OWNER'S MANUAL

UI BUTTON

STANDALONE

3.0

UI BUTTON.ANIMATOR

UI BUTTON.SOUND

UI BUTTON.EFFECT

DOOZY UI

Contents

Thank you for buying our asset and for supporting its further development. This plugin was created to extend the functionality of Unity's native UI system. Should you need help, find issues or have any suggestions, don't hesitate to send us a message at support@doozyentertainment.com.

Please read the quick setup guide before you start using this asset.

Solutions – UIButton solves the following problems

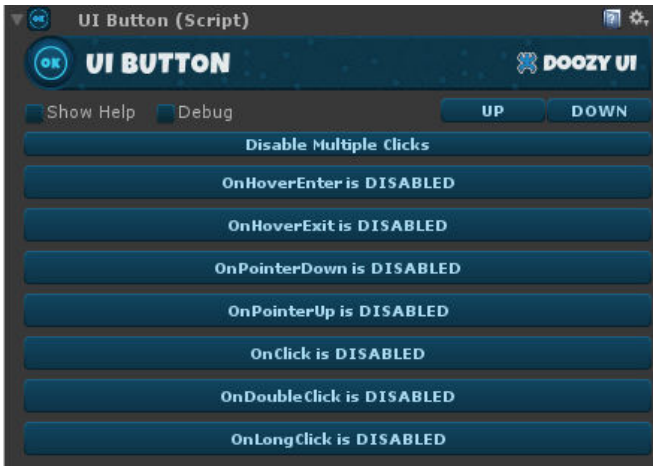
- When using Unity's native Button component you may find yourself needing a few more options than the basic ones it provides. This asset comes with most of the missing options and makes them readily available to you.
- Every action should have a reaction, thus when you interact with a button you should be able to hear a sound (auditive reaction), trigger an animation (visual reaction) and/or an effect (visual reaction).
- Also, for every button state: OnHoverEnter, OnHoverExit, OnPointerDown, OnPointerUp, OnClick, OnDoubleClick and OnLongClick – you can trigger one or more custom methods through the help of UnityEvents. You can do that directly from the Inspector (persistent listener) or from code (non-persistent listener).

Quick Setup Guide

1. Import DOTween (free asset found @UnityAssetStore)
2. Setup DOTween (Tools/Demigiant/DOTween Utility Panel → Setup DOTween…)
3. Import UIButton (from @UnityAssetStore)
4. Watch the introduction video @YouTube - https://youtu.be/c2_ItexaG4o

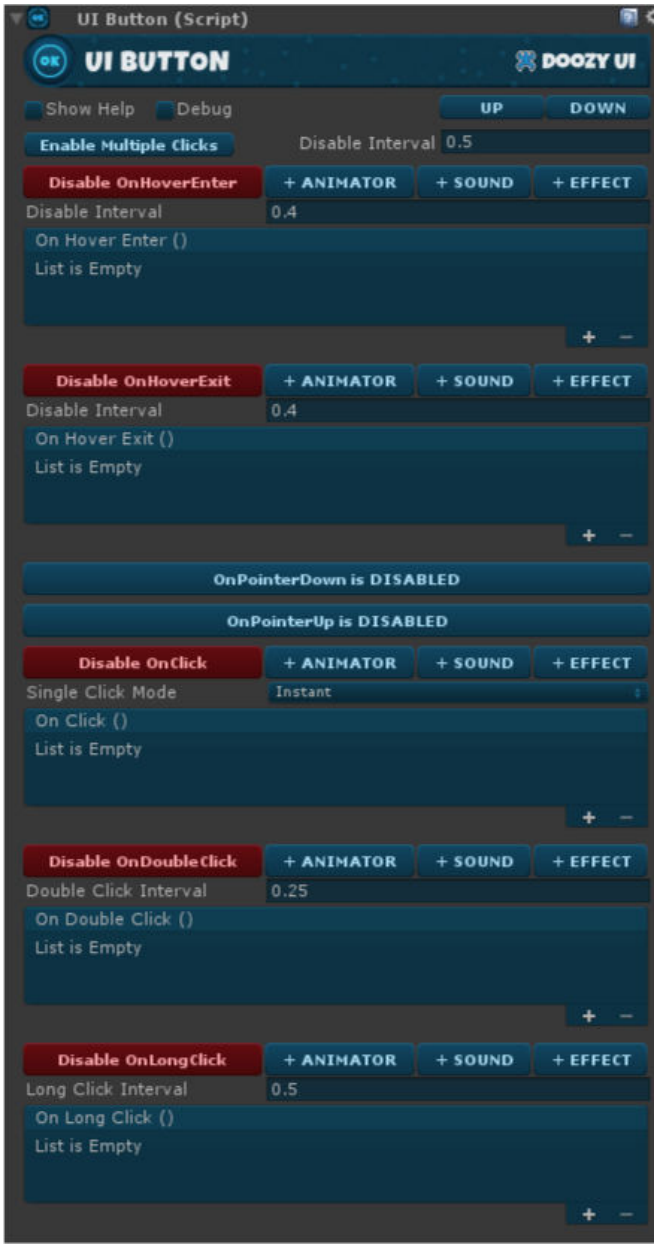
NOTE: Please follow the above steps and import&setup DOTween before importing our asset to avoid any missing methods issues.

