut (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a back (overshooting cubic easing: (s+1)*t^3 - s*t^2) easing in/out: acceleration until halfway, then deceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.BackEaseOut (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a back (overshooting cubic easing: (s+1)*t^3 - s*t^2) easing out: decelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.BackEaseOutIn (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a back (overshooting cubic easing: (s+1)*t^3 - s*t^2) easing out/in: deceleration until halfway, then acceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.BounceEaseIn (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a bounce (exponentially decaying parabolic bounce) easing in: accelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.BounceEaseInOut (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a bounce (exponentially decaying parabolic bounce) easing in/out: acceleration until halfway, then deceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.BounceEaseOut (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a bounce (exponentially decaying parabolic bounce) easing out: decelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.BounceEaseOutIn (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a bounce (exponentially decaying parabolic bounce) easing out/in: deceleration until halfway, then acceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.CircEaseIn (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a circular (sqrt(1-t^2)) easing in: accelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.CircEaseInOut (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a circular (sqrt(1-t^2)) easing in/out: acceleration until halfway, then deceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.CircEaseOut (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a circular (sqrt(1-t^2)) easing out: decelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.

d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.CircEaseOutIn (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a circular (sqrt(1-t^2)) easing in/out: acceleration until halfway, then deceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

IEnumerator FluxAnimation.ColorFunc (bool *forward*)

Color Enumerator Function

Parameters:

forward	True if it's an open animation, False if its a Close Animation

Returns:

The IEnumerator

static float FluxAnimation.CubicEaseIn (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a cubic (t^3) easing in: accelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.CubicEaseInOut (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a cubic (t^3) easing in/out: acceleration until halfway, then deceleration.

Parameters:

t	Current time in seconds.
b	Starting value.

c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.CubicEaseOut (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a cubic (t^3) easing out: decelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.CubicEaseOutIn (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a cubic (t^3) easing out/in: deceleration until halfway, then acceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.ElasticEaseIn (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for an elastic (exponentially decaying sine wave) easing in: accelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.ElasticEaseInOut (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for an elastic (exponentially decaying sine wave) easing in/out: acceleration until halfway, then deceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.ElasticEaseOut (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for an elastic (exponentially decaying sine wave) easing out: decelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.ElasticEaseOutIn (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for an elastic (exponentially decaying sine wave) easing out/in: deceleration until halfway, then acceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

delegate float FluxAnimation.EquationFunction (float *t*, float *b*, float *c*, float *d*)

static float FluxAnimation.ExpoEaseIn (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for an exponential (2^t) easing in: accelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.ExpoEaseInOut (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for an exponential (2^t) easing in/out: acceleration until halfway, then deceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.ExpoEaseOut (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for an exponential (2^t) easing out: decelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.ExpoEaseOutIn (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for an exponential (2^t) easing out/in: deceleration until halfway, then acceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

float FluxAnimation.GetDeltaTime ()

Returns the delta time based on Current Time Mode

Returns:

Returns DeltaTime

static float FluxAnimation.Linear (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a simple linear tweening, with no easing.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

IEnumerator FluxAnimation.MoveFunc (bool *forward*)

Movement Enumerator Function

Parameters:

forward	True if it's an open animation, False if its a Close Animation

Returns:

The IEnumerator

void FluxAnimation.OnAnimationFinished ()

Calls Automatically when each Animation is finished

static float FluxAnimation.QuadEaseIn (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a quadratic (t^2) easing in: accelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.QuadEaseInOut (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a quadratic (t^2) easing in/out: acceleration until halfway, then deceleration.

