



DOOZY UI
COMPLETE
UI MANAGEMENT
SYSTEM

Api Documentation

Table of Contents

Api Documentation

DoozyUI

Anim

Anim.AnimationType

AnimData

DUI

DUI.EventType

DUISettings

EditorNavigationPointer

EditorNavigationPointerData

Fade

FadeLoop

Loop

Loop.LoopType

LoopData

Move

Move.MoveDirection

MoveLoop

NamesDatabase

NavigationPointer

NavigationPointerData

OrientationManager

OrientationManager.Orientation

OrientationManager.OrientationChange

PlaymakerEventDispatcher

Punch

PunchData

PunchMove

PunchRotate

PunchScale

RadialLayout

Rotate

RotateLoop

Scale
ScaleLoop
SceneLoader
SoundType
Soundy
UIAnimationManager
UIAnimationManager.AnimationType
UIAnimationManager.ButtonLoopsAnimations
UIAnimationManager.ButtonLoopType
UIAnimationManager.InAnimations
UIAnimationManager.LoopAnimations
UIAnimationManager.OnClickAnimations
UIAnimationManager.OutAnimations
UIAnimator
UIAnimator.AnimationTarget
UIAnimator.ButtonAnimationType
UIAnimator.EaseType
UIAnimator.FadeIn
UIAnimator.FadeLoop
UIAnimator.FadeOut
UIAnimator.InitialData
UIAnimator.MoveDetails
UIAnimator.MoveIn
UIAnimator.MoveLoop
UIAnimator.MoveOut
UIAnimator.ResetType
UIAnimator.RotationIn
UIAnimator.RotationLoop
UIAnimator.RotationOut
UIAnimator.ScaleIn
UIAnimator.ScaleLoop
UIAnimator.ScaleOut
UIAnimator.SoundDetails
UIAnimator.SoundOutput
UIAnimator.TweenIdAnimation
UIAnimator.TweenIdType
UIAnimatorUtil

- UIButton
 - UIButton.ButtonActionType
 - UIButton.ButtonClickType
 - UIButton.ButtonName
 - UIButton.ButtonSound
 - UIButton.SingleClickMode
- UICanvas
- UIEffect
 - UIEffect.EffectPosition
- UIElement
 - UIElement.ElementName
- UIManager
 - UIManager.Orientation
- UINavigationController
- UINotification
 - UINotification.NotificationData
- UINotificationManager
 - UINotificationManager.NotificationItem
- UISound
- UITrigger
 - UITrigger.TriggerEvent
- QuickEngine
 - UIColor
 - QColors
 - QResources
- QuickEngine.Common
 - Singleton<T>
- QuickEngine.Extensions
 - ArrayAndListExtensions
 - AudioExtensions
 - BoolExtensions
 - CameraExtensions
 - ColorExtensions
 - DateTimeExtensions
 - DictionaryExtensions
 - FloatExtensions
 - ICollectionExtensions

RectTransformExtensions

StringExtensions

TransformExtensions

VectorExtensions

QuickEngine.IO

File

QuickEngine.Utils

QAssets

QEmailValidator

QIPValidator

QReflection

Namespace DoozyUI

Classes

[Anim](#)

Base class for all the In and Out animations.

[AnimData](#)

[DUI](#)

[DUISettings](#)

[EditorNavigationPointer](#)

[EditorNavigationPointerData](#)

[Fade](#)

Animation settings for Fade (alpha value)

[FadeLoop](#)

[Loop](#)

Base class for all the Loop animations.

[LoopData](#)

[Move](#)

Animation settings for Movement

[MoveLoop](#)

[NamesDatabase](#)

[NavigationPointer](#)

Helper class for the UI Navigation.

[NavigationPointerData](#)

Helper class for the UINavigation.

[OrientationManager](#)

[OrientationManager.OrientationChange](#)

[PlaymakerEventDispatcher](#)

[Punch](#)

Base class for all the Punch animations.

[PunchData](#)

[PunchMove](#)

Punches a Transform's anchoredPosition towards the given direction and then back to the starting one as if it was connected to the starting scale via an elastic.

[PunchRotate](#)

Punches a Transform's localRotation towards the given rotation and then back to the starting one as if it was connected to the starting scale via an elastic.

[PunchScale](#)

Punches a Transform's localScale towards the given size and then back to the starting one as if it was connected to the starting scale via an elastic.

[RadialLayout](#)

[Rotate](#)

Animation settings for Rotation

[RotateLoop](#)

[Scale](#)

Animation settings for Scale

[ScaleLoop](#)

[SceneLoader](#)

[Soundy](#)

[UIAnimationManager](#)

[UIAnimationManager.ButtonLoopsAnimations](#)

[UIAnimationManager.InAnimations](#)

[UIAnimationManager.LoopAnimations](#)

[UIAnimationManager.OnClickAnimations](#)

[UIAnimationManager.OutAnimations](#)

[UIAnimator](#)

[UIAnimator.FadeIn](#)

[UIAnimator.FadeLoop](#)

[UIAnimator.FadeOut](#)

[UIAnimator.InitialData](#)

[UIAnimator.MoveIn](#)

[UIAnimator.MoveLoop](#)

[UIAnimator.MoveOut](#)

[UIAnimator.RotationIn](#)

[UIAnimator.RotationLoop](#)

[UIAnimator.RotationOut](#)

[UIAnimator.ScaleIn](#)

[UIAnimator.ScaleLoop](#)

[UIAnimator.ScaleOut](#)

[UIAnimator.SoundDetails](#)

[UIAnimatorUtil](#)

[UIButton](#)

[UIButton.ButtonName](#)

[UIButton.ButtonSound](#)

[UICanvas](#)

[UIEffect](#)

[UIElement](#)

[UIElement.ElementName](#)

[UIManager](#)

[UINavigation](#)

[UINotification](#)

[UINotification.NotificationData](#)

Helper class that holds all the Notification settings.

[UINotificationManager](#)

[UINotificationManager.NotificationItem](#)

Helper class for the NotificationManager.

[UISound](#)

[UITrigger](#)

[UITrigger.TriggerEvent](#)

Helper class for an UnityEvent with one string parameter.

Enums

[Anim.AnimationType](#)

Type of Animation. This changes the way the Animator percieves the set values.

[DUI.EventType](#)

[Loop.LoopType](#)

Types of loop

[Move.MoveDirection](#)

[OrientationManager.Orientation](#)

Orientation type

[SoundType](#)

[UIAnimationManager.AnimationType](#)

[UIAnimationManager.ButtonLoopType](#)

[UIAnimator.AnimationTarget](#)

[UIAnimator.ButtonAnimationType](#)

[UIAnimator.EaseType](#)

Type of ease an animation, loop or punch should use.

[UIAnimator.MoveDetails](#)

[UIAnimator.ResetType](#)

[UIAnimator.SoundOutput](#)

[UIAnimator.TweenIdAnimation](#)

Used to map the tween ids.

[UIAnimator.TweenIdType](#)

Used to map the tween ids.

[UIButton.ButtonActionType](#)

All the action types a button can perform.

[UIButton.ButtonClickType](#)

All the click types actions a button can perform.

[UIButton.SingleClickMode](#)

Setting for the OnClick trigger that marks if it should be registered instantly without checking if it's a double click or not.

[UIEffect.EffectPosition](#)

Determines the sorting order.

[UIManager.Orientation](#)

Types of orientation used by DoozyUI. Unknown is used for initialization purposes.

Class Anim

Base class for all the In and Out animations.

Inheritance

System.Object
Anim

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class Anim
```

Constructors

Anim(Anim.AnimationType)

Declaration

```
public Anim(Anim.AnimationType aType)
```

Parameters

TYPE	NAME	DESCRIPTION
Anim.AnimationType	aType	

Fields

animationType

Determines what type of animation this is. This changes the way the Animator percieves the set values.

Declaration

```
public Anim.AnimationType animationType
```

Field Value

TYPE	DESCRIPTION
Anim.AnimationType	

fade

Fade (alpha) animation settings.

Declaration

```
public Fade fade
```

Field Value

TYPE	DESCRIPTION
Fade	

move

Movement animation settings.

Declaration

<code>public Move move</code>

Field Value

TYPE	DESCRIPTION
Move	

rotate

Rotation animation settings.

Declaration

<code>public Rotate rotate</code>

Field Value

TYPE	DESCRIPTION
Rotate	

scale

Scale animation settings.

Declaration

<code>public Scale scale</code>

Field Value

TYPE	DESCRIPTION
Scale	

Properties

Enabled

Declaration

<code>public bool Enabled { get; }</code>

Property Value

TYPE	DESCRIPTION
System.Boolean	

StartDelay

Declaration

```
public float StartDelay { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

TotalDuration

Declaration

```
public float TotalDuration { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Copy()

Declaration

```
public Anim Copy()
```

Returns

TYPE	DESCRIPTION
Anim	

Reset(Anim.AnimationType)

Declaration

```
public void Reset(Anim.AnimationType aType)
```

Parameters

TYPE	NAME	DESCRIPTION
Anim.AnimationType	aType	

Reverse()

Declaration

```
public Anim Reverse()
```

Returns

TYPE	DESCRIPTION
Anim	

Reverse(Anim.AnimationType)

Declaration

```
public static Anim.AnimationType Reverse(Anim.AnimationType animationType)
```

Parameters

TYPE	NAME	DESCRIPTION
Anim.AnimationType	animationType	

Returns

TYPE	DESCRIPTION
Anim.AnimationType	

UpdateValues(Anim)

Declaration

```
public void UpdateValues(Anim a)
```

Parameters

TYPE	NAME	DESCRIPTION
Anim	a	

Enum Anim.AnimationType

Type of Animation. This changes the way the Animator percieves the set values.

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public enum AnimationType
```

Fields

NAME	DESCRIPTION
In	
Out	

Class AnimData

Inheritance

System.Object

AnimData

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class AnimData : ScriptableObject
```

Constructors

AnimData(Anim.AnimationType)

Declaration

```
public AnimData(Anim.AnimationType aType)
```

Parameters

TYPE	NAME	DESCRIPTION
Anim.AnimationType	aType	

Fields

data

Declaration

```
public Anim data
```

Field Value

TYPE	DESCRIPTION
Anim	

presetCategory

Declaration

```
public string presetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

presetName

Declaration

```
public string presetName
```

Field Value

TYPE	DESCRIPTION
System.String	

Properties

LoadDefaultValues

Declaration

```
public bool LoadDefaultValues { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Class DUI

Inheritance

System.Object
DUI

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class DUI
```

Fields

BACK_BUTTON_NAME

Declaration

```
public const string BACK_BUTTON_NAME = "Back"
```

Field Value

TYPE	DESCRIPTION
System.String	

CANVAS_DATABASE_FILENAME

Declaration

```
public const string CANVAS_DATABASE_FILENAME = "CanvasDatabase"
```

Field Value

TYPE	DESCRIPTION
System.String	

CanvasNamesDatabase

Declaration

```
public static NamesDatabase CanvasNamesDatabase
```

Field Value

TYPE	DESCRIPTION
NamesDatabase	

COMPONENT_MENU_PLAYMAKER_EVENT_DISPATCHER

Declaration

```
public const string COMPONENT_MENU_PLAYMAKER_EVENT_DISPATCHER = "DoozyUI/Playmaker/Event Dispatcher"
```

Field Value

TYPE	DESCRIPTION
System.String	

COMPONENT_MENU_UIBUTTON

Declaration

```
public const string COMPONENT_MENU_UIBUTTON = "DoozyUI/UI Button"
```

Field Value

TYPE	DESCRIPTION
System.String	

COMPONENT_MENU_UICANVAS

Declaration

```
public const string COMPONENT_MENU_UICANVAS = "DoozyUI/UI Canvas"
```

Field Value

TYPE	DESCRIPTION
System.String	

COMPONENT_MENU_UIEFFECT

Declaration

```
public const string COMPONENT_MENU_UIEFFECT = "DoozyUI/UI Effect"
```

Field Value

TYPE	DESCRIPTION
System.String	

COMPONENT_MENU_UIELEMENT

Declaration

```
public const string COMPONENT_MENU_UIELEMENT = "DoozyUI/UI Element"
```

Field Value

TYPE	DESCRIPTION
System.String	

COMPONENT_MENU_UINOTIFICATION

Declaration

```
public const string COMPONENT_MENU_UINOTIFICATION = "DoozyUI/UI Notification"
```

Field Value

TYPE	DESCRIPTION
System.String	

COMPONENT_MENU_UITRIGGER

Declaration

```
public const string COMPONENT_MENU_UITRIGGER = "DoozyUI/UI Trigger"
```

Field Value

TYPE	DESCRIPTION
System.String	

CUSTOM_NAME

Declaration

```
public const string CUSTOM_NAME = "~Custom Name~"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_BUTTON_NAME

Declaration

```
public const string DEFAULT_BUTTON_NAME = "~Button Name~"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_CANVAS_NAME

Declaration

```
public const string DEFAULT_CANVAS_NAME = "MasterCanvas"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_CATEGORY_NAME

Declaration

```
public const string DEFAULT_CATEGORY_NAME = "Uncategorized"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_ELEMENT_NAME

Declaration

```
public const string DEFAULT_ELEMENT_NAME = "~Element Name~"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_SOUND_NAME

Declaration

```
public const string DEFAULT_SOUND_NAME = "~No Sound~"
```

Field Value

TYPE	DESCRIPTION
System.String	

DISPATCH_ALL

Declaration

```
public const string DISPATCH_ALL = "~Dispatch All~"
```

Field Value

TYPE	DESCRIPTION
System.String	

GAMEOBJECT_MENU_ORIENTATION_MANAGER

Declaration

```
public const string GAMEOBJECT_MENU_ORIENTATION_MANAGER = "GameObject/DoozyUI/Managers/Orientation Manager"
```

Field Value

TYPE	DESCRIPTION
System.String	

GAMEOBJECT_MENU_PLAYMAKER_EVENT_DISPATCHER

Declaration

```
public const string GAMEOBJECT_MENU_PLAYMAKER_EVENT_DISPATCHER = "GameObject/DoozyUI/Playmaker/Event Dispatcher"
```

Field Value

TYPE	DESCRIPTION
System.String	

GAMEOBJECT_MENU_SCENE_LOADER

Declaration

```
public const string GAMEOBJECT_MENU_SCENE_LOADER = "GameObject/DoozyUI/Managers/Scene Loader"
```

Field Value

TYPE	DESCRIPTION
System.String	

GAMEOBJECT_MENU_UIBUTTON

Declaration

```
public const string GAMEOBJECT_MENU_UIBUTTON = "GameObject/DoozyUI/UI Button"
```

Field Value

TYPE	DESCRIPTION
System.String	

GAMEOBJECT_MENU_UICANVAS

Declaration

```
public const string GAMEOBJECT_MENU_UICANVAS = "GameObject/DoozyUI/UI Canvas"
```

Field Value

TYPE	DESCRIPTION
System.String	

GAMEOBJECT_MENU_UIEFFECT

Declaration

```
public const string GAMEOBJECT_MENU_UIEFFECT = "GameObject/DoozyUI/UI Effect"
```

Field Value

TYPE	DESCRIPTION
System.String	

GAMEOBJECT_MENU_UIELEMENT

Declaration

```
public const string GAMEOBJECT_MENU_UIELEMENT = "GameObject/DoozyUI/UI Element"
```

Field Value

TYPE	DESCRIPTION
System.String	

GAMEOBJECT_MENU_UIMANAGER

Declaration

```
public const string GAMEOBJECT_MENU_UIMANAGER = "GameObject/DoozyUI/Managers/UI Manager"
```

Field Value

TYPE	DESCRIPTION
System.String	

GAMEOBJECT_MENU_UINOTIFICATION

Declaration

```
public const string GAMEOBJECT_MENU_UINOTIFICATION = "GameObject/DoozyUI/UI Notification"
```

Field Value

TYPE	DESCRIPTION
System.String	

GAMEOBJECT_MENU_UITRIGGER

Declaration

```
public const string GAMEOBJECT_MENU_UITRIGGER = "GameObject/DoozyUI/UI Trigger"
```

Field Value

TYPE	DESCRIPTION
System.String	

MENU_PRIORITY_ORIENTATION_MANAGER

Declaration

```
public const int MENU_PRIORITY_ORIENTATION_MANAGER = 101
```

Field Value

TYPE	DESCRIPTION
System.Int32	

MENU_PRIORITY_PLAYMAKER_EVENT_DISPATCHER

Declaration

```
public const int MENU_PRIORITY_PLAYMAKER_EVENT_DISPATCHER = 50
```

Field Value

TYPE	DESCRIPTION
System.Int32	

MENU_PRIORITY_SCENE_LOADER

Declaration

```
public const int MENU_PRIORITY_SCENE_LOADER = 102
```

Field Value

TYPE	DESCRIPTION
System.Int32	

MENU_PRIORITY_UIBUTTON

Declaration

```
public const int MENU_PRIORITY_UIBUTTON = 2
```

Field Value

TYPE	DESCRIPTION
System.Int32	

MENU_PRIORITY_UICANVAS

Declaration

```
public const int MENU_PRIORITY_UICANVAS = 0
```

Field Value

TYPE	DESCRIPTION
System.Int32	

MENU_PRIORITY_UIEFFECT

Declaration

```
public const int MENU_PRIORITY_UIEFFECT = 4
```

Field Value

TYPE	DESCRIPTION
System.Int32	

MENU_PRIORITY_UIELEMENT

Declaration

```
public const int MENU_PRIORITY_UIELEMENT = 1
```

Field Value

TYPE	DESCRIPTION
System.Int32	

MENU_PRIORITY_UIMANAGER

Declaration

public const int MENU_PRIORITY_UIMANAGER = 100
--

Field Value

TYPE	DESCRIPTION
System.Int32	

MENU_PRIORITY_UINOTIFICATION

Declaration

public const int MENU_PRIORITY_UINOTIFICATION = 6

Field Value

TYPE	DESCRIPTION
System.Int32	

MENU_PRIORITY_UITRIGGER

Declaration

public const int MENU_PRIORITY_UITRIGGER = 5
--

Field Value

TYPE	DESCRIPTION
System.Int32	

RESOURCES_PATH_CANVAS_DATABASE

Declaration

public const string RESOURCES_PATH_CANVAS_DATABASE = "DUI/Canvases/"
--

Field Value

TYPE	DESCRIPTION
System.String	

RESOURCES_PATH_SETTINGS

Declaration

public const string RESOURCES_PATH_SETTINGS = "DUI/Settings/"

Field Value

TYPE	DESCRIPTION
System.String	

RESOURCES_PATH_UIBUTTONS

Declaration

```
public const string RESOURCES_PATH_UIBUTTONS = "DUI/UIButtonns/"
```

Field Value

TYPE	DESCRIPTION
System.String	

RESOURCES_PATH_UIELEMENTS

Declaration

```
public const string RESOURCES_PATH_UIELEMENTS = "DUI/UIElements/"
```

Field Value

TYPE	DESCRIPTION
System.String	

RESOURCES_PATH_UISOUNDS

Declaration

```
public const string RESOURCES_PATH_UISOUNDS = "DUI/UISounds/"
```

Field Value

TYPE	DESCRIPTION
System.String	

SETTINGS_FILENAME

Declaration

```
public const string SETTINGS_FILENAME = "DUISettings"
```

Field Value

TYPE	DESCRIPTION
System.String	

SYMBOL_DOOZYUI

Declaration

```
public const string SYMBOL_DOOZYUI = "dUI_DoozyUI"
```

Field Value

TYPE	DESCRIPTION
System.String	

SYMBOL_ENERGY_BAR_TOOLKIT

Declaration

```
public const string SYMBOL_ENERGY_BAR_TOOLKIT = "dUI_EnergyBarToolkit"
```

Field Value

TYPE	DESCRIPTION
System.String	

SYMBOL_MASTER_AUDIO

Declaration

```
public const string SYMBOL_MASTER_AUDIO = "dUI_MasterAudio"
```

Field Value

TYPE	DESCRIPTION
System.String	

SYMBOL_NAVIGATION_SYSTEM

Declaration

```
public const string SYMBOL_NAVIGATION_SYSTEM = "dUI_NavigationDisabled"
```

Field Value

TYPE	DESCRIPTION
System.String	

SYMBOL_ORIENTATION_MANAGER

Declaration

```
public const string SYMBOL_ORIENTATION_MANAGER = "dUI_UseOrientationManager"
```

Field Value

TYPE	DESCRIPTION
System.String	

SYMBOL_PLAYMAKER

Declaration

```
public const string SYMBOL_PLAYMAKER = "dUI_PlayMaker"
```

Field Value

TYPE	DESCRIPTION
System.String	

TOOLS_MENU_ORIENTATION_MANAGER

Declaration

```
public const string TOOLS_MENU_ORIENTATION_MANAGER = "Tools/DoozyUI/Managers/Orientation Manager"
```

Field Value

TYPE	DESCRIPTION
System.String	

TOOLS_MENU_SCENE_LOADER

Declaration

```
public const string TOOLS_MENU_SCENE_LOADER = "Tools/DoozyUI/Managers/Scene Loader"
```

Field Value

TYPE	DESCRIPTION
System.String	

TOOLS_MENU_UIMANAGER

Declaration

```
public const string TOOLS_MENU_UIMANAGER = "Tools/DoozyUI/Managers/UI Manager"
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButtonsDatabase

Declaration

```
public static Dictionary<string, NamesDatabase> UIButtonsDatabase
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, DoozyUI.NamesDatabase>	

UIElementsDatabase

Declaration

```
public static Dictionary<string, NamesDatabase> UIElementsDatabase
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, DoozyUI.NamesDatabase>	

UISoundsDatabase

Declaration

public static List<UISound> UISoundsDatabase
--

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<DoozyUI.UISound>	

Properties

CanvasNames

Declaration

public static List<string> CanvasNames { get; }

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

DUISettings

Declaration

public static DUISettings DUISettings { get; }
--

Property Value

TYPE	DESCRIPTION
DUISettings	

RELATIVE_PATH_CANVAS_DATABASE

Declaration

public static string RELATIVE_PATH_CANVAS_DATABASE { get; }

Property Value

TYPE	DESCRIPTION
System.String	

RELATIVE_PATH_SETTINGS

Declaration

public static string RELATIVE_PATH_SETTINGS { get; }
--

Property Value

TYPE	DESCRIPTION
System.String	

RELATIVE_PATH_UIBUTTONS

Declaration

```
public static string RELATIVE_PATH_UIBUTTONS { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

RELATIVE_PATH_UIELEMENTS

Declaration

```
public static string RELATIVE_PATH_UIELEMENTS { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

RELATIVE_PATH_UISOUNDS

Declaration

```
public static string RELATIVE_PATH_UISOUNDS { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

UIButtonCategories

Declaration

```
public static List<string> UIButtonCategories { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

UIElementCategories

Declaration

```
public static List<string> UIElementCategories { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

UISoundNamesAll

Declaration

```
public static List<string> UISoundNamesAll { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

UISoundNamesUIButtons

Declaration

```
public static List<string> UISoundNamesUIButtons { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

UISoundNamesUIElements

Declaration

```
public static List<string> UISoundNamesUIElements { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

Methods

AddCanvasName(String)

Declaration

```
public static void AddCanvasName(string canvasName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	canvasName	

AddUIButtonName(String, String)

Declaration

```
public static void AddUIButtonName(string categoryName, string elementName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	
System.String	elementName	

AddUIElementName(String, String)

Declaration

```
public static void AddUIElementName(string categoryName, string elementName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	
System.String	elementName	

CreateAsset<T>(String, String, String)

Declaration

```
public static T CreateAsset<T>(string relativePath, string fileName, string extension = ".asset")where T : ScriptableObject
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	relativePath	
System.String	fileName	
System.String	extension	

Returns

TYPE	DESCRIPTION
T	

Type Parameters

NAME	DESCRIPTION
T	

CreatedUISettings()

Declaration

```
public static void CreatedUISettings()
```

CreateUIButtonsCategory(String)

Declaration

```
public static void CreateUIButtonsCategory(string categoryName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	

CreateUICategory(String)

Declaration

```
public static void CreateUICategory(string categoryName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	

CreateUISound(String, SoundType, AudioClip)

Declaration

```
public static void CreateUISound(string soundName, SoundType soundType, AudioClip audioClip = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	
SoundType	soundType	
AudioClip	audioClip	

DeleteUIButtonCategory(String)

Declaration

```
public static void DeleteUIButtonCategory(string categoryName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	

DeleteUICategory(String)

Declaration

```
public static void DeleteUICategory(string categoryName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	

DeleteUISound(String)

Declaration

```
public static void DeleteUISound(string soundName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	

GetCanvasNamesDatabase()

Declaration

```
public static NamesDatabase GetCanvasNamesDatabase()
```

Returns

TYPE	DESCRIPTION
NamesDatabase	

GetResource<T>(String, String)

Declaration

```
public static T GetResource<T>(string resourcesPath, string fileName)where T : ScriptableObject
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	resourcesPath	
System.String	fileName	

Returns

TYPE	DESCRIPTION
T	

Type Parameters

NAME	DESCRIPTION
T	

GetUIButtonNames(String)

Declaration

```
public static List<string> GetUIButtonNames(string categoryName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

GetUIButtonsDatabase(String)

Declaration

```
public static NamesDatabase GetUIButtonsDatabase(string category)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	category	

Returns

TYPE	DESCRIPTION
NamesDatabase	

GetUIElementNames(String)

Declaration

```
public static List<string> GetUIElementNames(string categoryName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

GetUIElementsDatabase(String)

Declaration

```
public static NamesDatabase GetUIElementsDatabase(string category)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	category	

Returns

TYPE	DESCRIPTION
NamesDatabase	

GetUISound(String)

Declaration

```
public static UISound GetUISound(string soundName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	

Returns

TYPE	DESCRIPTION
UISound	

RefreshCanvasNames()

Declaration

```
public static void RefreshCanvasNames()
```

RefreshCanvasNamesDatabase()

Declaration

```
public static void RefreshCanvasNamesDatabase()
```

RefreshUIButtonCategories()

Declaration

```
public static void RefreshUIButtonCategories()
```

RefreshUIButtonsDatabase()

Declaration

```
public static void RefreshUIButtonsDatabase()
```

RefreshUIElementCategories()

Declaration

```
public static void RefreshUIElementCategories()
```

RefreshUIElementsDatabase()

Declaration

```
public static void RefreshUIElementsDatabase()
```

RefreshUISoundNames()

Declaration

```
public static void RefreshUISoundNames()
```

RefreshUISoundsDatabase()

Declaration

```
public static void RefreshUISoundsDatabase()
```

RemoveUIButtonName(String, String)

Declaration

```
public static void RemoveUIButtonName(string categoryName, string elementName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	
System.String	elementName	

RemoveUIElementName(String, String)

Declaration

```
public static void RemoveUIElementName(string categoryName, string elementName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	
System.String	elementName	

RenameName(NamesDatabase, String, String)

Declaration

```
public static void RenameName(NamesDatabase targetDatabase, string oldName, string newName)
```

Parameters

TYPE	NAME	DESCRIPTION
NamesDatabase	targetDatabase	
System.String	oldName	
System.String	newName	

RenameUIButtonsCategory(String, String)

Declaration

```
public static void RenameUIButtonsCategory(string oldName, string newName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	oldName	
System.String	newName	

RenameUIElementsCategory(String, String)

Declaration

```
public static void RenameUIElementsCategory(string oldName, string newName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	oldName	
System.String	newName	

SortUIButtonCategories(String)

Declaration

```
public static void SortUIButtonCategories(string categoryName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	

SortUIElementsCategories(String)

Declaration

```
public static void SortUIElementsCategories(string categoryName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	

UIButtonCategoryExists(String)

Declaration

```
public static bool UIButtonCategoryExists(string categoryName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	

Returns

TYPE	DESCRIPTION
System.Boolean	

UIButtonNameExists(String, String)

Declaration

```
public static bool UIButtonNameExists(string categoryName, string elementName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	
System.String	elementName	

Returns

TYPE	DESCRIPTION
System.Boolean	

UIElementCategoryExists(String)

Declaration

```
public static bool UIElementCategoryExists(string categoryName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	

Returns

TYPE	DESCRIPTION
System.Boolean	

UIElementNameExists(String, String)

Declaration

```
public static bool UIElementNameExists(string categoryName, string elementName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	categoryName	
System.String	elementName	

Returns

TYPE	DESCRIPTION
System.Boolean	

UISoundNameExists(String, SoundType)

Declaration

```
public static bool UISoundNameExists(string soundName, SoundType soundType = SoundType.All)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	
SoundType	soundType	

Returns

TYPE	DESCRIPTION
System.Boolean	

Enum DUI.EventType

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public enum EventType
```

Fields

NAME	DESCRIPTION
ButtonClick	
ButtonDoubleClick	
ButtonLongClick	
GameEvent	

Class DUISettings

Inheritance

System.Object

DUISettings

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class DUISettings : ScriptableObject
```

Fields

HierarchyManager_Enabled

Declaration

```
public bool HierarchyManager_Enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_OrientationManager_ShowIcon

Declaration

```
public bool HierarchyManager_OrientationManager_ShowIcon
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_PlaymakerEventDispatcher_ShowIcon

Declaration

```
public bool HierarchyManager_PlaymakerEventDispatcher_ShowIcon
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_SceneLoader_ShowIcon

Declaration

```
public bool HierarchyManager_SceneLoader_ShowIcon
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_Soundy_ShowIcon

Declaration

public bool HierarchyManager_Soundy_ShowIcon
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UIButton_ShowButtonCategory

Declaration

public bool HierarchyManager_UIButton_ShowButtonCategory
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UIButton_ShowButtonName

Declaration

public bool HierarchyManager_UIButton_ShowButtonName
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UIButton_ShowIcon

Declaration

public bool HierarchyManager_UIButton_ShowIcon
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UICanvas_ShowCanvasName

Declaration

public bool HierarchyManager_UICanvas_ShowCanvasName
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UICanvas_ShowIcon

Declaration

public bool HierarchyManager_UICanvas_ShowIcon
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UICanvas_ShowSortingLayerNameAndOrder

Declaration

public bool HierarchyManager_UICanvas_ShowSortingLayerNameAndOrder
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UIEffect_ShowIcon

Declaration

public bool HierarchyManager_UIEffect_ShowIcon
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UIEffect_ShowSortingLayerNameAndOrder

Declaration

public bool HierarchyManager_UIEffect_ShowSortingLayerNameAndOrder
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UIElement_ShowElementCategory

Declaration

public bool HierarchyManager_UIElement_ShowElementCategory
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UIElement_ShowElementName

Declaration

public bool HierarchyManager_UIElement_ShowElementName
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UIElement_ShowIcon

Declaration

public bool HierarchyManager_UIElement_ShowIcon

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UIElement_ShowSortingLayerNameAndOrder

Declaration

public bool HierarchyManager_UIElement_ShowSortingLayerNameAndOrder

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UIManager_ShowIcon

Declaration

public bool HierarchyManager_UIManager_ShowIcon

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UINotification_ShowIcon

Declaration

public bool HierarchyManager_UINotification_ShowIcon
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UINotificationManager_ShowIcon

Declaration

public bool HierarchyManager_UINotificationManager_ShowIcon

Field Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UITrigger_ShowIcon

Declaration

public bool HierarchyManager_UITrigger_ShowIcon

Field Value

TYPE	DESCRIPTION
System.Boolean	

InternalSettings_ExecutedUpgrade

Declaration

public bool InternalSettings_ExecutedUpgrade
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_addToNavigationHistory

Declaration

public bool UIButton_addToNavigationHistory

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_allowMultipleClicks

Declaration

public bool UIButton_allowMultipleClicks
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_customOnClickSound

Declaration

```
public bool UIButton_customOnClickSound
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_customOnDoubleClickSound

Declaration

```
public bool UIButton_customOnDoubleClickSound
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_customOnLongClickSound

Declaration

```
public bool UIButton_customOnLongClickSound
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_customOnPointerDownSound

Declaration

```
public bool UIButton_customOnPointerDownSound
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_customOnPointerEnterSound

Declaration

```
public bool UIButton_customOnPointerEnterSound
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_customOnPointerExitSound

Declaration

public bool UIButton_customOnPointerExitSound

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_customOnPointerUpSound

Declaration

public bool UIButton_customOnPointerUpSound

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_deselectButtonOnClick

Declaration

public bool UIButton_deselectButtonOnClick
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_disableButtonInterval

Declaration

public float UIButton_disableButtonInterval

Field Value

TYPE	DESCRIPTION
System.Single	

UIButton_doubleClickRegisterInterval

Declaration

public float UIButton_doubleClickRegisterInterval

Field Value

TYPE	DESCRIPTION
System.Single	

UIButton_Inspector_HideNormalLoop

Declaration

public bool UIButton_Inspector_HideNormalLoop

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_Inspector_HideOnClick

Declaration

public bool UIButton_Inspector_HideOnClick
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_Inspector_HideOnDoubleClick

Declaration

public bool UIButton_Inspector_HideOnDoubleClick
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_Inspector_HideOnLongClick

Declaration

public bool UIButton_Inspector_HideOnLongClick
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_Inspector_HideOnPointerDown

Declaration

public bool UIButton_Inspector_HideOnPointerDown
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_Inspector_HideOnPointerEnter

Declaration

public bool UIButton_Inspector_HideOnPointerEnter

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_Inspector_HideOnPointerExit

Declaration

public bool UIButton_Inspector_HideOnPointerExit
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_Inspector_HideOnPointerUp

Declaration

public bool UIButton_Inspector_HideOnPointerUp
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_Inspector_HideSelectedLoop

Declaration

public bool UIButton_Inspector_HideSelectedLoop

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_Inspector_RenameGameObjectPrefix

Declaration

public string UIButton_Inspector_RenameGameObjectPrefix

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_Inspector_RenameGameObjectSuffix

Declaration

```
public string UIButton_Inspector_RenameGameObjectSuffix
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_Inspector_ShowButtonRenameGameObject

Declaration

```
public bool UIButton_Inspector_ShowButtonRenameGameObject
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_loadNormalLoopPresetAtRuntime

Declaration

```
public bool UIButton_loadNormalLoopPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_loadOnClickPunchPresetAtRuntime

Declaration

```
public bool UIButton_loadOnClickPunchPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_loadOnDoubleClickPunchPresetAtRuntime

Declaration

```
public bool UIButton_loadOnDoubleClickPunchPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_loadOnLongClickPunchPresetAtRuntime

Declaration

```
public bool UIButton_loadOnLongClickPunchPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_loadOnPointerDownPunchPresetAtRuntime

Declaration

```
public bool UIButton_loadOnPointerDownPunchPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_loadOnPointerEnterPunchPresetAtRuntime

Declaration

```
public bool UIButton_loadOnPointerEnterPunchPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_loadOnPointerExitPunchPresetAtRuntime

Declaration

```
public bool UIButton_loadOnPointerExitPunchPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_loadOnPointerUpPunchPresetAtRuntime

Declaration

```
public bool UIButton_loadOnPointerUpPunchPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_loadSelectedLoopPresetAtRuntime

Declaration

```
public bool UIButton_loadSelectedLoopPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_longClickRegisterInterval

Declaration

```
public float UIButton_longClickRegisterInterval
```

Field Value

TYPE	DESCRIPTION
System.Single	

UIButton_normalLoopPresetCategory

Declaration

```
public string UIButton_normalLoopPresetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_normalLoopPresetName

Declaration

```
public string UIButton_normalLoopPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onClickPunchPresetCategory

Declaration

```
public string UIButton_onClickPunchPresetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onClickPunchPresetName

Declaration

```
public string UIButton_onClickPunchPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onClickSound

Declaration

```
public string UIButton_onClickSound
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onDoubleClickPunchPresetCategory

Declaration

```
public string UIButton_onDoubleClickPunchPresetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onDoubleClickPunchPresetName

Declaration

```
public string UIButton_onDoubleClickPunchPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onDoubleClickSound

Declaration

```
public string UIButton_onDoubleClickSound
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onLongClickPunchPresetCategory

Declaration

```
public string UIButton_onLongClickPunchPresetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onLongClickPunchPresetName

Declaration

```
public string UIButton_onLongClickPunchPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onLongClickSound

Declaration

```
public string UIButton_onLongClickSound
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onPointerDownPunchPresetCategory

Declaration

```
public string UIButton_onPointerDownPunchPresetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onPointerDownPunchPresetName

Declaration

```
public string UIButton_onPointerDownPunchPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onPointerDownSound

Declaration

public string UIButton_onPointerDownSound

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onPointerEnterDisableInterval

Declaration

public float UIButton_onPointerEnterDisableInterval

Field Value

TYPE	DESCRIPTION
System.Single	

UIButton_onPointerEnterPunchPresetCategory

Declaration

public string UIButton_onPointerEnterPunchPresetCategory
--

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onPointerEnterPunchPresetName

Declaration

public string UIButton_onPointerEnterPunchPresetName
--

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onPointerEnterSound

Declaration

public string UIButton_onPointerEnterSound
--

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onPointerExitDisableInterval

Declaration

public float UIButton_onPointerExitDisableInterval
--

Field Value

TYPE	DESCRIPTION
System.Single	

UIButton_onPointerExitPunchPresetCategory

Declaration

public string UIButton_onPointerExitPunchPresetCategory

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onPointerExitPunchPresetName

Declaration

public string UIButton_onPointerExitPunchPresetName

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onPointerExitSound

Declaration

public string UIButton_onPointerExitSound

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onPointerUpPunchPresetCategory