UIButton Component



Show Help: shows inspector tooltips on each setting
Debug: prints to Debug.Log all the relevant functionality informations needed for debug purposes
Up/Down: Moves the component UP or DOWN in the Inspector.

Disable Multiple Clicks: disables mulitple clicks for this button by disabling it for a set disable interval
OnHoverEnter is DISABLED: enables the OnHoverEnter trigger
OnHoverExit is DISABLED: enables the OnHoverExit trigger
OnPointerDown is DISABLED: enables the OnPointerDown trigger
OnPointerUp is DISABLED: enables the OnPointerUp trigger
OnClick is DISABLED: enables the OnClick trigger
OnDoubleClick is DISABLED: enables the OnDouble Click trigger
OnLongClick is DISABLED: enables the OnLongClick trigger



Enable Multiple Clicks: enables mulitple clicks for this button
Disable Interval: how long will the button get disabled after each click
OnHoverEnter - Disable Interval: the time interval between two OnHoverEnter event triggers. This disables the OnHoverEnter for a set time interval
OnHoverExit - Disable Interval: the time interval between two OnHoverExit event triggers. This disables the OnHoverExit for a set time interval
OnClick – Single Click Mode – Instant: the click will get registered instantly without checking if it's a double click or not. This is the normal behaviour of a single click in any OS. Use this if you want to make sure a single click will get executed before a double click (dual actions). (usage example: SingleClick - selects, DoubleClick - executes an action)
OnClick – Single Click Mode – Delayed: the click will get registered after checking if it's a double click or not. If it's a double click, the single click will not get triggered. Use this if you want to make sure the user does not execute a single click before a double click. The donwside is that there is a delay when executing the single click (the delay is the doulbe click register interval), so make sure you take that into account
OnDoubleClick – Double Click Interval: the time inverval between two sequential clicks that makes them count as a DoubleClick
OnLongClick – Long Click Interval: the time interval the button has to be pressed down in order to trigger a LongClick

+ANIMATOR: Adds an UIButton.Animator component, to the selected gameObject, that is linked to the source trigger
-ANIMATOR: Remones the UIButton.Animator component , from the selected gameObjectm that is linked to the source trigger

+SOUND: Adds an UIButton.Sound component, to the selected gameObject, that is linked to the source trigger
-SOUND: Remones the UIButton.Sound component , from the selected gameObjectm that is linked to the source trigger

+EFFECT: Adds an UIButton.Effect component, to the selected gameObject, that is linked to the source trigger
-EFFECT: Remones the UIButton.Effect component , from the selected gameObjectm that is linked to the source trigger

UIButton.Animator Component

Show Help: shows inspector tooltips on each setting
Debug: prints to Debug.Log all the relevant functionality informations needed for debug purposes
Up/Down: Moves the component UP or DOWN in the Inspector.
Disable Multiple Clicks: disables mulitple clicks for this button by disabling it for a set disable interval

PRESETS - All the presets are saved as .dat files. All the categories are subfolder names found under UIButton/Resources/Presets/
Category: Select/Create a preset category
Preset Name: Select/Create a preset

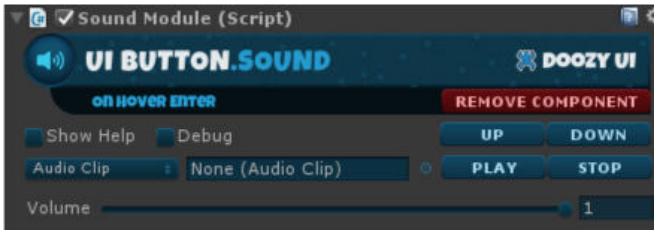
PUNCH MOVE - Punches a RectTransform's anchoredPosition towards the given direction and then back to the starting one as if it was connected to the starting position via an elastic
PUNCH ROTATE - Punches a Transform's localRotation towards the given size and then back to the starting one as if it was connected to the starting rotation via an elastic
PUNCH SCALE - 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