Parameters:

t	Current time in seconds.

b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.QuadEaseOut (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a quadratic (t^2) easing out: decelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.QuadEaseOutIn (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a quadratic (t^2) easing out/in: deceleration until halfway, then acceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.QuartEaseIn (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a quartic (t^4) easing in: accelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.QuartEaseInOut (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a quartic (t^4) easing in/out: acceleration until halfway, then deceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.QuartEaseOut (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a quartic (t^4) easing out: decelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.QuartEaseOutIn (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a quartic (t^4) easing out/in: deceleration until halfway, then acceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.QuintEaseIn (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a quintic (t^5) easing in: accelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.QuintEaseInOut (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a quintic (t^5) easing in/out: acceleration until halfway, then deceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.QuintEaseOut (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a quintic (t^5) easing out: decelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.QuintEaseOutIn (float *t*, float *b*, float *c*, float *d*) [static]

Easing equation function for a quintic (t^5) easing in/out: acceleration until halfway, then deceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

IEnumerator FluxAnimation.RotationFunc (bool *forward*)

Rotation Enumerator Function

Parameters:

forward	True if it's an open animation, False if its a Close Animation

Returns:

The IEnumerator

IEnumerator FluxAnimation.ScaleFunc (bool *forward*)

Scale Enumerator Function

Parameters:

forward	True if it's an open animation, False if its a Close Animation

Returns:

The IEnumerator

static float FluxAnimation.SineEaseIn (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a sinusoidal (sin(t)) easing in: accelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.SineEaseInOut (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a sinusoidal (sin(t)) easing in/out: acceleration until halfway, then deceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.SineEaseOut (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a sinusoidal (sin(t)) easing out: decelerating from zero velocity.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

static float FluxAnimation.SineEaseOutIn (float *t*, float *b*, float *c*, float *d*) `[static]`

Easing equation function for a sinusoidal (sin(t)) easing in/out: deceleration until halfway, then acceleration.

Parameters:

t	Current time in seconds.
b	Starting value.
c	Final value.
d	Duration of animation.

Returns:

The correct value.

void FluxAnimation.StartAlphaCloseAnimation ()

Starts Alpha Close Animation

void FluxAnimation.StartAlphaOpenAnimation ()

Starts Alpha Open Animation

void FluxAnimation.StartClosingAnimation ()

Starts all the closing animations

void FluxAnimation.StartColorCloseAnimation ()

Starts Color Close Animation

void FluxAnimation.StartColorOpenAnimation ()

Starts Color Open Animation

void FluxAnimation.StartMovementCloseAnimation ()

Starts Movement Close Animation

void FluxAnimation.StartMovementOpenAnimation ()

Starts Movement Open Animation

void FluxAnimation.StartOpeningAnimation ()

Starts All the Opening Animations

void FluxAnimation.StartRotationCloseAnimation ()

Starts Rotation Close Animation

void FluxAnimation.StartRotationOpenAnimation ()

Starts Rotation Open Animation

void FluxAnimation.StartScaleCloseAnimation ()

Starts Scale Close Animation

void FluxAnimation.StartScaleOpenAnimation ()

Starts Scale Open Animation

void FluxAnimation.StopAlphaAnimation ()

Stop Alpha Animation

void FluxAnimation.StopAnimation ()

Stops All the Open Animations

void FluxAnimation.StopColorAnimation ()

Stop Color Animation

void FluxAnimation.StopMovementAnimation ()

Stop Movement Animation

void FluxAnimation.StopRotationAnimation ()

Stop Rotation Animation

void FluxAnimation.StopScaleAnimation ()

Stop Scale Animation

IEnumerator FluxAnimation.WaitForRealSeconds (float *time*)

Wait for Seconds Based on the Time Mode

Parameters:

time	Time to wait for

Returns:

Member Data Documentation

VCAlphaData FluxAnimation.Alpha

FluxColorData FluxAnimation.ColorData

FluxAnimationData FluxAnimation.Movement

AnimNotificationEvent FluxAnimation.onAlphaCloseEnd = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onAlphaCloseStart = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onAlphaOpenEnd = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onAlphaOpenStart = new AnimNotificationEvent()

Alpha.

AnimNotificationEvent FluxAnimation.onCloseEnd = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onCloseStart = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onColorCloseEnd = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onColorCloseStart = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onColorOpenEnd = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onColorOpenStart = new AnimNotificationEvent()

Color.