Declaration

public string UIButton_onPointerUpPunchPresetCategory

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onPointerUpPunchPresetName

Declaration

public string UIButton_onPointerUpPunchPresetName

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_onPointerUpSound

Declaration

public string UIButton_onPointerUpSound

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_selectedLoopPresetCategory

Declaration

public string UIButton_selectedLoopPresetCategory

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_selectedLoopPresetName

Declaration

public string UIButton_selectedLoopPresetName

Field Value

TYPE	DESCRIPTION
System.String	

UIButton_singleClickMode

Declaration

public UIButton.SingleClickMode UIButton_singleClickMode
--

Field Value

TYPE	DESCRIPTION
UIButton.SingleClickMode	

UIButton_useOnClickAnimations

Declaration

public bool UIButton_useOnClickAnimations

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_useOnDoubleClick

Declaration

public bool UIButton_useOnDoubleClick

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_useOnLongClick

Declaration

public bool UIButton_useOnLongClick

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_useOnPointerDown

Declaration

public bool UIButton_useOnPointerDown

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_useOnPointerEnter

Declaration

public bool UIButton_useOnPointerEnter
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_useOnPointerExit

Declaration

public bool UIButton_useOnPointerExit

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_useOnPointerUp

Declaration

public bool UIButton_useOnPointerUp

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_waitForOnClickAnimation

Declaration

public bool UIButton_waitForOnClickAnimation
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_waitForOnDoubleClickAnimation

Declaration

public bool UIButton_waitForOnDoubleClickAnimation
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIButton_waitForOnLongClickAnimation

Declaration

public bool UIButton_waitForOnLongClickAnimation
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIEffect_customOrderInLayer

Declaration

```
public int UIEffect_customOrderInLayer
```

Field Value

TYPE	DESCRIPTION
System.Int32	

UIEffect_customSortingLayerName

Declaration

```
public string UIEffect_customSortingLayerName
```

Field Value

TYPE	DESCRIPTION
System.String	

UIEffect_effectPosition

Declaration

```
public UIEffect.EffectPosition UIEffect_effectPosition
```

Field Value

TYPE	DESCRIPTION
UIEffect.EffectPosition	

UIEffect_Inspector_RenameGameObjectPrefix

Declaration

```
public string UIEffect_Inspector_RenameGameObjectPrefix
```

Field Value

TYPE	DESCRIPTION
System.String	

UIEffect_Inspector_RenameGameObjectSuffix

Declaration

```
public string UIEffect_Inspector_RenameGameObjectSuffix
```

Field Value

TYPE	DESCRIPTION
System.String	

UIEffect_Inspector_ShowButtonRenameGameObject

Declaration

public bool UIEffect_Inspector_ShowButtonRenameGameObject

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIEffect_playOnAwake

Declaration

public bool UIEffect_playOnAwake

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIEffect_sortingOrderStep

Declaration

public int UIEffect_sortingOrderStep

Field Value

TYPE	DESCRIPTION
System.Int32	

UIEffect_stopInstantly

Declaration

public bool UIEffect_stopInstantly

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIEffect_useCustomOrderInLayer

Declaration

public bool UIEffect_useCustomOrderInLayer
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIEffect_useCustomSortingLayerName

Declaration

public bool UIEffect_useCustomSortingLayerName
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIElement_animateAtStart

Declaration

public bool UIElement_animateAtStart

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIElement_customStartAnchoredPosition

Declaration

public Vector3 UIElement_customStartAnchoredPosition
--

Field Value

TYPE	DESCRIPTION
Vector3	

UIElement_disableWhenHidden

Declaration

public bool UIElement_disableWhenHidden

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIElement_executeLayoutFix

Declaration

public bool UIElement_executeLayoutFix
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIElement_inAnimationsPresetCategoryName

Declaration

public string UIElement_inAnimationsPresetCategoryName
--

Field Value

TYPE	DESCRIPTION
System.String	

UIElement_inAnimationsPresetName

Declaration

public string UIElement_inAnimationsPresetName
--

Field Value

TYPE	DESCRIPTION
System.String	

UIElement_Inspector_HideLoopAnimations

Declaration

public bool UIElement_Inspector_HideLoopAnimations
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIElement_Inspector_RenameGameObjectPrefix

Declaration

public string UIElement_Inspector_RenameGameObjectPrefix
--

Field Value

TYPE	DESCRIPTION
System.String	

UIElement_Inspector_RenameGameObjectSuffix

Declaration

public string UIElement_Inspector_RenameGameObjectSuffix
--

Field Value

TYPE	DESCRIPTION
System.String	

UIElement_Inspector_ShowButtonRenameGameObject

Declaration

public bool UIElement_Inspector_ShowButtonRenameGameObject
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIElement_LANDSCAPE

Declaration

public bool UIElement_LANDSCAPE

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIElement_loadInAnimationsPresetAtRuntime

Declaration

public bool UIElement_loadInAnimationsPresetAtRuntime

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIElement_loadLoopAnimationsPresetAtRuntime

Declaration

public bool UIElement_loadLoopAnimationsPresetAtRuntime

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIElement_loadOutAnimationsPresetAtRuntime

Declaration

public bool UIElement_loadOutAnimationsPresetAtRuntime
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIElement_loopAnimationsPresetCategoryName

Declaration

```
public string UIElement_loopAnimationsPresetCategoryName
```

Field Value

TYPE	DESCRIPTION
System.String	

UIElement_loopAnimationsPresetName

Declaration

```
public string UIElement_loopAnimationsPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

UIElement_outAnimationsPresetCategoryName

Declaration

```
public string UIElement_outAnimationsPresetCategoryName
```

Field Value

TYPE	DESCRIPTION
System.String	

UIElement_outAnimationsPresetName

Declaration

```
public string UIElement_outAnimationsPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

UIElement_PORTRAIT

Declaration

```
public bool UIElement_PORTRAIT
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIElement_startHidden

Declaration

public bool UIElement_startHidden

Field Value

TYPE	DESCRIPTION
System.Boolean	

UIElement_useCustomStartAnchoredPosition

Declaration

public bool UIElement_useCustomStartAnchoredPosition
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

Properties

HierarchyManager_UIButton_Enabled

Declaration

public bool HierarchyManager_UIButton_Enabled { get; }
--

Property Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UICanvas_Enabled

Declaration

public bool HierarchyManager_UICanvas_Enabled { get; }
--

Property Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UIEffect_Enabled

Declaration

public bool HierarchyManager_UIEffect_Enabled { get; }
--

Property Value

TYPE	DESCRIPTION
System.Boolean	

HierarchyManager_UIElement_Enabled

Declaration

```
public bool HierarchyManager_UIElement_Enabled { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Class EditorNavigationPointer

Inheritance

System.Object
EditorNavigationPointer

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class EditorNavigationPointer
```

Constructors

EditorNavigationPointer(Int32, Int32)

Declaration

```
public EditorNavigationPointer(int CategoryIndex, int NameIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	CategoryIndex	
System.Int32	NameIndex	

Fields

categoryIndex

Declaration

```
public int categoryIndex
```

Field Value

TYPE	DESCRIPTION
System.Int32	

nameIndex

Declaration

```
public int nameIndex
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

Copy()

Declaration

```
public EditorNavigationPointer Copy()
```

Returns

TYPE	DESCRIPTION
EditorNavigationPointer	

Class EditorNavigationPointerData

Inheritance

System.Object
EditorNavigationPointerData

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class EditorNavigationPointerData
```

Constructors

EditorNavigationPointerData()

Declaration

```
public EditorNavigationPointerData()
```

Fields

hideIndex

Declaration

```
public List<EditorNavigationPointer> hideIndex
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<DoozyUI.EditorNavigationPointer>	

showIndex

Declaration

```
public List<EditorNavigationPointer> showIndex
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<DoozyUI.EditorNavigationPointer>	

Methods

Copy()

Declaration

public EditorNavigationPointerData Copy()

Returns

TYPE	DESCRIPTION
EditorNavigationPointerData	

Class Fade

Animation settings for Fade (alpha value)

Inheritance

System.Object
Fade

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class Fade
```

Constructors

Fade(Anim.AnimationType)

Declaration

```
public Fade(Anim.AnimationType aType)
```

Parameters

TYPE	NAME	DESCRIPTION
Anim.AnimationType	aType	

Fields

alpha

Depending on the animation type, this is considered either the TO or the FROM scale.

Declaration

```
public float alpha
```

Field Value

TYPE	DESCRIPTION
System.Single	

animationCurve

If the easeType is set to AnimationCurve, this will be used in order to calculate the rate of change of the animation over time.

Declaration

```
public AnimationCurve animationCurve
```


Field Value

TYPE	DESCRIPTION
AnimationCurve	

animationType

Select if this data is for an IN or an OUT animation.

Declaration

public Anim.AnimationType animationType

Field Value

TYPE	DESCRIPTION
Anim.AnimationType	

duration

The duration of the animation.

Declaration

public float duration

Field Value

TYPE	DESCRIPTION
System.Single	

ease

Sets the ease of the tween. Easing functions specify the rate of change of a parameter over time.

To see how default ease curves look, check out easings.net

Declaration

public Ease ease

Field Value

TYPE	DESCRIPTION
Ease	

easeType

Use an Ease or an AnimationCurve in order to calculate the rate of change of the animation over time.

Declaration

public UIAnimator.EaseType easeType

Field Value

TYPE	DESCRIPTION
UIAnimator.EaseType	

enabled

If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default: false).

Declaration

public bool enabled

Field Value

TYPE	DESCRIPTION
System.Boolean	

startDelay

Start delay for the animation.

Declaration

public float startDelay

Field Value

TYPE	DESCRIPTION
System.Single	

Properties

TotalDuration

Declaration

public float TotalDuration { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Copy()

Declaration

public Fade Copy()

Returns

TYPE	DESCRIPTION
Fade	

Reset(Anim.AnimationType)

Declaration

```
public void Reset(Anim.AnimationType aType)
```

Parameters

TYPE	NAME	DESCRIPTION
Anim.AnimationType	aType	

Reverse()

Declaration

```
public Fade Reverse()
```

Returns

TYPE	DESCRIPTION
Fade	

UpdateValues(Fade)

Declaration

```
public void UpdateValues(Fade f)
```

Parameters

TYPE	NAME	DESCRIPTION
Fade	f	

Class FadeLoop

Inheritance

System.Object
FadeLoop

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class FadeLoop
```

Fields

animationCurve

If the easeType is set to AnimationCurve, this will be used in order to calculate the rate of change of the animation over time.

Declaration

```
public AnimationCurve animationCurve
```

Field Value

TYPE	DESCRIPTION
AnimationCurve	

DEFAULT_MAX

Declaration

```
public static float DEFAULT_MAX
```

Field Value

TYPE	DESCRIPTION
System.Single	

DEFAULT_MIN

Declaration

```
public static float DEFAULT_MIN
```

Field Value

TYPE	DESCRIPTION
System.Single	

duration

The duration of the animation.

Declaration

```
public float duration
```

Field Value

TYPE	DESCRIPTION
System.Single	

ease

Sets the ease of the tween. Easing functions specify the rate of change of a parameter over time.

To see how default ease curves look, check out easings.net

Declaration

```
public Ease ease
```

Field Value

TYPE	DESCRIPTION
Ease	

easeType

Use an Ease or an AnimationCurve in order to calculate the rate of change of the animation over time.

Declaration

```
public UIManimator.EaseType easeType
```

Field Value

TYPE	DESCRIPTION
UIManimator.EaseType	

enabled

If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default: false).

Declaration

```
public bool enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

loops

Number of loops (-1 = infinite loops).

Declaration

```
public int loops
```

Field Value

TYPE	DESCRIPTION
System.Int32	

loopType

Types of loop.

Declaration

```
public Loop.LoopType loopType
```

Field Value

TYPE	DESCRIPTION
Loop.LoopType	

max

The maximum alpha value for the fade animation loop (default: 1).

Declaration

```
public float max
```

Field Value

TYPE	DESCRIPTION
System.Single	

min

The minimum alpha value for the fade animation loop (default: 0).

Declaration

```
public float min
```

Field Value

TYPE	DESCRIPTION
System.Single	

startDelay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

Properties

TotalDuration

Declaration

```
public float TotalDuration { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Copy()

Declaration

```
public FadeLoop Copy()
```

Returns

TYPE	DESCRIPTION
FadeLoop	

Reset()

Declaration

```
public void Reset()
```

Class Loop

Base class for all the Loop animations.

Inheritance

System.Object
Loop

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class Loop
```

Constructors

Loop()

Declaration

```
public Loop()
```

Fields

autoStart

This deternimes if the loop should start from the get go (after being initialized) or on demand.

Declaration

```
public bool autoStart
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

fade

Declaration

```
public FadeLoop fade
```

Field Value

TYPE	DESCRIPTION
FadeLoop	

move

Declaration

```
public MoveLoop move
```

Field Value

TYPE	DESCRIPTION
MoveLoop	

rotate

Declaration

```
public RotateLoop rotate
```

Field Value

TYPE	DESCRIPTION
RotateLoop	

scale

Declaration

```
public ScaleLoop scale
```

Field Value

TYPE	DESCRIPTION
ScaleLoop	

Properties

Enabled

Declaration

```
public bool Enabled { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

TotalDuration

Declaration

```
public float TotalDuration { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Copy()

Declaration

```
public Loop Copy()
```

Returns

TYPE	DESCRIPTION
Loop	

GetLoopType(DG.Tweening.LoopType)

Declaration

```
public static Loop.LoopType GetLoopType(DG.Tweening.LoopType loopType)
```

Parameters

TYPE	NAME	DESCRIPTION
DG.Tweening.LoopType	loopType	

Returns

TYPE	DESCRIPTION
Loop.LoopType	

GetLoopType(Loop.LoopType)

Declaration

```
public static DG.Tweening.LoopType GetLoopType(Loop.LoopType loopType)
```

Parameters

TYPE	NAME	DESCRIPTION
Loop.LoopType	loopType	

Returns

TYPE	DESCRIPTION
DG.Tweening.LoopType	

Reset()

Declaration

```
public void Reset()
```

Enum Loop.LoopType

Types of loop

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public enum LoopType
```

Fields

NAME	DESCRIPTION
Restart	Each loop cycle restarts from the beginning
Yoyo	The tween moves forward and backwards at alternate cycles

Class LoopData

Inheritance

System.Object

LoopData

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class LoopData : ScriptableObject
```

Constructors

LoopData()

Declaration

```
public LoopData()
```

Fields

data

Declaration

```
public Loop data
```

Field Value

TYPE	DESCRIPTION
Loop	

presetCategory

Declaration

```
public string presetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

presetName

Declaration

```
public string presetName
```

Field Value

TYPE	DESCRIPTION
System.String	

Properties

LoadDefaultValues

Declaration

```
public bool LoadDefaultValues { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Class Move

Animation settings for Movement

Inheritance

System.Object

Move

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class Move
```

Constructors

Move(Anim.AnimationType)

Declaration

```
public Move(Anim.AnimationType aType)
```

Parameters

TYPE	NAME	DESCRIPTION
Anim.AnimationType	aType	

Fields

animationCurve

If the easeType is set to AnimationCurve, this will be used in order to calculate the rate of change of the animation over time.

Declaration

```
public AnimationCurve animationCurve
```

Field Value

TYPE	DESCRIPTION
AnimationCurve	

animationType

Select if this data is for an IN or an OUT animation.

Declaration

```
public Anim.AnimationType animationType
```

Field Value

TYPE	DESCRIPTION
Anim.AnimationType	

customPosition

Depending on the animation type, this is considered either the TO or the FROM position (when moveDirection is set to CustomPosition).

Declaration

<code>public Vector3 customPosition</code>
--

Field Value

TYPE	DESCRIPTION
Vector3	

duration

The duration of the animation.

Declaration

<code>public float duration</code>

Field Value

TYPE	DESCRIPTION
System.Single	

ease

Sets the ease of the tween. Easing functions specify the rate of change of a parameter over time.

To see how default ease curves look, check out easings.net

Declaration

<code>public Ease ease</code>

Field Value

TYPE	DESCRIPTION
Ease	

easeType

Use an Ease or an AnimationCurve in order to calculate the rate of change of the animation over time.

Declaration

<code>public UIManipulator.EaseType easeType</code>

Field Value

TYPE	DESCRIPTION
UIAnimator.EaseType	

enabled

If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default: false).

Declaration

<code>public bool enabled</code>

Field Value

TYPE	DESCRIPTION
System.Boolean	

moveDirection

Depending on the animation type, the direction is considered either IN (eg. Move IN from Left) or OUT (eg. Move OUT to Left)

Declaration

<code>public Move.MoveDirection moveDirection</code>
--

Field Value

TYPE	DESCRIPTION
Move.MoveDirection	

startDelay

Start delay for the animation.

Declaration

<code>public float startDelay</code>

Field Value

TYPE	DESCRIPTION
System.Single	

Properties

TotalDuration

Declaration

<code>public float TotalDuration { get; }</code>
--

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Copy()

Declaration

```
public Move Copy()
```

Returns

TYPE	DESCRIPTION
Move	

Reset(Anim.AnimationType, Move.MoveDirection)

Declaration

```
public void Reset(Anim.AnimationType aType, Move.MoveDirection mDirection = Move.MoveDirection.Left)
```

Parameters

TYPE	NAME	DESCRIPTION
Anim.AnimationType	aType	
Move.MoveDirection	mDirection	

Reverse()

Declaration

```
public Move Reverse()
```

Returns

TYPE	DESCRIPTION
Move	

Reverse(Move.MoveDirection)

Declaration

```
public static Move.MoveDirection Reverse(Move.MoveDirection moveDirection)
```

Parameters

TYPE	NAME	DESCRIPTION
Move.MoveDirection	moveDirection	

Returns

TYPE	DESCRIPTION
Move.MoveDirection	

UpdateValues(Move)

Declaration

```
public void UpdateValues(Move m)
```

Parameters

TYPE	NAME	DESCRIPTION
Move	m	

Enum Move.MoveDirection

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public enum MoveDirection
```

Fields

NAME	DESCRIPTION
Bottom	
BottomCenter	
BottomLeft	
BottomRight	
CustomPosition	
Left	
MiddleCenter	
MiddleLeft	
MiddleRight	
Right	
Top	
TopCenter	
TopLeft	
TopRight	

Class MoveLoop

Inheritance

System.Object
MoveLoop

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class MoveLoop
```

Fields

animationCurve

If the easeType is set to AnimationCurve, this will be used in order to calculate the rate of change of the animation over time.

Declaration

```
public AnimationCurve animationCurve
```

Field Value

TYPE	DESCRIPTION
AnimationCurve	

duration

The duration of the animation.

Declaration

```
public float duration
```

Field Value

TYPE	DESCRIPTION
System.Single	

ease

Sets the ease of the tween. Easing functions specify the rate of change of a parameter over time.

To see how default ease curves look, check out [easings.net](#)

Declaration

```
public Ease ease
```

Field Value

TYPE	DESCRIPTION
Ease	

easeType

Use an Ease or an AnimationCurve in order to calculate the rate of change of the animation over time.

Declaration

```
public UIManimator.EaseType easeType
```

Field Value

TYPE	DESCRIPTION
UIManimator.EaseType	

enabled

If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default: false).

Declaration

```
public bool enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

loops

Number of loops (-1 = infinite loops).

Declaration

```
public int loops
```

Field Value

TYPE	DESCRIPTION
System.Int32	

loopType

Types of loop.

Declaration

```
public Loop.LoopType loopType
```

Field Value

TYPE	DESCRIPTION
Loop.LoopType	

movement

This movement is calculated startAnchoredPosition-movement for min and startAnchoredPosition+movment for max

Declaration

```
public Vector3 movement
```

Field Value

TYPE	DESCRIPTION
Vector3	

startDelay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

Properties

TotalDuration

Declaration

```
public float TotalDuration { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Copy()

Declaration

```
public MoveLoop Copy()
```

Returns

TYPE	DESCRIPTION
MoveLoop	

Reset()

Declaration

```
public void Reset()
```

Class NamesDatabase

Inheritance

System.Object

NamesDatabase

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class NamesDatabase : ScriptableObject
```

Fields

data

Declaration

```
public List<string> data
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

Properties

Count

Declaration

```
public int Count { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

IsEmpty

Declaration

```
public bool IsEmpty { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

IsNull

Declaration

```
public bool IsNull { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

Add(String)

Declaration

```
public void Add(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	

Clear()

Declaration

```
public void Clear()
```

Contains(String)

Declaration

```
public bool Contains(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	

Returns

TYPE	DESCRIPTION
System.Boolean	

GetName(Int32)

Declaration

```
public string GetName(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

Returns

TYPE	DESCRIPTION
System.String	

IndexOf(String)

Declaration

```
public int IndexOf(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	

Returns

TYPE	DESCRIPTION
System.Int32	

Init()

Declaration

```
public void Init()
```

Remove(String)

Declaration

```
public void Remove(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	

RemoveAt(Int32)

Declaration

```
public void RemoveAt(int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

RemoveEmpty()

Declaration

```
public void RemoveEmpty()
```

Reverse()

Declaration

```
public void Reverse()
```

Sort()

Declaration

```
public void Sort()
```

ToArray()

Declaration

```
public string[] ToArray()
```

Returns

TYPE	DESCRIPTION
System.String[]	

Class NavigationPointer

Helper class for the UI Navigation.

Inheritance

System.Object
NavigationPointer

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class NavigationPointer
```

Constructors

NavigationPointer()

Declaration

```
public NavigationPointer()
```

NavigationPointer(String, String)

Declaration

```
public NavigationPointer(string Category, string Name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	Category	
System.String	Name	

Fields

category

Element Category

Declaration

```
public string category
```

Field Value

TYPE	DESCRIPTION
System.String	

name

Element Name

Declaration

```
public string name
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

Copy()

Declaration

```
public NavigationPointer Copy()
```

Returns

TYPE	DESCRIPTION
NavigationPointer	

Class NavigationPointerData

Helper class for the UINavigation.

Inheritance

System.Object
NavigationPointerData

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class NavigationPointerData
```

Constructors

NavigationPointerData()

Declaration

```
public NavigationPointerData()
```

NavigationPointerData(Boolean)

Declaration

```
public NavigationPointerData(bool AddToNavigationHitory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	AddToNavigationHitory	

Fields

addToNavigationHistory

Should the Navigation Pointers from the Show list be added to the UI Navigation History? Default is set to false.

Declaration

```
public bool addToNavigationHistory
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

hide

A list of Navigation Pointers used for all the UIElements that need to he hidden.

Declaration

```
public List<NavigationPointer> hide
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<DoozyUI.NavigationPointer>	

show

A list of Navigation Pointers used for all the UIElements that need to be shown.

Declaration

```
public List<NavigationPointer> show
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<DoozyUI.NavigationPointer>	

Methods

Copy()

Declaration

```
public NavigationPointerData Copy()
```

Returns

TYPE	DESCRIPTION
NavigationPointerData	

Class OrientationManager

Inheritance

System.Object
QuickEngine.Common.Singleton<DoozyUI.OrientationManager>
OrientationManager

Inherited Members

[Singleton<OrientationManager>.Instance](#)
[Singleton<OrientationManager>.OnDestroy\(\)](#)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class OrientationManager : Singleton<OrientationManager>
```

Constructors

OrientationManager()

Declaration

```
protected OrientationManager()
```

Fields

debug

Prints to Debug.Log all the relevant functionality informations needed for debug purposes

Declaration

```
public bool debug
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

onOrientationChange

UnityEvent that sends an OrientaionManager.Orientation parameter when the device's orientation changes.

Declaration

```
public OrientationManager.OrientationChange onOrientationChange
```

Field Value

TYPE	DESCRIPTION
OrientationManager.OrientationChange	

Properties

Canvas

Declaration

```
public Canvas Canvas { get; }
```


Property Value

TYPE	DESCRIPTION
Canvas	

CurrentOrientation

Retruns the current orientation of the device.

Declaration

```
public OrientationManager.Orientation CurrentOrientation { get; }
```

Property Value

TYPE	DESCRIPTION
OrientationManager.Orientation	

RectTransform

Declaration

```
public RectTransform RectTransform { get; }
```

Property Value

TYPE	DESCRIPTION
RectTransform	

Methods

AddOrientationManagerToScene()

Declaration

```
public static OrientationManager AddOrientationManagerToScene()
```

Returns

TYPE	DESCRIPTION
OrientationManager	

ChangeOrientation(OrientationManager.Orientation)

Updates the currentOrientation to the specified value and sends an UnityEvent to signal the change.

Declaration

```
public void ChangeOrientation(OrientationManager.Orientation newOrientation)
```

Parameters

TYPE	NAME	DESCRIPTION
OrientationManager.Orientation	newOrientation	

CheckDeviceOrientation()

Checks the current orientation and updates it if it changed since the last check. You do not need to call this yourself as this is called automatically by the OrientationManager in the most efficient way.

Declaration

```
public void CheckDeviceOrientation()
```

Enum OrientationManager.Orientation

Orientation type

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public enum Orientation
```

Fields

NAME	DESCRIPTION
Landscape	Landscape mode
Portrait	Portrait mode
Unknown	Unknown mode. Used for calibration purposes

Class OrientationManager.OrientationChange

Inheritance

System.Object

OrientationManager.OrientationChange

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class OrientationChange : UnityEvent<OrientationManager.Orientation>
```

Class PlaymakerEventDispatcher

Inheritance

System.Object

PlaymakerEventDispatcher

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class PlaymakerEventDispatcher : MonoBehaviour
```

Class Punch

Base class for all the Punch animations.

Inheritance

System.Object
Punch

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class Punch
```

Constructors

Punch()

Declaration

```
public Punch()
```

Fields

DEFAULT_PUNCH_DURATION

Declaration

```
public const float DEFAULT_PUNCH_DURATION = 0.3F
```

Field Value

TYPE	DESCRIPTION
System.Single	

DEFAULT_PUNCH_ELASTICITY

Declaration

```
public const float DEFAULT_PUNCH_ELASTICITY = 0.5F
```

Field Value

TYPE	DESCRIPTION
System.Single	

DEFAULT_PUNCH_MOVE_PUNCH

Declaration

```
public static Vector3 DEFAULT_PUNCH_MOVE_PUNCH
```

Field Value

TYPE	DESCRIPTION
Vector3	

DEFAULT_PUNCH_ROTATE_PUNCH

Declaration

```
public static Vector3 DEFAULT_PUNCH_ROTATE_PUNCH
```

Field Value

TYPE	DESCRIPTION
Vector3	

DEFAULT_PUNCH_SCALE_PUNCH

Declaration

```
public static Vector3 DEFAULT_PUNCH_SCALE_PUNCH
```

Field Value

TYPE	DESCRIPTION
Vector3	

DEFAULT_PUNCH_START_DELAY

Declaration

```
public const float DEFAULT_PUNCH_START_DELAY = 0F
```

Field Value

TYPE	DESCRIPTION
System.Single	

DEFAULT_PUNCH_VIBRATO

Declaration

```
public const int DEFAULT_PUNCH_VIBRATO = 4
```

Field Value

TYPE	DESCRIPTION
System.Int32	

move

Declaration

```
public PunchMove move
```

Field Value

TYPE	DESCRIPTION
PunchMove	

rotate

Declaration

```
public PunchRotate rotate
```

Field Value

TYPE	DESCRIPTION
PunchRotate	

scale

Declaration

```
public PunchScale scale
```

Field Value

TYPE	DESCRIPTION
PunchScale	

Properties

Enabled

Declaration

```
public bool Enabled { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

TotalDuration

Declaration

```
public float TotalDuration { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Copy()

Declaration


```
public Punch Copy()
```

Returns

TYPE	DESCRIPTION
Punch	

Reset()

Declaration

```
public void Reset()
```

Class PunchData

Inheritance

System.Object

PunchData

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class PunchData : ScriptableObject
```

Constructors

PunchData()

Declaration

```
public PunchData()
```

Fields

data

Declaration

```
public Punch data
```

Field Value

TYPE	DESCRIPTION
Punch	

presetCategory

Declaration

```
public string presetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

presetName

Declaration

```
public string presetName
```

Field Value

TYPE	DESCRIPTION
System.String	

Properties

LoadDefaultValues

Declaration

```
public bool LoadDefaultValues { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Class PunchMove

Punches a Transform's anchoredPosition towards the given direction and then back to the starting one as if it was connected to the starting scale via an elastic.

Inheritance

System.Object
PunchMove

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class PunchMove
```

Constructors

PunchMove()

Declaration

```
public PunchMove()
```

Fields

duration

The duration of the animation.

Declaration

```
public float duration
```

Field Value

TYPE	DESCRIPTION
System.Single	

elasticity

Represents how much (0 to 1) the vector will go beyond the starting position when bouncing backwards. 1 creates a full oscillation between the punch position and the opposite position, while 0 oscillates only between the punch position and the start position.

Declaration

```
public float elasticity
```

Field Value

TYPE	DESCRIPTION
System.Single	

enabled

If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default: false).

Declaration

```
public bool enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

punch

The punch strength (added to the Transform's current position).

Declaration

```
public Vector3 punch
```

Field Value

TYPE	DESCRIPTION
Vector3	

startDelay

Start delay for the animation.

Declaration

```
public float startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

vibrato

Indicates how much will the punch vibrate.

Declaration

```
public int vibrato
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Properties

TotalDuration

Declaration

```
public float TotalDuration { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Copy()

Declaration

```
public PunchMove Copy()
```

Returns

TYPE	DESCRIPTION
PunchMove	

Reset()

Declaration

```
public void Reset()
```

UpdateValues(PunchMove)

Declaration

```
public void UpdateValues(PunchMove p)
```

Parameters

TYPE	NAME	DESCRIPTION
PunchMove	p	

Class PunchRotate

Punches a Transform's localRotation towards the given rotation and then back to the starting one as if it was connected to the starting scale via an elastic.

Inheritance

System.Object
PunchRotate

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class PunchRotate
```

Constructors

PunchRotate()

Declaration

```
public PunchRotate()
```

Fields

duration

The duration of the animation.

Declaration

```
public float duration
```

Field Value

TYPE	DESCRIPTION
System.Single	

elasticity

Represents how much (0 to 1) the vector will go beyond the starting size when bouncing backwards. 1 creates a full oscillation between the punch scale and the opposite scale, while 0 oscillates only between the punch scale and the start scale.

Declaration

```
public float elasticity
```

Field Value

TYPE	DESCRIPTION
System.Single	

enabled

If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default: false).

Declaration

```
public bool enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

punch

The punch strength (added to the Transform's current position / rotation / scale).

Declaration

```
public Vector3 punch
```

Field Value

TYPE	DESCRIPTION
Vector3	

startDelay

Start delay for the animation.

Declaration

```
public float startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

vibrato

Indicates how much will the punch vibrate.

Declaration

```
public int vibrato
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Properties

TotalDuration

Declaration

```
public float TotalDuration { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Copy()

Declaration

```
public PunchRotate Copy()
```

Returns

TYPE	DESCRIPTION
PunchRotate	

Reset()

Declaration

```
public void Reset()
```

UpdateValues(PunchRotate)

Declaration

```
public void UpdateValues(PunchRotate r)
```

Parameters

TYPE	NAME	DESCRIPTION
PunchRotate	r	

Class PunchScale

Punches a Transform's localScale towards the given size and then back to the starting one as if it was connected to the starting scale via an elastic.

Inheritance

System.Object
PunchScale

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class PunchScale
```

Constructors

PunchScale()

Declaration

```
public PunchScale()
```

Fields

duration

The duration of the animation.

Declaration

```
public float duration
```

Field Value

TYPE	DESCRIPTION
System.Single	

elasticity

Represents how much (0 to 1) the vector will go beyond the starting size when bouncing backwards. 1 creates a full oscillation between the punch scale and the opposite scale, while 0 oscillates only between the punch scale and the start scale.

Declaration

```
public float elasticity
```

Field Value

TYPE	DESCRIPTION
System.Single	

enabled

If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default: false).

Declaration

```
public bool enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

punch

The punch strength (added to the Transform's current position / rotation / scale).

Declaration

```
public Vector3 punch
```

Field Value

TYPE	DESCRIPTION
Vector3	

startDelay

Start delay for the animation.

Declaration

```
public float startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

vibrato

Indicates how much will the punch vibrate.

Declaration

```
public int vibrato
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Properties

TotalDuration

Declaration

```
public float TotalDuration { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Copy()

Declaration

```
public PunchScale Copy()
```

Returns

TYPE	DESCRIPTION
PunchScale	

Reset()

Declaration

```
public void Reset()
```

UpdateValues(PunchScale)

Declaration

```
public void UpdateValues(PunchScale s)
```

Parameters

TYPE	NAME	DESCRIPTION
PunchScale	s	

Class RadialLayout

Inheritance

System.Object

RadialLayout

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class RadialLayout : LayoutGroup
```

Fields

fDistance

Declaration

```
public float fDistance
```

Field Value

TYPE	DESCRIPTION
System.Single	

MaxAngle

Declaration

```
public float MaxAngle
```

Field Value

TYPE	DESCRIPTION
System.Single	

MinAngle

Declaration

```
public float MinAngle
```

Field Value

TYPE	DESCRIPTION
System.Single	

StartAngle

Declaration

```
public float StartAngle
```

Field Value

TYPE	DESCRIPTION
System.Single	

XMultiplier

Declaration

```
public float XMultiplier
```

Field Value

TYPE	DESCRIPTION
System.Single	

YMultiplier

Declaration

```
public float YMultiplier
```

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

CalculateLayoutInputHorizontal()

Declaration

```
public override void CalculateLayoutInputHorizontal()
```

CalculateLayoutInputVertical()

Declaration

```
public override void CalculateLayoutInputVertical()
```

OnEnable()

Declaration

```
protected override void OnEnable()
```

OnValidate()

Declaration

```
protected override void OnValidate()
```

SetLayoutHorizontal()

Declaration

```
public override void SetLayoutHorizontal()
```

SetLayoutVertical()

Declaration

```
public override void SetLayoutVertical()
```

Class Rotate

Animation settings for Rotation

Inheritance

System.Object

Rotate

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class Rotate
```

Constructors

Rotate(Anim.AnimationType)

Declaration

```
public Rotate(Anim.AnimationType aType)
```

Parameters

TYPE	NAME	DESCRIPTION
Anim.AnimationType	aType	

Fields

animationCurve

If the easeType is set to AnimationCurve, this will be used in order to calculate the rate of change of the animation over time.

Declaration

```
public AnimationCurve animationCurve
```

Field Value

TYPE	DESCRIPTION
AnimationCurve	

animationType

Select if this data is for an IN or an OUT animation.

Declaration

```
public Anim.AnimationType animationType
```

Field Value

TYPE	DESCRIPTION
Anim.AnimationType	

duration

The duration of the animation.

Declaration

<code>public float duration</code>

Field Value

TYPE	DESCRIPTION
System.Single	

ease

Sets the ease of the tween. Easing functions specify the rate of change of a parameter over time.

To see how default ease curves look, check out easings.net

Declaration

<code>public Ease ease</code>

Field Value

TYPE	DESCRIPTION
Ease	

easeType

Use an Ease or an AnimationCurve in order to calculate the rate of change of the animation over time.

Declaration

<code>public UIAnimator.EaseType easeType</code>
--

Field Value

TYPE	DESCRIPTION
UIAnimator.EaseType	

enabled

If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default: false).

Declaration

<code>public bool enabled</code>

Field Value

TYPE	DESCRIPTION
System.Boolean	

rotateMode

What type of rotation should this animation have: Fast, FastBeyond360, LocalAxisAdd or WorldAxisAdd. Default is RotateMode.FastBeyond360.

Declaration

```
public RotateMode rotateMode
```

Field Value

TYPE	DESCRIPTION
RotateMode	

rotation

Depending on the animation type, this is considered either the TO or the FROM rotation.

Declaration

```
public Vector3 rotation
```

Field Value

TYPE	DESCRIPTION
Vector3	

startDelay

Start delay for the animation.

Declaration

```
public float startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

Properties

TotalDuration

Declaration

```
public float TotalDuration { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Copy()

Declaration

```
public Rotate Copy()
```

Returns

TYPE	DESCRIPTION
Rotate	

Reset(Anim.AnimationType)

Declaration

```
public void Reset(Anim.AnimationType aType)
```

Parameters

TYPE	NAME	DESCRIPTION
Anim.AnimationType	aType	

Reverse()

Declaration

```
public Rotate Reverse()
```

Returns

TYPE	DESCRIPTION
Rotate	

UpdateValues(Rotate)

Declaration

```
public void UpdateValues(Rotate r)
```

Parameters

TYPE	NAME	DESCRIPTION
Rotate	r	

Class RotateLoop

Inheritance

System.Object
RotateLoop

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class RotateLoop
```

Fields

animationCurve

If the easeType is set to AnimationCurve, this will be used in order to calculate the rate of change of the animation over time.

Declaration

```
public AnimationCurve animationCurve
```

Field Value

TYPE	DESCRIPTION
AnimationCurve	

duration

The duration of the animation.

Declaration

```
public float duration
```

Field Value

TYPE	DESCRIPTION
System.Single	

ease

Sets the ease of the tween. Easing functions specify the rate of change of a parameter over time.

To see how default ease curves look, check out [easings.net](#)

Declaration

```
public Ease ease
```

Field Value

TYPE	DESCRIPTION
Ease	

easeType

Use an Ease or an AnimationCurve in order to calculate the rate of change of the animation over time.

Declaration

```
public UIManimator.EaseType easeType
```

Field Value

TYPE	DESCRIPTION
UIManimator.EaseType	

enabled

If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default: false).

Declaration

```
public bool enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

loops

Number of loops (-1 = infinite loops).

Declaration

```
public int loops
```

Field Value

TYPE	DESCRIPTION
System.Int32	

loopType

Types of loop.