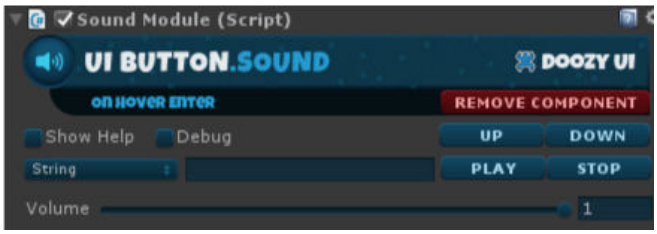
Punch: The direction and strength of the punch (added to the Transform's current position/rotation/scale)
Snap: If TRUE the tween will smoothly snap all values to integers
Duration: The duration of the tween
Delay: Start delay for the tween
Vibrato: Indicates how much will the punch vibrate
Elasticity: Represents how much (0 to 1) the vector will go beyond the starting position/rotation/scale when bouncing backwards. 1 creates a full oscillation between the punch direction and the opposite direction, while 0 oscillates only between the punch and the start position/rotation/scale

MOVE - Moves the button to a target position
ROTATE - Rotates the button to a target rotation
SCALE - Scales the button to a target scale
FADE - Fade the button to a target alpha value

To: Target position/rotation/scale/alpha vale
Snap: If TRUE the tween will smoothly snap all values to integers
Relative: If TRUE sets the tween as relative (the endValue will be calculated as startValue + endValue instead of being used directly)
Duration: The duration of the tween
Delay: Start delay for the tween
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
Reverse Animation?: Should there be another tween that reverts to the initial values
rDuration: The duration of the reverse tween
rDelay: Start delay for the revese tween
rEase: The ease of the reverse tween

UIButton.Sound Component

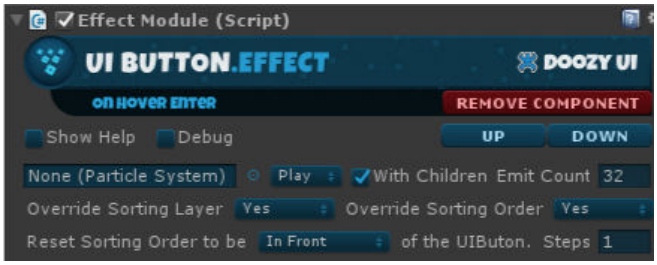




Show Help: shows inspector tooltips on each setting
Debug: prints to Debug.Log all the relevant functionality informations needed for debug purposes
Up/Down: Moves the component UP or DOWN in the Inspector.
Disable Multiple Clicks: disables mulitple clicks for this button by disabling it for a set disable interval

AudioClip: Reference to a sound clip that you would like to play when this event is fired
String: A sound filename (without the extension .wav or .mp3 or any other sound format) that you would like to play when this event is fired. Note that the file should be located under a Resources folder
Play/Stop: Plays/Stops the referenced sound file
Volume: The volume of the sound

UIButton.Effect Component



Show Help: shows inspector tooltips on each setting
Debug: prints to Debug.Log all the relevant functionality informations needed for debug purposes
Up/Down: Moves the component UP or DOWN in the Inspector.
Disable Multiple Clicks: disables mulitple clicks for this button by disabling it for a set disable interval

Particle System: Reference the Particle System you would like to be affected by this button
Effect Action: select if this event should -play-, -stop- or emit a -burst- of particles on trigger
With Children: also trigger the child particle systems under the main one
Emit Count: the number of particles that should be emitted on a -burst-
Clear On Stop: particles dissapear instantly on -stop- if true or or after lifetime if false
Override Sorting Layer: set the particle system's sorting layer the same as the UIButton's
Override Sorting Order: change the particle system's order in layer with the specified number of steps
Reset Sorting Order to be: In Front (+) or Behind (-) of the UIButton with the set number of steps
Steps: the number of steps that the sorting order should be changed to depending on the direction (In Front (+) or Behind (-))

CODE EXAMPLES

UIButton – Variables

`InitialData GetInitialData`	Returns a class that contains the startAnchoredPosition3D, startLocalRotation, startLocalScale and startAlpha for this UIButton
`RectTransform rectTransform`	Reference to the RectTransform component of this UIButton
`Button button`	Reference to the Button component of this UIButton
`bool interactable`	Returns TRUE if the button is interactable and FALSE otherwise

`bool allowMultipleClicks`	Toggles if the UIButton should allow multiple clicks or if it should disable the UIButton after each click for a set time interval. (default:true)
`float disableButtonInterval`	How long will the button get disabled after each click