AnimNotificationEvent FluxAnimation.onMovementCloseEnd = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onMovementCloseStart = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onMovementOpenEnd = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onMovementOpenStart = new AnimNotificationEvent()

Movement.

AnimNotificationEvent FluxAnimation.onOpenEnd = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onOpenStart = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onRotationCloseEnd = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onRotationCloseStart = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onRotationOpenEnd = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onRotationOpenStart = new AnimNotificationEvent()

Rotation.

AnimNotificationEvent FluxAnimation.onScaleCloseEnd = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onScaleCloseStart = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onScaleOpenEnd = new AnimNotificationEvent()

AnimNotificationEvent FluxAnimation.onScaleOpenStart = new AnimNotificationEvent()

Scale.

FluxAnimationData FluxAnimation.Rotation

FluxAnimationData FluxAnimation.Scale

bool FluxAnimation.ShowAlphaEvents = false

bool FluxAnimation.ShowAnimations = false

bool FluxAnimation.ShowColorEvents = false

bool FluxAnimation.ShowEvents = false

bool FluxAnimation.ShowMovementEvents = false

bool FluxAnimation.ShowNormalEvents = false

bool FluxAnimation.ShowRotationEvents = false

bool FluxAnimation.ShowScaleEvents = false

VCTimeMode FluxAnimation.TimeMode

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

- **FluxAnimation.cs**

FluxAnimation.FluxAnimationData Class Reference

Public Attributes

- **VCAnimationInfo OpenAnimation**
- **VCAnimationInfo CloseAnimation**
- bool **Enable** = false
- bool **Autoclose** = false
- float **AutocloseDelay** = 0
- bool **Autoplay** = false
- bool **Loop** = false
- bool **ShowInspector** = false

Member Data Documentation

bool FluxAnimation.FluxAnimationData.Autoclose = false

float FluxAnimation.FluxAnimationData.AutocloseDelay = 0

bool FluxAnimation.FluxAnimationData.Autoplay = false

VCAnimationInfo FluxAnimation.FluxAnimationData.CloseAnimation

bool FluxAnimation.FluxAnimationData.Enable = false

bool FluxAnimation.FluxAnimationData.Loop = false

VCAnimationInfo FluxAnimation.FluxAnimationData.OpenAnimation

bool FluxAnimation.FluxAnimationData.ShowInspector = false

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

- **FluxAnimation.cs**

FluxAnimation.FluxColorData Class Reference

Public Attributes

- **VCColorInfo OpenAnimation**
- **VCColorInfo CloseAnimation**
- bool **Enable** = false
- bool **Autoclose** = false
- float **AutocloseDelay** = 0
- bool **Autoplay** = false
- bool **Loop** = false
- bool **ShowInspector** = false

Member Data Documentation

bool FluxAnimation.FluxColorData.Autoclose = false

float FluxAnimation.FluxColorData.AutocloseDelay = 0

bool FluxAnimation.FluxColorData.Autoplay = false

VCColorInfo FluxAnimation.FluxColorData.CloseAnimation

bool FluxAnimation.FluxColorData.Enable = false

bool FluxAnimation.FluxColorData.Loop = false

VCColorInfo FluxAnimation.FluxColorData.OpenAnimation

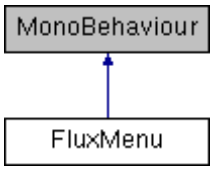
bool FluxAnimation.FluxColorData.ShowInspector = false

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

- **FluxAnimation.cs**

FluxMenu Class Reference

Inheritance diagram for FluxMenu:



Public Types

- enum **AnimationState** { **AnimationState.Opening**, **AnimationState.OpenIdle**, **AnimationState.Closing**, **AnimationState.CloseIdle** }