Declaration

```
public Loop.LoopType loopType
```

Field Value

TYPE	DESCRIPTION
Loop.LoopType	

rotation

This rotation is calculated startRotation-rotation for min and startRotation+rotation for max

Declaration

```
public Vector3 rotation
```

Field Value

TYPE	DESCRIPTION
Vector3	

startDelay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

Properties

TotalDuration

Declaration

```
public float TotalDuration { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Copy()

Declaration

```
public RotateLoop Copy()
```

Returns

TYPE	DESCRIPTION
RotateLoop	

Reset()

Declaration

```
public void Reset()
```

Class Scale

Animation settings for Scale

Inheritance

System.Object

Scale

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class Scale
```

Constructors

Scale(Anim.AnimationType)

Declaration

```
public Scale(Anim.AnimationType aType)
```

Parameters

TYPE	NAME	DESCRIPTION
Anim.AnimationType	aType	

Fields

animationCurve

If the easeType is set to AnimationCurve, this will be used in order to calculate the rate of change of the animation over time.

Declaration

```
public AnimationCurve animationCurve
```

Field Value

TYPE	DESCRIPTION
AnimationCurve	

animationType

Select if this data is for an IN or an OUT animation.

Declaration

```
public Anim.AnimationType animationType
```

Field Value

TYPE	DESCRIPTION
Anim.AnimationType	

duration

The duration of the animation.

Declaration

<code>public float duration</code>

Field Value

TYPE	DESCRIPTION
System.Single	

ease

Sets the ease of the tween. Easing functions specify the rate of change of a parameter over time.

To see how default ease curves look, check out easings.net

Declaration

<code>public Ease ease</code>

Field Value

TYPE	DESCRIPTION
Ease	

easeType

Use an Ease or an AnimationCurve in order to calculate the rate of change of the animation over time.

Declaration

<code>public UIAnimator.EaseType easeType</code>
--

Field Value

TYPE	DESCRIPTION
UIAnimator.EaseType	

enabled

If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default: false).

Declaration

<code>public bool enabled</code>

Field Value

TYPE	DESCRIPTION
System.Boolean	

scale

Depending on the animation type, this is considered either the TO or the FROM scale.

Declaration

```
public Vector3 scale
```

Field Value

TYPE	DESCRIPTION
Vector3	

startDelay

Start delay for the animation.

Declaration

```
public float startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

Properties

TotalDuration

Declaration

```
public float TotalDuration { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Copy()

Declaration

```
public Scale Copy()
```

Returns

TYPE	DESCRIPTION
Scale	

Reset(Anim.AnimationType)

Declaration

```
public void Reset(Anim.AnimationType aType)
```

Parameters

TYPE	NAME	DESCRIPTION
Anim.AnimationType	aType	

Reverse()

Declaration

```
public Scale Reverse()
```

Returns

TYPE	DESCRIPTION
Scale	

UpdateValues(Scale)

Declaration

```
public void UpdateValues(Scale s)
```

Parameters

TYPE	NAME	DESCRIPTION
Scale	s	

Class ScaleLoop

Inheritance

System.Object
ScaleLoop

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class ScaleLoop
```

Fields

animationCurve

If the easeType is set to AnimationCurve, this will be used in order to calculate the rate of change of the animation over time.

Declaration

```
public AnimationCurve animationCurve
```

Field Value

TYPE	DESCRIPTION
AnimationCurve	

DEFAULT_MAX

Declaration

```
public static Vector3 DEFAULT_MAX
```

Field Value

TYPE	DESCRIPTION
Vector3	

DEFAULT_MIN

Declaration

```
public static Vector3 DEFAULT_MIN
```

Field Value

TYPE	DESCRIPTION
Vector3	

duration

The duration of the animation.

Declaration

```
public float duration
```

Field Value

TYPE	DESCRIPTION
System.Single	

ease

Sets the ease of the tween. Easing functions specify the rate of change of a parameter over time.

To see how default ease curves look, check out easings.net

Declaration

```
public Ease ease
```

Field Value

TYPE	DESCRIPTION
Ease	

easeType

Use an Ease or an AnimationCurve in order to calculate the rate of change of the animation over time.

Declaration

```
public UIManimator.EaseType easeType
```

Field Value

TYPE	DESCRIPTION
UIManimator.EaseType	

enabled

If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default: false).

Declaration

```
public bool enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

loops

Number of loops (-1 = infinite loops).

Declaration

```
public int loops
```

Field Value

TYPE	DESCRIPTION
System.Int32	

loopType

Types of loop.

Declaration

```
public Loop.LoopType loopType
```

Field Value

TYPE	DESCRIPTION
Loop.LoopType	

max

The maximum values for the scale factor of the scale loop animation (default: 1.05).

Declaration

```
public Vector3 max
```

Field Value

TYPE	DESCRIPTION
Vector3	

min

The minimum values for the scale factor of the scale loop animation (default: 1).

Declaration

```
public Vector3 min
```

Field Value

TYPE	DESCRIPTION
Vector3	

startDelay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float startDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

Properties

TotalDuration

Declaration

```
public float TotalDuration { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Copy()

Declaration

```
public ScaleLoop Copy()
```

Returns

TYPE	DESCRIPTION
ScaleLoop	

Reset()

Declaration

```
public void Reset()
```

Class SceneLoader

Inheritance

System.Object
QuickEngine.Common.Singleton<DoozyUI.SceneLoader>
SceneLoader

Inherited Members

[Singleton<SceneLoader>.Instance](#)
[Singleton<SceneLoader>.OnDestroy\(\)](#)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class SceneLoader : Singleton<SceneLoader>
```

Fields

command_LoadLevel

Declaration

```
public string command_LoadLevel
```

Field Value

TYPE	DESCRIPTION
System.String	

command_LoadSceneAdditiveAsync_SceneBuildIndex

Declaration

```
public string command_LoadSceneAdditiveAsync_SceneBuildIndex
```

Field Value

TYPE	DESCRIPTION
System.String	

command_LoadSceneAdditiveAsync_SceneName

Declaration

```
public string command_LoadSceneAdditiveAsync_SceneName
```

Field Value

TYPE	DESCRIPTION
System.String	

command_LoadSceneAsync_SceneBuildIndex

Declaration

```
public string command_LoadSceneAsync_SceneBuildIndex
```

Field Value

TYPE	DESCRIPTION
System.String	

command_LoadSceneAsync_SceneName

Declaration

```
public string command_LoadSceneAsync_SceneName
```

Field Value

TYPE	DESCRIPTION
System.String	

command_UnloadLevel

Declaration

```
public string command_UnloadLevel
```

Field Value

TYPE	DESCRIPTION
System.String	

command_UnloadScene_SceneBuildIndex

Declaration

```
public string command_UnloadScene_SceneBuildIndex
```

Field Value

TYPE	DESCRIPTION
System.String	

command_UnloadScene_SceneName

Declaration

```
public string command_UnloadScene_SceneName
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_LEVEL_LOADED

Declaration

```
public const string DEFAULT_LEVEL_LOADED = "LevelLoaded"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_LEVEL_SCENE_NAME

Declaration

```
public const string DEFAULT_LEVEL_SCENE_NAME = "Level_"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_LOAD_LEVEL

Declaration

```
public const string DEFAULT_LOAD_LEVEL = "LoadLevel_"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_LOAD_SCENE_ADDITIVE_ASYNC_SCENE_BUILD_INDEX

Declaration

```
public const string DEFAULT_LOAD_SCENE_ADDITIVE_ASYNC_SCENE_BUILD_INDEX = "LoadSceneAdditiveAsync_ID_"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_LOAD_SCENE_ADDITIVE_ASYNC_SCENE_NAME

Declaration

```
public const string DEFAULT_LOAD_SCENE_ADDITIVE_ASYNC_SCENE_NAME = "LoadSceneAdditiveAsync_Name_"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_LOAD_SCENE_ASYNC_SCENE_BUILD_INDEX

Declaration

```
public const string DEFAULT_LOAD_SCENE_ASYNC_SCENE_BUILD_INDEX = "LoadSceneAsync_ID_"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_LOAD_SCENE_ASYNC_SCENE_NAME

Declaration

```
public const string DEFAULT_LOAD_SCENE_ASYNC_SCENE_NAME = "LoadSceneAsync_Name_"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_UNLOAD_LEVEL

Declaration

```
public const string DEFAULT_UNLOAD_LEVEL = "UnloadLevel_"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_UNLOAD_SCENE_SCENE_BUILD_INDEX

Declaration

```
public const string DEFAULT_UNLOAD_SCENE_SCENE_BUILD_INDEX = "UnloadScene_ID_"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_UNLOAD_SCENE_SCENE_NAME

Declaration

```
public const string DEFAULT_UNLOAD_SCENE_SCENE_NAME = "UnloadScene_Name_"
```

Field Value

TYPE	DESCRIPTION
System.String	

levelLoadedGameEvent

Declaration

```
public string levelLoadedGameEvent
```

Field Value

TYPE	DESCRIPTION
System.String	

levelSceneName

Declaration

public string levelSceneName

Field Value

TYPE	DESCRIPTION
System.String	

Methods

LoadLevel(Int32)

Declaration

public void LoadLevel(int levelNumber)
--

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	levelNumber	

LoadLevelAdditiveAsync(Int32)

Declaration

public void LoadLevelAdditiveAsync(int sceneBuildIndex)

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sceneBuildIndex	

LoadLevelAdditiveAsync(String)

Declaration

public void LoadLevelAdditiveAsync(string sceneName)
--

Parameters

TYPE	NAME	DESCRIPTION
System.String	sceneName	

LoadSceneAsync(Int32)

Declaration

public void LoadSceneAsync(int sceneBuildIndex)

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sceneBuildIndex	

LoadSceneAsync(String)

Declaration

```
public void LoadSceneAsync(string sceneName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	sceneName	

OnGameEvent(String)

Declaration

```
public void OnGameEvent(string gameEvent)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	gameEvent	

UnloadLevel(Int32)

Declaration

```
public void UnloadLevel(int levelNumber)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	levelNumber	

UnloadScene(Int32)

Declaration

```
public void UnloadScene(int sceneBuildIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	sceneBuildIndex	

UnloadScene(String)

Declaration

```
public void UnloadScene(string sceneName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	sceneName	

Enum SoundType

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public enum SoundType
```

Fields

NAME	DESCRIPTION
All	
UIButtons	
UIElements	

Class Soundy

Inheritance

System.Object

Soundy

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class Soundy : MonoBehaviour
```

Fields

masterPitch

Declaration

```
public float masterPitch
```

Field Value

TYPE	DESCRIPTION
System.Single	

masterVolume

Declaration

```
public float masterVolume
```

Field Value

TYPE	DESCRIPTION
System.Single	

numberOfChannels

Declaration

```
public int numberOfChannels
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

PlaySound(AudioClip)

Declaration

```
public void PlaySound(AudioClip aClip)
```

Parameters

TYPE	NAME	DESCRIPTION
AudioClip	aClip	

PlaySound(AudioClip, Single)

Declaration

```
public void PlaySound(AudioClip aClip, float volumePercentage)
```

Parameters

TYPE	NAME	DESCRIPTION
AudioClip	aClip	
System.Single	volumePercentage	

PlaySound(AudioClip, Single, Single)

Declaration

```
public void PlaySound(AudioClip aClip, float volumePercentage, float pitch)
```

Parameters

TYPE	NAME	DESCRIPTION
AudioClip	aClip	
System.Single	volumePercentage	
System.Single	pitch	

PlaySound(String)

Declaration

```
public void PlaySound(string soundName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	

PlaySound(String, Single)

Declaration

```
public void PlaySound(string soundName, float volumePercentage)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	
System.Single	volumePercentage	

PlaySound(String, Single, Single)

Declaration

```
public void PlaySound(string soundName, float volumePercentage, float pitch)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	
System.Single	volumePercentage	
System.Single	pitch	

PlaySoundFromResources(String)

Declaration

```
public void PlaySoundFromResources(string soundName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	

PlaySoundFromResources(String, Single)

Declaration

```
public void PlaySoundFromResources(string soundName, float volume)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	
System.Single	volume	

PlaySoundFromResources(String, Single, Single)

Declaration

```
public void PlaySoundFromResources(string soundName, float volume, float pitch)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	
System.Single	volume	
System.Single	pitch	

Class UIAnimationManager

Inheritance

System.Object
UIAnimationManager

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class UIAnimationManager : MonoBehaviour
```

Fields

DEFAULT_PRESET_NAME

Declaration

```
public const string DEFAULT_PRESET_NAME = "DefaultPreset"
```

Field Value

TYPE	DESCRIPTION
System.String	

Properties

GetUIButton

Declaration

```
public UIButton GetUIButton { get; }
```

Property Value

TYPE	DESCRIPTION
UIButton	

GetUIElement

Declaration

```
public UIElement GetUIElement { get; }
```

Property Value

TYPE	DESCRIPTION
UIElement	

Methods

DeletePreset(String, UIAnimationManager.AnimationType, UIAnimationManager.ButtonLoopType)

Declaration

```
public void DeletePreset(string presetName, UIAnimationManager.AnimationType animationType,  
UIAnimationManager.ButtonLoopType buttonLoopType = UIAnimationManager.ButtonLoopType.None)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetName	
UIAnimationManager.AnimationType	animationType	
UIAnimationManager.ButtonLoopType	buttonLoopType	

LoadPreset(String, UIAnimationManager.AnimationType, UIAnimationManager.ButtonLoopType)

Declaration

```
public void LoadPreset(string presetName, UIAnimationManager.AnimationType animationType,
UIAnimationManager.ButtonLoopType buttonLoopType = UIAnimationManager.ButtonLoopType.None)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetName	
UIAnimationManager.AnimationType	animationType	
UIAnimationManager.ButtonLoopType	buttonLoopType	

LoadPresetList(UIAnimationManager.AnimationType)

Declaration

```
public void LoadPresetList(UIAnimationManager.AnimationType animationType)
```

Parameters

TYPE	NAME	DESCRIPTION
UIAnimationManager.AnimationType	animationType	

SavePreset(String, UIAnimationManager.AnimationType, UIAnimationManager.ButtonLoopType)

Declaration

```
public void SavePreset(string presetName, UIAnimationManager.AnimationType animationType,
UIAnimationManager.ButtonLoopType buttonLoopType = UIAnimationManager.ButtonLoopType.None)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetName	
UIAnimationManager.AnimationType	animationType	
UIAnimationManager.ButtonLoopType	buttonLoopType	

Enum UIAnimationManager.AnimationType

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public enum AnimationType
```

Fields

NAME	DESCRIPTION
ButtonLoops	
IN	
LOOP	
OnClick	
OUT	

Class UIAnimationManager.ButtonLoopsAnimations

Inheritance

System.Object
UIAnimationManager.ButtonLoopsAnimations

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class ButtonLoopsAnimations
```

Fields

animationsPresetName

Declaration

```
public string animationsPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

fadeLoop

Declaration

```
public UIAnimator.FadeLoop fadeLoop
```

Field Value

TYPE	DESCRIPTION
UIAnimator.FadeLoop	

moveLoop

Declaration

```
public UIAnimator.MoveLoop moveLoop
```

Field Value

TYPE	DESCRIPTION
UIAnimator.MoveLoop	

rotationLoop

Declaration

```
public UIAnimator.RotationLoop rotationLoop
```

Field Value

TYPE	DESCRIPTION
UIAnimator.RotationLoop	

scaleLoop

Declaration

```
public UIAnimator.ScaleLoop scaleLoop
```

Field Value

TYPE	DESCRIPTION
UIAnimator.ScaleLoop	

Enum UIAnimationManager.ButtonLoopType

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public enum ButtonLoopType
```

Fields

NAME	DESCRIPTION
Highlighted	
None	
Normal	

Class UIAnimationManager.InAnimations

Inheritance

System.Object
UIAnimationManager.InAnimations

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Obsolete]  
[Serializable]  
public class InAnimations
```

Fields

fadeIn

Declaration

```
public UIAnimator.FadeIn fadeIn
```

Field Value

TYPE	DESCRIPTION
UIAnimator.FadeIn	

inAnimationsPresetName

Declaration

```
public string inAnimationsPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

moveIn

Declaration

```
public UIAnimator.MoveIn moveIn
```

Field Value

TYPE	DESCRIPTION
UIAnimator.MoveIn	

rotationIn

Declaration

```
public UIAnimator.RotationIn rotationIn
```

Field Value

TYPE	DESCRIPTION
UIAnimator.RotationIn	

scaleIn

Declaration

```
public UIAnimator.ScaleIn scaleIn
```

Field Value

TYPE	DESCRIPTION
UIAnimator.ScaleIn	

Class UIAnimationManager.LoopAnimations

Inheritance

System.Object
UIAnimationManager.LoopAnimations

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Obsolete]  
[Serializable]  
public class LoopAnimations
```

Fields

fadeLoop

Declaration

```
public UIAnimator.FadeLoop fadeLoop
```

Field Value

TYPE	DESCRIPTION
UIAnimator.FadeLoop	

loopAnimationsPresetName

Declaration

```
public string loopAnimationsPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

moveLoop

Declaration

```
public UIAnimator.MoveLoop moveLoop
```

Field Value

TYPE	DESCRIPTION
UIAnimator.MoveLoop	

rotationLoop

Declaration

```
public UIAnimator.RotationLoop rotationLoop
```

Field Value

TYPE	DESCRIPTION
UIAnimator.RotationLoop	

scaleLoop

Declaration

```
public UIAnimator.ScaleLoop scaleLoop
```

Field Value

TYPE	DESCRIPTION
UIAnimator.ScaleLoop	

Class UIAnimationManager.OnClickAnimations

Inheritance

System.Object
UIAnimationManager.OnClickAnimations

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Obsolete]  
[Serializable]  
public class OnClickAnimations
```

Fields

onClickAnimationsPresetName

Declaration

```
public string onClickAnimationsPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

punchPositionDelay

Declaration

```
public float punchPositionDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

punchPositionDuration

Declaration

```
public float punchPositionDuration
```

Field Value

TYPE	DESCRIPTION
System.Single	

punchPositionElasticity

Declaration

```
public float punchPositionElasticity
```

Field Value

TYPE	DESCRIPTION
System.Single	

punchPositionEnabled

Declaration

```
public bool punchPositionEnabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

punchPositionPunch

Declaration

```
public Vector2 punchPositionPunch
```

Field Value

TYPE	DESCRIPTION
Vector2	

punchPositionSnapping

Declaration

```
public bool punchPositionSnapping
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

punchPositionVibrato

Declaration

```
public int punchPositionVibrato
```

Field Value

TYPE	DESCRIPTION
System.Int32	

punchRotationDelay

Declaration

```
public float punchRotationDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

punchRotationDuration

Declaration

```
public float punchRotationDuration
```

Field Value

TYPE	DESCRIPTION
System.Single	

punchRotationElasticity

Declaration

```
public float punchRotationElasticity
```

Field Value

TYPE	DESCRIPTION
System.Single	

punchRotationEnabled

Declaration

```
public bool punchRotationEnabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

punchRotationPunch

Declaration

```
public Vector3 punchRotationPunch
```

Field Value

TYPE	DESCRIPTION
Vector3	

punchRotationVibrato

Declaration

```
public int punchRotationVibrato
```

Field Value

TYPE	DESCRIPTION
System.Int32	

punchScaleDelay

Declaration

```
public float punchScaleDelay
```

Field Value

TYPE	DESCRIPTION
System.Single	

punchScaleDuration

Declaration

```
public float punchScaleDuration
```

Field Value

TYPE	DESCRIPTION
System.Single	

punchScaleElasticity

Declaration

```
public float punchScaleElasticity
```

Field Value

TYPE	DESCRIPTION
System.Single	

punchScaleEnabled

Declaration

```
public bool punchScaleEnabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

punchScalePunch

Declaration

```
public Vector3 punchScalePunch
```

Field Value

TYPE	DESCRIPTION
Vector3	

punchScaleVibrato

Declaration

```
public int punchScaleVibrato
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Class UIManager.OutAnimations

Inheritance

System.Object
UIAnimationManager.OutAnimations

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Obsolete]  
[Serializable]  
public class OutAnimations
```

Fields

fadeOut

Declaration

```
public UIAnimator.FadeOut fadeOut
```

Field Value

TYPE	DESCRIPTION
UIAnimator.FadeOut	

moveOut

Declaration

```
public UIAnimator.MoveOut moveOut
```

Field Value

TYPE	DESCRIPTION
UIAnimator.MoveOut	

outAnimationsPresetName

Declaration

```
public string outAnimationsPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

rotationOut

Declaration

```
public UIAnimator.RotationOut rotationOut
```

Field Value

TYPE	DESCRIPTION
UIAnimator.RotationOut	

scaleOut

Declaration

```
public UIAnimator.ScaleOut scaleOut
```

Field Value

TYPE	DESCRIPTION
UIAnimator.ScaleOut	

Class UIAnimator

Inheritance

System.Object
UIAnimator

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class UIAnimator
```

Fields

DEFAULT_DURATION

Default duration set to an animation

Declaration

```
public const float DEFAULT_DURATION = 0.5F
```

Field Value

TYPE	DESCRIPTION
System.Single	

DEFAULT_DURATION_INIT_LOOP

Default loop setup duration. This is the time a loop animation is setup for it's cycle to start.

Declaration

```
public const float DEFAULT_DURATION_INIT_LOOP = 0.2F
```

Field Value

TYPE	DESCRIPTION
System.Single	

DEFAULT_DURATION_ONCOMPLETE

Default reset duration after a punch animation. This reset is needed to be sure the animation's initial values are restored.

Declaration

```
public const float DEFAULT_DURATION_ONCOMPLETE = 0.1F
```

Field Value

TYPE	DESCRIPTION
System.Single	

DEFAULT_DURATION_RESET_TARGET

Default target reset. This is the time a 'target' (rectTransfrom) is reset to it's start values (runtime values).

Declaration

```
public const float DEFAULT_DURATION_RESET_TARGET = 0.1F
```

Field Value

TYPE	DESCRIPTION
System.Single	

DEFAULT_EASE

Default ease set to an animations

Declaration

```
public const Ease DEFAULT_EASE = null
```

Field Value

TYPE	DESCRIPTION
Ease	

DEFAULT_LOOPS

Default loops set to a loop animation. -1 means infinite loops.

Declaration

```
public const int DEFAULT_LOOPS = -1
```

Field Value

TYPE	DESCRIPTION
System.Int32	

DEFAULT_START_DELAY

Default start delay set to an animation

Declaration

```
public const float DEFAULT_START_DELAY = 0F
```

Field Value

TYPE	DESCRIPTION
System.Single	

isTimeScaleIndependent

Should the UI ignore game timescale and work in realtime? Default is true.

Declaration

```
public static bool isTimeScaleIndependent
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Methods

Fade(RectTransform, Single, Anim, UnityAction, UnityAction, Boolean, Boolean)

Fades in or out a RectTransform (and it's children) by animating the alpha value of it's attached CanvasGroup. If a CanvasGroup is not attached to the target then it will automatically attach one for you.

Declaration

```
public static void Fade(RectTransform target, float startAlpha, Anim animation, UnityAction OnStart, UnityAction OnComplete, bool instantAnimation = false, bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	Target RectTransform.
System.Single	startAlpha	CanvasGroup's start alpha. This is the animation's center.
Anim	animation	The animation settings.
UnityAction	OnStart	Callback listener.
UnityAction	OnComplete	Callback listener.
System.Boolean	instantAnimation	If true, the animation will happen instantly (without creating a tween).
System.Boolean	forced	If true, it will initiate this animation, regardless if it's enabled or not.

GetDirection(UIAnimator.MoveDetails)

Declaration

```
public static Move.MoveDirection GetDirection(UIAnimator.MoveDetails moveDetails)
```

Parameters

TYPE	NAME	DESCRIPTION
UIAnimator.MoveDetails	moveDetails	

Returns

TYPE	DESCRIPTION
Move.MoveDirection	

GetTweenId(RectTransform, UIAnimator.TweenIdType, UIAnimator.TweenIdAnimation)

Returns the tween id of the given target with the given idType and idAnimation. This is a quick id generator.

Declaration

```
public static string GetTweenId(RectTransform target, UIAnimator.TweenIdType idType,
UIAnimator.TweenIdAnimation idAnimation)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	
UIAnimator.TweenIdType	idType	
UIAnimator.TweenIdAnimation	idAnimation	

Returns

TYPE	DESCRIPTION
System.String	

LoopFade(RectTransform, Single, Loop, UnityAction, UnityAction, String, Boolean, Boolean)

Creates a fade (alpha) Loop animation, but it does not start automatically unless the loop's autoStart variable is set to true.

Declaration

```
public static void LoopFade(RectTransform target, float startAlpha, Loop loop, UnityAction OnStart,
UnityAction OnComplete, string id = "", bool blocksRaycasts = false, bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	Target RectTransform.
System.Single	startAlpha	The initial rotation of the target. This is the animation's center.
Loop	loop	The loop animation settings.

TYPE	NAME	DESCRIPTION
UnityAction	OnStart	Callback listener.
UnityAction	OnComplete	Callback listener.
System.String	id	Adds an extra string to the loop's tween id. Used to differentiate several loops animations created for the same target.
System.Boolean	blocksRaycasts	
System.Boolean	forced	If true, it will initiate this animation, regardless if it's enabled or not.

LoopMove(RectTransform, Vector3, Loop, UnityAction, UnityAction, String, Boolean)

Creates a move Loop animation, but it does not start automatically unless the loop's autoStart variable is set to true.

Declaration

```
public static void LoopMove(RectTransform target, Vector3 startPosition, Loop loop, UnityAction OnStart,
UnityAction OnComplete, string id = "", bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	Target RectTransform.
Vector3	startPosition	The initial position of the target. This is the animation's center.
Loop	loop	The loop animation settings.
UnityAction	OnStart	Callback listener.
UnityAction	OnComplete	Callback listener.
System.String	id	Adds an extra string to the loop's tween id. Used to differentiate several loops animations created for the same target.
System.Boolean	forced	If true, it will initiate this animation, regardless if it's enabled or not.

LoopRotate(RectTransform, Vector3, Loop, UnityAction, UnityAction, String, Boolean)

Creates a rotation Loop animation, but it does not start automatically unless the loop's autoStart variable is set to true.

Declaration

```
public static void LoopRotate(RectTransform target, Vector3 startRotation, Loop loop, UnityAction OnStart,
UnityAction OnComplete, string id = "", bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	Target RectTransform.
Vector3	startRotation	The initial rotation of the target. This is the animation's center.
Loop	loop	The loop animation settings.
UnityAction	OnStart	Callback listener.
UnityAction	OnComplete	Callback listener.
System.String	id	Adds an extra string to the loop's tween id. Used to differentiate several loops animations created for the same target.
System.Boolean	forced	If true, it will initiate this animation, regardless if it's enabled or not.

LoopScale(RectTransform, Vector3, Loop, UnityAction, UnityAction, String, Boolean)

Creates a scale Loop animation, but it does not start automatically unless the loop's autoStart variable is set to true.

Declaration

```
public static void LoopScale(RectTransform target, Vector3 startScale, Loop loop, UnityAction OnStart,
UnityAction OnComplete, string id = "", bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	Target RectTransform.
Vector3	startScale	The initial rotation of the target. This is the animation's center.
Loop	loop	The loop animation settings.

TYPE	NAME	DESCRIPTION
UnityAction	OnStart	Callback listener.
UnityAction	OnComplete	Callback listener.
System.String	id	Adds an extra string to the loop's tween id. Used to differentiate several loops animations created for the same target.
System.Boolean	forced	If true, it will initiate this animation, regardless if it's enabled or not.

Move(RectTransform, Vector3, Anim, UnityAction, UnityAction, Boolean, Boolean)

Moves in or out a RectTransform by animating the anchoredPosition3D value.

Declaration

```
public static void Move(RectTransform target, Vector3 startPosition, Anim animation, UnityAction OnStart,
UnityAction OnComplete, bool instantAnimation = false, bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	Target RectTransform.
Vector3	startPosition	The initial position of the target.
Anim	animation	The animation settings.
UnityAction	OnStart	Callback listener.
UnityAction	OnComplete	Callback listener.
System.Boolean	instantAnimation	If true, the animation will happen instantly (without creating a tween).
System.Boolean	forced	If true, it will initiate this animation, regardless if it's enabled or not.

PlayLoops(RectTransform, String)

Plays all the loops that have been previously set up for the target RectTransform. This means that you should have called the SetupLoops method, for the target RectTransform, before you called this method.

Declaration

```
public static void PlayLoops(RectTransform target, string id = "")
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	Target RectTransform.
System.String	id	Adds an extra string to the loop's tween id. Used to differentiate several loops animations created for the same target.

PunchMove(RectTransform, Vector3, Punch, UnityAction, UnityAction, Boolean)

Punches a RectTransform's anchoredPosition towards the given direction and then back to the starting one as if it was connected to the starting position via an elastic.

You can force an execution of this animation (regardless if it's enabled or not) by setting forced as true.

Declaration

```
public static void PunchMove(RectTransform target, Vector3 startPosition, Punch punch, UnityAction OnStart, UnityAction OnComplete, bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	Target RectTransform.
Vector3	startPosition	RectTranform's start position (target.anchoredPosition). This will also be its end position.
Punch	punch	The punch animation settings.
UnityAction	OnStart	
UnityAction	OnComplete	
System.Boolean	forced	If true, it will fire this animation, regardless if it's enabled or not.

PunchRotate(RectTransform, Vector3, Punch, UnityAction, UnityAction, Boolean)

Punches a Transform's localRotation towards the given size and then back to the starting one as if it was connected to the starting rotation via an elastic.

You can force an execution of this animation (regardless if it's enabled or not) by setting forced as true.

Declaration

```
public static void PunchRotate(RectTransform target, Vector3 startRotation, Punch punch, UnityAction OnStart, UnityAction OnComplete, bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	Target RectTransform.
Vector3	startRotation	RectTranform's start localRotation. This will also be its end localRotation.
Punch	punch	The punch animation settings.
UnityAction	OnStart	
UnityAction	OnComplete	
System.Boolean	forced	If true, it will fire this animation, regardless if it's enabled or not.

PunchScale(RectTransform, Vector3, Punch, UnityAction, UnityAction, Boolean)

Punches a Transform's localScale towards the given size and then back to the starting one as if it was connected to the starting scale via an elastic.

Declaration

```
public static void PunchScale(RectTransform target, Vector3 startScale, Punch punch, UnityAction OnStart, UnityAction OnComplete, bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	Target RectTransform.
Vector3	startScale	RectTranform's start localScale. This will also be its end localScale.
Punch	punch	The punch animation settings.
UnityAction	OnStart	
UnityAction	OnComplete	
System.Boolean	forced	If true, it will fire this animation, regardless if it's enabled or not.

ResetTarget(RectTransform, Vector3, Vector3, Vector3, Single, Boolean)

Resets the given target (RectTransform) to the given start parameters (position, rotation, scale and alpha). By default this is an instant reset, but you can override the DEFAULT_DURATION_RESET_TARGET value and set instantAnimation to false, in order to

animate this reset (not recommended).

Declaration

```
public static void ResetTarget(RectTransform target, Vector3 startPosition, Vector3 startRotation, Vector3 startScale, float startAlpha, bool instantAnimation = true)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	
Vector3	startPosition	
Vector3	startRotation	
Vector3	startScale	
System.Single	startAlpha	
System.Boolean	instantAnimation	

Reverse(Ease)

Returns the reverse of the given ease.

Declaration

```
public static Ease Reverse(Ease ease)
```

Parameters

TYPE	NAME	DESCRIPTION
Ease	ease	

Returns

TYPE	DESCRIPTION
Ease	

Rotate(RectTransform, Vector3, Anim, UnityAction, UnityAction, Boolean, Boolean)

Rotates in or out a RectTransform by animating the localRotation value.

Declaration

```
public static void Rotate(RectTransform target, Vector3 startRotation, Anim animation, UnityAction OnStart, UnityAction OnComplete, bool instantAnimation = false, bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	Target RectTransform.

TYPE	NAME	DESCRIPTION
Vector3	startRotation	The initial rotation of the target.
Anim	animation	The animation settings.
UnityAction	OnStart	Callback listener.
UnityAction	OnComplete	Callback listener.
System.Boolean	instantAnimation	If true, the animation will happen instantly (without creating a tween).
System.Boolean	forced	If true, it will initiate this animation, regardless if it's enabled or not.

Scale(RectTransform, Vector3, Anim, UnityAction, UnityAction, Boolean, Boolean)

Scales in or out a RectTransform by animating the localScale value.

Declaration

```
public static void Scale(RectTransform target, Vector3 startScale, Anim animation, UnityAction OnStart,
UnityAction OnComplete, bool instantAnimation = false, bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	Target RectTransform.
Vector3	startScale	The initial scale of the target.
Anim	animation	The animation settings.
UnityAction	OnStart	Callback listener.
UnityAction	OnComplete	Callback listener.
System.Boolean	instantAnimation	If true, the animation will happen instantly (without creating a tween).

--	--	--

TYPE	NAME	DESCRIPTION
System.Boolean	forced	If true, it will initiate this animation, regardless if it's enabled or not.

SetupLoops(RectTransform, Vector3, Vector3, Vector3, Single, Loop, UnityAction, UnityAction, UnityAction, UnityAction, UnityAction, String, Boolean, Boolean)

Creates all the loops and pauses them. It plays only the ones that are set to autoStart.

Declaration

```
public static void SetupLoops(RectTransform target, Vector3 startPosition, Vector3 startRotation, Vector3 startScale, float startAlpha, Loop loop, UnityAction OnStartMoveLoop, UnityAction OnCompleteMoveLoop, UnityAction OnStartRotateLoop, UnityAction OnCompleteRotateLoop, UnityAction OnStartScaleLoop, UnityAction OnCompleteScaleLoop, UnityAction OnStartFadeLoop, UnityAction OnCompleteFadeLoop, string id = "", bool blocksRaycasts = false, bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	Target RectTransform.
Vector3	startPosition	RectTranform's start position. This is the animation's center.
Vector3	startRotation	RectTranform's start rotation. This is the animation's center.
Vector3	startScale	RectTranform's start scale. This is the animation's center.
System.Single	startAlpha	CanvasGroup's start alpha. This is the animation's center.
Loop	loop	The loop animation settings.
UnityAction	OnStartMoveLoop	Callback listener.
UnityAction	OnCompleteMoveLoop	Callback listener.
UnityAction	OnStartRotateLoop	Callback listener.
UnityAction	OnCompleteRotateLoop	Callback listener.

TYPE	NAME	DESCRIPTION
UnityAction	OnStartScaleLoop	Callback listener.
UnityAction	OnCompleteScaleLoop	Callback listener.
UnityAction	OnStartFadeLoop	Callback listener.
UnityAction	OnCompleteFadeLoop	Callback listener.
System.String	id	Adds an extra string to the loop's tween id. Used to differentiate several loops animations created for the same target.
System.Boolean	blocksRaycasts	Does the CanvasGroup (that is attached automatically to this target) block raycasting (allow collision). Or, in other words, false means that it ignores clicks (for UIElement) and true means that it registers clicks (for UIButtons).
System.Boolean	forced	If true, it will initiate this animation, regardless if it's enabled or not.

StopAnimations(RectTransform, Anim.AnimationType)

Stops all the running animations In and Out on the target (RectTransform). It uses the GetTweenId generator in order to get valid ids.

Declaration

```
public static void StopAnimations(RectTransform target, Anim.AnimationType aType)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	
Anim.AnimationType	aType	

StopLoops(RectTransform, String)

Stops (kills) all the loops that are playing on the target RectTransform. This means that you called the PlayLoops method, for the target RectTransform, before you called this method.

Note: Some loops might play even if PlayLoops was not called. This can happen if autoStart is true for those certain loops and the SetupLoops method was called.

Declaration

```
public static void StopLoops(RectTransform target, string id = "")
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	target	Target RectTransform.
System.String	id	Adds an extra string to the loop's tween id. Used to differentiate several loops animations created for the same target.

Enum UIAnimator.AnimationTarget

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Obsolete]  
public enum AnimationTarget
```

Fields

NAME	DESCRIPTION
None	
UIButton	
UIElement	

Enum UIAnimator.ButtonAnimationType

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
[Obsolete]
public enum ButtonAnimationType
```

Fields

NAME	DESCRIPTION
None	
PunchPosition	
PunchRotation	
PunchScale	

Enum UIAnimator.EaseType

Type of ease an animation, loop or punch should use.

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public enum EaseType
```

Fields

NAME	DESCRIPTION
AnimationCurve	
Ease	

Class UIAnimator.FadeIn

Inheritance

System.Object
UIAnimator.FadeIn

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class FadeIn
```

Fields

delay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

easeType

Easing is the rate of change of animation over time

Declaration

```
public DG.Tweening.Ease easeType
```

Field Value

TYPE	DESCRIPTION
DG.Tweening.Ease	

enabled

Is the animation enabled?

Declaration

```
public bool enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

soundAtFinish

Sends trigger sounds

Declaration

```
public string soundAtFinish
```

Field Value

TYPE	DESCRIPTION
System.String	

soundAtStart

Declaration

```
public string soundAtStart
```

Field Value

TYPE	DESCRIPTION
System.String	

time

Time is amount (seconds) that the animation will take to complete

Declaration

```
public float time
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class UIAnimator.FadeLoop

Inheritance

System.Object
UIAnimator.FadeLoop

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class FadeLoop
```

Fields

autoStart

If you want this animation to ignore IN and OUT animations and auto start then select this as true

Declaration

```
public bool autoStart
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

delay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

easeType

Easing is the rate of change of animation over time

Declaration

```
public DG.Tweening.Ease easeType
```

Field Value

TYPE	DESCRIPTION
DG.Tweening.Ease	

enabled

Is the animation enabled?