`bool useOnHoverEnter`	Toggles the active state of this trigger (default:false)
`float onHoverEnterDisableInterval`	The time interval between two OnHoverEnter event triggers. This disables the OnHoverEnter for a set time interval
`Animator.AnimatorModule onHoverEnterAnimatorModule`	Reference to the UIButton.Animator Component linked to this trigger
`Sound.SoundModule onHoverEnterSoundModule`	Reference to the UIButton.Sound Component linked to this trigger
`Effect.EffectModule onHoverEnterEffectModule`	Reference to the UIButton.Effect Component linked to this trigger
`UnityEvent OnHoverEnter`	UnityEvent that is Invoked every time this trigger is fired. You can set non-persistent listeners to it from code and a persistent listeners from the Inspector

`bool useOnHoverExit`	Toggles the active state of this trigger (default:false)
`float onHoverExitDisableInterval`	The time interval between two OnHoverExit event triggers. This disables the OnHoverExit for a set time interval
`Animator.AnimatorModule onHoverExitAnimatorModule`	Reference to the UIButton.Animator Component linked to this trigger
`Sound.SoundModule onHoverExitSoundModule`	Reference to the UIButton.Sound Component linked to this trigger
`Effect.EffectModule onHoverExitEffectModule`	Reference to the UIButton.Effect Component linked to this trigger
`UnityEvent OnHoverExit`	UnityEvent that is Invoked every time this trigger is fired. You can set non-persistent listeners to it from code and a persistent listeners from the Inspector

bool useOnPointerDown	Toggles the active state of this trigger (default:false)
Animator.AnimatorModule onPointerDownAnimatorModule	Reference to the UIButton.Animator Component linked to this trigger
Sound.SoundModule onPointerDownSoundModule	Reference to the UIButton.Sound Component linked to this trigger
Effect.EffectModule onPointerDownEffectModule	Reference to the UIButton.Effect Component linked to this trigger
UnityEvent OnPointerDown	UnityEvent that is Invoked every time this trigger is fired. You can set non-persistent listeners to it from code and a persistent listeners from the Inspector

bool useOnPointerUp	Toggles the active state of this trigger (default:false)
Animator.AnimatorModule onPointerUpAnimatorModule	Reference to the UIButton.Animator Component linked to this trigger
Sound.SoundModule onPointerUpSoundModule	Reference to the UIButton.Sound Component linked to this trigger
Effect.EffectModule onPointerUpEffectModule	Reference to the UIButton.Effect Component linked to this trigger
UnityEvent OnPointerUp	UnityEvent that is Invoked every time this trigger is fired. You can set non-persistent listeners to it from code and a persistent listeners from the Inspector

bool useOnClick	Toggles the active state of this trigger (default:false)
SingleClickMode singleClickMode	Set if the click should get registered instantly without checking if it's a double click or not
Animator.AnimatorModule onClickAnimatorModule	Reference to the UIButton.Animator Component linked to this trigger
Sound.SoundModule onClickSoundModule	Reference to the UIButton.Sound Component linked to this trigger
Effect.EffectModule onClickEffectModule	Reference to the UIButton.Effect Component linked to this trigger
UnityEvent OnClick	UnityEvent that is Invoked every time this trigger is fired. You can set non-persistent listeners to it from code and a persistent listeners from the Inspector

bool useOnDoubleClick	Toggles the active state of this trigger (default:false)
float doubleClickRegisterInterval	The time inverval between two sequential clicks that makes them count as a DoubleClick
Animator.AnimatorModule onDoubleClickAnimatorModule	Reference to the UIButton.Animator Component linked to this trigger
Sound.SoundModule onDoubleClickSoundModule	Reference to the UIButton.Sound Component linked to this trigger
Effect.EffectModule onDoubleClickEffectModule	Reference to the UIButton.Effect Component linked to this trigger
UnityEvent OnDoubleClick	UnityEvent that is Invoked every time this trigger is fired. You can set non-persistent listeners to it from code and a persistent listeners from the Inspector

`bool useOnLongClick`	Toggles the active state of this trigger (default:false)
`float longClickRegisterInterval`	The time interval the button has to be pressed down in order to trigger a LongClick
`Animator.AnimatorModule onLongClickAnimatorModule`	Reference to the UIButton.Animator Component linked to this trigger
`Sound.SoundModule onLongClickSoundModule`	Reference to the UIButton.Sound Component linked to this trigger
`Effect.EffectModule onLongClickEffectModule`	Reference to the UIButton.Effect Component linked to this trigger
`UnityEvent OnLongClick`	UnityEvent that is Invoked every time this trigger is fired. You can set non-persistent listeners to it from code and a persistent listeners from the Inspector