Public Member Functions

- **AnimationState GetAnimationState** ()
  Returns the Current Animation State
- virtual void **Awake** ()
- virtual void **Show** ()
  Show the Menu
- virtual void **Hide** ()
  Hide the Menu
- virtual void **Update** ()
  Calls once per Frame, Don't Call Explicitly
- **FluxMenuStateManager GetMenuManager** ()
  Returns the Menu State Manager attached to this menu
- void **MakeDefault** ()
  Make this the Default Menu
- void **MakeExit** ()
  Make this the Exit Menu

Public Attributes

- **FluxMenuStateManager StateManager**
- bool **Popup** = false

Properties

- virtual bool **IsOpen** `[get, set]`
  Returns if the Menu is Open or Closed

Member Enumeration Documentation

enum FluxMenu.AnimationState `[strong]`

Enumerator

Opening
OpenIdle
Closing

CloseIdle

Member Function Documentation

virtual void FluxMenu.Awake () `[virtual]`

AnimationState FluxMenu.GetAnimationState ()

Returns the Current Animation State

Returns:

FluxMenuStateManager FluxMenu.GetMenuManager ()

Returns the Menu State Manager attached to this menu

Returns:
FluxMenuStateManager

virtual void FluxMenu.Hide () `[virtual]`

Hide the Menu

void FluxMenu.MakeDefault ()

Make this the Default Menu

void FluxMenu.MakeExit ()

Make this the Exit Menu

virtual void FluxMenu.Show () `[virtual]`

Show the Menu

virtual void FluxMenu.Update () `[virtual]`

Calls once per Frame, Don't Call Explicitly

Member Data Documentation

bool FluxMenu.Popup = false

FluxMenuStateManager FluxMenu.StateManager

Property Documentation

virtual bool FluxMenu.IsOpen `[get]`, `[set]`

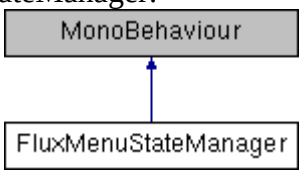
Returns if the Menu is Open or Closed

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

- **FluxMenu.cs**

FluxMenuStateManager Class Reference

Inheritance diagram for FluxMenuStateManager:



Public Member Functions

- void **AddStackedMenu** (**FluxMenu** menu)
  Adds a Stacked Menu to the Hierarchy
- void **SetCurrentMenu** (**FluxMenu** menu)
  Removes all the Previous Hierarachy and Adds a Stacked Menu
- void **GotoPrevious** ()
  Go to the Previous Menu of the Hierarchy
- GameObject **AddMenuObject** ()
  Adds a New Menu to the Scene Hierarchy

Public Attributes

- **FluxMenu DefaultMenu**
- **FluxMenu ExitMenu**

Static Public Attributes

- static **FluxMenuStateManager Instance**

Member Function Documentation

GameObject FluxMenuStateManager.AddMenuObject ()

Adds a New Menu to the Scene Hierarchy

Returns:

Returns the GameObject of the New Menu

void FluxMenuStateManager.AddStackedMenu (FluxMenu *menu*)

Adds a Stacked Menu to the Hierarchy

Parameters:

menu	**FluxMenu** to Add to the Hierarchy

void FluxMenuStateManager.GotoPrevious ()

Go to the Previous Menu of the Hierarchy

void FluxMenuStateManager.SetCurrentMenu (FluxMenu *menu*)

Removes all the Previous Hierarachy and Adds a Stacked Menu

Parameters:

menu	**FluxMenu** to Add to the Hierarchy

Member Data Documentation

FluxMenu FluxMenuStateManager.DefaultMenu

FluxMenu FluxMenuStateManager.ExitMenu

FluxMenuStateManager FluxMenuStateManager.Instance `[static]`

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

- **FluxMenuStateManager.cs**

FluxAnimation.VCAlphaData Class Reference

Public Attributes

- **VCAlphaInfo OpenAnimation**
- **VCAlphaInfo CloseAnimation**
- bool **Enable** = false
- bool **Autoclose** = false
- float **AutocloseDelay** = 0
- bool **Autoplay** = false
- bool **Loop** = false
- bool **ShowInspector** = false