Declaration

public bool enabled

Field Value

TYPE	DESCRIPTION
System.Boolean	

loops

Number of loops (-1 = infinite loops)

Declaration

public int loops

Field Value

TYPE	DESCRIPTION
System.Int32	

loopType

Types of loop

Declaration

public LoopType loopType

Field Value

TYPE	DESCRIPTION
LoopType	

max

The maximum alpha value for the fade animation loop

Declaration

public float max

Field Value

TYPE	DESCRIPTION
System.Single	

min

The minimum alpha value for the fade animation loop

Declaration

```
public float min
```

Field Value

TYPE	DESCRIPTION
System.Single	

soundAtFinish

Sends trigger sounds

Declaration

```
public string soundAtFinish
```

Field Value

TYPE	DESCRIPTION
System.String	

soundAtFinishReference

(deprecated) Sends trigger sounds

Declaration

```
public UIAnimator.SoundDetails soundAtFinishReference
```

Field Value

TYPE	DESCRIPTION
UIAnimator.SoundDetails	

soundAtStart

Sends trigger sounds

Declaration

```
public string soundAtStart
```

Field Value

TYPE	DESCRIPTION
System.String	

soundAtStartReference

(deprecated) Sends trigger sounds

Declaration


```
public UIAnimator.SoundDetails soundAtStartReference
```

Field Value

TYPE	DESCRIPTION
UIAnimator.SoundDetails	

time

Time is amount (seconds) that the animation will take to complete

Declaration

```
public float time
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class UIAnimator.FadeOut

Inheritance

System.Object
UIAnimator.FadeOut

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class FadeOut
```

Fields

delay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

easeType

Easing is the rate of change of animation over time

Declaration

```
public DG.Tweening.Ease easeType
```

Field Value

TYPE	DESCRIPTION
DG.Tweening.Ease	

enabled

Is the animation enabled?

Declaration

```
public bool enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

soundAtFinish

Sends trigger sounds

Declaration

public string soundAtFinish

Field Value

TYPE	DESCRIPTION
System.String	

soundAtFinishReference

(deprecated) Sends trigger sounds

Declaration

public UIAnimator.SoundDetails soundAtFinishReference

Field Value

TYPE	DESCRIPTION
UIAnimator.SoundDetails	

soundAtStart

Sends trigger sounds

Declaration

public string soundAtStart

Field Value

TYPE	DESCRIPTION
System.String	

soundAtStartReference

(deprecated) Sends trigger sounds

Declaration

public UIAnimator.SoundDetails soundAtStartReference
--

Field Value

TYPE	DESCRIPTION
UIAnimator.SoundDetails	

time

Time is amount (seconds) that the animation will take to complete

Declaration

```
public float time
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class UIAnimator.InitialData

Inheritance

System.Object
UIAnimator.InitialData

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class InitialData
```

Fields

soundOn

Declaration

```
public bool soundOn
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

startAlpha

Declaration

```
public float startAlpha
```

Field Value

TYPE	DESCRIPTION
System.Single	

startPosition

Declaration

```
public Vector3 startPosition
```

Field Value

TYPE	DESCRIPTION
Vector3	

startRotation

Declaration

```
public Vector3 startRotation
```

Field Value

TYPE	DESCRIPTION
Vector3	

startScale

Declaration

```
public Vector3 startScale
```

Field Value

TYPE	DESCRIPTION
Vector3	

Enum UIAnimator.MoveDetails

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Obsolete]  
public enum MoveDetails
```

Fields

NAME	DESCRIPTION
BottomCenter	
BottomLeft	
BottomRight	
BottomScreenEdge	
LeftScreenEdge	
LocalPosition	
MiddleCenter	
MiddleLeft	
MiddleRight	
ParentPosition	
RightScreenEdge	
TopCenter	
TopLeft	
TopRight	
TopScreenEdge	

Class UIAnimator.MoveIn

Inheritance

System.Object
UIAnimator.MoveIn

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class MoveIn
```

Fields

delay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

easeType

Easing is the rate of change of animation over time

Declaration

```
public DG.Tweening.Ease easeType
```

Field Value

TYPE	DESCRIPTION
DG.Tweening.Ease	

enabled

Is the animation enabled?

Declaration

```
public bool enabled
```


Field Value

TYPE	DESCRIPTION
System.Boolean	

moveFrom

Where does the animation begin from?

Declaration

<code>public UIMoveDetails moveFrom</code>
--

Field Value

TYPE	DESCRIPTION
UIMoveDetails	

positionAdjustment

Use this if you need to adjust the target position. You add or subtract (if the number is negative) values to the position of the target location

Declaration

<code>public Vector3 positionAdjustment</code>
--

Field Value

TYPE	DESCRIPTION
Vector3	

positionFrom

This is used when the Move From LocalPosition is selected

Declaration

<code>public Vector3 positionFrom</code>
--

Field Value

TYPE	DESCRIPTION
Vector3	

soundAtFinish

Sends trigger sounds

Declaration

<code>public string soundAtFinish</code>
--

Field Value

TYPE	DESCRIPTION
System.String	

soundAtStart

Sends trigger sounds

Declaration

```
public string soundAtStart
```

Field Value

TYPE	DESCRIPTION
System.String	

time

Time is amount (seconds) that the animation will take to complete

Declaration

```
public float time
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class UIAnimator.MoveLoop

Inheritance

System.Object
UIAnimator.MoveLoop

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class MoveLoop
```

Fields

autoStart

If you want this animation to ignore IN and OUT animations and auto start then select this as true

Declaration

```
public bool autoStart
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

delay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

easeType

Easing is the rate of change of animation over time

Declaration

```
public DG.Tweening.Ease easeType
```

Field Value

TYPE	DESCRIPTION
DG.Tweening.Ease	

enabled

Is the animation enabled?

Declaration

public bool enabled

Field Value

TYPE	DESCRIPTION
System.Boolean	

loops

Number of loops (-1 = infinite loops)

Declaration

public int loops

Field Value

TYPE	DESCRIPTION
System.Int32	

loopType

Types of loop

Declaration

public LoopType loopType

Field Value

TYPE	DESCRIPTION
LoopType	

movement

This movement is calculated startAnchoredPosition-movement for min and startAnchoredPosition+movment for max

Declaration

public Vector3 movement

Field Value

TYPE	DESCRIPTION
Vector3	

soundAtFinish

Sends trigger sounds

Declaration

```
public string soundAtFinish
```

Field Value

TYPE	DESCRIPTION
System.String	

soundAtFinishReference

(deprecated) Sends trigger sounds

Declaration

```
public UIAnimator.SoundDetails soundAtFinishReference
```

Field Value

TYPE	DESCRIPTION
UIAnimator.SoundDetails	

soundAtStart

Sends trigger sounds

Declaration

```
public string soundAtStart
```

Field Value

TYPE	DESCRIPTION
System.String	

soundAtStartReference

(deprecated) Sends trigger sounds

Declaration

```
public UIAnimator.SoundDetails soundAtStartReference
```

Field Value

TYPE	DESCRIPTION
UIAnimator.SoundDetails	

time

Time is amount (seconds) that the animation will take to complete

Declaration

public float time

Field Value

TYPE	DESCRIPTION
System.Single	

Class UIAnimator.MoveOut

Inheritance

System.Object
UIAnimator.MoveOut

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class MoveOut
```

Fields

delay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

easeType

Easing is the rate of change of animation over time

Declaration

```
public DG.Tweening.Ease easeType
```

Field Value

TYPE	DESCRIPTION
DG.Tweening.Ease	

enabled

Is the animation enabled?

Declaration

```
public bool enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

moveTo

Where does the animation end?

Declaration

<code>public UIMoveDetails moveTo</code>
--

Field Value

TYPE	DESCRIPTION
UIMoveDetails	

positionAdjustment

Use this if you need to adjust the target position. You add or subtract (if the number is negative) values to the position of the target location

Declaration

<code>public Vector3 positionAdjustment</code>
--

Field Value

TYPE	DESCRIPTION
Vector3	

positionTo

This is used when the Move From LocalPosition is selected

Declaration

<code>public Vector3 positionTo</code>
--

Field Value

TYPE	DESCRIPTION
Vector3	

soundAtFinish

Sends trigger sounds

Declaration

<code>public string soundAtFinish</code>
--

Field Value

TYPE	DESCRIPTION
System.String	

soundAtFinishReference

(deprecated) Sends trigger sounds

Declaration

```
public UIManimator.SoundDetails soundAtFinishReference
```

Field Value

TYPE	DESCRIPTION
UIManimator.SoundDetails	

soundAtStart

Sends trigger sounds

Declaration

```
public string soundAtStart
```

Field Value

TYPE	DESCRIPTION
System.String	

soundAtStartReference

(deprecated) Sends trigger sounds

Declaration

```
public UIManimator.SoundDetails soundAtStartReference
```

Field Value

TYPE	DESCRIPTION
UIManimator.SoundDetails	

time

Time is amount (seconds) that the animation will take to complete

Declaration

```
public float time
```

Field Value

TYPE	DESCRIPTION
System.Single	

Enum UIAnimator.ResetType

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Obsolete]  
public enum ResetType
```

Fields

NAME	DESCRIPTION
All	
Fade	
Position	
Rotation	
Scale	

Class UIAnimator.RotationIn

Inheritance

System.Object
UIAnimator.RotationIn

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class RotationIn
```

Fields

delay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

easeType

Easing is the rate of change of animation over time

Declaration

```
public DG.Tweening.Ease easeType
```

Field Value

TYPE	DESCRIPTION
DG.Tweening.Ease	

enabled

Is the animation enabled?

Declaration

```
public bool enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

rotateFrom

Where does the animation begin from?

Declaration

```
public Vector3 rotateFrom
```

Field Value

TYPE	DESCRIPTION
Vector3	

soundAtFinish

Sends trigger sounds

Declaration

```
public string soundAtFinish
```

Field Value

TYPE	DESCRIPTION
System.String	

soundAtStart

Sends trigger sounds

Declaration

```
public string soundAtStart
```

Field Value

TYPE	DESCRIPTION
System.String	

time

Time is amount (seconds) that the animation will take to complete

Declaration

```
public float time
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class UIAnimator.RotationLoop

Inheritance

System.Object
UIAnimator.RotationLoop

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class RotationLoop
```

Fields

autoStart

If you want this animation to ignore IN and OUT animations and auto start then select this as true

Declaration

```
public bool autoStart
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

delay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

easeType

Easing is the rate of change of animation over time

Declaration

```
public DG.Tweening.Ease easeType
```

Field Value

TYPE	DESCRIPTION
DG.Tweening.Ease	

enabled

Is the animation enabled?

Declaration

public bool enabled

Field Value

TYPE	DESCRIPTION
System.Boolean	

loops

Number of loops (-1 = infinite loops)

Declaration

public int loops

Field Value

TYPE	DESCRIPTION
System.Int32	

loopType

Types of loop

Declaration

public LoopType loopType

Field Value

TYPE	DESCRIPTION
LoopType	

rotation

This rotation is calculated startRotation-rotation for min and startRotation+rotation for max

Declaration

public Vector3 rotation

Field Value

TYPE	DESCRIPTION
Vector3	

soundAtFinish

Sends trigger sounds

Declaration

```
public string soundAtFinish
```

Field Value

TYPE	DESCRIPTION
System.String	

soundAtFinishReference

(deprecated) Sends trigger sounds

Declaration

```
public UIAnimator.SoundDetails soundAtFinishReference
```

Field Value

TYPE	DESCRIPTION
UIAnimator.SoundDetails	

soundAtStart

Sends trigger sounds

Declaration

```
public string soundAtStart
```

Field Value

TYPE	DESCRIPTION
System.String	

soundAtStartReference

(deprecated) Sends trigger sounds

Declaration

```
public UIAnimator.SoundDetails soundAtStartReference
```

Field Value

TYPE	DESCRIPTION
UIAnimator.SoundDetails	

time

Time is amount (seconds) that the animation will take to complete

Declaration

public float time

Field Value

TYPE	DESCRIPTION
System.Single	

Class UIAnimator.RotationOut

Inheritance

System.Object
UIAnimator.RotationOut

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class RotationOut
```

Fields

delay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

easeType

Easing is the rate of change of animation over time

Declaration

```
public DG.Tweening.Ease easeType
```

Field Value

TYPE	DESCRIPTION
DG.Tweening.Ease	

enabled

Is the animation enabled?

Declaration

```
public bool enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

rotateTo

Where does the animation end?

Declaration

```
public Vector3 rotateTo
```

Field Value

TYPE	DESCRIPTION
Vector3	

soundAtFinish

Sends trigger sounds

Declaration

```
public string soundAtFinish
```

Field Value

TYPE	DESCRIPTION
System.String	

soundAtFinishReference

(deprecated) Sends trigger sounds

Declaration

```
public UIAnimator.SoundDetails soundAtFinishReference
```

Field Value

TYPE	DESCRIPTION
UIAnimator.SoundDetails	

soundAtStart

Sends trigger sounds

Declaration

```
public string soundAtStart
```

Field Value

TYPE	DESCRIPTION
System.String	

soundAtStartReference

(deprecated) Sends trigger sounds

Declaration

```
public UIAnimator.SoundDetails soundAtStartReference
```

Field Value

TYPE	DESCRIPTION
UIAnimator.SoundDetails	

time

Time is amount (seconds) that the animation will take to complete

Declaration

```
public float time
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class UIAnimator.ScaleIn

Inheritance

System.Object
UIAnimator.ScaleIn

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class ScaleIn
```

Fields

delay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

easeType

Easing is the rate of change of animation over time

Declaration

```
public DG.Tweening.Ease easeType
```

Field Value

TYPE	DESCRIPTION
DG.Tweening.Ease	

enabled

Is the animation enabled?

Declaration

```
public bool enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

scaleBegin

From what scale factor does the animation begin? (default: 0)

Declaration

<pre>public Vector3 scaleBegin</pre>

Field Value

TYPE	DESCRIPTION
Vector3	

soundAtFinish

Sends trigger sounds

Declaration

<pre>public string soundAtFinish</pre>
--

Field Value

TYPE	DESCRIPTION
System.String	

soundAtStart

Declaration

<pre>public string soundAtStart</pre>

Field Value

TYPE	DESCRIPTION
System.String	

time

Time is amount (seconds) that the animation will take to complete

Declaration

<pre>public float time</pre>

Field Value

TYPE	DESCRIPTION
System.Single	

Class UIAnimator.ScaleLoop

Inheritance

System.Object
UIAnimator.ScaleLoop

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class ScaleLoop
```

Fields

autoStart

If you want this animation to ignore IN and OUT animations and auto start then select this as true

Declaration

```
public bool autoStart
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

delay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

easeType

Easing is the rate of change of animation over time

Declaration

```
public DG.Tweening.Ease easeType
```

Field Value

TYPE	DESCRIPTION
DG.Tweening.Ease	

enabled

Is the animation enabled?

Declaration

public bool enabled

Field Value

TYPE	DESCRIPTION
System.Boolean	

loops

Number of loops (-1 = infinite loops)

Declaration

public int loops

Field Value

TYPE	DESCRIPTION
System.Int32	

loopType

Types of loop

Declaration

public LoopType loopType

Field Value

TYPE	DESCRIPTION
LoopType	

max

The maximum values for the scale factor of the scale loop animation (default: 1.05)

Declaration

public Vector3 max

Field Value

TYPE	DESCRIPTION
Vector3	

min

The minimum values for the scale factor of the scale loop animation (default: 1)

Declaration

```
public Vector3 min
```

Field Value

TYPE	DESCRIPTION
Vector3	

soundAtFinish

Sends trigger sounds

Declaration

```
public string soundAtFinish
```

Field Value

TYPE	DESCRIPTION
System.String	

soundAtFinishReference

(deprecated) Sends trigger sounds

Declaration

```
public UIAnimator.SoundDetails soundAtFinishReference
```

Field Value

TYPE	DESCRIPTION
UIAnimator.SoundDetails	

soundAtStart

Sends trigger sounds

Declaration

```
public string soundAtStart
```

Field Value

TYPE	DESCRIPTION
System.String	

soundAtStartReference

(deprecated) Sends trigger sounds

Declaration


```
public UIAnimator.SoundDetails soundAtStartReference
```

Field Value

TYPE	DESCRIPTION
UIAnimator.SoundDetails	

time

Time is amount (seconds) that the animation will take to complete

Declaration

```
public float time
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class UIAnimator.ScaleOut

Inheritance

System.Object
UIAnimator.ScaleOut

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class ScaleOut
```

Fields

delay

Delay is amount (seconds) that the animation will wait before beginning

Declaration

```
public float delay
```

Field Value

TYPE	DESCRIPTION
System.Single	

easeType

Easing is the rate of change of animation over time

Declaration

```
public DG.Tweening.Ease easeType
```

Field Value

TYPE	DESCRIPTION
DG.Tweening.Ease	

enabled

Is the animation enabled?

Declaration

```
public bool enabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

scaleEnd

This is the scale factor at which the animation ends at

Declaration

public Vector3 scaleEnd

Field Value

TYPE	DESCRIPTION
Vector3	

soundAtFinish

Sends trigger sounds

Declaration

public string soundAtFinish

Field Value

TYPE	DESCRIPTION
System.String	

soundAtFinishReference

(deprecated) Sends trigger sounds

Declaration

public UIAnimator.SoundDetails soundAtFinishReference

Field Value

TYPE	DESCRIPTION
UIAnimator.SoundDetails	

soundAtStart

Sends trigger sounds

Declaration

public string soundAtStart

Field Value

TYPE	DESCRIPTION
System.String	

soundAtStartReference

(deprecated) Sends trigger sounds

Declaration

```
public UIAnimator.SoundDetails soundAtStartReference
```

Field Value

TYPE	DESCRIPTION
UIAnimator.SoundDetails	

time

Time is amount (seconds) that the animation will take to complete

Declaration

```
public float time
```

Field Value

TYPE	DESCRIPTION
System.Single	

Class UIAnimator.SoundDetails

Inheritance

System.Object
UIAnimator.SoundDetails

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class SoundDetails
```

Fields

soundName

Declaration

```
public string soundName
```

Field Value

TYPE	DESCRIPTION
System.String	

Enum UIAnimator.SoundOutput

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Obsolete]  
public enum SoundOutput
```

Fields

NAME	DESCRIPTION
AudioSource	
MasterAudioFireCustomEvent	
MasterAudioPlaySoundAndForget	

Enum UIAnimator.TweenIdAnimation

Used to map the tween ids.

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public enum TweenIdAnimation
```

Fields

NAME	DESCRIPTION
In	
Loop	
Out	
Punch	

Enum UIAnimator.TweenIdType

Used to map the tween ids.

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public enum TweenIdType
```

Fields

NAME	DESCRIPTION
Fade	
Move	
Rotate	
Scale	

Class UIAnimatorUtil

Inheritance

System.Object
UIAnimatorUtil

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class UIAnimatorUtil
```

Fields

DEFAULT_PRESET_CATEGORY

Declaration

```
public const string DEFAULT_PRESET_CATEGORY = "Uncategorized"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_PRESET_NAME

Declaration

```
public const string DEFAULT_PRESET_NAME = "DefaultPreset"
```

Field Value

TYPE	DESCRIPTION
System.String	

FOLDER_NAME_IN

Declaration

```
public const string FOLDER_NAME_IN = "In/"
```

Field Value

TYPE	DESCRIPTION
System.String	

FOLDER_NAME_LOOP

Declaration

```
public const string FOLDER_NAME_LOOP = "Loop/"
```

Field Value

TYPE	DESCRIPTION
System.String	

FOLDER_NAME_OUT

Declaration

```
public const string FOLDER_NAME_OUT = "Out/"
```

Field Value

TYPE	DESCRIPTION
System.String	

FOLDER_NAME_PUNCH

Declaration

```
public const string FOLDER_NAME_PUNCH = "Punch/"
```

Field Value

TYPE	DESCRIPTION
System.String	

InAnimDataPresetsDatabase

Declaration

```
public static Dictionary<string, List<AnimData>> InAnimDataPresetsDatabase
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Collections.Generic.List<DoozyUI.AnimData>>	

LoopDataPresetsDatabase

Declaration

```
public static Dictionary<string, List<LoopData>> LoopDataPresetsDatabase
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Collections.Generic.List<DoozyUI.LoopData>>	

OutAnimDataPresetsDatabase

Declaration

```
public static Dictionary<string, List<AnimData>> OutAnimDataPresetsDatabase
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Collections.Generic.List<DoozyUI.AnimData>>	

PunchDataPresetsDatabase

Declaration

```
public static Dictionary<string, List<PunchData>> PunchDataPresetsDatabase
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Collections.Generic.List<DoozyUI.PunchData>>	

RESOURCES_PATH_ANIMATIONS

Declaration

```
public const string RESOURCES_PATH_ANIMATIONS = "DUI/Animations/"
```

Field Value

TYPE	DESCRIPTION
System.String	

RESOURCES_PATH_IN_ANIM_DATA

Declaration

```
public const string RESOURCES_PATH_IN_ANIM_DATA = "DUI/Animations/In/"
```

Field Value

TYPE	DESCRIPTION
System.String	

RESOURCES_PATH_LOOP_DATA

Declaration

```
public const string RESOURCES_PATH_LOOP_DATA = "DUI/Animations/Loop/"
```

Field Value

TYPE	DESCRIPTION
System.String	

RESOURCES_PATH_OUT_ANIM_DATA

Declaration

```
public const string RESOURCES_PATH_OUT_ANIM_DATA = "DUI/Animations/Out/"
```

Field Value

TYPE	DESCRIPTION
System.String	

RESOURCES_PATH_PUNCH_DATA

Declaration

```
public const string RESOURCES_PATH_PUNCH_DATA = "DUI/Animations/Punch/"
```

Field Value

TYPE	DESCRIPTION
System.String	

Properties

InAnimPresetCategories

Declaration

```
public static List<string> InAnimPresetCategories { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

LoopPresetCategories

Declaration

```
public static List<string> LoopPresetCategories { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

OutAnimPresetCategories

Declaration

```
public static List<string> OutAnimPresetCategories { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

PunchPresetCategories

Declaration

```
public static List<string> PunchPresetCategories { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

RELATIVE_PATH_ANIMATIONS

Declaration

```
public static string RELATIVE_PATH_ANIMATIONS { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

RELATIVE_PATH_IN_ANIM_DATA

Declaration

```
public static string RELATIVE_PATH_IN_ANIM_DATA { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

RELATIVE_PATH_LOOP_DATA

Declaration

```
public static string RELATIVE_PATH_LOOP_DATA { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

RELATIVE_PATH_OUT_ANIM_DATA

Declaration

```
public static string RELATIVE_PATH_OUT_ANIM_DATA { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

RELATIVE_PATH_PUNCH_DATA

Declaration

```
public static string RELATIVE_PATH_PUNCH_DATA { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

CreateInAnimPreset(String, String, Anim)

Declaration

```
public static void CreateInAnimPreset(string presetCategory, string presetName, Anim anim)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	
Anim	anim	

CreateLoopPreset(String, String, Loop)

Declaration

```
public static void CreateLoopPreset(string presetCategory, string presetName, Loop loop)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	
Loop	loop	

CreateOutAnimPreset(String, String, Anim)

Declaration

```
public static void CreateOutAnimPreset(string presetCategory, string presetName, Anim anim)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	
Anim	anim	

CreatePunchPreset(String, String, Punch)

Declaration

```
public static void CreatePunchPreset(string presetCategory, string presetName, Punch punch)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	
Punch	punch	

DeleteInAnimCategory(String)

Declaration

```
public static void DeleteInAnimCategory(string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	

DeleteInAnimPreset(String, String)

Declaration

```
public static void DeleteInAnimPreset(string presetCategory, string presetName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

DeleteLoopCategory(String)

Declaration

```
public static void DeleteLoopCategory(string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	

DeleteLoopPreset(String, String)

Declaration

```
public static void DeleteLoopPreset(string presetCategory, string presetName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	

TYPE	NAME	DESCRIPTION
System.String	presetName	

DeleteOutAnimCategory(String)

Declaration

```
public static void DeleteOutAnimCategory(string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	

DeleteOutAnimPreset(String, String)

Declaration

```
public static void DeleteOutAnimPreset(string presetCategory, string presetName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

DeletePunchCategory(String)

Declaration

```
public static void DeletePunchCategory(string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	

DeletePunchPreset(String, String)

Declaration

```
public static void DeletePunchPreset(string presetCategory, string presetName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

GetInAnim(String, String)

Declaration


```
public static Anim GetInAnim(string presetCategory, string presetName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

Returns

TYPE	DESCRIPTION
Anim	

GetInAnimData(String, String)

Declaration

```
public static AnimData GetInAnimData(string presetCategory, string presetName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

Returns

TYPE	DESCRIPTION
AnimData	

GetInAnimPresetNames(String)

Declaration

```
public static List<string> GetInAnimPresetNames(string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

GetLoop(String, String)

Declaration

```
public static Loop GetLoop(string presetCategory, string presetName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

Returns

TYPE	DESCRIPTION
Loop	

GetLoopData(String, String)

Declaration

<pre>public static LoopData GetLoopData(string presetCategory, string presetName)</pre>

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

Returns

TYPE	DESCRIPTION
LoopData	

GetLoopPresetNames(String)

Declaration

<pre>public static List<string> GetLoopPresetNames(string presetCategory)</pre>

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

GetOutAnim(String, String)

Declaration

<pre>public static Anim GetOutAnim(string presetCategory, string presetName)</pre>
--

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

Returns

TYPE	DESCRIPTION
Anim	

GetOutAnimData(String, String)

Declaration

```
public static AnimData GetOutAnimData(string presetCategory, string presetName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

Returns

TYPE	DESCRIPTION
AnimData	

GetOutAnimPresetNames(String)

Declaration

```
public static List<string> GetOutAnimPresetNames(string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

GetPunch(String, String)

Declaration

```
public static Punch GetPunch(string presetCategory, string presetName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

Returns

TYPE	DESCRIPTION
Punch	

GetPunchData(String, String)

Declaration

```
public static PunchData GetPunchData(string presetCategory, string presetName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

Returns

TYPE	DESCRIPTION
PunchData	

GetPunchPresetNames(String)

Declaration

```
public static List<string> GetPunchPresetNames(string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

GetResource<T>(String, String)

Declaration

```
public static T GetResource<T>(string resourcesPath, string fileName)where T : ScriptableObject
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	resourcesPath	
System.String	fileName	

Returns

TYPE	DESCRIPTION
T	

Type Parameters

NAME	DESCRIPTION
T	

InAnimPresetCategoryExists(String)

Declaration

```
public static bool InAnimPresetCategoryExists(string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	

Returns

TYPE	DESCRIPTION
System.Boolean	

InAnimPresetExists(String, String)

Declaration

```
public static bool InAnimPresetExists(string presetCategory, string presetName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

Returns

TYPE	DESCRIPTION
System.Boolean	

LoopPresetCategoryExists(String)

Declaration

```
public static bool LoopPresetCategoryExists(string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	

Returns

TYPE	DESCRIPTION
System.Boolean	

LoopPresetExists(String, String)

Declaration

```
public static bool LoopPresetExists(string presetCategory, string presetName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

Returns

TYPE	DESCRIPTION
System.Boolean	

OutAnimPresetCategoryExists(String)

Declaration

```
public static bool OutAnimPresetCategoryExists(string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	

Returns

TYPE	DESCRIPTION
System.Boolean	

OutAnimPresetExists(String, String)

Declaration

```
public static bool OutAnimPresetExists(string presetCategory, string presetName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

Returns

TYPE	DESCRIPTION
System.Boolean	

PunchPresetCategoryExists(String)

Declaration

```
public static bool PunchPresetCategoryExists(string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	

Returns

TYPE	DESCRIPTION
System.Boolean	

PunchPresetExists(String, String)

Declaration

```
public static bool PunchPresetExists(string presetCategory, string presetName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	presetCategory	
System.String	presetName	

Returns

TYPE	DESCRIPTION
System.Boolean	

RefreshInAnimDataPresetsDatabase()

Declaration

```
public static void RefreshInAnimDataPresetsDatabase()
```

RefreshLoopDataPresetsDatabase()

Declaration

```
public static void RefreshLoopDataPresetsDatabase()
```

RefreshOutAnimDataPresetsDatabase()

Declaration

```
public static void RefreshOutAnimDataPresetsDatabase()
```

RefreshPunchDataPresetsDatabase()

Declaration

```
public static void RefreshPunchDataPresetsDatabase()
```

RenameInAnimPreset(String, String, String)

Declaration

```
public static void RenameInAnimPreset(string oldName, string newName, string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	oldName	
System.String	newName	
System.String	presetCategory	

RenameLoopPreset(String, String, String)

Declaration

```
public static void RenameLoopPreset(string oldName, string newName, string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	oldName	
System.String	newName	
System.String	presetCategory	

RenameOutAnimPreset(String, String, String)

Declaration

```
public static void RenameOutAnimPreset(string oldName, string newName, string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	oldName	
System.String	newName	

TYPE	NAME	DESCRIPTION
System.String	presetCategory	

RenamePunchPreset(String, String, String)

Declaration

```
public static void RenamePunchPreset(string oldName, string newName, string presetCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	oldName	
System.String	newName	
System.String	presetCategory	

Class UIButton

Inheritance

System.Object
UIButton

Namespace: **DoozyUI**
Assembly: Assembly-CSharp.dll

Syntax

```
public class UIButton : MonoBehaviour, IPointerEnterHandler, IPointerExitHandler, IPointerDownHandler, IPointerUpHandler, IPointerClickHandler, ISelectHandler, IDeselectHandler
```

Fields

addToNavigationHistory

This was used by the old navigation system that only worked for OnClick. The new system has new button actions and this old value is the equivalent of the new `onClickNavigation.addToNavigationHistory` value

Declaration

```
[Obsolete]  
public bool addToNavigationHistory
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

allowMultipleClicks

Should the button get disabled for a set interval (`disableButtonInterval`) between each click. By default we allow the user to press the button multiple times.

Declaration

```
public bool allowMultipleClicks
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

backButton

Declaration

```
[Obsolete]  
public bool backButton
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

BETWEEN_CLICKS_DISABLE_INTERVAL

Default value used to disable button after each click. Used when allow multiple clicks is set to false.

Declaration

```
public const float BETWEEN_CLICKS_DISABLE_INTERVAL = 0.4F
```

Field Value

TYPE	DESCRIPTION
System.Single	

buttonCategory

The category this button name belongs to. The category is used only for database sorting purposes only. It does not matter when registering a button action.

Declaration

```
public string buttonCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

buttonName

The name of this button. This is the value the system looks at when this button issues an action.

Declaration

```
public string buttonName
```

Field Value

TYPE	DESCRIPTION
System.String	

buttonNameReference

Declaration

```
[Obsolete]  
public UIButton.ButtonName buttonNameReference
```

Field Value

TYPE	DESCRIPTION
UIButton.ButtonName	

customOnClickSound

Used by the custom inspector to allow you to type a sound name instead of selecting it from the UISounds Database.

Declaration

```
public bool customOnClickSound
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

customOnDoubleClickSound

Used by the custom inspector to allow you to type a sound name instead of selecting it from the UISounds Database.

Declaration

```
public bool customOnDoubleClickSound
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

customOnLongClickSound

Used by the custom inspector to allow you to type a sound name instead of selecting it from the UISounds Database.

Declaration

```
public bool customOnLongClickSound
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

customOnPointerDownSound

Used by the custom inspector to allow you to type a sound name instead of selecting it from the UISounds Database.

Declaration

```
public bool customOnPointerDownSound
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

customOnPointerEnterSound

Used by the custom inspector to allow you to type a sound name instead of selecting it from the UISounds Database.

Declaration

```
public bool customOnPointerEnterSound
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

customOnPointerExitSound

Used by the custom inspector to allow you to type a sound name instead of selecting it from the UISounds Database.

Declaration

```
public bool customOnPointerExitSound
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

customOnPointerUpSound

Used by the custom inspector to allow you to type a sound name instead of selecting it from the UISounds Database.

Declaration

```
public bool customOnPointerUpSound
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

debug

Enables debug logs.

Declaration

```
public bool debug
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

DESELECT_BUTTON_DELAY

Special time interval added when deselecting a button. It fixes some anomalies.

Declaration

```
public const float DESELECT_BUTTON_DELAY = 0.1F
```

Field Value

TYPE	DESCRIPTION
System.Single	

deselectButtonOnClick

Should the button get deselected after each click. This is useful if you do not want this button to get selected after a click.

Declaration

```
public bool deselectButtonOnClick
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

disableButtonInterval

If allowMultipleClicks is false, then this is the interval that this button will be disabled for between each click.

Declaration

```
public float disableButtonInterval
```

Field Value

TYPE	DESCRIPTION
System.Single	

DOUBLE_CLICK_REGISTER_INTERVAL

Default time interval used to register a double click. This is the time interval calculated between two sequential clicks to determine if either a double click or two separate clicks occurred.

Declaration

```
public const float DOUBLE_CLICK_REGISTER_INTERVAL = 0.2F
```

Field Value

TYPE	DESCRIPTION
System.Single	

doubleClickRegisterInterval

Time interval used to register a double click. This is the time interval calculated between two sequential clicks to determine if either a double click or two separate clicks occurred.

Declaration

```
public float doubleClickRegisterInterval
```

Field Value

TYPE	DESCRIPTION
System.Single	

gameEvents

Declaration

```
[Obsolete]  
public List<string> gameEvents
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

hideElements

Declaration

[Obsolete] public List<string> hideElements
--

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

highlightedAnimationSettings

Declaration

[Obsolete] public UIAnimationManager.ButtonLoopsAnimations highlightedAnimationSettings
--

Field Value

TYPE	DESCRIPTION
UIAnimationManager.ButtonLoopsAnimations	

loadNormalLoopPresetAtRuntime

Should the system load, at runtime, the Loop Preset with the set Preset Category and Preset Name. This overrides any values set in the inspector.

Declaration

public bool loadNormalLoopPresetAtRuntime

Field Value

TYPE	DESCRIPTION
System.Boolean	

loadOnClickPunchPresetAtRuntime

Should the system load, at runtime, the Punch Preset with the set Preset Category and Preset Name. This overrides any values set in the inspector.

Declaration

public bool loadOnClickPunchPresetAtRuntime

Field Value

TYPE	DESCRIPTION
System.Boolean	

loadOnDoubleClickPunchPresetAtRuntime

Should the system load, at runtime, the Punch Preset with the set Preset Category and Preset Name. This overrides any values set in the inspector.

Declaration

```
public bool loadOnDoubleClickPunchPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

loadOnLongClickPunchPresetAtRuntime

Should the system load, at runtime, the Punch Preset with the set Preset Category and Preset Name. This overrides any values set in the inspector.

Declaration

```
public bool loadOnLongClickPunchPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

loadOnPointerDownPunchPresetAtRuntime

Should the system load, at runtime, the Punch Preset with the set Preset Category and Preset Name. This overrides any values set in the inspector.

Declaration

```
public bool loadOnPointerDownPunchPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

loadOnPointerEnterPunchPresetAtRuntime

Should the system load, at runtime, the Punch Preset with the set Preset Category and Preset Name. This overrides any values set in the inspector.

Declaration

```
public bool loadOnPointerEnterPunchPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

loadOnPointerExitPunchPresetAtRuntime

Should the system load, at runtime, the Punch Preset with the set Preset Category and Preset Name. This overrides any values set in the inspector.

Declaration

```
public bool loadOnPointerExitPunchPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

loadOnPointerUpPunchPresetAtRuntime

Should the system load, at runtime, the Punch Preset with the set Preset Category and Preset Name. This overrides any values set in the inspector.

Declaration

```
public bool loadOnPointerUpPunchPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

loadSelectedLoopPresetAtRuntime

Should the system load, at runtime, the Loop Preset with the set Preset Category and Preset Name. This overrides any values set in the inspector.

Declaration

```
public bool loadSelectedLoopPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

LONG_CLICK_REGISTER_INTERVAL

Default time interval used to register a long click. This is the time interval a button has to be pressed down to be considered a long click.

Declaration

```
public const float LONG_CLICK_REGISTER_INTERVAL = 0.5F
```

Field Value

TYPE	DESCRIPTION
System.Single	

longClickRegisterInterval

Time interval used to register a long click. This is the time interval a button has to be pressed down to be considered a long click.

Declaration

```
public float longClickRegisterInterval
```

Field Value

TYPE	DESCRIPTION
System.Single	

NORMAL_LOOP_ID

This is an extra id tag given to the tweener in order to locate the proper tween that manages the normal loop animations.

Declaration

```
public const string NORMAL_LOOP_ID = "ButtonNormalLoop"
```

Field Value

TYPE	DESCRIPTION
System.String	

normalAnimationSettings

Declaration

```
[Obsolete]  
public UIManager.ButtonLoopsAnimations normalAnimationSettings
```

Field Value

TYPE	DESCRIPTION
UIAnimator.ButtonLoopsAnimations	

normalLoop

Loop Animation Settings

Declaration

```
public Loop normalLoop
```

Field Value

TYPE	DESCRIPTION
Loop	

normalLoopPresetCategory

Loop Animation Preset Category Name

Declaration

```
public string normalLoopPresetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

normalLoopPresetName

Loop Animation Preset Name

Declaration

<pre>public string normalLoopPresetName</pre>

Field Value

TYPE	DESCRIPTION
System.String	

ON_POINTER_ENTER_DISABLE_INTERVAL

Default value used to disable the on pointer enter capture functionality after it has been triggered. Useful for certain cases.

Declaration

<pre>public const float ON_POINTER_ENTER_DISABLE_INTERVAL = 0.4F</pre>
--

Field Value

TYPE	DESCRIPTION
System.Single	

ON_POINTER_EXIT_DISABLE_INTERVAL

Default value used to disable the on pointer exit capture functionality after it has been triggered. Useful for certain cases.

Declaration

<pre>public const float ON_POINTER_EXIT_DISABLE_INTERVAL = 0.4F</pre>

Field Value

TYPE	DESCRIPTION
System.Single	

OnClick

UnityEvent invoked when on click has been captured by the system.