`DisableButtonClicks()`	Disables this button by setting the interactable value to FALSE
`DisableButtonClicks(float time)`	Disables this button by setting the interactable value to FALSE. And it enables it after the set time interval
`EnableButtonClicks()`	Enables this button by setting the interactable value to TRUE
`ExecuteHoverEnter(bool forcedExecution = false)`	Executes the OnHoverEnter trigger. You can force an execution of this trigger (regardless if it's enabled or not) by calling this method with forcedExecution set to TRUE
`ExecuteHoverExit(bool forcedExecution = false)`	Executes the OnHoverExit trigger. You can force an execution of this trigger (regardless if it's enabled or not) by calling this method with forcedExecution set to TRUE
`ExecutePointerDown(bool forcedExecution = false)`	Executes the OnPointerDown trigger. You can force an execution of this trigger (regardless if it's enabled or not) by calling this method with forcedExecution set to TRUE
`ExecutePointerUp(bool forcedExecution = false)`	Executes the OnPointerUp trigger. You can force an execution of this trigger (regardless if it's enabled or not) by calling this method with forcedExecution set to TRUE
`ExecuteClick(bool forcedExecution = false)`	Executes the OnClick trigger. You can force an execution of this trigger (regardless if it's enabled or not) by calling this method with forcedExecution set to TRUE
`ExecuteDoubleClick(bool forcedExecution = false)`	Executes the OnDoubleClick trigger. You can force an execution of this trigger (regardless if it's enabled or not) by calling this method with forcedExecution set to TRUE
`ExecuteLongClick(bool forcedExecution = false)`	Executes the OnLongClick trigger. You can force an execution of this trigger (regardless if it's enabled or not) by calling this method with forcedExecution set to TRUE

`UIButton.ReactTo reactTo`	Selects the UIButton's trigger that this module reacts to. You should not change this in code as you might break the UIButton
`string presetCategoryName`	The preset category name (this is a folder name)
`string presetName`	The preset name (this is a file name)
`bool UsesPreset`	Returns TRUE if this Animator uses a preset and FALSE otherwise

`PunchMove punchMove`	Animation settings for PunchMove
- `bool enabled`	If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default:false)
- `Vector3 punch`	The direction and strength of the punch (added to the Transform's current position)
- `bool snapping`	If TRUE the tween will smoothly snap all values to integers (default:false)
- `float duration`	The duration of the tween
- `float delay`	Start delay for the tween
- `int vibrato`	Indicates how much will the punch vibrate
- `float 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 direction and the opposite direction, while 0 oscillates only between the punch and the start position

`PunchRotate punchRotate`	Animation settings for PunchRotate
- `bool enabled`	If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default:false)
- `Vector3 punch`	The direction and strength of the punch (added to the Transform's current rotation)
- `float duration`	The duration of the tween
- `float delay`	Start delay for the tween
- `int vibrato`	Indicates how much will the punch vibrate
- `float elasticity`	Represents how much (0 to 1) the vector will go beyond the starting rotation when bouncing backwards. 1 creates a full oscillation between the punch rotation and the opposite rotation, while 0 oscillates only between the punch and the start rotation

PunchScale punchScale	Animations settings for PunchScale
- bool enabled	If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default:false)
- Vector3 punch	The punch strength (added to the Transform's current scale)
- float duration	The duration of the tween
- float delay	Start delay for the tween
- int vibrato	Indicates how much will the punch vibrate
- float 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

Move move	Animation settings for Move
- bool enabled	If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default:false)
- Vector3 to	Target position
- bool relative	If TRUE sets the tween as relative (the endValue will be calculated as startValue + endValue instead of being used directly) (default:true)
- bool snapping	If TRUE the tween will smoothly snap all values to integers
- float duration	The duration of the tween
- float delay	Start delay for the tween
- Ease 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
- bool reverseAfterTime	Should there be another tween that reverts to the initial values
- float reverseDuration	The duration of the reverse tween
- float reverseDelay	Start delay for the revese tween
- Ease reverseEase	The ease of the reverse tween

Rotate rotate	Animation settings for Rotate
- bool enabled	If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default:false)
- Vector3 to	Target rotation
- bool relative	If TRUE sets the tween as relative (the endValue will be calculated as startValue + endValue instead of being used directly) (default:true)
- float duration	The duration of the tween
- float delay	Start delay for the tween
- Ease 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
- bool reverseAfterTime	Should there be another tween that reverts to the initial values
- float reverseDuration	The duration of the reverse tween
- float reverseDelay	Start delay for the revese tween
- Ease reverseEase	The ease of the reverse tween