Member Data Documentation

bool FluxAnimation.VCAlphaData.Autoclose = false

float FluxAnimation.VCAlphaData.AutocloseDelay = 0

bool FluxAnimation.VCAlphaData.Autoplay = false

VCAlphaInfo FluxAnimation.VCAlphaData.CloseAnimation

bool FluxAnimation.VCAlphaData.Enable = false

bool FluxAnimation.VCAlphaData.Loop = false

VCAlphaInfo FluxAnimation.VCAlphaData.OpenAnimation

bool FluxAnimation.VCAlphaData.ShowInspector = false

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

- **FluxAnimation.cs**

FluxAnimation.VCAlphaInfo Class Reference

Public Member Functions

- **VCAlphaInfo** ()

Public Attributes

- float **Duration**
- float **Delay**
- float **Start**
- float **End**
- bool **ShowInspector** = false

Constructor & Destructor Documentation

FluxAnimation.VCAlphaInfo.VCAlphaInfo ()

Member Data Documentation

float FluxAnimation.VCAlphaInfo.Delay

float FluxAnimation.VCAlphaInfo.Duration

float FluxAnimation.VCAlphaInfo.End

bool FluxAnimation.VCAlphaInfo.ShowInspector = false

float FluxAnimation.VCAlphaInfo.Start

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

- **FluxAnimation.cs**

FluxAnimation.VCAnimationInfo Class Reference

Public Member Functions

- **VCAnimationInfo** ()

Public Attributes

- **VCAnimationMethod AnimMethod**
- AnimationCurve **CurveX**
- AnimationCurve **CurveY**
- AnimationCurve **CurveZ**
- **Equations XEquation**
- **Equations YEquation**
- **Equations ZEquation**
- float **Duration**
- float **Delay**
- Vector3 **Start**
- Vector3 **End**
- bool **ShowInspector** = false

Constructor & Destructor Documentation

FluxAnimation.VCAnimationInfo.VCAnimationInfo ()

Member Data Documentation

VCAnimationMethod FluxAnimation.VCAnimationInfo.AnimMethod

AnimationCurve FluxAnimation.VCAnimationInfo.CurveX

AnimationCurve FluxAnimation.VCAnimationInfo.CurveY

AnimationCurve FluxAnimation.VCAnimationInfo.CurveZ

float FluxAnimation.VCAnimationInfo.Delay

float FluxAnimation.VCAnimationInfo.Duration

Vector3 FluxAnimation.VCAnimationInfo.End

bool FluxAnimation.VCAnimationInfo.ShowInspector = false

Vector3 FluxAnimation.VCAnimationInfo.Start

Equations FluxAnimation.VCAnimationInfo.XEquation

Equations FluxAnimation.VCAnimationInfo.YEquation

Equations FluxAnimation.VCAnimationInfo.ZEquation

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

- **FluxAnimation.cs**

FluxAnimation.VCColorInfo Class Reference

Public Member Functions

- **VCColorInfo** ()

Public Attributes

- **VCAnimationMethod AnimMethod**
- AnimationCurve **Curve**
- **Equations Equation**
- float **Duration**
- float **Delay**
- Color **Start**
- Color **End**
- bool **ShowInspector** = false

Constructor & Destructor Documentation

FluxAnimation.VCColorInfo.VCColorInfo ()

Member Data Documentation

VCAnimationMethod FluxAnimation.VCColorInfo.AnimMethod

AnimationCurve FluxAnimation.VCColorInfo.Curve

float FluxAnimation.VCColorInfo.Delay

float FluxAnimation.VCColorInfo.Duration

Color FluxAnimation.VCColorInfo.End

Equations FluxAnimation.VCColorInfo.Equation

bool FluxAnimation.VCColorInfo.ShowInspector = false

Color FluxAnimation.VCColorInfo.Start

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

- **FluxAnimation.cs**