Declaration

<pre>public UnityEvent OnClick</pre>

Field Value

TYPE	DESCRIPTION
UnityEvent	

onClickAnimationSettings

Declaration

```
[Obsolete]
public UIManager.OnClickAnimations onClickAnimationSettings
```

Field Value

TYPE	DESCRIPTION
UIAnimationManager.OnClickAnimations	

onClickGameEvents

A list of game events that are sent when on click has been triggered.

Declaration

```
public List<string> onClickGameEvents
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

onClickNavigation

UINavigation settings.

Declaration

```
public NavigationPointerData onClickNavigation
```

Field Value

TYPE	DESCRIPTION
NavigationPointerData	

onClickPunch

Punch Animation Settings

Declaration

```
public Punch onClickPunch
```

Field Value

TYPE	DESCRIPTION
Punch	

onClickPunchPresetCategory

Punch Animation Preset Category Name

Declaration

```
public string onClickPunchPresetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

onClickPunchPresetName

Punch Animation Preset Name

Declaration

```
public string onClickPunchPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

onClickSound

The sound name of the sound that gets played on click.

Declaration

```
public string onClickSound
```

Field Value

TYPE	DESCRIPTION
System.String	

onClickSoundReference

Declaration

```
[Obsolete]  
public UIButton.ButtonSound onClickSoundReference
```

Field Value

TYPE	DESCRIPTION
UIButton.ButtonSound	

OnDoubleClick

UnityEvent invoked when on double click has been captured by the system.

Declaration

```
public UnityEvent OnDoubleClick
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

onDoubleClickGameEvents

A list of game events that are sent when on double click has been triggered.

Declaration

<code>public List<string> onDoubleClickGameEvents</code>
--

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

onDoubleClickNavigation

UINavigation settings.

Declaration

<code>public NavigationPointerData onDoubleClickNavigation</code>

Field Value

TYPE	DESCRIPTION
NavigationPointerData	

onDoubleClickPunch

Punch Animation Settings

Declaration

<code>public Punch onDoubleClickPunch</code>
--

Field Value

TYPE	DESCRIPTION
Punch	

onDoubleClickPunchPresetCategory

Punch Animation Preset Category Name

Declaration

<code>public string onDoubleClickPunchPresetCategory</code>

Field Value

TYPE	DESCRIPTION
System.String	

onDoubleClickPunchPresetName

Punch Animation Preset Name

Declaration

```
public string onDoubleClickPunchPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

onDoubleClickSound

The sound name of the sound that gets played on click.

Declaration

```
public string onDoubleClickSound
```

Field Value

TYPE	DESCRIPTION
System.String	

OnLongClick

UnityEvent invoked when on long click has been captured by the system.

Declaration

```
public UnityEvent OnLongClick
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

onLongClickGameEvents

A list of game events that are sent when on long click has been triggered.

Declaration

```
public List<string> onLongClickGameEvents
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

onLongClickNavigation

UINavigation settings.

Declaration

```
public NavigationPointerData onLongClickNavigation
```

Field Value

TYPE	DESCRIPTION
NavigationPointerData	

onLongClickPunch

Punch Animation Settings

Declaration

```
public Punch onLongClickPunch
```

Field Value

TYPE	DESCRIPTION
Punch	

onLongClickPunchPresetCategory

Punch Animation Preset Category Name

Declaration

```
public string onLongClickPunchPresetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

onLongClickPunchPresetName

Punch Animation Preset Name

Declaration

```
public string onLongClickPunchPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

onLongClickSound

The sound name of the sound that gets played on click.

Declaration

```
public string onLongClickSound
```

Field Value

TYPE	DESCRIPTION
System.String	

OnPointerDown

UnityEvent invoked when on pointer down has been captured by the system.

Declaration

```
public UnityEvent OnPointerDown
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

onPointerDownGameEvents

A list of game events that are sent when on pointer down has been triggered.

Declaration

```
public List<string> onPointerDownGameEvents
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

onPointerDownNavigation

UINavigation settings.

Declaration

```
public NavigationPointerData onPointerDownNavigation
```

Field Value

TYPE	DESCRIPTION
NavigationPointerData	

onPointerDownPunch

Punch Animation Settings

Declaration

```
public Punch onPointerDownPunch
```

Field Value

TYPE	DESCRIPTION
Punch	

onPointerDownPunchPresetCategory

Punch Animation Preset Category Name

Declaration

```
public string onPointerDownPunchPresetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

onPointerDownPunchPresetName

Punch Animation Preset Name

Declaration

```
public string onPointerDownPunchPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

onPointerDownSound

The sound name of the sound that gets played on pointer down.

Declaration

```
public string onPointerDownSound
```

Field Value

TYPE	DESCRIPTION
System.String	

OnPointerEnter

UnityEvent invoked when on pointer enter has been captured by the system.

Declaration

```
public UnityEvent OnPointerEnter
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

onPointerEnterDisableInterval

Time interval when the on pointer enter functionality is disabled after it has been triggered. Useful in certain cases.

Declaration

```
public float onPointerEnterDisableInterval
```

Field Value

TYPE	DESCRIPTION
System.Single	

onPointerEnterGameEvents

A list of game events that are sent when on pointer enter has been triggered.

Declaration

```
public List<string> onPointerEnterGameEvents
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

onPointerEnterNavigation

UINavigation settings.

Declaration

```
public NavigationPointerData onPointerEnterNavigation
```

Field Value

TYPE	DESCRIPTION
NavigationPointerData	

onPointerEnterPunch

Punch Animation Settings

Declaration

```
public Punch onPointerEnterPunch
```

Field Value

TYPE	DESCRIPTION
Punch	

onPointerEnterPunchPresetCategory

Punch Animation Preset Category Name

Declaration

```
public string onPointerEnterPunchPresetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

onPointerEnterPunchPresetName

Punch Animation Preset Name

Declaration

<code>public string onPointerEnterPunchPresetName</code>
--

Field Value

TYPE	DESCRIPTION
System.String	

onPointerEnterSound

The sound name of the sound that gets played on pointer enter.

Declaration

<code>public string onPointerEnterSound</code>
--

Field Value

TYPE	DESCRIPTION
System.String	

OnPointerExit

UnityEvent invoked when on pointer exit has been captured by the system.

Declaration

<code>public UnityEvent OnPointerExit</code>
--

Field Value

TYPE	DESCRIPTION
UnityEvent	

onPointerExitDisableInterval

Time interval when the on pointer exit functionality is disabled after it has been triggered. Useful in certain cases.

Declaration

<code>public float onPointerExitDisableInterval</code>
--

Field Value

TYPE	DESCRIPTION
System.Single	

onPointerExitGameEvents

A list of game events that are sent when on pointer exit has been triggered.

Declaration

```
public List<string> onPointerExitGameEvents
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

onPointerExitNavigation

UINavigation settings.

Declaration

```
public NavigationPointerData onPointerExitNavigation
```

Field Value

TYPE	DESCRIPTION
NavigationPointerData	

onPointerExitPunch

Punch Animation Settings

Declaration

```
public Punch onPointerExitPunch
```

Field Value

TYPE	DESCRIPTION
Punch	

onPointerExitPunchPresetCategory

Punch Animation Preset Category Name

Declaration

```
public string onPointerExitPunchPresetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

onPointerExitPunchPresetName

Punch Animation Preset Name

Declaration

```
public string onPointerExitPunchPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

onPointerExitSound

The sound name of the sound that gets played on pointer exit.

Declaration

```
public string onPointerExitSound
```

Field Value

TYPE	DESCRIPTION
System.String	

OnPointerUp

UnityEvent invoked when on pointer up has been captured by the system.

Declaration

```
public UnityEvent OnPointerUp
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

onPointerUpGameEvents

A list of game events that are sent when on pointer up has been triggered.

Declaration

```
public List<string> onPointerUpGameEvents
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

onPointerUpNavigation

UINavigation settings.

Declaration

```
public NavigationPointerData onPointerUpNavigation
```

Field Value

TYPE	DESCRIPTION
NavigationPointerData	

onPointerUpPunch

Punch Animation Settings

Declaration

<code>public Punch onPointerUpPunch</code>
--

Field Value

TYPE	DESCRIPTION
Punch	

onPointerUpPunchPresetCategory

Punch Animation Preset Category Name

Declaration

<code>public string onPointerUpPunchPresetCategory</code>

Field Value

TYPE	DESCRIPTION
System.String	

onPointerUpPunchPresetName

Punch Animation Preset Name

Declaration

<code>public string onPointerUpPunchPresetName</code>

Field Value

TYPE	DESCRIPTION
System.String	

onPointerUpSound

The sound name of the sound that gets played on pointer up.

Declaration

<code>public string onPointerUpSound</code>

Field Value

TYPE	DESCRIPTION
System.String	

SELECTED_LOOP_ID

This is an extra id tag given to the tweener in order to locate the proper tween that manages the selected loop animations.

Declaration

```
public const string SELECTED_LOOP_ID = "ButtonSelectedLoop"
```

Field Value

TYPE	DESCRIPTION
System.String	

selectedLoop

Loop Animation Settings

Declaration

```
public Loop selectedLoop
```

Field Value

TYPE	DESCRIPTION
Loop	

selectedLoopPresetCategory

Loop Animation Preset Category Name

Declaration

```
public string selectedLoopPresetCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

selectedLoopPresetName

Loop Animation Preset Name

Declaration

```
public string selectedLoopPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

showElements

Declaration

```
[Obsolete]  
public List<string> showElements
```


Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

singleClickMode

Determines if on click is triggered instantly or after it checks if it's a double click or not. Depending on your use case, you might need the Instant or Delayed mode. Default is set to Instant.

Declaration

<pre>public UIButton.SingleClickMode singleClickMode</pre>
--

Field Value

TYPE	DESCRIPTION
UIButton.SingleClickMode	

useHighlightedStateAnimations

Declaration

<pre>[Obsolete] public bool useHighlightedStateAnimations</pre>

Field Value

TYPE	DESCRIPTION
System.Boolean	

useNormalStateAnimations

Declaration

<pre>[Obsolete] public bool useNormalStateAnimations</pre>
--

Field Value

TYPE	DESCRIPTION
System.Boolean	

useOnClickAnimations

Toggles the OnClick functionality. Not recommended to be disabled. If you disable this functionality, do some tests to be sure that the button behaves as you want it to.

Declaration

<pre>public bool useOnClickAnimations</pre>

Field Value

TYPE	DESCRIPTION
System.Boolean	

useOnDoubleClick

Toggles the OnDoubleClick functionality.

Declaration

```
public bool useOnDoubleClick
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

useOnLongClick

Toggles the OnLongClick functionality.

Declaration

```
public bool useOnLongClick
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

useOnPointerDown

Toggles the OnPointerDown functionality.

Declaration

```
public bool useOnPointerDown
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

useOnPointerEnter

Toggles the OnPointerEnter functionality.

Declaration

```
public bool useOnPointerEnter
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

useOnPointerExit

Toggles the OnPointerExit functionality.

Declaration

```
public bool useOnPointerExit
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

useOnPointerUp

Toggles the OnPointerUp functionality.

Declaration

```
public bool useOnPointerUp
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

waitForOnClickAnimation

If enabled, the button action and game events are sent after the on click punch animation has finished playing. This is useful if you want be sure the uses sees the button animation.

Declaration

```
public bool waitForOnClickAnimation
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

waitForOnDoubleClickAnimation

If enabled, the button action and game events are sent after the on double click punch animation has finished playing. This is useful if you want be sure the uses sees the button animation.

Declaration

```
public bool waitForOnDoubleClickAnimation
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

waitForOnLongClickAnimation

If enabled, the button action and game events are sent after the on long click punch animation has finished playing. This is useful

if you want be sure the uses sees the button animation.

Declaration

```
public bool WaitForOnLongClickAnimation
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Properties

Button

Returns the Button component.

Declaration

```
public Button Button { get; }
```

Property Value

TYPE	DESCRIPTION
Button	

interactable

Use Interactable instead.

Declaration

```
[Obsolete]  
public bool interactable { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Interactable

Returns true if the button's Button component is interactable. This also toggles this button's interactability.

Declaration

```
public bool Interactable { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

IsBackButton

Returns true if this button's name is 'Back'

Declaration

```
public bool IsBackButton { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

IsSelected

Returns true if this button is selected, by checking the `EventSystem.current.currentSelectedGameObject`

Declaration

```
public bool IsSelected { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

RectTransform

Returns the RectTransform component.

Declaration

```
public RectTransform RectTransform { get; }
```

Property Value

TYPE	DESCRIPTION
RectTransform	

Methods

AddGameEvent(String, UIButton.ButtonActionType)

Add a game event to the target action type gameEvents list.

Declaration

```
public void AddGameEvent(string eventName, UIButton.ButtonActionType buttonActionType =  
UIButton.ButtonActionType.OnClick)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	eventName	
UIButton.ButtonActionType	buttonActionType	

DisableButton()

Sets Interactable to false.

Declaration

```
public void DisableButton()
```

DisableButton(Single)

Sets Interactable to false for the set duration. After that it sets Interactable to true.

Declaration

```
public void DisableButton(float duration)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	duration	

EnableButton()

Sets Interactable to true.

Declaration

```
public void EnableButton()
```

ExecuteButtonClick()

Executes the button click by playing the button sound (if set), starting the OnClick animation (if enabled) and sending the ButtonClick and GameEvents to the UIManager

Declaration

```
[Obsolete]  
public void ExecuteButtonClick()
```

ExecuteClick(Boolean)

Executes the OnClick trigger. You can force an execution of this trigger (regardless if it's enabled or not) by calling this method with forced set to TRUE

Declaration

```
public void ExecuteClick(bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	forced	Fires this trigger regardless if it is enabled or not (default:false)

ExecuteDoubleClick(Boolean)

Executes the OnDoubleClick trigger. You can force an execution of this trigger (regardless if it's enabled or not) by calling this method with forced set to TRUE

Declaration

```
public void ExecuteDoubleClick(bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	forced	Fires this trigger regardless if it is enabled or not (default:false)

ExecuteLongClick(Boolean)

Executes the OnLongClick trigger. You can force an execution of this trigger (regardless if it's enabled or not) by calling this method with forced set to TRUE

Declaration

```
public void ExecuteLongClick(bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	forced	Fires this trigger regardless if it is enabled or not (default:false)

ExecuteOnPointerEnter(Boolean)

Executes the OnPointerEnter trigger. You can force an execution of this trigger (regardless if it's enabled or not) by calling this method with forced set to TRUE

Declaration

```
public void ExecuteOnPointerEnter(bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	forced	Fires this trigger regardless if it is enabled or not (default:false)

ExecutePointerDown(Boolean)

Executes the OnPointerDown trigger. You can force an execution of this trigger (regardless if it's enabled or not) by calling this method with forced set to TRUE

Declaration

```
public void ExecutePointerDown(bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	forced	Fires this trigger regardless if it is enabled or not (default:false)

ExecutePointerUp(Boolean)

Executes the OnPointerUp trigger. You can force an execution of this trigger (regardless if it's enabled or not) by calling this method with forced set to TRUE

Declaration

```
public void ExecutePointerUp(bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	forced	Fires this trigger regardless if it is enabled or not (default:false)

GetButtonType(UIButton.ButtonClickType)

Converts enum type from ButtonClickType to ButtonActionType

Declaration

```
public static UIButton.ButtonActionType GetButtonType(UIButton.ButtonClickType clickType)
```

Parameters

TYPE	NAME	DESCRIPTION
UIButton.ButtonClickType	clickType	

Returns

TYPE	DESCRIPTION
UIButton.ButtonActionType	

GetButtonClickType(UIButton.ButtonActionType)

Converts enum type from ButtonActionType to ButtonClickType

Declaration

```
public static UIButton.ButtonClickType GetButtonClickType(UIButton.ButtonActionType actionType)
```

Parameters

TYPE	NAME	DESCRIPTION
UIButton.ButtonActionType	actionType	

Returns

TYPE	DESCRIPTION
UIButton.ButtonClickType	

OnDeselect(BaseEventData)

Used by IDeselectHandler.

Declaration

```
public void OnDeselect(BaseEventData eventData)
```

Parameters

TYPE	NAME	DESCRIPTION
BaseEventData	eventData	

OnSelect(BaseEventData)

Used by ISelectHandler.

Declaration

```
public void OnSelect(BaseEventData eventData)
```

Parameters

TYPE	NAME	DESCRIPTION
BaseEventData	eventData	

RemoveGameEvent(String, UIButton.ButtonActionType)

Remove game event from the target action type gameEvents list.

Declaration

```
public void RemoveGameEvent(string eventName, UIButton.ButtonActionType buttonActionType =
UIButton.ButtonActionType.OnClick)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	eventName	
UIButton.ButtonActionType	buttonActionType	

ResetAnimations()

Declaration

```
public void ResetAnimations()
```

SendButtonClick()

Simulates this button's click action, without playing the set on click sound and punch animation.

Declaration

```
public void SendButtonClick()
```

SendButtonClick(Boolean, Boolean, Boolean, Boolean)

Simulates this button's click action and plays the set on click sound and punch animation.

Declaration

```
public void SendButtonClick(bool playSound, bool animate, bool sendGameEvents, bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	playSound	
System.Boolean	animate	
System.Boolean	sendGameEvents	
System.Boolean	forced	

SendButtonClickAndGameEvents()

Sends the ButtonClick and the GameEvents to the UIManager without starting the OnClick animation (if enabled) and playing the button sound (if set)

Declaration

```
[Obsolete]
public void SendButtonClickAndGameEvents()
```

SendButtonDoubleClick()

Simulates this button's double click action, without playing the set on double click sound and punch animation.

Declaration

```
public void SendButtonDoubleClick()
```

SendButtonDoubleClick(Boolean, Boolean, Boolean, Boolean)

Simulates this button's double click action and plays the set on double click sound and punch animation.

Declaration

```
public void SendButtonDoubleClick(bool playSound, bool animate, bool sendGameEvents, bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	playSound	
System.Boolean	animate	
System.Boolean	sendGameEvents	
System.Boolean	forced	

SendButtonLongClick()

Simulates this button's long click action, without playing the set on long click sound and punch animation.

Declaration

```
public void SendButtonLongClick()
```

SendButtonLongClick(Boolean, Boolean, Boolean, Boolean)

Simulates this button's long click action and plays the set on long click sound and punch animation.

Declaration

```
public void SendButtonLongClick(bool playSound, bool animate, bool sendGameEvents, bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	playSound	
System.Boolean	animate	
System.Boolean	sendGameEvents	
System.Boolean	forced	

StartHighlightedStateAnimations()

This method is obsolete. Use StartSelectedLoop instead.

Declaration

```
[Obsolete]  
public void StartHighlightedStateAnimations()
```

StartNormalStateAnimations()

This method is obsolete. Use StartNormalLoop instead.

Declaration

```
[Obsolete]  
public void StartNormalStateAnimations()
```

StartOnClickAnimations()

This method is obsolete.

Declaration

```
[Obsolete]  
public void StartOnClickAnimations()
```

StopHighlightedSteateAnimations()

This method is obsolete. Use StopSelectedLoop instead.

Declaration

```
[Obsolete]  
public void StopHighlightedSteateAnimations()
```

StopNormalStateAnimations()

This method is obsolete. Use StopNormalLoop instead.

Declaration

```
[Obsolete]  
public void StopNormalStateAnimations()
```

Enum UIButton.ButtonActionType

All the action types a button can perform.

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public enum ButtonActionType
```

Fields

NAME	DESCRIPTION
OnClick	
OnDoubleClick	
OnLongClick	
OnPointerDown	
OnPointerEnter	
OnPointerExit	
OnPointerUp	

Enum UIButton.ButtonClickType

All the click types actions a button can perform.

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public enum ButtonClickType
```

Fields

NAME	DESCRIPTION
OnClick	
OnDoubleClick	
OnLongClick	

Class UIButton.ButtonName

Inheritance

System.Object
UIButton.ButtonName

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class ButtonName
```

Fields

buttonName

Declaration

```
public string buttonName
```

Field Value

TYPE	DESCRIPTION
System.String	

Class UIButton.ButtonSound

Inheritance

System.Object
UIButton.ButtonSound

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class ButtonSound
```

Fields

onClickSound

Declaration

```
public string onClickSound
```

Field Value

TYPE	DESCRIPTION
System.String	

Enum UIButton.SingleClickMode

Setting for the OnClick trigger that marks if it should be registered instantly without checking if it's a double click or not.

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public enum SingleClickMode
```

Fields

NAME	DESCRIPTION
Delayed	<p>The click will get registered after checking if it's a double click or not.</p> <p>If it's a double click, the single click will not get triggered.</p> <p>Use this if you want to make sure the user does not execute a single click before a double click.</p> <p>The downside is that there is a delay when executing the single click (the delay is the double click register interval), so make sure you take that into account</p>
Instant	<p>The click will get registered instantly without checking if it's a double click or not.</p> <p>This is the normal behaviour of a single click in any OS.</p> <p>Use this if you want to make sure a single click will get executed before a double click (dual actions).</p> <p>(usage example: SingleClick - selects, DoubleClick - executes an action)</p>

Class UICanvas

Inheritance

System.Object

UICanvas

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UICanvas : MonoBehaviour
```

Fields

canvasName

The name of this canvas.

Declaration

```
public string canvasName
```

Field Value

TYPE	DESCRIPTION
System.String	

customCanvasName

Used by the custom inspector to allow you to type a canvas name instead of selecting it from the Canvas Names Database.

Declaration

```
public bool customCanvasName
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

DEFAULT_CANVAS_NAME

Default name given to a new canvas. The name is 'MasterCanvas' and you should have ONLY ONE per scene as this is considere your main/default canvas.

Declaration

```
public const string DEFAULT_CANVAS_NAME = "MasterCanvas"
```

Field Value

TYPE	DESCRIPTION
System.String	

Properties

Canvas

Returns the Canvas component.

Declaration

```
public Canvas Canvas { get; }
```

Property Value

TYPE	DESCRIPTION
Canvas	

IsMasterCanvas

Returns true if this canvas name is 'MasterCanvas' and if it has been registered to the UIManager as the MasterCanvas

Declaration

```
public bool IsMasterCanvas { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

RectTransform

Returns the RectTransform component.

Declaration

```
public RectTransform RectTransform { get; }
```

Property Value

TYPE	DESCRIPTION
RectTransform	

Methods

RegisterToUIManager()

Registeres this UICanvas to the UIManager.

Declaration

```
public void RegisterToUIManager()
```

UnregisterFromUIManager()

Unregisteres this UICanvas from the UIManager.

Declaration

```
public void UnregisterFromUIManager()
```

Class UIEffect

Inheritance

System.Object

UIEffect

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UIEffect : MonoBehaviour
```

Fields

autoRegister

Used by the UINotification. If this effect is used by a notification, then the notification should handle it's registration process in order to use an auto generated name. Do not change this value yourself.

Declaration

```
public bool autoRegister
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

customOrderInLayer

Used by the custom inspector to set your custom order in layer. Use this only if you know what you are doing.

Declaration

```
public int customOrderInLayer
```

Field Value

TYPE	DESCRIPTION
System.Int32	

customSortingLayerName

Used by the custom inspector to set your custom layer name. Use this only if you know what you are doing.

Declaration

```
public string customSortingLayerName
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_CUSTOM_ORDER_IN_LAYER

Default order in layer value.

Declaration

```
public const int DEFAULT_CUSTOM_ORDER_IN_LAYER = 0
```

Field Value

TYPE	DESCRIPTION
System.Int32	

DEFAULT_CUSTOM_SORTING_LAYER_NAME

Default sorting layer name.

Declaration

```
public const string DEFAULT_CUSTOM_SORTING_LAYER_NAME = "Default"
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_DEFAULT_SORTING_ORDER_STEP

Default sorting order step value.

Declaration

```
public const int DEFAULT_DEFAULT_SORTING_ORDER_STEP = 1
```

Field Value

TYPE	DESCRIPTION
System.Int32	

effectPosition

Determines the order in layer by adding (if InFrontOfTarget) or subtracting (if BehindTarget) the set number of sorting order steps to the order in layer value.

Declaration

```
public UIEffect.EffectPosition effectPosition
```

Field Value

TYPE	DESCRIPTION
UIEffect.EffectPosition	

isVisible

Keeps track if this UIEffect is visible or not. Do not change this value yourself.

Declaration

```
public bool isVisible
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

playOnAwake

Should this effect start playing on awake or not. Default is set to false.

Declaration

<pre>public bool playOnAwake</pre>

Field Value

TYPE	DESCRIPTION
System.Boolean	

sortingOrderStep

Taking into account the target's [Canvas][Order in Layer][value] - we adjust the [ParticleSystem][Renderer][Order in Layer][value] with this sorting step (by adding, if set to InFrontOfTarget or subtracting, if set BehindTarget)

Declaration

<pre>public int sortingOrderStep</pre>
--

Field Value

TYPE	DESCRIPTION
System.Int32	

startDelay

Time interval to wait to play this effect, after the show command has been sent for the target UIElement.

Declaration

<pre>public float startDelay</pre>

Field Value

TYPE	DESCRIPTION
System.Single	

stopInstantly

Should the effect stop instantly and clear, after the hide command has been sent, or should it stop and let the particles disappear after their set lifetime. Default is set to false.

Declaration

<pre>public bool stopInstantly</pre>

Field Value

TYPE	DESCRIPTION
System.Boolean	

targetParticleSystem

The particle system that thie UIEffect is controlling.

Declaration

```
public ParticleSystem targetParticleSystem
```

Field Value

TYPE	DESCRIPTION
ParticleSystem	

targetUIElement

The target UIElement that controls this UIEffect.

Declaration

```
public UIElement targetUIElement
```

Field Value

TYPE	DESCRIPTION
UIElement	

useCustomOrderInLayer

Used by the custom inspector to allow you to type a order in layer instead of getting it automatically set by this UIEffect. Use this only if you know what you are doing.

Declaration

```
public bool useCustomOrderInLayer
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

useCustomSortingLayerName

Used by the custom inspector to allow you to type a layer name instead of selecting it from the layers dropdown list. Use this only if you know what you are doing.

Declaration

```
public bool useCustomSortingLayerName
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Methods

Hide(Boolean)

Hides this UIEffect (similar to the Hide method of the UIElement).

Declaration

```
public void Hide(bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	forced	

RegisterToUIManager()

Registers this UIEffect to the UIManager.

Declaration

```
public void RegisterToUIManager()
```

Show(Boolean)

Shows this UIEffect (similar to the Show method of the UIElement).

Declaration

```
public void Show(bool forced = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	forced	

UnregisterFromUIManager()

Unregisters this UIEffect from the UIManager.

Declaration

```
public void UnregisterFromUIManager()
```

UpdateSorting()

Updates the sorting of this effect to the set and calculated values.

Declaration

```
public void UpdateSorting()
```

Enum UIEffect.EffectPosition

Determines the sorting order.

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public enum EffectPosition
```

Fields

NAME	DESCRIPTION
BehindTarget	
InFrontOfTarget	

Class UIElement

Inheritance

System.Object

UIElement

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UIElement : MonoBehaviour
```

Fields

animateAtStart

Animate the UIElement at runtime at start. Initiates a Show, thus playing an In animation. Default is set to false.

Declaration

```
public bool animateAtStart
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

autoRegister

Used by the UINotification. If this element is linked to a notification, then the notification should handle it's registration process in order to use an auto generated name. Do not change this value yourself.

Declaration

```
public bool autoRegister
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

customInAnimationsSoundAtFinish

Used by the custom inspector to allow you to type a sound name instead of selecting it from the UISounds Database.

Declaration

```
public bool customInAnimationsSoundAtFinish
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

customInAnimationsSoundAtStart

Used by the custom inspector to allow you to type a sound name instead of selecting it from the UISounds Database.

Declaration

```
public bool customInAnimationsSoundAtStart
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

customOutAnimationsSoundAtFinish

Used by the custom inspector to allow you to type a sound name instead of selecting it from the UISounds Database.

Declaration

```
public bool customOutAnimationsSoundAtFinish
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

customOutAnimationsSoundAtStart

Used by the custom inspector to allow you to type a sound name instead of selecting it from the UISounds Database.

Declaration

```
public bool customOutAnimationsSoundAtStart
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

customStartAnchoredPosition

The custom anchored position that this UIElement comes from or goes to when an In or Out animation is played. You can use this in code to customize on the fly this position.

Declaration

```
public Vector3 customStartAnchoredPosition
```

Field Value

TYPE	DESCRIPTION
Vector3	

disableWhenHidden

Disables this UIElement when it is not visible (it is hidden) by setting its active state to false.

Use this only if you have scripts that you need to disable. Otherwise you don't need it as the system handles the drawcalls in an efficient manner.

Declaration

```
public bool disableWhenHidden
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

elementCategory

The category this element name belongs to. The category is important when showing or hiding an UIElement as it is taken into account.

Declaration

```
public string elementCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

elementName

The name of this element. The name is important when showing or hiding an UIElement as it is taken into account.

Declaration

```
public string elementName
```

Field Value

TYPE	DESCRIPTION
System.String	

executeLayoutFix

This fixes a very strange issue inside Unity. When setting a VerticalLayoutGroup or a HorizontalLayoutGroup, the Image bounds get moved (the image appears in a different place).

If you have this issue, just set this to true. Default is set to false.

If you are curious about what this does, look at the ExecuteLayoutFix method.

Declaration

```
public bool executeLayoutFix
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

fadeOut

Declaration

[Obsolete]
public UIAnimator.FadeIn fadeIn

Field Value

TYPE	DESCRIPTION
UIAnimator.FadeIn	

fadeOut

Declaration

[Obsolete]
public UIAnimator.FadeLoop fadeLoop

Field Value

TYPE	DESCRIPTION
UIAnimator.FadeLoop	

fadeOut

Declaration

[Obsolete]
public UIAnimator.FadeOut fadeOut

Field Value

TYPE	DESCRIPTION
UIAnimator.FadeOut	

inAnimations

In Animation Settings

Declaration

public Anim inAnimations

Field Value

TYPE	DESCRIPTION
Anim	

inAnimationsPresetCategoryName

Out Animations Preset Category Name

Declaration

public string inAnimationsPresetCategoryName

Field Value

TYPE	DESCRIPTION
System.String	

inAnimationsPresetName

Out Animations Preset Name

Declaration

```
public string inAnimationsPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

inAnimationsSoundAtFinish

The sound name of the sound that gets played when the in animations finished.

Declaration

```
public string inAnimationsSoundAtFinish
```

Field Value

TYPE	DESCRIPTION
System.String	

inAnimationsSoundAtStart

The sound name of the sound that gets played when the in animations start.

Declaration

```
public string inAnimationsSoundAtStart
```

Field Value

TYPE	DESCRIPTION
System.String	

isVisible

Keeps track if this UIElement is visible or not. Do not change this value yourself.

Declaration

```
public bool isVisible
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

LANDSCAPE

Use this UIElement for LANDSCAPE orientation. Default is true.

If Orientation Manager is disabled, this setting does nothing.

Declaration

```
public bool LANDSCAPE
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

linkedToNotification

If this UIElement is linked to an UINotification then it will have an auto-generated element name. Do not change this value yourself.

Declaration

```
public bool linkedToNotification
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

loadInAnimationsPresetAtRuntime

Should the system load, at runtime, the Animation Preset with the set Preset Category and Preset Name. This overrides any values set in the inspector.

Declaration

```
public bool loadInAnimationsPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

loadLoopAnimationsPresetAtRuntime

Should the system load, at runtime, the Loop Preset with the set Preset Category and Preset Name. This overrides any values set in the inspector.

Declaration

```
public bool loadLoopAnimationsPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

loadOutAnimationsPresetAtRuntime

Should the system load, at runtime, the Animation Preset with the set Preset Category and Preset Name. This overrides any values set in the inspector.

Declaration

```
public bool loadOutAnimationsPresetAtRuntime
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

LOOP_ANIMATIONS_ID

This is an extra id tag given to the tweener in order to locate the proper tween that manages the loop animations.

Declaration

```
public const string LOOP_ANIMATIONS_ID = "UIElementLoopAnimations"
```

Field Value

TYPE	DESCRIPTION
System.String	

loopAnimations

Loop Animation Settings

Declaration

```
public Loop loopAnimations
```

Field Value

TYPE	DESCRIPTION
Loop	

loopAnimationsPresetCategoryName

Loop Animations Preset Category Name

Declaration

```
public string loopAnimationsPresetCategoryName
```

Field Value

TYPE	DESCRIPTION
System.String	

loopAnimationsPresetName

Loop Animations Preset Name

Declaration

```
public string loopAnimationsPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

moveIn

Declaration

```
[Obsolete]  
public UIAnimator.MoveIn moveIn
```

Field Value

TYPE	DESCRIPTION
UIAnimator.MoveIn	

moveLoop

Declaration

```
[Obsolete]  
public UIAnimator.MoveLoop moveLoop
```

Field Value

TYPE	DESCRIPTION
UIAnimator.MoveLoop	

moveOut

Declaration

```
[Obsolete]  
public UIAnimator.MoveOut moveOut
```

Field Value

TYPE	DESCRIPTION
UIAnimator.MoveOut	

OnInAnimationsFinish

UnityEvent invoked when In animations finished.

Declaration

```
public UnityEvent OnInAnimationsFinish
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

OnInAnimationsStart

UnityEvent invoked when In animations start.

Declaration

```
public UnityEvent OnInAnimationsStart
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

OnOutAnimationsFinish

UnityEvent invoked when Out animations finished.

Declaration

```
public UnityEvent OnOutAnimationsFinish
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

OnOutAnimationsStart

UnityEvent invoked when Out animations start.

Declaration

```
public UnityEvent OnOutAnimationsStart
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

outAnimations

Out Animation Settings

Declaration

```
public Anim outAnimations
```

Field Value

TYPE	DESCRIPTION
Anim	

outAnimationsPresetCategoryName

Out Animations Preset Category Name

Declaration

```
public string outAnimationsPresetCategoryName
```

Field Value

TYPE	DESCRIPTION
System.String	

outAnimationsPresetName

Out Animations Preset Name

Declaration

```
public string outAnimationsPresetName
```

Field Value

TYPE	DESCRIPTION
System.String	

outAnimationsSoundAtFinish

The sound name of the sound that gets played when the out animations finished.

Declaration

```
public string outAnimationsSoundAtFinish
```

Field Value

TYPE	DESCRIPTION
System.String	

outAnimationsSoundAtStart

The sound name of the sound that gets played when the out animations start.

Declaration

```
public string outAnimationsSoundAtStart
```

Field Value

TYPE	DESCRIPTION
System.String	

PORTRAIT

Use this UIElement for PORTRAIT orientation. Default is true.

If Orientation Manager is disabled, this setting does nothing.

Declaration

```
public bool PORTRAIT
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

rotationIn

Declaration

```
[Obsolete]
public UIAnimator.RotationIn rotationIn
```

Field Value

TYPE	DESCRIPTION
UIAnimator.RotationIn	

rotationLoop

Declaration

```
[Obsolete]
public UIAnimator.RotationLoop rotationLoop
```

Field Value

TYPE	DESCRIPTION
UIAnimator.RotationLoop	

rotationOut

Declaration

```
[Obsolete]
public UIAnimator.RotationOut rotationOut
```

Field Value

TYPE	DESCRIPTION
UIAnimator.RotationOut	

scaleIn

Declaration

```
[Obsolete]
public UIAnimator.ScaleIn scaleIn
```

Field Value

TYPE	DESCRIPTION
UIAnimator.ScaleIn	

scaleLoop

Declaration

```
[Obsolete]
public UIAnimator.ScaleLoop scaleLoop
```

Field Value

TYPE	DESCRIPTION
UIAnimator.ScaleLoop	

scaleOut

Declaration

```
[Obsolete]
public UIAnimator.ScaleOut scaleOut
```

Field Value

TYPE	DESCRIPTION
UIAnimator.ScaleOut	

selectedButton

The button that gets selected when this UIElement gets shown; if null then no button will get auto selected. Default is set to null.

Declaration

```
public GameObject selectedButton
```

Field Value

TYPE	DESCRIPTION
GameObject	

startHidden

Hide the UIElement at runtime at start. Initiates an instant Hide. Default is set to false.

Declaration

```
public bool startHidden
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

useCustomStartAnchoredPosition

Should this UIElement come from or go to a set custom position every time an In or Out animation is played? Default is set to false.

Declaration

```
public bool useCustomStartAnchoredPosition
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Properties

Canvas

Returns the Canvas component.

Declaration

```
public Canvas Canvas { get; }
```

Property Value

TYPE	DESCRIPTION
Canvas	

CanvasGroup

Returns the CanvasGroup component.

Declaration

```
public CanvasGroup CanvasGroup { get; }
```

Property Value

TYPE	DESCRIPTION
CanvasGroup	

GraphicRaycaster

Returns the GraphicRaycaster component.

Declaration

```
public GraphicRaycaster GraphicRaycaster { get; }
```

Property Value

TYPE	DESCRIPTION
GraphicRaycaster	

InAnimationsEnabled

Retruns true if at least one In animation is enabled. This means that if either move or rotate or scale or fade are enabled it will return true and false otherwise.

Declaration

```
public bool InAnimationsEnabled { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

InitialData

Declaration

```
[Obsolete]
public UIAnimator.InitialData InitialData { get; }
```

Property Value

TYPE	DESCRIPTION
UIAnimator.InitialData	

LoopAnimationsEnabled

Retruns true if at least one Loop animation is enabled. This means that if either move or rotate or scale or fade are enabled it will return true and false otherwise.

Declaration

```
public bool LoopAnimationsEnabled { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

OutAnimationsEnabled

Retruns true if at least one Out animation is enabled. This means that if either move or rotate or scale or fade are enabled it will return true and false otherwise.

Declaration

```
public bool OutAnimationsEnabled { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

RectTransform

Returns the RectTransform component.

Declaration

```
public RectTransform RectTransform { get; }
```

Property Value

TYPE	DESCRIPTION

TYPE	DESCRIPTION
RectTransform	

Methods

Hide(Boolean)

Hides the element.