Scale scale	Animation settings for Scale
- bool enabled	If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default:false)
- Vector3 to	Target scale
- bool relative	If TRUE sets the tween as relative (the endValue will be calculated as startValue + endValue instead of being used directly) (default:true)
- float duration	The duration of the tween
- float delay	Start delay for the tween
- Ease 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
- bool reverseAfterTime	Should there be another tween that reverts to the initial values
- float reverseDuration	The duration of the reverse tween
- float reverseDelay	Start delay for the revese tween
- Ease reverseEase	The ease of the reverse tween

`Fade fade`	Animation settings for Fade
- `bool enabled`	If TRUE, this animation will get executed by the Animator when triggered, FALSE otherwise (default:true)
- `float to`	Target fade (alpha value)
- `bool relative`	If TRUE sets the tween as relative (the endValue will be calculated as startValue + endValue instead of being used directly) (default:false)
- `float duration`	The duration of the tween
- `float delay`	Start delay for the tween
- `Ease 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
- `bool reverseAfterTime`	Should there be another tween that reverts to the initial values
- `float reverseDuration`	The duration of the reverse tween
- `float reverseDelay`	Start delay for the revese tween
- `Ease reverseEase`	The ease of the reverse tween

`RectTransform GetRectTransform`	Reference to the RectTransform component of this UIButton
`UIButton GetUIButton`	Reference of the UIButton component that this module reacts to
`bool IsAnimatorModuleEnabled`	Returns TRUE it this module will get triggered by the UIButton and FALSE otherwise

`ExecutePunchMove(bool forcedExecution = false)`	Executes the PunchMove animation. You can force an execution of this animation (regardless if it's enabled or not) by calling this method with forcedExecution set to TRUE
`ExecutePunchRotate(bool forcedExecution = false)`	Executes the PunchRotate animation. You can force an execution of this animation (regardless if it's enabled or not) by calling this method with forcedExecution set to TRUE
`ExecutePunchScale(bool forcedExecution = false)`	Executes the PunchScale animation. You can force an execution of this animation (regardless if it's enabled or not) by calling this method with forcedExecution set to TRUE
`ExecuteMove(bool forcedExecution = false)`	Executes the Move animation. You can force an execution of this animation (regardless if it's enabled or not) by calling this method with forcedExecution set to TRUE
`ExecuteRotate(bool forcedExecution = false)`	Executes the Rotate animation. You can force an execution of this animation (regardless if it's enabled or not) by calling this method with forcedExecution set to TRUE
`ExecuteScale(bool forcedExecution = false)`	Executes the Scale animation. You can force an execution of this animation (regardless if it's enabled or not) by calling this method with forcedExecution set to TRUE
`ExecuteFade(bool forcedExecution = false)`	Executes the Fade animation. You can force an execution of this animation (regardless if it's enabled or not) by calling this method with forcedExecution set to TRUE
`ExecuteAllAnimations()`	Executes all the enabled animations PuchMove, PunchRotate, PunchScale, Move, Rotate, Scale and Fade

UIButton.Sound – Variables

`UIButton.ReactTo reactTo`	Selects the UIButton's trigger that this module reacts to. You should not change this in code as you might break the UIButton
`SoundSource soundSource`	Selected sound source to play
`Sound sound`	Contains the sound references
`float volume`	Returns TRUE if this Animator uses a preset and FALSE otherwise
`RectTransform GetRectTransform`	Reference to the RectTransform component of this UIButton
`UIButton GetUIButton`	Reference of the UIButton component that this module reacts to
`bool IsSoundModuleEnabled`	Returns TRUE it this module will get triggered by the UIButton and FALSE otherwise

UIButton.Sound – Methods

`PlaySound()`	Plays the loaded sound
`AudioClip GetLoadedSound()`	Returns an audioClip from the selected sound source

`UIButton.ReactTo reactTo`	Selects the UIButton's trigger that this module reacts to. You should not change this in code as you might break the UIButton
`ParticleSystem pSystem`	Reference the Particle System you would like to be affected by this module
`EffectAction effectAction`	Select the action this module should execute -play-, -stop- or emit a -burst- of particles on trigger
`bool clearOnStop`	If TRUE, the particles will dissapear instantly on -stop- otherwise they will dissapear after lifetime (default:false)
`bool withChildren`	If TRUE it will also trigger the child particle systems under the main one (default:true)
`int emitCount`	The number of particles that should be emitted on a -burst-
`OverrideSortingLayer overrideSortingLayer`	Set the particle system's sorting layer the same as the UIButton's
`OverrideSortingOrder overrideSortingOrder`	Change the particle system's order in layer with the specified number of steps
`ResetSortingOrderToBe resetSortingOrderToBe`	In Front (+) or Behind (-) of the UIButton with the set number of steps
`int orderInLayerSteps`	The number of steps that the sorting order should be changed to depending on the direction (In Front (+) or Behind (-))
`RectTransform GetRectTransform`	Reference to the RectTransform component of this UIButton
`UIButton GetUIButton`	Reference of the UIButton component that this module reacts to
`bool IsEffectModuleEnabled`	Returns TRUE it this module will get triggered by the UIButton and FALSE otherwise