Declaration

```
public void Hide(bool instantAction)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	instantAction	If set to <code>true</code> it will execute the animations in 0 seconds and with 0 delay

Hide(Boolean, Boolean)

Hides the element.

Declaration

```
public void Hide(bool instantAction, bool shouldDisable)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	instantAction	If set to <code>true</code> it will execute the animations in 0 seconds and with 0 delay
System.Boolean	shouldDisable	

RegisterToUIManager()

Registers this UIElement to the UIManager.

Declaration

```
public void RegisterToUIManager()
```

Reset()

Declaration

```
public void Reset()
```

Show(Boolean)

Shows the element.

Declaration

```
public void Show(bool instantAction)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	instantAction	If set to <code>true</code> it will execute the animations in 0 seconds and with 0 delay

UnregisterFromUIManager()

Unregisters this UIElement from the UIManager.

Declaration

```
public void UnregisterFromUIManager()
```


Class UIElement.ElementName

Inheritance

System.Object
UIElement.ElementName

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
[Obsolete]  
public class ElementName
```

Fields

elementName

Declaration

```
public string elementName
```

Field Value

TYPE	DESCRIPTION
System.String	

Class UIManager

Inheritance

System.Object
QuickEngine.Common.Singleton<DoozyUI.UIManager>
.UIManager

Inherited Members

[Singleton<UIManager>.Instance](#)
[Singleton<UIManager>.OnDestroy\(\)](#)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class UIManager : Singleton<UIManager>
```

Constructors

UIManager()

Declaration

```
protected UIManager()
```

Fields

autoDisableButtonClicks

Should the system disable button clicks when an UIElement is in transition (an In or Out animation is running). Default is true.

Declaration

```
public bool autoDisableButtonClicks
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

currentGameTimeScale

Every time the user pauses the game, this variable stores the current Time.timeScale value. This is needed so that when the game needs to get unpaused, UIManager will know at what timescale should the game return to.

Declaration

```
public static float currentGameTimeScale
```

Field Value

TYPE	DESCRIPTION
System.Single	

currentOrientation

Returns the current orientation of the device. Default is Orientation.Unknown because that triggers an orientation check/update.

Declaration

```
public static UIManager.Orientation currentOrientation
```

Field Value

TYPE	DESCRIPTION
UIManager.Orientation	

debugGameEvents

Prints debug messages related to game events at runtime.

Declaration

```
public bool debugGameEvents
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

debugUIButtons

Prints debug messages related to UIButtons at runtime.

Declaration

```
public bool debugUIButtons
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

debugUICanvases

Prints debug messages related to UICanvases at runtime.

Declaration

```
public bool debugUICanvases
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

debugUIElements

Prints debug messages related to UIElements at runtime.

Declaration

```
public bool debugUIElements
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

debugUINotifications

Prints debug messages related to UINotifications at runtime.

Declaration

```
public bool debugUINotifications
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

gamePaused

Returns true if the game has been paused (by the UIManager) and false otherwise.

Declaration

```
public static bool gamePaused
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

isMusicOn

Returns true if the music is on and false otherwise. This variable knows only if the music is on for DoozyUI and not for anything else as it checks a PlayerPrefs int value named 'musicState'.

Declaration

```
public static bool isMusicOn
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

isSoundOn

Returns true if the sound is on and false otherwise. This variable knows only if the sound is on for DoozyUI and not for anything else as it checks a PlayerPrefs int value named 'soundState'.

Declaration

```
public static bool isSoundOn
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

sceneLoader

Reference to the SceneLoader. The SceneLoader registeres itself on OnEnable.

Declaration

```
public static SceneLoader sceneLoader
```

Field Value

TYPE	DESCRIPTION
SceneLoader	

uiCamera

Declaration

```
[Obsolete]
public Camera uiCamera
```

Field Value

TYPE	DESCRIPTION
Camera	

useOrientationManager

Determines if the Orientation Manager should be used. This value is automatically set to true when the 'dUI_UseOrientationManager' Scripting Define Symbol has been added to the current active platform.

Declaration

```
public static bool useOrientationManager
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

usesTMPPro

Global static variable that determines if the UINotification look for TextMeshProUGUI component instead of a Text componenet when looking for text.

TextMeshPro support is currently in limbo as we wait to see what Unity does with it.

Declaration

```
public static bool usesTMPPro
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

useTextMeshPro

TextMeshPro support is currently in limbo as we wait to see what Unity does with it.

Declaration

```
public bool useTextMeshPro
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Properties

BackButtonDisabled

Returns true if the 'Back' button is disabled and false otherwise.

Declaration

```
public static bool BackButtonDisabled { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

ButtonClicksDisabled

Returns true if button clicks are disabled and false otherwise. This is mostly used when an UIElement is in transition and, in order to prevent accidental clicks, the buttons need to be disabled.

Declaration

```
public static bool ButtonClicksDisabled { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

ButtonClicksTriggerDatabase

Returns a registry of all the registered UITriggers that listens for button clicks.

Declaration

```
public static Dictionary<string, List<UITrigger>> ButtonClicksTriggerDatabase { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Collections.Generic.List<DoozyUI.UITrigger> >	

ButtonDoubleClicksTriggerDatabase

Returns a registry of all the registered UITriggers that listens for button clicks.

Declaration

```
public static Dictionary<string, List<UITrigger>> ButtonDoubleClicksTriggerDatabase { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Collections.Generic.List<DoozyUI.UITrigger> >	

ButtonLongClicksTriggerRegistry

Returns a registry of all the registered UITriggers that listens for button clicks.

Declaration

```
public static Dictionary<string, List<UITrigger>> ButtonLongClicksTriggerRegistry { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Collections.Generic.List<DoozyUI.UITrigger> >	

CanvasDatabase

Returns a registry of all the registered UICanvases.

Declaration

```
public static Dictionary<string, UICanvas> CanvasDatabase { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, DoozyUI.UICanvas>	

EffectDatabase

Returns a registry of all the registered UIEffects.

Declaration

```
public static Dictionary<string, List<UIEffect>> EffectDatabase { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Collections.Generic.List<DoozyUI.UIEffect> >	

ElementDatabase

Returns a registry of all the registered UIElements.

Declaration

```
public static Dictionary<string, List<UIElement>> ElementDatabase { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Collections.Generic.List<DoozyUI.UIElement>>	

GameEventsTriggerDatabase

Returns a registry of all the registered UITriggers that listens for game events.

Declaration

```
public static Dictionary<string, List<UITrigger>> GameEventsTriggerDatabase { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Collections.Generic.List<DoozyUI.UITrigger>>	

GetUICamera

Returns the main camera.

Declaration

```
[Obsolete]  
public static Camera GetUICamera { get; }
```

Property Value

TYPE	DESCRIPTION
Camera	

GetUiContainer

Obsolete method. Use GetMasterCanvas() or GetCanvas(canvasName) instead

Declaration

```
[Obsolete]  
public static Transform GetUiContainer { get; }
```

Property Value

TYPE	DESCRIPTION
Transform	

IsNavigationEnabled

Returns true if the UI Navigation is enabled and false otherwise. It is set to false if Scripting Define Symbols, for the current active

platform, contain the 'dUI_NavigationDisabled' symbol.

In you want to handle the UI Navigation yourself just disable the UI Navigation from the Control Panel.

Declaration

```
public static bool IsNavigationEnabled { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

NotificationManager

Returns the UINotificationManager component.

Declaration

```
public static UINotificationManager NotificationManager { get; }
```

Property Value

TYPE	DESCRIPTION
UINotificationManager	

OrientationManager

Returns the OrientationManager reference.

Declaration

```
public OrientationManager OrientationManager { get; }
```

Property Value

TYPE	DESCRIPTION
OrientationManager	

Soundy

Returns the Soundy component.

Declaration

```
public static Soundy Soundy { get; }
```

Property Value

TYPE	DESCRIPTION
Soundy	

UICamera

Declaration

```
[Obsolete]  
public Camera UICamera { get; }
```

Property Value

TYPE	DESCRIPTION
Camera	

Methods

ApplicationQuit()

Exits play mode (if in editor) or quits the application if in build mode

Declaration

```
public static void ApplicationQuit()
```

BackButtonEvent()

The 'back' button was pressed (or escape key)

Declaration

```
public static void BackButtonEvent()
```

ChangeOrientation(UIManager.Orientation)

Updates the current orientation to the new given one.

Declaration

```
public void ChangeOrientation(UIManager.Orientation newOrientation)
```

Parameters

TYPE	NAME	DESCRIPTION
UIManager.Orientation	newOrientation	

CreateCanvas(String)

Creates an UICanvas with the given canvas name and returns the reference to it.

Declaration

```
public static UICanvas CreateCanvas(string canvasName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	canvasName	The canvas name for the new UICanvas.

Returns

TYPE	DESCRIPTION
UICanvas	

DisableBackButton()

Disables the 'Back' button functionality

Declaration

```
public static void DisableBackButton()
```

DisableButtonClicks()

Disables all the button clicks. This is triggered by the system when an UIElement started a transition (IN/OUT animations).

Declaration

```
public static void DisableButtonClicks()
```

EnableBackButton()

Enables the 'Back' button functionality

Declaration

```
public static void EnableBackButton()
```

EnableBackButtonByForce()

Enables the 'Back' button functionality by resetting the additive bool to zero. backButtonDisableLevel = 0. Use this ONLY for special cases when something wrong happens and the back button is stuck in disabled mode.

Declaration

```
public static void EnableBackButtonByForce()
```

EnableButtonClicks()

Enables all the button clicks. This is triggered by the system when an UIElement finished a transition (IN/OUT animations).

Declaration

```
public static void EnableButtonClicks()
```

EnableButtonClicksByForce()

Enables the button clicks by resetting the additive bool to zero. buttonClicksDisableLevel = 0. Use this ONLY for special cases when something unexpected happens and the button clicks are stuck in disabled mode.

Declaration

```
public static void EnableButtonClicksByForce()
```

GetCanvas(String, Boolean, Boolean)

Retruns a reference to an UICanvas that has the given canvas name. It can also create the canvas you are searching for or just return the 'MasterCanvas' UICanvas.

Declaration

```
public static UICanvas GetCanvas(string canvasName, bool createCanvasIfNotFound = false, bool  
returnMasterCanvasIfTargetCanvasNotFound = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	canvasName	The canvas name you are looking for.
System.Boolean	createCanvasIfNotFound	Should the system create an UICanvas with the canvas name you are looking for?
System.Boolean	returnMasterCanvasIfTargetCanvasNotFound	Should this method return a reference to the 'MasterCanvas' UICanvas if the canvas name you are looking for was not found?

Returns

TYPE	DESCRIPTION
UICanvas	

GetMasterCanvas(Boolean)

Returns a reference to an UICanvas that is considered and used as a 'MasterCanvas'. If no such canvas exists, one will get created automatically by default.

Declaration

```
public static UICanvas GetMasterCanvas(bool createMasterCanvasIfNotFound = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	createMasterCanvasIfNotFound	Should a 'MasterCanvas' be created if it is missing.

Returns

TYPE	DESCRIPTION
UICanvas	

GetUiEffects(String, String)

Returns a List of all UIEffects that are linked to an UIElement with a given name and category. If no UIEffect was found, it will return an empty list.

Declaration

```
public static List<UIEffect> GetUiEffects(string elementName, string elementCategory = "Uncategorized")
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	elementName	

TYPE	NAME	DESCRIPTION
System.String	elementCategory	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<DoozyUI.UIEffect>	

GetUiElements(String, String)

Returns a List of all UIElements that have a given name and category. If no UIElement was found, it will return an empty list.

Declaration

```
public static List<UIElement> GetUiElements(string elementName, string elementCategory = "Uncategorized")
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	elementName	
System.String	elementCategory	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<DoozyUI.UIElement>	

GetUiTriggers(String, DUI.EventType)

Returns a list of all the UITriggers that are linked to the given triggerValue and of the given triggerType.

Declaration

```
public static List<UITrigger> GetUiTriggers(string triggerValue, DUI.EventType triggerType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	triggerValue	This can be either a game event or a button name or the special DUI.DISPATCH_ALL value.
DUI.EventType	triggerType	Depending on the triggerType, this method will search in a different registry.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<DoozyUI.UITrigger>	

GetVisibleUIElements()

Returns a list of all the UIElements that are visible on the screen. An UIElement is considered visible if isVisible = true.

Returns a List of all the elements that are visible on the screen. An element is considered visible if visible == true.

If eDatabase is null or empty or if no UIElements are visible, it will return an empty list.

Declaration

```
public static List<UIElement> GetVisibleUIElements()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<DoozyUI.UIElement>	

HideUiElement(String, Boolean)

Hides all the UIElements that have the given name and the DEFAULT CATEGORY name.

Declaration

```
public static void HideUiElement(string elementName, bool instantAction = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	elementName	
System.Boolean	instantAction	Should the animation play instantly (in zero seconds)

HideUiElement(String, String)

Hides all the UIElements that have the given name and category.

Declaration

```
public static void HideUiElement(string elementName, string elementCategory)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	elementName	
System.String	elementCategory	

HideUiElement(String, String, Boolean)

Hides all the UIElements that have the given name and category.

Declaration

```
public static void HideUiElement(string elementName, string elementCategory, bool instantAction)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	elementName	

TYPE	NAME	DESCRIPTION
System.String	elementCategory	
System.Boolean	instantAction	Should the animation play instantly (in zero seconds)

MusicCheck()

Checks the musicState when the game starts in the PlayerPrefs

Declaration

```
public static void MusicCheck()
```

PlaySound(AudioClip)

Plays the given audio clip, through Soundy. You can also use Soundy.PlaySound...

Declaration

```
public static void PlaySound(AudioClip aClip)
```

Parameters

TYPE	NAME	DESCRIPTION
AudioClip	aClip	

PlaySound(AudioClip, Single)

Plays the given audio clip at the given volume level, through Soundy. You can also use Soundy.PlaySound...

Declaration

```
public static void PlaySound(AudioClip aClip, float volume)
```

Parameters

TYPE	NAME	DESCRIPTION
AudioClip	aClip	
System.Single	volume	

PlaySound(AudioClip, Single, Single)

Plays the given audio clip at the given volume and pitch levels, through Soundy. You can also use Soundy.PlaySound...

Declaration

```
public static void PlaySound(AudioClip aClip, float volume, float pitch)
```

Parameters

TYPE	NAME	DESCRIPTION
AudioClip	aClip	

TYPE	NAME	DESCRIPTION
System.Single	volume	
System.Single	pitch	

PlaySound(String)

Plays the given sound name, through Soundy. You can also use Soundy.PlaySound...

Note: If support for MasterAudio is enabled it will play the sound name from the MasterAudio sounds database.

Declaration

```
public static void PlaySound(string soundName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	

PlaySound(String, Single)

Plays the given sound name, through Soundy. You can also use Soundy.PlaySound...

Note: If support for MasterAudio is enabled it will play the sound name from the MasterAudio sounds database.

Declaration

```
public static void PlaySound(string soundName, float volume)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	
System.Single	volume	

PlaySound(String, Single, Single)

Plays the given sound name, through Soundy. You can also use Soundy.PlaySound...

Note: If support for MasterAudio is enabled it will play the sound name from the MasterAudio sounds database.

Declaration

```
public static void PlaySound(string soundName, float volume, float pitch)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	
System.Single	volume	
System.Single	pitch	

PlaySoundFromResources(String)

Plays the given sound name by searching the Resources folder for it, through Soundy. You can also use Soundy.PlaySound...

Declaration

```
public static void PlaySoundFromResources(string soundName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	

PlaySoundFromResources(String, Single)

Plays the given sound name by searching the Resources folder for it, through Soundy. You can also use Soundy.PlaySound...

Declaration

```
public static void PlaySoundFromResources(string soundName, float volume)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	
System.Single	volume	

PlaySoundFromResources(String, Single, Single)

Plays the given sound name by searching the Resources folder for it, through Soundy. You can also use Soundy.PlaySound...

Declaration

```
public static void PlaySoundFromResources(string soundName, float volume, float pitch)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	soundName	
System.Single	volume	
System.Single	pitch	

RegisterToNotificationQueue(UINotification.NotificationData)

Every notification that needs to enter the Notification Queue will be added to the notificatioQueue list as the last item.

Declaration

```
public void RegisterToNotificationQueue(UINotification.NotificationData nData)
```

Parameters

TYPE	NAME	DESCRIPTION
UINotification.NotificationData	nData	

SendButtonAction(UIButton, UIButton.ButtonActionType)

Sends a button action with a reference to the UIButton that sent it and what type of action it is.

Declaration

```
public void SendButtonAction(UIButton uiButton, UIButton.ButtonActionType actionType)
```

Parameters

TYPE	NAME	DESCRIPTION
UIButton	uiButton	
UIButton.ButtonActionType	actionType	

SendButtonAction(String, UIButton.ButtonActionType)

Sends a button action with just a button name and what type of action it is. This method is used to simulate a button action since it does not have an UIButton reference.

Declaration

```
public void SendButtonAction(string buttonName, UIButton.ButtonActionType actionType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	buttonName	
UIButton.ButtonActionType	actionType	

SendButtonAction(String, UIButton.ButtonClickType)

Sends a button action with just a button name and what type of click it is. This method is used to simulate a button action since it does not have an UIButton reference.

Declaration

```
public void SendButtonAction(string buttonName, UIButton.ButtonClickType clickType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	buttonName	
UIButton.ButtonClickType	clickType	

SendButtonClick(String, Boolean, List<String>, List<String>, List<String>)

Use SendButtonAction instead.

Declaration

[Obsolete]

```
public static void SendButtonClick(string buttonName, bool addToNavigationHistory = false, List<string> showElements = null, List<string> hideElements = null, List<string> gameEvents = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	buttonName	
System.Boolean	addToNavigationHistory	
System.Collections.Generic.List<System.String>	showElements	
System.Collections.Generic.List<System.String>	hideElements	
System.Collections.Generic.List<System.String>	gameEvents	

SendGameEvent(String)

Sends the given game event.

Declaration

```
public static void SendGameEvent(string gameEvent)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	gameEvent	

SendGameEvents(List<String>)

Sends the given list of game events.

Declaration

```
public static void SendGameEvents(List<string> gameEvents)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.String>	gameEvents	

ShowNotification(GameObject, Single, Boolean, Sprite, UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public static UINotification ShowNotification(GameObject _prefab, float _lifetime, bool _addToNotificationQueue, Sprite _icon, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
Sprite	_icon	The sprite you want the notification icon to have (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String, Sprite, UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public static UINotification ShowNotification(GameObject _prefab, float _lifetime, bool _addToNotificationQueue, string _title, Sprite _icon, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String, String, Sprite, String[], String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public static UINotification ShowNotification(GameObject _prefab, float _lifetime, bool
_addToNotificationQueue, string _title, string _message, Sprite _icon, string[] _buttonNames, string[]
_buttonTexts, UnityAction[] _buttonCallback = null, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String	_message	The message you want to show in the message area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
System.String[]	_buttonTexts	The text on the buttons (example: 'OK', 'Cancel', 'Yes', 'No' and so on)
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String, String, Sprite, String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public static UINotification ShowNotification(GameObject _prefab, float _lifetime, bool
_addToNotificationQueue, string _title, string _message, Sprite _icon, string[] _buttonNames, UnityAction[]
_buttonCallback = null, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String	_message	The message you want to show in the message area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String, String, Sprite, UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public static UINotification ShowNotification(GameObject _prefab, float _lifetime, bool
_addToNotificationQueue, string _title, string _message, Sprite _icon, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String	_message	The message you want to show in the message area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UnityEngine.UI.Notification	

ShowNotification(GameObject, Single, Boolean, String, String, UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public static UnityEngine.UI.Notification ShowNotification(GameObject _prefab, float _lifetime, bool
_addToNotificationQueue, string _title, string _message, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)

TYPE	NAME	DESCRIPTION
System.String	_message	The message you want to show in the message area (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String, String[], String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public static UINotification ShowNotification(GameObject _prefab, float _lifetime, bool
_addToNotificationQueue, string _title, string[] _buttonNames, string[] _buttonTexts, UnityAction[]
_buttonCallback = null, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
System.String[]	_buttonTexts	The text on the buttons (example: 'OK', 'Cancel', 'Yes', 'No' and so on)
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String, String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public static UINotification ShowNotification(GameObject _prefab, float _lifetime, bool
_addToNotificationQueue, string _title, string[] _buttonNames, UnityAction[] _buttonCallback = null,
UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String, UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public static UINotification ShowNotification(GameObject _prefab, float _lifetime, bool
_addToNotificationQueue, string _title, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String[], String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```

public static UINotification ShowNotification(GameObject _prefab, float _lifetime, bool
_addToNotificationQueue, string[] _buttonNames, string[] _buttonTexts, UnityAction[] _buttonCallback = null,
UnityAction _hideCallback = null)

```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
System.String[]	_buttonTexts	The text on the buttons (example: 'OK', 'Cancel', 'Yes', 'No' and so on)
UnityAction[]	_buttonCallback	

TYPE	NAME	DESCRIPTION
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public static UINotification ShowNotification(GameObject _prefab, float _lifetime, bool
_addToNotificationQueue, string[] _buttonNames, UnityAction[] _buttonCallback = null, UnityAction
_hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public static UINotification ShowNotification(GameObject _prefab, float _lifetime, bool
_addToNotificationQueue, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, Sprite, UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public static UINotification ShowNotification(string _prefabName, float _lifetime, bool
_addToNotificationQueue, Sprite _icon, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
Sprite	_icon	The sprite you want the notification icon to have (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String, Sprite, UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public static UINotification ShowNotification(string _prefabName, float _lifetime, bool
_addToNotificationQueue, string _title, Sprite _icon, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String, String, Sprite, String[], String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public static UINotification ShowNotification(string _prefabName, float _lifetime, bool
_addToNotificationQueue, string _title, string _message, Sprite _icon, string[] _buttonNames, string[]
_buttonTexts, UnityAction[] _buttonCallback = null, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway

TYPE	NAME	DESCRIPTION
System.String	_title	The text you want to show in the title area (if linked)
System.String	_message	The message you want to show in the message area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
System.String[]	_buttonTexts	The text on the buttons (example: 'OK', 'Cancel', 'Yes', 'No' and so on)
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String, String, Sprite, String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public static UINotification ShowNotification(string _prefabName, float _lifetime, bool
_addToNotificationQueue, string _title, string _message, Sprite _icon, string[] _buttonNames, UnityAction[]
_buttonCallback = null, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)

TYPE	NAME	DESCRIPTION
System.String	_message	The message you want to show in the message area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
UnityEngine.UI.Action[]	_buttonCallback	
UnityEngine.UI.Action	_hideCallback	

Returns

TYPE	DESCRIPTION
UnityEngine.UI.Notification	

ShowNotification(String, Single, Boolean, String, String, Sprite, UnityEngine.UI.Action)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public static UnityEngine.UI.Notification ShowNotification(string _prefabName, float _lifetime, bool
_addToNotificationQueue, string _title, string _message, Sprite _icon, UnityEngine.UI.Action _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String	_message	The message you want to show in the message area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
UnityEngine.UI.Action	_hideCallback	

TYPE	NAME	DESCRIPTION

Returns

TYPE	DESCRIPTION
UnityEngine.UI.Notification	

ShowNotification(String, Single, Boolean, String, String, UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public static UnityEngine.UI.Notification ShowNotification(string _prefabName, float _lifetime, bool
_addToNotificationQueue, string _title, string _message, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String	_message	The message you want to show in the message area (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UnityEngine.UI.Notification	

ShowNotification(String, Single, Boolean, String, String[], String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public static UnityEngine.UI.Notification ShowNotification(string _prefabName, float _lifetime, bool
_addToNotificationQueue, string _title, string[] _buttonNames, string[] _buttonTexts, UnityAction[]
_buttonCallback = null, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
System.String[]	_buttonTexts	The text on the buttons (example: 'OK', 'Cancel', 'Yes', 'No' and so on)
UnityEngine.UI.Action[]	_buttonCallback	
UnityEngine.UI.Action	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String, String[], UnityEngine.UI.Action[], UnityEngine.UI.Action)

Show a premade notification with the given settings, using a prefabName.

Declaration

```

public static UINotification ShowNotification(string _prefabName, float _lifetime, bool
_addToNotificationQueue, string _title, string[] _buttonNames, UnityEngine.UI.Action[] _buttonCallback = null,
UnityEngine.UI.Action _hideCallback = null)

```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway

TYPE	NAME	DESCRIPTION
System.String	_title	The text you want to show in the title area (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String, UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public static UINotification ShowNotification(string _prefabName, float _lifetime, bool
_addToNotificationQueue, string _title, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String[], String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public static UINotification ShowNotification(string _prefabName, float _lifetime, bool
_addToNotificationQueue, string[] _buttonNames, string[] _buttonTexts, UnityAction[] _buttonCallback = null,
UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
System.String[]	_buttonTexts	The text on the buttons (example: 'OK', 'Cancel', 'Yes', 'No' and so on)
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public static UINotification ShowNotification(string _prefabName, float _lifetime, bool
_addToNotificationQueue, string[] _buttonNames, UnityAction[] _buttonCallback = null, UnityAction
_hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1

TYPE	NAME	DESCRIPTION
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public static UINotification ShowNotification(string _prefabName, float _lifetime, bool
_addToNotificationQueue, UnityAction[] _buttonCallback = null, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowUiElement(String, Boolean)

Shows all the UIElements that have the given name and the DEFAULT CATEGORY name.

Declaration

```
public static void ShowUiElement(string elementName, bool instantAction)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	elementName	
System.Boolean	instantAction	Should the animation play instantly (in zero seconds)

ShowUiElement(String, String)

Shows all the UIElements that have the given name and category.

Declaration

```
public static void ShowUiElement(string elementName, string elementCategory = "Uncategorized")
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	elementName	
System.String	elementCategory	

ShowUiElement(String, String, Boolean)

Shows all the UIElements that have the given name and category.

Declaration

```
public static void ShowUiElement(string elementName, string elementCategory, bool instantAction)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	elementName	
System.String	elementCategory	
System.Boolean	instantAction	Should the animation play instantly (in zero seconds)

SoundCheck()

Checks the soundState when the game starts in the PlayerPrefs

Declaration

```
public static void SoundCheck()
```

ToggleMusic()

Toggles the musicState and saves it to the PlayerPrefs

Declaration

```
public static void ToggleMusic()
```

TogglePause()

Pauses or Unpauses the application

Declaration

```
public static void TogglePause()
```

ToggleSound()

Toggles the soundState and saves it to the PlayerPrefs

Declaration

```
public static void ToggleSound()
```

TriggerTheTriggers(String, DUI.EventType)

Triggers all the UITriggers that are listening for the given triggerValue and are of the given triggerType.

Declaration

```
public static void TriggerTheTriggers(string triggerValue, DUI.EventType triggerType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	triggerValue	
DUI.EventType	triggerType	

UnregisterFromNotificationQueue(UINotification.NotificationData)

Unregisters a notification, by removing the notification data that started it.

Declaration

```
public void UnregisterFromNotificationQueue(UINotification.NotificationData nData)
```

Parameters

TYPE	NAME	DESCRIPTION
UINotification.NotificationData	nData	

UpdateCanvasSortingLayerName(GameObject, String)

Updates the sorting layer for all the canvases on and under the target gameObject

Declaration

```
public static void UpdateCanvasSortingLayerName(GameObject targetObject, string sortingLayerName)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	targetObject	

TYPE	NAME	DESCRIPTION
System.String	sortingLayerName	

UpdateRendererSortingLayerName(GameObject, String)

Updates all the sorting layer for all the renderers on and under the target gameObject

Declaration

```
public static void UpdateRendererSortingLayerName(GameObject targetObject, string sortingLayerName)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	targetObject	
System.String	sortingLayerName	

Enum UIManager.Orientation

Types of orientation used by DoozyUI. Unknown is used for initialization purposes.

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public enum Orientation
```

Fields

NAME	DESCRIPTION
Landscape	
Portrait	
Unknown	

Class UINavigation

Inheritance

System.Object
UINavigation

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class UINavigation
```

Fields

m_isNavigationEnabled

Internal variable that determines if the UI Navigation is enabled or not. Default is set to true.

Declaration

```
public static bool m_isNavigationEnabled
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Properties

IsNavigationEnabled

Returns true if the UI Navigation is enabled and false otherwise. It is set to false if Scripting Define Symbols, for the current active platform, contain the 'dUI_NavigationDisabled' symbol.

In you want to handle the UI Navigation yourself just disable the UI Navigation from the Control Panel.

Declaration

```
public static bool IsNavigationEnabled { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

AddItemToHistory(NavigationPointerData)

Adds a navigation item to the end of Navigation History (FILO - First In Last Out).

Declaration

```
public static void AddItemToHistory(NavigationPointerData data)
```

Parameters

TYPE	NAME	DESCRIPTION
NavigationPointerData	data	

ClearNavigationHistory()

Clears the Navigation History.

Declaration

```
public static void ClearNavigationHistory()
```

GetLastItemFromNavigationHistory(Boolean)

Returns the last item in the Navigation History. It removes the data from History by default.

Declaration

```
public static NavigationPointerData GetLastItemFromNavigationHistory(bool removeFromHistory = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	removeFromHistory	

Returns

TYPE	DESCRIPTION
NavigationPointerData	

Hide(List<NavigationPointer>, Boolean, Boolean)

Executes the Hide for the given list of Navigation Pointers.

Declaration

```
public static void Hide(List<NavigationPointer> hide, bool instantAction = false, bool disableWhenHidden = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<DoozyUI.NavigationPointer>	hide	
System.Boolean	instantAction	
System.Boolean	disableWhenHidden	

RemoveLastItemFromHistory()

Removes the last item from the Navigation History (FIFO - First In Last Out).

Declaration

```
public static void RemoveLastItemFromHistory()
```

Show(List<NavigationPointer>, Boolean)

Executes the Show for the given list of Navigation Pointers.

Declaration

```
public static void Show(List<NavigationPointer> show, bool instantAction = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<DoozyUI.NavigationPointer>	show	
System.Boolean	instantAction	

UpdateTheNavigationHistory(NavigationPointerData)

Updates the Navigation History while showing and hiding the relevant UIElements.

Declaration

```
public static void UpdateTheNavigationHistory(NavigationPointerData navData)
```

Parameters

TYPE	NAME	DESCRIPTION
NavigationPointerData	navData	

Class UINotification

Inheritance

System.Object

UINotification

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UINotification : MonoBehaviour
```

Fields

buttons

An array of references of child UIButtons that will be used as the notification's buttons.

Declaration

```
public UIButton[] buttons
```

Field Value

TYPE	DESCRIPTION
DoozyUI.UIButton[]	

closeButton

This is a reference to a Button component, that is attached by default to the notification's gameObject. Upon clicking the notification, it will auto close (by calling Hide on itself).

Declaration

```
public Button closeButton
```

Field Value

TYPE	DESCRIPTION
Button	

customTargetCanvasName

Used by the custom inspector to allow you to type a canvas name instead of selecting it from the Canvas Names Database.

Declaration

```
public bool customTargetCanvasName
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

data

Used by the notification when it gets set up.

Declaration

```
public UnityEngine.NotificationData data
```

Field Value

TYPE	DESCRIPTION
UnityEngine.NotificationData	

DEFAULT_ADD_TO_NOTIFICATION_QUEUE

Default behaviour if a notification should be added to the notification queue or be shown right away.

Declaration

```
public const bool DEFAULT_ADD_TO_NOTIFICATION_QUEUE = true
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

DEFAULT_BUTTON_NAMES

Default notification array of button names (the button name is used to distinguish buttons one from the other).

Declaration

```
public const string[] DEFAULT_BUTTON_NAMES = null
```

Field Value

TYPE	DESCRIPTION
System.String[]	

DEFAULT_BUTTON_TEXT

Default notification array of button texts (the button texts are the text values shown on the buttons).

Declaration

```
public const string[] DEFAULT_BUTTON_TEXT = null
```

Field Value

TYPE	DESCRIPTION
System.String[]	

DEFAULT_ICON

Default notification icon.

Declaration

```
public const Sprite DEFAULT_ICON = null
```

Field Value

TYPE	DESCRIPTION
Sprite	

DEFAULT_LIFETIME

Default time interval of how long should a notification be seen on screen before the Hide command is automatically issued.

Declaration

```
public const float DEFAULT_LIFETIME = 3F
```

Field Value

TYPE	DESCRIPTION
System.Single	

DEFAULT_MESSAGE

Default notification message.

Declaration

```
public const string DEFAULT_MESSAGE = null
```

Field Value

TYPE	DESCRIPTION
System.String	

DEFAULT_TITLE

Default notification title.

Declaration

```
public const string DEFAULT_TITLE = null
```

Field Value

TYPE	DESCRIPTION
System.String	

effects

An array of references to any child UIEffects used by this notification.

Declaration

```
public UIEffect[] effects
```

Field Value

TYPE	DESCRIPTION
DoozyUI.UIEffect[]	

icon

Reference to a child GameObject with an Image attached. This Image component will get it's sprite value set to the notification's icon.

Declaration

```
public Image icon
```

Field Value

TYPE	DESCRIPTION
Image	

listenForBackButton

Should this notification listen for the 'Back' button? If yes, upon pressing the 'Back' button, the notification will close by automaically calling Hide on itself. Default is true.

Declaration

```
public bool listenForBackButton
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

message

Reference to a child GameObject that has a Text component atatched. This Text component will get it's text value set to the notification's message.

Declaration

```
public GameObject message
```

Field Value

TYPE	DESCRIPTION
GameObject	

notificationContainer

Reference to the main UIElement that holds everything.

Declaration

```
public UIElement notificationContainer
```

Field Value

TYPE	DESCRIPTION
UIElement	

overlay

Reference to an UIElement that can be used as a background image.

Declaration

```
public UIElement overlay
```

Field Value

TYPE	DESCRIPTION
UIElement	

specialElements

An array of references to any other child UIElements that need to be controlled by this notification. I allows for a lot of flexibility design wise.

For example if you have 3 stars with different animations, you can create an UIElement gameGbject for each, set up their respective animations and reference them to this array.

Declaration

```
public UIElement[] specialElements
```

Field Value

TYPE	DESCRIPTION
DoozyUI.UIElement[]	

targetCanvasName

The target canvas where this notification will be shown.

Declaration

```
public string targetCanvasName
```

Field Value

TYPE	DESCRIPTION
System.String	

title

Reference to a child GameObject that has a Text component atatched. This Text component will get it's text value set to the notification's title.

Declaration

```
public GameObject title
```

Field Value

TYPE	DESCRIPTION
GameObject	

Properties

RectTransform

Returns the RectTransform component.

Declaration

```
public RectTransform RectTransform { get; }
```

Property Value

TYPE	DESCRIPTION
RectTransform	

Methods

HideNotification(Boolean)

Hides the notification with a destroy option. Default betification behaviour is to get automatically destroyed.

Declaration

```
public void HideNotification(bool hideAndDestroy = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	hideAndDestroy	

Initialize()

Executes the initial setup of this notification.

Declaration

```
public void Initialize()
```

ShowNotification(UINotification.NotificationData, UICanvas)

Shows the notification taking into account the NotificationData value.

Declaration

```
public void ShowNotification(UINotification.NotificationData ndata, UICanvas targetCanvas)
```

Parameters

TYPE	NAME	DESCRIPTION
UINotification.NotificationData	ndata	
UICanvas	targetCanvas	

Class UINotification.NotificationData

Helper class that holds all the Notification settings.

Inheritance

System.Object
UINotification.NotificationData

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class NotificationData
```

Fields

addToNotificationQueue

Should this notification be added to the Norification Queue or should it ignore it? (default: true)

Declaration

```
public bool addToNotificationQueue
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

buttonCallback

Callback action for every button

Declaration

```
public UnityEngine.UI.Button.ButtonClickedHandler buttonCallback
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UI.Button.ButtonClickedHandler	

buttonNames

If the notification has buttons, these are the buttonNames that will be sent on Button Click. If there are 3 buttons available and you enter only 2 buttonNames, only those 2 buttons will be visible and active (the 3rd will not appear, nor work).

Declaration

```
public string[] buttonNames
```

Field Value

TYPE	DESCRIPTION
System.String[]	

buttonTexts

If the notification has buttons and those buttons have a Text or a TextMeshProUGUI compoment attached to them or one of their children, then these are the button text that will appear on the buttons. If there are 3 buttons available and active, and you enter the button text for only 2 of them, only the first 2 buttons well have a text, and the third will have nothing. You can leave this null if your buttons show pre-set icons instead of text.

Declaration

```
public string[] buttonTexts
```

Field Value

TYPE	DESCRIPTION
System.String[]	

hideCallback

Callback action @Hide

Declaration

```
public UnityAction hideCallback
```

Field Value

TYPE	DESCRIPTION
UnityAction	

icon

If the notification has a custom icon, this sprite will appear there.

Declaration

```
public Sprite icon
```

Field Value

TYPE	DESCRIPTION
Sprite	

lifetime

The lifetime of the norification. Excluding the IN and OUT animation times, as they are calculated separately.

Declaration

```
public float lifetime
```

Field Value

TYPE	DESCRIPTION
System.Single	

message

If the notification has a message, this text will appear there.

Declaration

public string message

Field Value

TYPE	DESCRIPTION
System.String	

prefab

The prefab GameObject

Declaration

public GameObject prefab

Field Value

TYPE	DESCRIPTION
GameObject	

prefabName

The name of the notification prefab in a 'Resources' folder or the notification name set up in the Inspector of the UI Notification Manager

Declaration

public string prefabName

Field Value

TYPE	DESCRIPTION
System.String	

targetCanvasName

The target canvas where this notification will be shown.

Declaration

public string targetCanvasName

Field Value

TYPE	DESCRIPTION
System.String	

title

If the notification has a title, this text will appear there.

Declaration

```
public string title
```

Field Value

TYPE	DESCRIPTION
System.String	

Class UINotificationManager

Inheritance

System.Object
UINotificationManager

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class UINotificationManager : MonoBehaviour
```

Fields

NotificationItems

List of notification items that have been set up in the Inspector.

Declaration

```
public List<UINotificationManager.NotificationItem> NotificationItems
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<DoozyUI.UINotificationManager.NotificationItem>	

Properties

NotificationQueue

The Notification Queue list.

Declaration

```
public List<UINotification.NotificationData> NotificationQueue { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<DoozyUI.UINotification.NotificationData>	

Methods

GetUINotification(String)

Declaration

```
public UINotification GetUINotification(string notificationName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	notificationName	

Returns

TYPE	DESCRIPTION
UINotification	

RegisterToNotificationQueue(UINotification.NotificationData)

Every notification that needs to enter the Notification Queue will be added to the notificatioQueue list as the last item.

Declaration

<pre>public void RegisterToNotificationQueue(UINotification.NotificationData nData)</pre>

Parameters

TYPE	NAME	DESCRIPTION
UINotification.NotificationData	nData	

ShowNotification(GameObject, Single, Boolean, Sprite, UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

<pre>public UINotification ShowNotification(GameObject _prefab, float _lifetime, bool _addToNotificationQueue, Sprite _icon, UnityAction _hideCallback = null)</pre>
--

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
Sprite	_icon	The sprite you want the notification icon to have (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String, Sprite, UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public UINotification ShowNotification(GameObject _prefab, float _lifetime, bool _addToNotificationQueue,
string _title, Sprite _icon, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String, String, Sprite, String[], String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public UINotification ShowNotification(GameObject _prefab, float _lifetime, bool _addToNotificationQueue,
string _title, string _message, Sprite _icon, string[] _buttonNames, string[] _buttonTexts, UnityAction[]
_buttonCallback = null, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway

TYPE	NAME	DESCRIPTION
System.String	_title	The text you want to show in the title area (if linked)
System.String	_message	The message you want to show in the message area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
System.String[]	_buttonTexts	The text on the buttons (example: 'OK', 'Cancel', 'Yes', 'No' and so on)
UnityEngine.UI.ButtonClickedEvent	_buttonCallback	
UnityEngine.UI.ButtonClickedEvent	_hideCallback	

Returns

TYPE	DESCRIPTION
UnityEngine.UI.Notification	

ShowNotification(GameObject, Single, Boolean, String, String, Sprite, String[], UnityEngine.UI.ButtonClickedEvent, UnityEngine.UI.ButtonClickedEvent)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public UnityEngine.UI.Notification ShowNotification(GameObject _prefab, float _lifetime, bool _addToNotificationQueue,
string _title, string _message, Sprite _icon, string[] _buttonNames, UnityEngine.UI.ButtonClickedEvent _buttonCallback = null,
UnityEngine.UI.ButtonClickedEvent _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)

TYPE	NAME	DESCRIPTION
System.String	_message	The message you want to show in the message area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
UnityEngine.UI.Button[]	_buttonCallback	
UnityEngine.UI.Button	_hideCallback	

Returns

TYPE	DESCRIPTION
UnityEngine.UI.Notification	

ShowNotification(GameObject, Single, Boolean, String, String, Sprite, UnityEngine.UI.Button)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public UnityEngine.UI.Notification ShowNotification(GameObject _prefab, float _lifetime, bool _addToNotificationQueue, string _title, string _message, Sprite _icon, UnityEngine.UI.Button _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String	_message	The message you want to show in the message area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
UnityEngine.UI.Button	_hideCallback	

TYPE	NAME	DESCRIPTION

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String, String, UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public UINotification ShowNotification(GameObject _prefab, float _lifetime, bool _addToNotificationQueue,
string _title, string _message, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String	_message	The message you want to show in the message area (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String, String[], String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public UINotification ShowNotification(GameObject _prefab, float _lifetime, bool _addToNotificationQueue,
string _title, string[] _buttonNames, string[] _buttonTexts, UnityAction[] _buttonCallback = null, UnityAction
_hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
System.String[]	_buttonTexts	The text on the buttons (example: 'OK', 'Cancel', 'Yes', 'No' and so on)
UnityEngine.UI.ButtonClickedEvent	_buttonCallback	
UnityEngine.UI.ButtonClickedEvent	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String, String[], UnityEngine.UI.ButtonClickedEvent, UnityEngine.UI.ButtonClickedEvent)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public UINotification ShowNotification(GameObject _prefab, float _lifetime, bool _addToNotificationQueue,
string _title, string[] _buttonNames, UnityEngine.UI.ButtonClickedEvent _buttonCallback = null, UnityEngine.UI.ButtonClickedEvent _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway

TYPE	NAME	DESCRIPTION
System.String	_title	The text you want to show in the title area (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
UnityEngine.UI.Button.OnClick[]	_buttonCallback	
UnityEngine.UI.Button.OnClick	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String, UnityEngine.UI.Button.OnClick)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public UINotification ShowNotification(GameObject _prefab, float _lifetime, bool _addToNotificationQueue, string _title, UnityEngine.UI.Button.OnClick _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
UnityEngine.UI.Button.OnClick	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String[], String[], UnityEngine.UI.Button.OnClick[], UnityEngine.UI.Button.OnClick)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public UINotification ShowNotification(GameObject _prefab, float _lifetime, bool _addToNotificationQueue,
string[] _buttonNames, string[] _buttonTexts, UnityAction[] _buttonCallback = null, UnityAction _hideCallback
= null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
System.String[]	_buttonTexts	The text on the buttons (example: 'OK', 'Cancel', 'Yes', 'No' and so on)
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public UINotification ShowNotification(GameObject _prefab, float _lifetime, bool _addToNotificationQueue,
string[] _buttonNames, UnityAction[] _buttonCallback = null, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1

TYPE	NAME	DESCRIPTION
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
UnityEngine.UI.Button[]	_buttonCallback	
UnityEngine.UI.Button	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(GameObject, Single, Boolean, UnityEngine.UI.Button)

Show a premade notification with the given settings, using a prefab GameObject reference.

Declaration

```
public UINotification ShowNotification(GameObject _prefab, float _lifetime, bool _addToNotificationQueue,
UnityEngine.UI.Button _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
GameObject	_prefab	The prefab GameObject reference
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
UnityEngine.UI.Button	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, Sprite, UnityEngine.UI.Button)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public UINotification ShowNotification(string _prefabName, float _lifetime, bool _addToNotificationQueue,
Sprite _icon, UnityEngine.UI.Button _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
Sprite	_icon	The sprite you want the notification icon to have (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String, Sprite, UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public UINotification ShowNotification(string _prefabName, float _lifetime, bool _addToNotificationQueue, string _title, Sprite _icon, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String, String, Sprite, String[], String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public UINotification ShowNotification(string _prefabName, float _lifetime, bool _addToNotificationQueue,
string _title, string _message, Sprite _icon, string[] _buttonNames, string[] _buttonTexts, UnityAction[]
_buttonCallback = null, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	
System.Boolean	_addToNotificationQueue	
System.String	_title	
System.String	_message	
Sprite	_icon	
System.String[]	_buttonNames	
System.String[]	_buttonTexts	
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String, String, Sprite, String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public UINotification ShowNotification(string _prefabName, float _lifetime, bool _addToNotificationQueue,
string _title, string _message, Sprite _icon, string[] _buttonNames, UnityAction[] _buttonCallback = null,
UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String	_message	The message you want to show in the message area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String, String, Sprite, UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public UINotification ShowNotification(string _prefabName, float _lifetime, bool _addToNotificationQueue,
string _title, string _message, Sprite _icon, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1

TYPE	NAME	DESCRIPTION
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String	_message	The message you want to show in the message area (if linked)
Sprite	_icon	The sprite you want the notification icon to have (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String, String, UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public UINotification ShowNotification(string _prefabName, float _lifetime, bool _addToNotificationQueue,
string _title, string _message, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String	_message	The message you want to show in the message area (if linked)
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String, String[], String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public UINotification ShowNotification(string _prefabName, float _lifetime, bool _addToNotificationQueue,
string _title, string[] _buttonNames, string[] _buttonTexts, UnityAction[] _buttonCallback = null, UnityAction
_hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
System.String[]	_buttonTexts	The text on the buttons (example: 'OK', 'Cancel', 'Yes', 'No' and so on)
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String, String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public UINotification ShowNotification(string _prefabName, float _lifetime, bool _addToNotificationQueue,
string _title, string[] _buttonNames, UnityAction[] _buttonCallback = null, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String, UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

<pre>public UINotification ShowNotification(string _prefabName, float _lifetime, bool _addToNotificationQueue, string _title, UnityAction _hideCallback = null)</pre>

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String	_title	The text you want to show in the title area (if linked)

TYPE	NAME	DESCRIPTION
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String[], String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public UINotification ShowNotification(string _prefabName, float _lifetime, bool _addToNotificationQueue,
string[] _buttonNames, string[] _buttonTexts, UnityAction[] _buttonCallback = null, UnityAction _hideCallback
= null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
System.String[]	_buttonTexts	The text on the buttons (example: 'OK', 'Cancel', 'Yes', 'No' and so on)
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, String[], UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public UINotification ShowNotification(string _prefabName, float _lifetime, bool _addToNotificationQueue,
string[] _buttonNames, UnityAction[] _buttonCallback = null, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
System.String[]	_buttonNames	The button names you want the notification to have (from left to right). These values are the ones that we listen to as button click
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

ShowNotification(String, Single, Boolean, UnityAction[], UnityAction)

Show a premade notification with the given settings, using a prefabName.

Declaration

```
public UINotification ShowNotification(string _prefabName, float _lifetime, bool _addToNotificationQueue,
UnityAction[] _buttonCallback = null, UnityAction _hideCallback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	_prefabName	The prefab name
System.Single	_lifetime	How long will the notification be on the screen. Infinite lifetime is -1
System.Boolean	_addToNotificationQueue	Should this notification be added to the NotificationQueue or shown rightaway
UnityAction[]	_buttonCallback	
UnityAction	_hideCallback	

Returns

TYPE	DESCRIPTION
UINotification	

`UnregisterFromNotificationQueue(UINotification.NotificationData)`

Unregisters a notification, by removing the notification data that started it.

Declaration

```
public void UnregisterFromNotificationQueue(UINotification.NotificationData nData)
```

Parameters

TYPE	NAME	DESCRIPTION
UINotification.NotificationData	nData	

Class UINotificationManager.NotificationItem

Helper class for the NotificationManager.

Inheritance

System.Object
UINotificationManager.NotificationItem

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class NotificationItem
```

Fields

notificationName

Notification Name

Declaration

```
public string notificationName
```

Field Value

TYPE	DESCRIPTION
System.String	

notificationPrefab

Notification Prefab

Declaration

```
public UINotification notificationPrefab
```

Field Value

TYPE	DESCRIPTION
UINotification	

Class UISound

Inheritance

System.Object
UISound

Namespace: [DoozyUI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class UISound : ScriptableObject
```

Constructors

UISound(String, AudioClip, SoundType)

Declaration

```
public UISound(string sName, AudioClip aClip, SoundType sType = SoundType.All)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	sName	
AudioClip	aClip	
SoundType	sType	

Fields

audioClip

Declaration

```
public AudioClip audioClip
```

Field Value

TYPE	DESCRIPTION
AudioClip	

soundName

Declaration

```
public string soundName
```

Field Value

TYPE	DESCRIPTION
System.String	

soundType

Declaration

public SoundType soundType

Field Value

TYPE	DESCRIPTION
SoundType	

Class UITrigger

Inheritance

System.Object

UITrigger

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UITrigger : MonoBehaviour
```

Fields

buttonCategory

Used by the custom inspector to allow you to select a button name from the UIButtons Database.

Declaration

```
public string buttonCategory
```

Field Value

TYPE	DESCRIPTION
System.String	

buttonName

If any of triggerOnButtonClick or triggerOnButtonDoubleClick or triggerOnButtonLongClick are true, this is the button name value that will make this UITrigger extcute its actions.

Declaration

```
public string buttonName
```

Field Value

TYPE	DESCRIPTION
System.String	

dispatchAll

If dispatch all is set to true, game event and button name are set to a special value that make this UITrigger execute its actions on every game event or button click/double click/long click.

Declaration

```
public bool dispatchAll
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

gameEvent

If triggerOnGameEvent is true, this is the game event value that will make this UITrigger extcute its actions.

Declaration

```
public string gameEvent
```

Field Value

TYPE	DESCRIPTION
System.String	

gameEvents

List of game events that are sent by the UITrigger when it executes its actions.

Declaration

```
public List<string> gameEvents
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

triggerOnButtonClick

Should this UITrigger execute its actions on button click? Default is false.

Declaration

```
public bool triggerOnButtonClick
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

triggerOnButtonDoubleClick

Should this UITrigger execute its actions on button double click? Default is false.

Declaration

```
public bool triggerOnButtonDoubleClick
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

triggerOnButtonLongClick

Should this UITrigger execute its actions on button long click? Default is false.

Declaration

```
public bool triggerOnButtonLongClick
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

triggerOnGameEvent

Should this UITrigger execute its actions on game event? Default is false.

Declaration

<pre>public bool triggerOnGameEvent</pre>

Field Value

TYPE	DESCRIPTION
System.Boolean	

Properties

Enabled

Retruns true if this UITrigger has proper settings set up and is operational.

Declaration

<pre>public bool Enabled { get; }</pre>

Property Value

TYPE	DESCRIPTION
System.Boolean	

ListeningFor

Returns the type of event that this UITrigger is listening for.

Declaration

<pre>public DUI.EventType ListeningFor { get; }</pre>

Property Value

TYPE	DESCRIPTION
DUI.EventType	

Methods

RegisterToUIManager()

Registers this UITrigger to the UIManager.

Declaration

<pre>protected void RegisterToUIManager()</pre>

TriggerTheTrigger(String)

Triggers the UITrigger to execute its actions.

Declaration

```
public void TriggerTheTrigger(string triggerValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	triggerValue	

UnregisterFromUIManager()

Unregisters this UITrigger from the UIManager.

Declaration

```
protected void UnregisterFromUIManager()
```

Class UITrigger.TriggerEvent

Helper class for an UnityEvent with one string parameter.

Inheritance

System.Object

UITrigger.TriggerEvent

Namespace: [DoozyUI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class TriggerEvent : UnityEvent<string>
```


Namespace QuickEngine

Classes

[QColor](#)

[QColors](#)

[QResources](#)

Class QColor

Inheritance

System.Object
QColor

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
[Serializable]  
public class QColor
```

Constructors

QColor(Color)

Declaration

```
public QColor(Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
Color	color	

QColor(Color, Single)

Declaration

```
public QColor(Color color, float alpha)
```

Parameters

TYPE	NAME	DESCRIPTION
Color	color	
System.Single	alpha	

QColor(Single, Single, Single, Boolean)

Declaration

```
public QColor(float r, float g, float b, bool from256 = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	r	
System.Single	g	
System.Single	b	
System.Boolean	from256	

QColor(Single, Single, Single, Single, Boolean)

Declaration

```
public QColor(float r, float g, float b, float a, bool from256 = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	r	
System.Single	g	
System.Single	b	
System.Single	a	
System.Boolean	from256	

Properties

Color

Declaration

```
public Color Color { get; }
```

Property Value

TYPE	DESCRIPTION
Color	

ColorBrightness

Returns the brightness of the color, defined as the average off the three color channels.

Declaration

```
public float ColorBrightness { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ColorDark

Declaration

```
public Color ColorDark { get; }
```

Property Value

TYPE	DESCRIPTION
Color	

ColorDarkBrightness

Returns the brightness of the color, defined as the average off the three color channels.

Declaration

```
public float ColorDarkBrightness { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ColorDarkInvert

Returns a new color that is this color inverted.

Declaration

```
public Color ColorDarkInvert { get; }
```

Property Value

TYPE	DESCRIPTION
Color	

ColorDarkOpaque

Returns an opaque version of the set color.

Declaration

```
public Color ColorDarkOpaque { get; }
```

Property Value

TYPE	DESCRIPTION
Color	

ColorInvert

Returns a new color that is this color inverted.

Declaration

```
public Color ColorInvert { get; }
```

Property Value

TYPE	DESCRIPTION
Color	

ColorLight

Declaration

```
public Color ColorLight { get; }
```

Property Value

TYPE	DESCRIPTION
Color	

ColorLightBrightness

Returns the brightness of the color, defined as the average off the three color channels.

Declaration

```
public float ColorLightBrightness { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

ColorLightInvert

Returns a new color that is this color inverted.

Declaration

```
public Color ColorLightInvert { get; }
```

Property Value

TYPE	DESCRIPTION
Color	

ColorLightOpaque

Returns an opaque version of the set color.

Declaration

```
public Color ColorLightOpaque { get; }
```

Property Value

TYPE	DESCRIPTION
Color	

ColorOpaque

Returns an opaque version of the set color.

Declaration

```
public Color ColorOpaque { get; }
```

Property Value

TYPE	DESCRIPTION
Color	

Methods

ColorDarkWithAlpha(Single)

Returns a new color with the given alpha.

Declaration

```
public Color ColorDarkWithAlpha(float alpha)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	alpha	

Returns

TYPE	DESCRIPTION
Color	

ColorDarkWithBrightness(Single)

Returns a new color with the RGB values scaled so that the color has the given brightness.

If the color is too dark, a grey is returned with the right brightness. The alpha is left uncanged.

Declaration

```
public Color ColorDarkWithBrightness(float brightness)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	brightness	

Returns

TYPE	DESCRIPTION
Color	

ColorLightWithAlpha(Single)

Returns a new color with the given alpha.

Declaration

```
public Color ColorLightWithAlpha(float alpha)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	alpha	

Returns

TYPE	DESCRIPTION
Color	

ColorLightWithBrightness(Single)

Returns a new color with the RGB values scaled so that the color has the given brightness.

If the color is too dark, a grey is returned with the right brightness. The alpha is left uncanged.

Declaration

```
public Color ColorLightWithBrightness(float brightness)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	brightness	

Returns

TYPE	DESCRIPTION
Color	

ColorWithAlpha(Single)

Returns a new color with the given alpha.

Declaration

```
public Color ColorWithAlpha(float alpha)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	alpha	

Returns

TYPE	DESCRIPTION
Color	

ColorWithBrightness(Single)

Returns a new color with the RGB values scaled so that the color has the given brightness.

If the color is too dark, a grey is returned with the right brightness. The alpha is left uncanged.

Declaration

```
public Color ColorWithBrightness(float brightness)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	brightness	

Returns

TYPE	DESCRIPTION
Color	

SetColor(Color)

Updates the color.

Declaration

```
public void SetColor(Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
Color	color	

Class QColor

Inheritance

System.Object
QColor

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public class QColor
```

Fields

UnityDark

Declaration

```
public static QColor UnityDark
```

Field Value

TYPE	DESCRIPTION
QColor	

UnityLight

Declaration

```
public static QColor UnityLight
```

Field Value

TYPE	DESCRIPTION
QColor	

UnityMild

Declaration

```
public static QColor UnityMild
```

Field Value

TYPE	DESCRIPTION
QColor	

WhiteDark

Declaration

```
public static QColor WhiteDark
```

Field Value

TYPE	DESCRIPTION
QColor	

WhiteLight

Declaration

```
public static QColor WhiteLight
```

Field Value

TYPE	DESCRIPTION
QColor	

Class QResources

Inheritance

System.Object
QResources

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public class QResources
```

Properties

FontAwesome

Declaration

```
public static Font FontAwesome { get; }
```

Property Value

TYPE	DESCRIPTION
Font	

Namespace QuickEngine.Common

Classes

[Singleton<T>](#)

Be aware this will not prevent a non singleton constructor such as `T myT = new T();` To prevent that, add `protected T () {}` to your singleton class.

As a note, this is made as MonoBehaviour because we need Coroutines.

Class Singleton<T>

Be aware this will not prevent a non singleton constructor such as `T myT = new T();` To prevent that, add `protected T () {}` to your singleton class.

As a note, this is made as MonoBehaviour because we need Coroutines.

Inheritance

System.Object

Singleton<T>

[OrientationManager](#)

[SceneLoader](#)

[UIManager](#)

Namespace: [QuickEngine.Common](#)

Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public class Singleton<T> : MonoBehaviour where T : MonoBehaviour
```

Type Parameters

NAME	DESCRIPTION
T	

Properties

Instance

Declaration

```
public static T Instance { get; }
```

Property Value

TYPE	DESCRIPTION
T	

Methods

OnDestroy()

When Unity quits, it destroys objects in a random order. In principle, a Singleton is only destroyed when application quits. If any script calls Instance after it have been destroyed, it will create a buggy ghost object that will stay on the Editor scene even after stopping playing the Application. Really bad! So, this was made to be sure we're not creating that buggy ghost object.

Declaration

```
public void OnDestroy()
```

Namespace QuickEngine.Extensions

Classes

[ArrayAndListExtensions](#)

[AudioExtensions](#)

[BoolExtensions](#)

[CameraExtensions](#)

[ColorExtensions](#)

[DateTimeExtensions](#)

[DictionaryExtensions](#)

[FloatExtensions](#)

[IListExtensions](#)

[RectTransformExtensions](#)

[StringExtensions](#)

[TransformExtensions](#)

[VectorExtensions](#)

Class ArrayAndListExtensions

Inheritance

System.Object
ArrayAndListExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Extensions](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class ArrayAndListExtensions
```

Methods

GetRandomElement<T>(T[])

Returns a random element of the array. Does NOT check if the array is empty or null!

Declaration

```
public static T GetRandomElement<T>(this T[] array)
```

Parameters

TYPE	NAME	DESCRIPTION
T[]	array	The array.

Returns

TYPE	DESCRIPTION
T	

Type Parameters

NAME	DESCRIPTION
T	Array Type.

GetRandomElement<T>(List<T>)

Returns a random element of the list. Does NOT check if the list is empty or null!

Declaration

```
public static T GetRandomElement<T>(this List<T> list)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<T>	list	The list.

Returns

TYPE	DESCRIPTION
T	Radom element from the list

Type Parameters

NAME	DESCRIPTION
T	List Type.

IsNullOrEmpty<T>(T[])

Returns true if the array is null or empty.

Declaration

```
public static bool IsNullOrEmpty<T>(this T[] array)
```

Parameters

TYPE	NAME	DESCRIPTION
T[]	array	The array.

Returns

TYPE	DESCRIPTION
System.Boolean	

Type Parameters

NAME	DESCRIPTION
T	Array Type.

IsNullOrEmpty<T>(List<T>)

Returns true if the list is null or empty.

Declaration

```
public static bool IsNullOrEmpty<T>(this List<T> list)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<T>	list	The list.

Returns

TYPE	DESCRIPTION
System.Boolean	

Type Parameters

NAME	DESCRIPTION
T	List Type.

IsEmpty<TKey, TValue>(Dictionary<TKey, TValue>)

Returns true if the dictionary is null or empty.

Declaration

```
public static bool IsNullOrEmpty<TKey, TValue>(this Dictionary<TKey, TValue> dict)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<TKey, TValue>	dict	

Returns

TYPE	DESCRIPTION
System.Boolean	

Type Parameters

NAME	DESCRIPTION
TKey	Key Type.
TValue	Value Type.

ShuffleArray<T>(T[])

Shuffle the array.

Declaration

```
public static void ShuffleArray<T>(this T[] array)
```

Parameters

TYPE	NAME	DESCRIPTION
T[]	array	The array.

Type Parameters

NAME	DESCRIPTION
T	Array Type.

ShuffleList<T>(List<T>)

Shuffle the list.

Declaration

```
public static void ShuffleList<T>(this List<T> list)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<T>	list	The list.

Type Parameters

NAME	DESCRIPTION
T	List type.

ToString<T>(T[], String)

Joins all the elements of the array into a string separated by the given separator string.

Declaration

```
public static string ToString<T>(this T[] array, string separator)
```

Parameters

TYPE	NAME	DESCRIPTION
T[]	array	The array.
System.String	separator	String separator.

Returns

TYPE	DESCRIPTION
System.String	

Type Parameters

NAME	DESCRIPTION
T	Array Type.

ToString<T>(List<T>, String)

Joins all the elements of the list into a string separated by the given separator string.

Declaration

public static string ToString<T>(this List<T> list, string separator)

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<T>	list	The list.
System.String	separator	String separator.

Returns

TYPE	DESCRIPTION
System.String	

Type Parameters

NAME	DESCRIPTION
T	List Type.

Class AudioExtensions

Inheritance

System.Object
AudioExtensions

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Extensions](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class AudioExtensions
```

Methods

PlatformAudioExtension()

Declaration

```
public static string PlatformAudioExtension()
```

Returns

TYPE	DESCRIPTION
System.String	

PlatformAudioType()

Declaration

```
public static AudioType PlatformAudioType()
```

Returns

TYPE	DESCRIPTION
AudioType	

PlatformFileProtocol()

Declaration

```
public static string PlatformFileProtocol()
```

Returns

TYPE	DESCRIPTION
System.String	

PlayOneShotDelayed(AudioSource, AudioClip, Single)

Declaration

```
public static IEnumerator PlayOneShotDelayed(this AudioSource anAudioSource, AudioClip anAudioClip, float aDelay)
```

Parameters

TYPE	NAME	DESCRIPTION
AudioSource	anAudioSource	
AudioClip	anAudioClip	
System.Single	aDelay	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

ToDecibel(Single)

Declaration

```
public static float ToDecibel(this float linear)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	linear	

Returns

TYPE	DESCRIPTION
System.Single	

ToLinear(Single)

Declaration

```
public static float ToLinear(this float dB)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	dB	

Returns

TYPE	DESCRIPTION
System.Single	

Class BoolExtensions

Inheritance

System.Object
BoolExtensions

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Extensions](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class BoolExtensions
```

Methods

IsFalse(Boolean)

Checks whether the given boolean item is false.

Declaration

```
public static bool IsFalse(this bool bool)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	bool	The boolean value.

Returns

TYPE	DESCRIPTION
System.Boolean	

IsTrue(Boolean)

Checks whether the given boolean item is true.

Declaration

```
public static bool IsTrue(this bool bool)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	bool	The boolean value.

Returns

TYPE	DESCRIPTION
System.Boolean	

Toggle(Boolean)

Toggles the given boolean item and returns the toggled value.

Declaration

<pre>public static bool Toggle(this bool bool)</pre>
--

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	bool	The boolean value.

Returns

TYPE	DESCRIPTION
System.Boolean	

ToInt(Boolean)

Converts the given boolean value to integer.

Declaration

<pre>public static int ToInt(this bool bool)</pre>
--

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	bool	The boolean value.

Returns

TYPE	DESCRIPTION
System.Int32	

ToLowerString(Boolean)

Returns the lower string representation of boolean.

Declaration

<pre>public static string ToLowerString(this bool bool)</pre>

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	bool	The boolean value.

Returns

TYPE	DESCRIPTION
System.String	

ToString(Boolean, String, String)

Returns the trueString or falseString based on the given boolean value.

Declaration

```
public static string ToString(this bool bool, string trueString, string falseString)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	bool	The boolean value.
System.String	trueString	String returned if the bool is true.
System.String	falseString	String returned if the bool is false.

Returns

TYPE	DESCRIPTION
System.String	

ToType<T>(Boolean, T, T)

Returns the trueValue or the falseValue based on the given boolean value.

Declaration

```
public static T ToType<T>(this bool bool, T trueValue, T falseValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	bool	The boolean value.
T	trueValue	Value returned if the bool is true.

TYPE	NAME	DESCRIPTION
T	falseValue	Value returned if the bool is false.

Returns

TYPE	DESCRIPTION
T	

Type Parameters

NAME	DESCRIPTION
T	Output type.

Class CameraExtensions

Inheritance

System.Object
CameraExtensions

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Extensions](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class CameraExtensions
```

Methods

EdgePosition(Camera, TextAnchor, Single)

Declaration

```
public static Vector3 EdgePosition(this Camera camera, TextAnchor point, float distance)
```

Parameters

TYPE	NAME	DESCRIPTION
Camera	camera	
TextAnchor	point	
System.Single	distance	

Returns

TYPE	DESCRIPTION
Vector3	

Pixel2Units2D(Camera)

Declaration

```
public static Vector2 Pixel2Units2D(this Camera c)
```

Parameters

TYPE	NAME	DESCRIPTION
Camera	c	

Returns

TYPE	DESCRIPTION
Vector2	

ToScreenRect(Camera, Renderer)

Declaration

```
public static Rect ToScreenRect(this Camera camera, Renderer renderer)
```

Parameters

TYPE	NAME	DESCRIPTION
Camera	camera	
Renderer	renderer	

Returns

TYPE	DESCRIPTION
Rect	

ToScreenSize(Camera, Bounds)

Declaration

```
public static Vector2 ToScreenSize(this Camera camera, Bounds bounds)
```

Parameters

TYPE	NAME	DESCRIPTION
Camera	camera	
Bounds	bounds	

Returns

TYPE	DESCRIPTION
Vector2	

ToWorldRect(Camera, Renderer)

Declaration

```
public static Rect ToWorldRect(this Camera camera, Renderer renderer)
```

Parameters

TYPE	NAME	DESCRIPTION
Camera	camera	
Renderer	renderer	

Returns

TYPE	DESCRIPTION
Rect	

ToWorldSize(Camera, Bounds)

Declaration

```
public static Vector2 ToWorldSize(this Camera camera, Bounds bounds)
```

Parameters

TYPE	NAME	DESCRIPTION
Camera	camera	
Bounds	bounds	

Returns

TYPE	DESCRIPTION
Vector2	

Unit2Pixels2D(Camera)

Declaration

```
public static Vector2 Unit2Pixels2D(this Camera c)
```

Parameters

TYPE	NAME	DESCRIPTION
Camera	c	

Returns

TYPE	DESCRIPTION
Vector2	

Class ColorExtensions

Inheritance

System.Object
ColorExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Extensions](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class ColorExtensions
```

Methods

Brightness(Color)

Returns the brightness of the Color, defined as the average off the three Color channels.

Declaration

```
public static float Brightness(this Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
Color	color	The Color.

Returns

TYPE	DESCRIPTION
System.Single	The new Color.

ColorFrom256(Color, Single, Single, Single, Single)

Returns a new Color with the given settings.

Declaration

```
public static Color ColorFrom256(this Color color, float r, float g, float b, float a = 256F)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
Color	color	The Color.
System.Single	r	red
System.Single	g	green
System.Single	b	blue
System.Single	a	alpha

Returns

TYPE	DESCRIPTION
Color	The new Color.

ColorFrom256(Single, Single, Single, Single)

Returns a new Color with the given settings.

Declaration

```
public static Color ColorFrom256(float r, float g, float b, float a = 256F)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	r	red
System.Single	g	green
System.Single	b	blue
System.Single	a	alpha

Returns

TYPE	DESCRIPTION
Color	

Darker(Color)

Returns a Color darker than the given color.

Declaration

```
public static Color Darker(this Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
Color	color	The Color.

Returns

TYPE	DESCRIPTION
Color	The new Color.

Invert(Color)

Returns a new Color that is the inversion of this Color.

Declaration

```
public static Color Invert(this Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
Color	color	The Color.

Returns

TYPE	DESCRIPTION
Color	

IsApproximatelyBlack(Color)

Returns true if the Color is black or almost black, false otherwise.

Declaration

```
public static bool IsApproximatelyBlack(this Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
Color	color	The Color.

Returns

TYPE	DESCRIPTION
System.Boolean	

IsApproximatelyWhite(Color)

Returns true if the Color is white or almost white, false otherwise.

Declaration

```
public static bool IsApproximatelyWhite(this Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
Color	color	The Color.

Returns

TYPE	DESCRIPTION
System.Boolean	

Lighter(Color)

Returns a Color lighter than the given color.

Declaration

```
public static Color Lighter(this Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
Color	color	The Color.

Returns

TYPE	DESCRIPTION
Color	The new Color.

Opaque(Color)

Returns an opaque (no transparency) version of the given Color.

Declaration

```
public static Color Opaque(this Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
Color	color	The Color.

Returns

TYPE	DESCRIPTION
Color	

WithAlpha(Color, Single)

Returns a new Color with the same settings and a new alpha.

Declaration

```
public static Color WithAlpha(this Color color, float alpha)
```

Parameters

TYPE	NAME	DESCRIPTION
Color	color	The Color.
System.Single	alpha	Alpha for the Color.

Returns

TYPE	DESCRIPTION
Color	

WithBrightness(Color, Single)

Returns a new Color with the RGB values scaled so that the color has the given brightness.

If the Color is too dark, a grey is returned with the right brightness.The alpha is left uncanged.

Declaration

```
public static Color WithBrightness(this Color color, float brightness)
```

Parameters

TYPE	NAME	DESCRIPTION
Color	color	The Color.
System.Single	brightness	New brightness.

Returns

TYPE	DESCRIPTION
Color	The new Color.

Class DateTimeExtensions

Inheritance

System.Object

DateTimeExtensions

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Extensions](#)

Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class DateTimeExtensions
```

Methods

EndOfMonth(DateTime)

Returns the last day of the month

Declaration

```
public static DateTime EndOfMonth(this DateTime date)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.

Returns

TYPE	DESCRIPTION
System.DateTime	

FirstOfMonth(DateTime)

Returns the first day of the month.

Declaration

```
public static DateTime FirstOfMonth(this DateTime date)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.

Returns

TYPE	DESCRIPTION
System.DateTime	

IsBetween(DateTime, DateTime, DateTime)

Returns true if a date is in a period between two others.

Declaration

<pre>public static bool IsBetween(this DateTime date, DateTime from, DateTime to)</pre>

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date checked.
System.DateTime	from	Start of period.
System.DateTime	to	End of period.

Returns

TYPE	DESCRIPTION
System.Boolean	

IsLaterDate(DateTime, DateTime)

Returns true if the date is greater than compareDate

Declaration

<pre>public static bool IsLaterDate(this DateTime date, DateTime compareDate)</pre>

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.
System.DateTime	compareDate	Date to compare.

Returns

TYPE	DESCRIPTION
System.Boolean	

IsOlderDate(DateTime, DateTime)

Returns true if the date is less than compareDate

Declaration

```
public static bool IsOlderDate(this DateTime date, DateTime compareDate)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.
System.DateTime	compareDate	Date to compare.

Returns

TYPE	DESCRIPTION
System.Boolean	

IsSameDay(DateTime, DateTime)

Returns true if both dates are on the same day.

Declaration

```
public static bool IsSameDay(this DateTime date, DateTime compareDate)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.
System.DateTime	compareDate	Date to be compared.

Returns

TYPE	DESCRIPTION
System.Boolean	

IsToday(DateTime)

Checks whether the given day is Today.

Declaration

```
public static bool IsToday(this DateTime date)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.

Returns

TYPE	DESCRIPTION
System.Boolean	True if the given day is Today, false otherwise.

IsTomorrow(DateTime)

Checks whether the given day is Tomorrow.

Declaration

```
public static bool IsTomorrow(this DateTime date)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.

Returns

TYPE	DESCRIPTION
System.Boolean	True if the given day is Tomorrow, false otherwise.

IsYesterday(DateTime)

Checks whether the given day is Yesterday.

Declaration

```
public static bool IsYesterday(this DateTime date)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.

Returns

TYPE	DESCRIPTION
System.Boolean	True if the given day is yesterday, false otherwise.

Midnight(DateTime)

Returns the date as midnight

Declaration

```
public static DateTime Midnight(this DateTime date)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.

Returns

TYPE	DESCRIPTION
System.DateTime	

ToDdMmYyDot(DateTime)

Formats the given DateTime to "dd.MM.yy".

Declaration

```
public static string ToDdMmYyDot(this DateTime date)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.

Returns

TYPE	DESCRIPTION
System.String	The string representation according to the format.

ToDdMmYyHyphen(DateTime)

Formats the given DateTime to "dd-MM-yy".

Declaration

```
public static string ToDdMmYyHyphen(this DateTime date)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.

Returns

TYPE	DESCRIPTION
System.String	The string representation according to the format.

ToDdMmYySlash(DateTime)

Formats the given DateTime to "dd/MM/yy".

Declaration

```
public static string ToDdMmYySlash(this DateTime date)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.

Returns

TYPE	DESCRIPTION
System.String	The string representation according to the format.

ToDdMmYyWithSep(DateTime, String)

Formats the given DateTime to "ddMMyy" by applying the given separator.

Declaration

```
public static string ToDdMmYyWithSep(this DateTime date, string separator)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.
System.String	separator	The given separator.

Returns

TYPE	DESCRIPTION
System.String	The string representation according to the format.

ToDdMmYyyyDot(DateTime)

Formats the given DateTime to "dd.MM.yyyy".

Declaration


```
public static string ToDdMmYyyyDot(this DateTime date)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.

Returns

TYPE	DESCRIPTION
System.String	The string representation according to the format.

ToDdMmYyyyHyphen(DateTime)

Formats the given DateTime to "dd-MM-yyyy".

Declaration

```
public static string ToDdMmYyyyHyphen(this DateTime date)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.

Returns

TYPE	DESCRIPTION
System.String	The string representation according to the format.

ToDdMmYyyySlash(DateTime)

Formats the given DateTime to "dd/MM/yyyy".

Declaration

```
public static string ToDdMmYyyySlash(this DateTime date)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
System.String	The string representation according to the format.

ToDdMmYyyyWithSep(DateTime, String)

Formats the given DateTime to "ddMMyyyy" by applying the given separator.