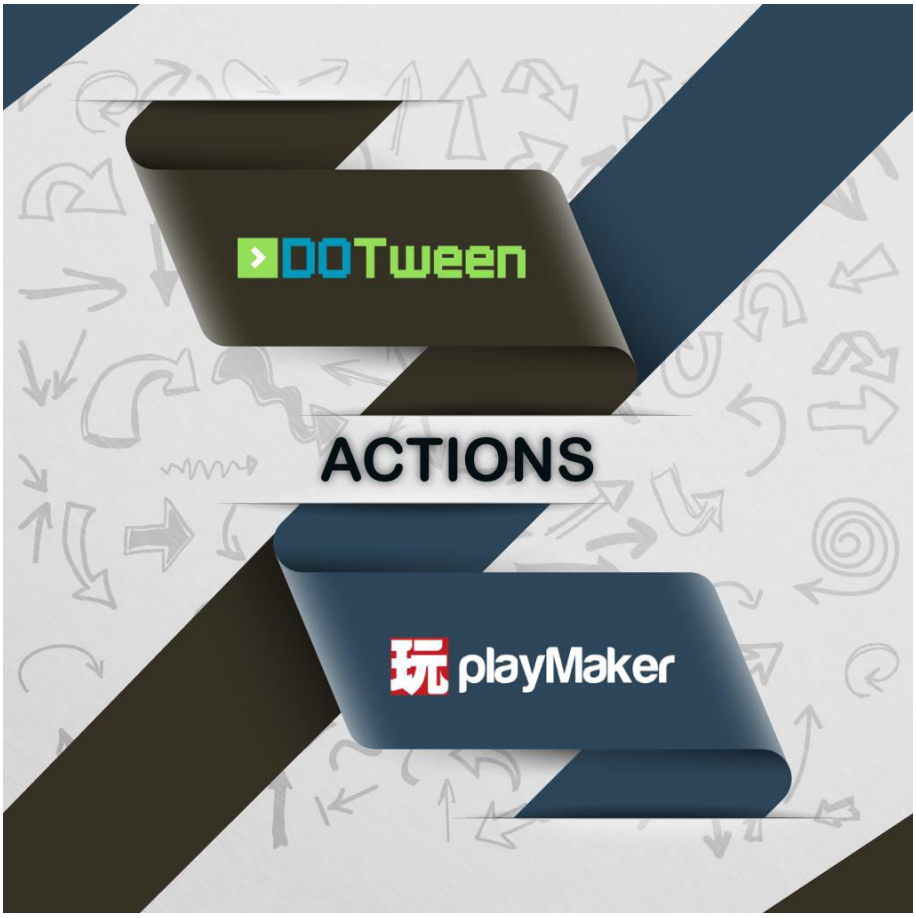
UIButton.Effect – Methods

`ExecutePlay()`	Executes the -play- action on the referenced ParticleSystem
`ExecuteStop()`	Executes the -stop- action on the referenced ParticleSystem
`ExecuteBurst()`	Executes the -burst- action on the referenced ParticleSystem