Declaration

```
public static string ToDdMmYyyyWithSep(this DateTime date, string separator)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.
System.String	separator	The preferred separator.

Returns

TYPE	DESCRIPTION
System.String	The string representation according to the format.

Tomorrow(DateTime)

Returns Tomorrow's date (keeps the time)

Declaration

```
public static DateTime Tomorrow(this DateTime date)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	

Returns

TYPE	DESCRIPTION
System.DateTime	

TomorrowMidnight(DateTime)

Returns Tomorrow's date at midnight

Declaration

```
public static DateTime TomorrowMidnight(this DateTime date)
```

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	

Returns

TYPE	DESCRIPTION
System.DateTime	

Yesterday(DateTime)

Returns yesterday's date (keeps the time)

Declaration

<pre>public static DateTime Yesterday(this DateTime date)</pre>

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.

Returns

TYPE	DESCRIPTION
System.DateTime	

YesterdayMidnight(DateTime)

Returns yesterday's date at midnight

Declaration

<pre>public static DateTime YesterdayMidnight(this DateTime date)</pre>

Parameters

TYPE	NAME	DESCRIPTION
System.DateTime	date	Date.

Returns

TYPE	DESCRIPTION
System.DateTime	

Class DictionaryExtensctions

Inheritance

System.Object
DictionaryExtensctions

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Extensions](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class DictionaryExtensctions
```

Methods

AddIfKeyNotPresent<TKey, TValue>(Dictionary<TKey, TValue>, TKey, TValue)

Adds the given key and value to the dictionary, if the key is not already present in the dictionary. Returns true if the key-value was added, false otherwise.

Declaration

```
public static bool AddIfKeyNotPresent<TKey, TValue>(this Dictionary<TKey, TValue> dict, TKey key, TValue value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<TKey, TValue>	dict	Dictionary.
TKey	key	The key to be added.
TValue	value	The value to be added.

Returns

TYPE	DESCRIPTION
System.Boolean	

Type Parameters

NAME	DESCRIPTION

NAME	DESCRIPTION
TKey	Key Type.
TValue	Value Type.

AddOrUpdate<TKey, TValue>(Dictionary<TKey, TValue>, TKey, TValue)

If the key is not present in the dictionary, it is added with the specified value. Otherwise, its value is changed to the one specified here.

Declaration

```
public static void AddOrUpdate<TKey, TValue>(this Dictionary<TKey, TValue> dict, TKey key, TValue value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<TKey, TValue>	dict	Dictionary.
TKey	key	The key to be added or updated.
TValue	value	The value for the key.

Type Parameters

NAME	DESCRIPTION
TKey	Key type.
TValue	Value type.

TryAddKey<TKey, TValue>(Dictionary<TKey, TValue>, TKey, TValue)

Tries to add the key-value to the dictionary. Returns true if successful, false otherwise.

This method is just a wrapper for the `AddIfKeyNotPresent` renamed to match the native TryGetValue().

Declaration

```
public static bool TryAddKey<TKey, TValue>(this Dictionary<TKey, TValue> dict, TKey key, TValue value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<TKey, TValue>	dict	Dictionary.

TYPE	NAME	DESCRIPTION
TKey	key	The key to be added.
TValue	value	The value to be added.

Returns

TYPE	DESCRIPTION
System.Boolean	

Type Parameters

NAME	DESCRIPTION
TKey	Key Type.
TValue	Value Type.

Class FloatExtensions

Inheritance

System.Object
FloatExtensions

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Extensions](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class FloatExtensions
```

Methods

Round(Single)

Returns the float rounded to the nearest integer.

Declaration

```
public static float Round(this float f)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	f	

Returns

TYPE	DESCRIPTION
System.Single	

Round(Single, Int32)

Returns a float rounded up to the set number of decimals.

Declaration

```
public static float Round(this float f, int decimals = 1)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	f	
System.Int32	decimals	

Returns

TYPE	DESCRIPTION
System.Single	

Class IListExtensions

Inheritance

System.Object
IListExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Extensions](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class IListExtensions
```

Methods

IsNullOrEmpty<T>(IList<T>)

Returns true if this list is null or empty.

Declaration

```
public static bool IsNullOrEmpty<T>(this IList<T> items)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IList<T>	items	

Returns

TYPE	DESCRIPTION
System.Boolean	

Type Parameters

NAME	DESCRIPTION
T	

NotNullOrEmpty<T>(IList<T>)

Returns true if this list is NOT null or empty.

Declaration

```
public static bool NotNullOrEmpty<T>(this IList<T> items)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IList<T>	items	

Returns

TYPE	DESCRIPTION
System.Boolean	

Type Parameters

NAME	DESCRIPTION
T	

Shuffle<T>(IList<T>)

Shuffle the list.

Declaration

```
public static void Shuffle<T>(this IList<T> list)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IList<T>	list	

Type Parameters

NAME	DESCRIPTION
T	

Class RectTransformExtensions

Inheritance

System.Object
RectTransformExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Extensions](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class RectTransformExtensions
```

Methods

GetHeight(RectTransform)

Declaration

```
public static float GetHeight(this RectTransform trans)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	trans	

Returns

TYPE	DESCRIPTION
System.Single	

GetSize(RectTransform)

Declaration

```
public static Vector2 GetSize(this RectTransform trans)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	trans	

Returns

TYPE	DESCRIPTION
Vector2	

GetWidth(RectTransform)

Declaration

```
public static float GetWidth(this RectTransform trans)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	trans	

Returns

TYPE	DESCRIPTION
System.Single	

SetAnchors(RectTransform, Vector2)

Declaration

```
public static void SetAnchors(this RectTransform trans, Vector2 aVec)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	trans	
Vector2	aVec	

SetDefaultScale(RectTransform)

Declaration

```
public static void SetDefaultScale(this RectTransform trans)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	trans	

SetHeight(RectTransform, Single)

Declaration

```
public static void SetHeight(this RectTransform trans, float newSize)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	trans	
System.Single	newSize	

SetLeftBottomPosition(RectTransform, Vector2)

Declaration

```
public static void SetLeftBottomPosition(this RectTransform trans, Vector2 newPos)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	trans	
Vector2	newPos	

SetLeftTopPosition(RectTransform, Vector2)

Declaration

```
public static void SetLeftTopPosition(this RectTransform trans, Vector2 newPos)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	trans	
Vector2	newPos	

SetPivotAndAnchors(RectTransform, Vector2)

Declaration

```
public static void SetPivotAndAnchors(this RectTransform trans, Vector2 aVec)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	trans	
Vector2	aVec	

SetPositionOfPivot(RectTransform, Vector2)

Declaration

```
public static void SetPositionOfPivot(this RectTransform trans, Vector2 newPos)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	trans	
Vector2	newPos	

SetRightBottomPosition(RectTransform, Vector2)

Declaration

```
public static void SetRightBottomPosition(this RectTransform trans, Vector2 newPos)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	trans	
Vector2	newPos	

SetRightTopPosition(RectTransform, Vector2)

Declaration

```
public static void SetRightTopPosition(this RectTransform trans, Vector2 newPos)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	trans	
Vector2	newPos	

SetSize(RectTransform, Vector2)

Declaration

```
public static void SetSize(this RectTransform trans, Vector2 newSize)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	trans	
Vector2	newSize	

SetWidth(RectTransform, Single)

Declaration

```
public static void SetWidth(this RectTransform trans, float newSize)
```

Parameters

TYPE	NAME	DESCRIPTION
RectTransform	trans	
System.Single	newSize	

Class StringExtensions

Inheritance

System.Object
StringExtensions

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Extensions](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class StringExtensions
```

Methods

Contains(String, String, StringComparison)

Declaration

```
public static bool Contains(this string source, string toCheck, StringComparison comp)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	source	
System.String	toCheck	
System.StringComparison	comp	

Returns

TYPE	DESCRIPTION
System.Boolean	

ContainsNumeric(String)

Returns true if the string contains a numeric sequence, false otherwise.

Declaration

```
public static bool ContainsNumeric(this string str)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.String	str	The string.

Returns

TYPE	DESCRIPTION
System.Boolean	

FixNewLine(String)

Declaration

```
public static string FixNewLine(this string s)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	s	

Returns

TYPE	DESCRIPTION
System.String	

GetAccent(String)

Declaration

```
public static char GetAccent(this string stIn)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	stIn	

Returns

TYPE	DESCRIPTION
System.Char	

IsDiacriticsed(String)

Declaration

```
public static bool IsDiacriticsed(this string stIn)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	stIn	

Returns

TYPE	DESCRIPTION
System.Boolean	

IsNullOrEmpty(String)

Extension for string.IsNullOrEmpty(). Returns true if the string is null or empty, false otherwise.

Declaration

```
public static bool IsNullOrEmpty(this string str)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	str	

Returns

TYPE	DESCRIPTION
System.Boolean	

IsNumeric(String)

Returns true if the entire string is numeric, false otherwise.

Declaration

```
public static bool IsNumeric(this string str)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	str	The string.

Returns

TYPE	DESCRIPTION
System.Boolean	

NthIndexOf(String, String, Int32)

Declaration

```
public static int NthIndexOf(this string target, string value, int n)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	target	
System.String	value	

TYPE	NAME	DESCRIPTION
System.Int32	n	

Returns

TYPE	DESCRIPTION
System.Int32	

OccurenceCount(String, String)

Declaration

```
public static int OccurenceCount(this string str, string val)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	str	
System.String	val	

Returns

TYPE	DESCRIPTION
System.Int32	

RemoveDiacritics(String)

Declaration

```
public static string RemoveDiacritics(this string stIn)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	stIn	

Returns

TYPE	DESCRIPTION
System.String	

Split(String, String)

Declaration

```
public static string[] Split(this string s, string separator)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	s	

TYPE	NAME	DESCRIPTION
System.String	separator	

Returns

TYPE	DESCRIPTION
System.String[]	

StripTagsCharArray(String)

Declaration

```
public static string StripTagsCharArray(this string source)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	source	

Returns

TYPE	DESCRIPTION
System.String	

StripTagsRegex(String)

Remove HTML from string with Regex.

Declaration

```
public static string StripTagsRegex(this string source)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	source	

Returns

TYPE	DESCRIPTION
System.String	

ToTitleCase(String)

Converts the specified string to title case (except for words that are entirely in uppercase, which are considered to be acronyms).

Declaration

```
public static string ToTitleCase(this string str)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	str	

Returns

TYPE	DESCRIPTION
System.String	

UnPascalCase(String)

A simple extension method based on Binary Worrier's code which will handle acronyms properly, and is repeatable (won't mangle already spaced words).

Declaration

```
public static string UnPascalCase(this string text)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	text	

Returns

TYPE	DESCRIPTION
System.String	

Class TransformExtensions

Inheritance

System.Object
TransformExtensions

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Extensions](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class TransformExtensions
```

Methods

FlipPostive(Transform)

Sets all scale values to the absolute values.

Declaration

```
public static void FlipPostive(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

FlipX(Transform)

Negates the X scale.

Declaration

```
public static void FlipX(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

FlipXY(Transform)

Negates the X and Y scale.

Declaration

```
public static void FlipXY(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

FlipXYZ(Transform)

Negates the X, Y and Z scale.

Declaration

```
public static void FlipXYZ(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

FlipXZ(Transform)

Negates the X and Z scale.

Declaration

```
public static void FlipXZ(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

FlipY(Transform)

Negates the Y scale.

Declaration

```
public static void FlipY(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

FlipYZ(Transform)

Negates the Y and Z scale.

Declaration

```
public static void FlipYZ(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

FlipZ(Transform)

Negates the Z scale.

Declaration

```
public static void FlipZ(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

Reset(Transform)

Resets position, rotation and scale.

Declaration

```
public static void Reset(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

ResetLocal(Transform)

Resets localPosition, localRotation and localScale.

Declaration

```
public static void ResetLocal(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

ResetLocalPosition(Transform)

Sets the local position to 0, 0, 0.

Declaration

```
public static void ResetLocalPosition(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

ResetLocalRotation(Transform)

Resets the local rotation to 0, 0, 0

Declaration

```
public static void ResetLocalRotation(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

ResetPosition(Transform)

Sets the position to 0, 0, 0.

Declaration

```
public static void ResetPosition(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

ResetRotation(Transform)

Resets the rotation to 0, 0, 0

Declaration

```
public static void ResetRotation(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

ResetScale(Transform)

Resets the local scale of this transform in to Vector3.one.

Declaration

```
public static void ResetScale(this Transform transform)
```


Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

RotateAroundX(Transform, Single)

Rotates the transform around the X axis.

Declaration

```
public static void RotateAroundX(this Transform transform, float angle)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	angle	Rotation angle.

RotateAroundY(Transform, Single)

Rotates the transform around the Y axis.

Declaration

```
public static void RotateAroundY(this Transform transform, float angle)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	angle	Rotation angle.

RotateAroundZ(Transform, Single)

Rotates the transform around the Z axis.

Declaration

```
public static void RotateAroundZ(this Transform transform, float angle)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

TYPE	NAME	DESCRIPTION
System.Single	angle	Rotation angle.

ScaleByX(Transform, Single)

Multiply the transform's X scale by the given amount.

Declaration

```
public static void ScaleByX(this Transform transform, float x)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	X scale multiplier.

ScaleByXY(Transform, Single)

Scale this transform in the X and Y directions.

Declaration

```
public static void ScaleByXY(this Transform transform, float r)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	r	X and Y scale multiplier.

ScaleByXY(Transform, Single, Single)

Scale this transform in the X, Y direction.

Declaration

```
public static void ScaleByXY(this Transform transform, float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

TYPE	NAME	DESCRIPTION
System.Single	x	X scale multiplier.
System.Single	y	Y scale multiplier.

ScaleByXYZ(Transform, Single)

Scale this transform in the X, Y and Z directions.

Declaration

```
public static void ScaleByXYZ(this Transform transform, float r)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	r	X, Y and Z scale multiplier.

ScaleByXYZ(Transform, Single, Single, Single)

Scale this transform in the X, Y and Z directions.

Declaration

```
public static void ScaleByXYZ(this Transform transform, float x, float y, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	X scale multiplier.
System.Single	y	Y scale multiplier.
System.Single	z	Z scale multiplier.

ScaleByXZ(Transform, Single)

Scale this transform in the X and Z directions.

Declaration

```
public static void ScaleByXZ(this Transform transform, float r)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	r	X and Z scale multiplier.

ScaleByXZ(Transform, Single, Single)

Scale this transform in the X, Z directions.

Declaration

```
public static void ScaleByXZ(this Transform transform, float x, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	X scale multiplier.
System.Single	z	Z scale multiplier.

ScaleByY(Transform, Single)

Scale this transform in the Y direction.

Declaration

```
public static void ScaleByY(this Transform transform, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	y	Y scale multiplier.

ScaleByYZ(Transform, Single)

Scale this transform in the Y and Z directions.

Declaration

```
public static void ScaleByYZ(this Transform transform, float r)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	r	Y and Z scale multiplier.

ScaleByYZ(Transform, Single, Single)

Scale this transform in the Y and Z directions.

Declaration

```
public static void ScaleByYZ(this Transform transform, float y, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	y	Y scale multiplier.
System.Single	z	Z scale multiplier.

ScaleByZ(Transform, Single)

Scale this transform in the Z direction.

Declaration

```
public static void ScaleByZ(this Transform transform, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	z	Z scale multiplier.

SetLocalRotationX(Transform, Single)

Sets the local X rotation.

Declaration

```
public static void SetLocalRotationX(this Transform transform, float angle)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	angle	Euler angle X.

SetLocalRotationY(Transform, Single)

Sets the local Y rotation.

Declaration

```
public static void SetLocalRotationY(this Transform transform, float angle)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	angle	Euler angle Y.

SetLocalRotationZ(Transform, Single)

Sets the local Z rotation.

Declaration

```
public static void SetLocalRotationZ(this Transform transform, float angle)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	angle	Euler angle Z.

SetLocalX(Transform, Single)

Sets the local X position of this transform.

Declaration

```
public static void SetLocalX(this Transform transform, float x)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	New localPosition X.

SetLocalXY(Transform, Single, Single)

Sets the local X and Y position of this transform.

Declaration

```
public static void SetLocalXY(this Transform transform, float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	New localPosition X.
System.Single	y	New localPosition Y.

SetLocalXYZ(Transform, Single, Single, Single)

Sets the local X, Y and Z position of this transform.

Declaration

```
public static void SetLocalXYZ(this Transform transform, float x, float y, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	New localPosition X.
System.Single	y	New localPosition Y.
System.Single	z	New localPosition Z.

SetLocalXZ(Transform, Single, Single)

Sets the local X and Z position of this transform.

Declaration

```
public static void SetLocalXZ(this Transform transform, float x, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	New localPosition X.
System.Single	z	New localPosition Z.

SetLocalY(Transform, Single)

Sets the local Y position of this transform.

Declaration

```
public static void SetLocalY(this Transform transform, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	y	New localPosition Y.

SetLocalYZ(Transform, Single, Single)

Sets the local Y and Z position of this transform.

Declaration

```
public static void SetLocalYZ(this Transform transform, float y, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	y	New localPosition Y.
System.Single	z	New localPosition Z.

SetLocalZ(Transform, Single)

Sets the local Z position of this transform.

Declaration

```
public static void SetLocalZ(this Transform transform, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	z	New localPosition Z.

SetRotationX(Transform, Single)

Sets the X rotation.

Declaration

```
public static void SetRotationX(this Transform transform, float angle)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	angle	Euler angle X.

SetRotationY(Transform, Single)

Sets the Y rotation.

Declaration

```
public static void SetRotationY(this Transform transform, float angle)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	angle	Euler angle Y.

SetRotationZ(Transform, Single)

Sets the Z rotation.

Declaration

```
public static void SetRotationZ(this Transform transform, float angle)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	angle	Euler angle Z.

SetScaleX(Transform, Single)

Sets the local X scale of this transform.

Declaration

```
public static void SetScaleX(this Transform transform, float x)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	New X scale.

SetScaleXY(Transform, Single, Single)

Sets the local X and Y scale of this transform.

Declaration

```
public static void SetScaleXY(this Transform transform, float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	New X scale.
System.Single	y	New Y scale.

SetScaleXYZ(Transform, Single, Single, Single)

Sets the local X, Y and Z scale of this transform.

Declaration

```
public static void SetScaleXYZ(this Transform transform, float x, float y, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	New X scale.
System.Single	y	New Y scale.
System.Single	z	New Z scale.

SetScaleXZ(Transform, Single, Single)

Sets the local X and Z scale of this transform.

Declaration

```
public static void SetScaleXZ(this Transform transform, float x, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	New X scale.
System.Single	z	New Z scale.

SetScaleY(Transform, Single)

Sets the local Y scale of this transform.

Declaration

```
public static void SetScaleY(this Transform transform, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.

TYPE	NAME	DESCRIPTION
System.Single	y	New Y scale.

SetScaleYZ(Transform, Single, Single)

Sets the local Y and Z scale of this transform.

Declaration

```
public static void SetScaleYZ(this Transform transform, float y, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	y	New Y scale.
System.Single	z	New Z scale.

SetScaleZ(Transform, Single)

Sets the local Z scale of this transform.

Declaration

```
public static void SetScaleZ(this Transform transform, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	z	New Z scale.

SetX(Transform, Single)

Sets the X position of this transform.

Declaration

```
public static void SetX(this Transform transform, float x)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	New position X.

SetXY(Transform, Single, Single)

Sets the X and Y position of this transform.

Declaration

```
public static void SetXY(this Transform transform, float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	New position X.
System.Single	y	New position Y.

SetXYZ(Transform, Single, Single, Single)

Sets the X, Y and Z position of this transform.

Declaration

```
public static void SetXYZ(this Transform transform, float x, float y, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	New position X.
System.Single	y	New position Y.
System.Single	z	New position Z.

SetXZ(Transform, Single, Single)

Sets the X and Z position of this transform.

Declaration

```
public static void SetXZ(this Transform transform, float x, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	New position X.
System.Single	z	New position Y.

SetY(Transform, Single)

Sets the Y position of this transform.

Declaration

```
public static void SetY(this Transform transform, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The Transform.
System.Single	y	New position Y.

SetYZ(Transform, Single, Single)

Sets the Y and Z position of this transform.

Declaration

```
public static void SetYZ(this Transform transform, float y, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	y	New position Y.
System.Single	z	New position Z.

SetZ(Transform, Single)

Sets the Z position of this transform.

Declaration

```
public static void SetZ(this Transform transform, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	z	New position Z.

TranslateX(Transform, Single)

Translates this transform along the X axis.

Declaration

```
public static void TranslateX(this Transform transform, float x)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	Distance on the X axis.

TranslateXYZ(Transform, Single, Single, Single)

Translates this transform along the X, Y and Z axis.

Declaration

```
public static void TranslateXYZ(this Transform transform, float x, float y, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	x	Distance on the X axis.
System.Single	y	Distance on the Y axis.

TYPE	NAME	DESCRIPTION
System.Single	z	Distance on the Z axis.

TranslateY(Transform, Single)

Translates this transform along the Y axis.

Declaration

```
public static void TranslateY(this Transform transform, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	y	Distance on the Y axis.

TranslateZ(Transform, Single)

Translates this transform along the Z axis.

Declaration

```
public static void TranslateZ(this Transform transform, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Transform	transform	The transform.
System.Single	z	Distance on the Z axis.

Class VectorExtensions

Inheritance

System.Object
VectorExtensions

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Extensions](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class VectorExtensions
```

Methods

AddXY(Vector2, Single, Single)

Adds X to vector.x, adds Y value to vector.y and returns the new Vector2.

Declaration

```
public static Vector2 AddXY(this Vector2 v2, float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector2	v2	The Vector2.
System.Single	x	Value to be added to X.
System.Single	y	Value to be added to Y.

Returns

TYPE	DESCRIPTION
Vector2	

AddXYZ(Vector3, Single, Single, Single)

Adds X to vector.x, adds Y value to vector.y, adds Z to vector.z and returns the new Vector3.

Declaration

```
public static Vector3 AddXYZ(this Vector3 v3, float x, float y, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector3	v3	The Vector3.
System.Single	x	Value to be added to X.
System.Single	y	Value to be added to Y.
System.Single	z	Value to be added to Z.

Returns

TYPE	DESCRIPTION
Vector3	

Proj(Vector2, Vector2)

Returns the projection of this vector onto the given base.

Declaration

```
public static Vector2 Proj(this Vector2 v2, Vector2 base)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector2	v2	The Vector2.
Vector2	base	Base Vector2.

Returns

TYPE	DESCRIPTION
Vector2	

Proj(Vector3, Vector3)

Returns the projection of this vector onto the given base.

Declaration

```
public static Vector3 Proj(this Vector3 v3, Vector3 base)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector3	v3	The Vector3.
Vector3	base	Base Vector3.

Returns

TYPE	DESCRIPTION
Vector3	

Rej(Vector2, Vector2)

Returns the rejection of this vector onto the given base. The sum of a vector's projection and rejection on a base is equal to the original vector.

Declaration

```
public static Vector2 Rej(this Vector2 v2, Vector2 base)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector2	v2	The Vector2.
Vector2	base	Base Vector2.

Returns

TYPE	DESCRIPTION
Vector2	

Rej(Vector3, Vector3)

Returns the rejection of this vector onto the given base. The sum of a vector's projection and rejection on a base is equal to the original vector.

Declaration

```
public static Vector3 Rej(this Vector3 v3, Vector3 base)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector3	v3	The Vector3.
Vector3	base	Base Vector3.

Returns

TYPE	DESCRIPTION
Vector3	

Round(Vector2, Int32)

Returns a Vector2 with rounded values to the set number of decimals.

Declaration

<pre>public static Vector2 Round(this Vector2 v2, int decimals = 1)</pre>

Parameters

TYPE	NAME	DESCRIPTION
Vector2	v2	The Vector2.
System.Int32	decimals	Number of decimals.

Returns

TYPE	DESCRIPTION
Vector2	

Round(Vector3, Int32)

Returns a Vector3 with rounded values to the set number of decimals.

Declaration

<pre>public static Vector3 Round(this Vector3 v3, int decimals = 1)</pre>

Parameters

TYPE	NAME	DESCRIPTION
Vector3	v3	The Vector3.
System.Int32	decimals	Number of decimals.

Returns

TYPE	DESCRIPTION
Vector3	

ScaleSizeBy(Rect, Single)

Declaration

<pre>public static Rect ScaleSizeBy(this Rect rect, float scale)</pre>
--

Parameters

TYPE	NAME	DESCRIPTION
Rect	rect	
System.Single	scale	

Returns

TYPE	DESCRIPTION
Rect	

ScaleSizeBy(Rect, Single, Vector2)

Declaration

```
public static Rect ScaleSizeBy(this Rect rect, float scale, Vector2 pivotPoint)
```

Parameters

TYPE	NAME	DESCRIPTION
Rect	rect	
System.Single	scale	
Vector2	pivotPoint	

Returns

TYPE	DESCRIPTION
Rect	

ScaleSizeBy(Rect, Vector2)

Declaration

```
public static Rect ScaleSizeBy(this Rect rect, Vector2 scale)
```

Parameters

TYPE	NAME	DESCRIPTION
Rect	rect	
Vector2	scale	

Returns

TYPE	DESCRIPTION
Rect	

ScaleSizeBy(Rect, Vector2, Vector2)

Declaration

```
public static Rect ScaleSizeBy(this Rect rect, Vector2 scale, Vector2 pivotPoint)
```

Parameters

TYPE	NAME	DESCRIPTION
Rect	rect	
Vector2	scale	
Vector2	pivotPoint	

Returns

TYPE	DESCRIPTION
Rect	

SetX(Vector2, Single)

Sets a new X value to the vector and returns it.

Declaration

```
public static Vector2 SetX(this Vector2 v2, float x)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector2	v2	The Vector2.
System.Single	x	New X value.

Returns

TYPE	DESCRIPTION
Vector2	

SetX(Vector3, Single)

Sets a new X value to the vector and returns it.

Declaration

```
public static Vector3 SetX(this Vector3 v3, float x)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector3	v3	The Vector3.

TYPE	NAME	DESCRIPTION
System.Single	x	New X value.

Returns

TYPE	DESCRIPTION
Vector3	

SetY(Vector2, Single)

Sets a new Y value to the vector and returns it.

Declaration

```
public static Vector2 SetY(this Vector2 v2, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector2	v2	The Vector2.
System.Single	y	New Y value.

Returns

TYPE	DESCRIPTION
Vector2	

SetY(Vector3, Single)

Sets a new Y value to the vector and returns it.

Declaration

```
public static Vector3 SetY(this Vector3 v3, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector3	v3	The Vector3.
System.Single	y	New Y value.

Returns

TYPE	DESCRIPTION
Vector3	

SetZ(Vector3, Single)

Sets a new Z value to the vector and returns it.

Declaration

```
public static Vector3 SetZ(this Vector3 v3, float z)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector3	v3	The Vector3.
System.Single	z	New Z value.

Returns

TYPE	DESCRIPTION
Vector3	

TopLeft(Rect)

Declaration

```
public static Vector2 TopLeft(this Rect rect)
```

Parameters

TYPE	NAME	DESCRIPTION
Rect	rect	

Returns

TYPE	DESCRIPTION
Vector2	

ToString(Vector2)

Returns a Vector3 to a string in X, Y format.

Declaration

```
public static string ToString(this Vector2 v2)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector2	v2	The Vector2.

Returns

TYPE	DESCRIPTION
System.String	

ToString(Vector2, Int32)

Returns a Vector3 to a string in X, Y format, rounded up to the set number of decimals.

Declaration

```
public static string ToString(this Vector2 v2, int decimals = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector2	v2	The Vector2.
System.Int32	decimals	Number of decimals.

Returns

TYPE	DESCRIPTION
System.String	

ToString(Vector3, Int32)

Returns a Vector3 to a string in X, Y format, rounded up to the set number of decimals.

Declaration

```
public static string ToString(this Vector3 v3, int decimals = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector3	v3	The Vector3.
System.Int32	decimals	Number of decimals.

Returns

TYPE	DESCRIPTION
System.String	

ToStringXY(Vector3)

Returns a Vector3 to a string in X, Y format.

Declaration

```
public static string ToStringXY(this Vector3 v3)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector3	v3	The Vector3.

Returns

TYPE	DESCRIPTION
System.String	

Namespace QuickEngine.IO

Classes

[File](#)

Class File

Inheritance

System.Object
File

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.IO](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class File
```

Methods

CreateDirectory(String)

Creates a Directory at the specified path.

Declaration

```
public static void CreateDirectory(string path)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	path	

Delete(String)

Declaration

```
public static void Delete(string path)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	path	

Exists(String)

Returns true if the file exists at the specified path.

Declaration

```
public static bool Exists(string path)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	path	

Returns

TYPE	DESCRIPTION
System.Boolean	

GetAbsoluteDirectoryPath(String, Boolean)

Searches for the directoryName in all the project's directories and returns the absolute path of the first one it encounters.

Declaration

```
public static string GetAbsoluteDirectoryPath(string directoryName, bool debug = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	directoryName	
System.Boolean	debug	

Returns

TYPE	DESCRIPTION
System.String	

GetDirectories(String)

Returns a DirectoryInfo array of all the directories (subfolders) found at the specified path.

Declaration

```
public static DirectoryInfo[] GetDirectories(string directoryPath)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	directoryPath	

Returns

TYPE	DESCRIPTION
System.IO.DirectoryInfo[]	

GetDirectoriesNames(String)

Returns a string array of all the directories names (subfolders) found at the specified path.

Declaration

```
public static string[] GetDirectoriesNames(string directoryPath)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	directoryPath	

Returns

TYPE	DESCRIPTION
System.String[]	

GetFiles(String)

Returns a FileInfo array of all the files found at the specified path.

Declaration

<pre>public static FileInfo[] GetFiles(string directoryPath)</pre>
--

Parameters

TYPE	NAME	DESCRIPTION
System.String	directoryPath	

Returns

TYPE	DESCRIPTION
System.IO.FileInfo[]	

GetFiles(String, String)

Returns a FileInfo array of all the files, with the given fileExtension, found at the specified path.

Declaration

<pre>public static FileInfo[] GetFiles(string directoryPath, string fileExtension)</pre>
--

Parameters

TYPE	NAME	DESCRIPTION
System.String	directoryPath	
System.String	fileExtension	

Returns

TYPE	DESCRIPTION
System.IO.FileInfo[]	

GetFilesNames(String)

Returns a string array of all the filenames found at the specified path.

Declaration

```
public static string[] GetFileNames(string directoryPath)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	directoryPath	

Returns

TYPE	DESCRIPTION
System.String[]	

GetFileNames(String, String)

Returns a string array of all the filenames, of the files with the given fileExtension, found at the specified path.

Declaration

```
public static string[] GetFileNames(string directoryPath, string fileExtension)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	directoryPath	
System.String	fileExtension	

Returns

TYPE	DESCRIPTION
System.String[]	

GetRelativeDirectoryPath(String)

Searches for the directoryName in all the project's directories and returns the relative path of the first one it encounters.

Declaration

```
public static string GetRelativeDirectoryPath(string directoryName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	directoryName	

Returns

TYPE	DESCRIPTION
System.String	

Move(String, String)

Declaration

```
public static void Move(string sourceFileName, string destFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	sourceFileName	
System.String	destFileName	

Rename(String, String)

Declaration

```
public static void Rename(string sourceFileName, string destFileName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	sourceFileName	
System.String	destFileName	

WriteFile<T>(String, T, Action<FileStream, T>)

Declaration

```
public static void WriteFile<T>(string filePath, T obj, Action<FileStream, T> serializeMethod)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	filePath	
T	obj	
System.Action<System.IO.FileStream, T>	serializeMethod	

Type Parameters

NAME	DESCRIPTION
T	

Namespace QuickEngine.Utils

Classes

[QAssets](#)

[QEmailValidator](#)

[QIPValidator](#)

[QReflection](#)

Class QAssets

Inheritance

System.Object
QAssets

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Utils](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class QAssets
```

Methods

GetScriptableObjectArray<T>(Object[])

Declaration

```
public static T[] GetScriptableObjectArray<T>(Object[] objects)where T : ScriptableObject
```

Parameters

TYPE	NAME	DESCRIPTION
Object[]	objects	

Returns

TYPE	DESCRIPTION
T[]	

Type Parameters

NAME	DESCRIPTION
T	

GetScriptableObjectFromResources<T>(String)

Returns the reference to a ScriptableObject found at the path. The path will consider the Resources folder as the root. So if we have an asset named 'myAsset' under Resources/Data/myAsset, then the path will be "Data/myAsset". Do not add the .asset file extension as it will not work.

Declaration

```
public static Object GetScriptableObjectFromResources<T>(string path)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	path	

Returns

TYPE	DESCRIPTION
Object	

Type Parameters

NAME	DESCRIPTION
T	

GetScriptableObjectsFromResources(String)

Returns all the references to a ScriptableObjects found at the path. The path will consider the Resources folder as the root. So if we have an asset named 'myAsset' under Resources/Data/myAsset, then the path will be "Data/myAsset". Do not add the .asset file extension as it will not work.

Declaration

```
public static Object[] GetScriptableObjectsFromResources(string path)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	path	

Returns

TYPE	DESCRIPTION
Object[]	

Class QEmailValidator

Inheritance

System.Object
QEmailValidator

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Utils](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class QEmailValidator
```

Methods

IsValidEmail(String)

True if email is valid, false otherwise.

Declaration

```
public static bool IsValidEmail(string emailString)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	emailString	Email address.

Returns

TYPE	DESCRIPTION
System.Boolean	

Class QIPValidator

Inheritance

System.Object
QIPValidator

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ToString()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Utils](#)

Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class QIPValidator
```

Methods

IsValidIPAddress(String)

Returns true if the string parameter is a valid IPv4 addrees, false otherwise.

Declaration

```
public static bool IsValidIPAddress(string str)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	str	IP string to be checked.

Returns

TYPE	DESCRIPTION
System.Boolean	

Class QReflection

Inheritance

System.Object
QReflection

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ToString()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [QuickEngine.Utils](#)
Assembly: Assembly-CSharp-firstpass.dll

Syntax

```
public static class QReflection
```

Properties

Assemblies

Declaration

```
public static Assembly[] Assemblies { get; }
```

Property Value

TYPE	DESCRIPTION
System.Reflection.Assembly[]	

AssemblyNames

Declaration

```
public static List<string> AssemblyNames { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

NameSpaceCache

Declaration

```
public static Dictionary<Assembly, List<string>> NameSpaceCache { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Reflection.Assembly, System.Collections.Generic.List<System.String>>	

TypeCache

Declaration

```
public static Dictionary<string, Type> TypeCache { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Type>	

Methods

GetAllAssemblies()

Declaration

```
public static Assembly[] GetAllAssemblies()
```

Returns

TYPE	DESCRIPTION
System.Reflection.Assembly[]	

GetNameSpaces(Assembly)

Declaration

```
public static List<string> GetNameSpaces(Assembly assembly)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.Assembly	assembly	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

GetQualifiedName(String, String)

Declaration

```
public static string GetQualifiedName(string name, string namespace = "")
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	
System.String	namespace	

Returns

TYPE	DESCRIPTION
System.String	

GetSingletonField(Type, String, String, Boolean, BindingFlags)

Declaration

```
public static object GetSingletonField(Type type, string singletonName, string fieldName, bool singletonIsProperty = true, BindingFlags singletonFlags = BindingFlags.Static | BindingFlags.Public)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	type	
System.String	singletonName	
System.String	fieldName	
System.Boolean	singletonIsProperty	
System.Reflection.BindingFlags	singletonFlags	

Returns

TYPE	DESCRIPTION
System.Object	

GetSingletonInstance(Type, String, Boolean, BindingFlags)

Declaration

```
public static object GetSingletonInstance(Type type, string singletonName, bool singletonIsProperty, BindingFlags flags = BindingFlags.Static | BindingFlags.Public)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	type	
System.String	singletonName	
System.Boolean	singletonIsProperty	
System.Reflection.BindingFlags	flags	

Returns

TYPE	DESCRIPTION
System.Object	

GetSingletonProperty(Type, String, String, Boolean, BindingFlags)

Declaration

```
public static object GetSingletonProperty(Type type, string singletonName, string propertyName, bool singletonIsProperty = true, BindingFlags singletonFlags = BindingFlags.Static | BindingFlags.Public)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	type	
System.String	singletonName	
System.String	propertyName	
System.Boolean	singletonIsProperty	
System.Reflection.BindingFlags	singletonFlags	

Returns

TYPE	DESCRIPTION
System.Object	

GetType(String, String)

Declaration

```
public static Type GetType(string name, string namespace = "")
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	
System.String	namespace	

Returns

TYPE	DESCRIPTION
System.Type	

GetTypeByQualifiedName(String)

Declaration

```
public static Type GetTypeByQualifiedName(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	

Returns

TYPE	DESCRIPTION
System.Type	

PrintManifestResources()

Declaration

```
public static void PrintManifestResources()
```

SetSingletonField(Type, String, String, Object, Boolean, BindingFlags)

Declaration

```
public static bool SetSingletonField(Type type, string singletonName, string fieldName, object value, bool singletonIsProperty = true, BindingFlags singletonFlags = BindingFlags.Static | BindingFlags.Public)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	type	
System.String	singletonName	
System.String	fieldName	
System.Object	value	
System.Boolean	singletonIsProperty	
System.Reflection.BindingFlags	singletonFlags	

Returns

TYPE	DESCRIPTION
System.Boolean	

SetSingletonProperty(Type, String, String, Object, Boolean, BindingFlags)

Declaration

```
public static bool SetSingletonProperty(Type type, string singletonName, string propertyName, object value, bool singletonIsProperty = true, BindingFlags singletonFlags = BindingFlags.Static | BindingFlags.Public)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	type	
System.String	singletonName	
System.String	propertyName	
System.Object	value	

TYPE	NAME	DESCRIPTION
System.Boolean	singletonIsProperty	
System.Reflection.BindingFlags	singletonFlags	

Returns

TYPE	DESCRIPTION
System.Boolean	