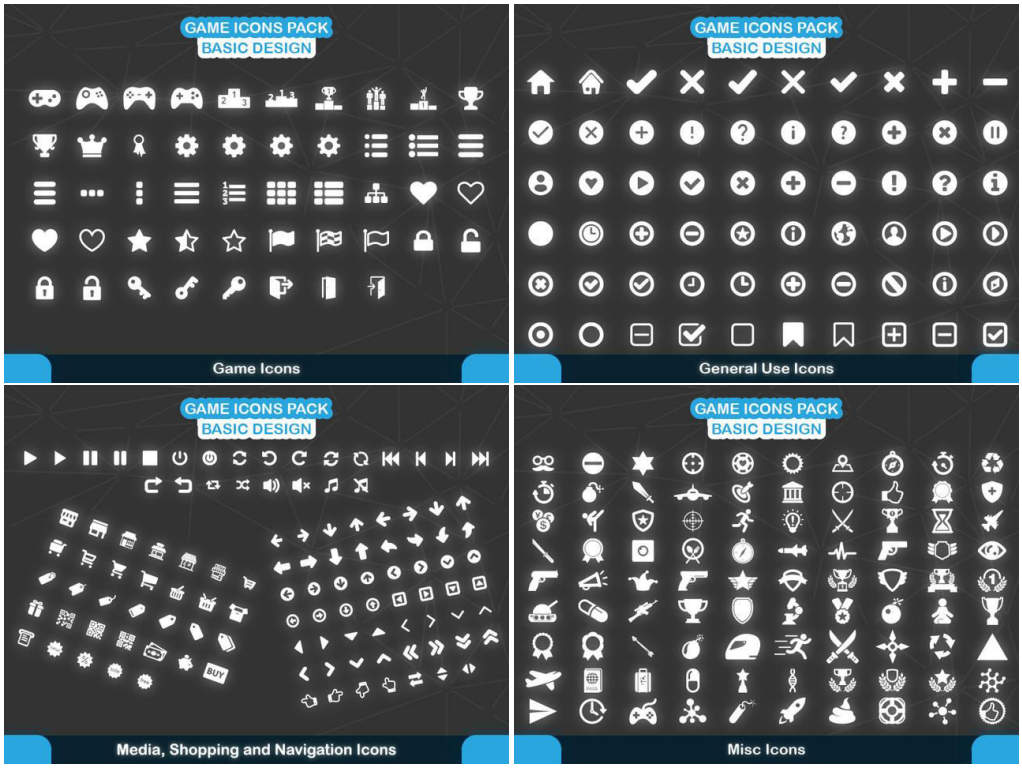
Final Words

- Support is available by emailing support@doozyentertainment.com
- Make sure you check out our other assets such as:
  - DoozyUI: Complete UI Management System - http://u3d.as/k82

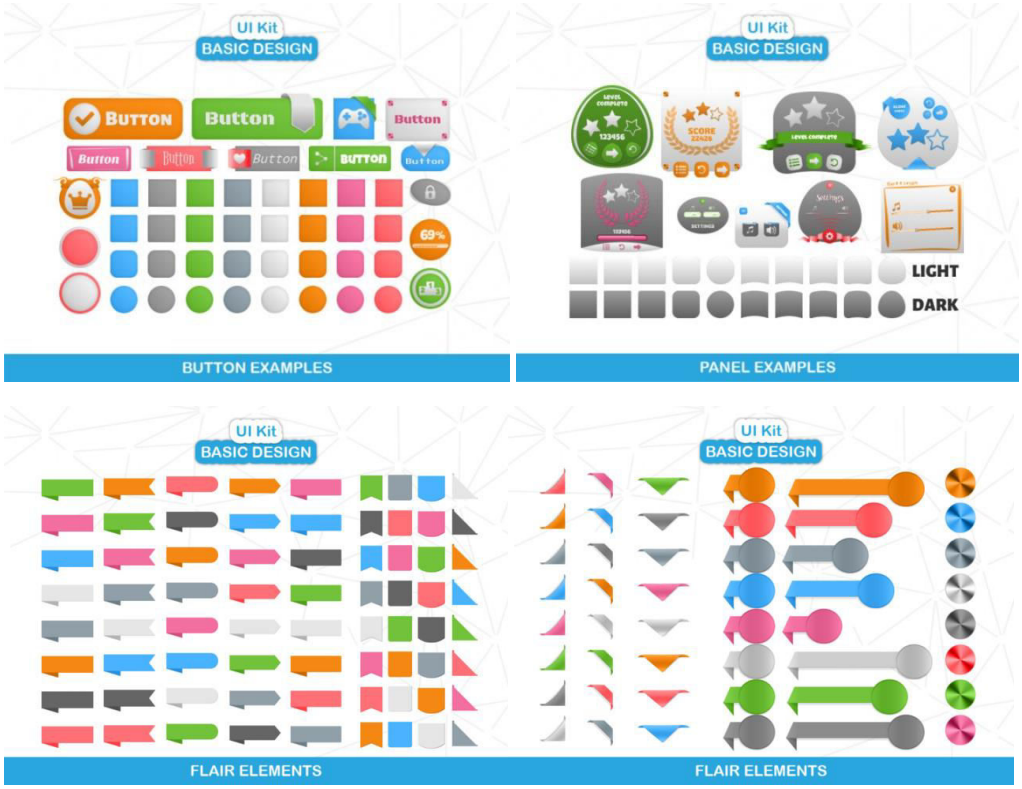
  - Playmaker Actions for DOTween by Doozy - http://u3d.as/kRs



- Game Icons Pack - Basic Design - http://u3d.as/crV

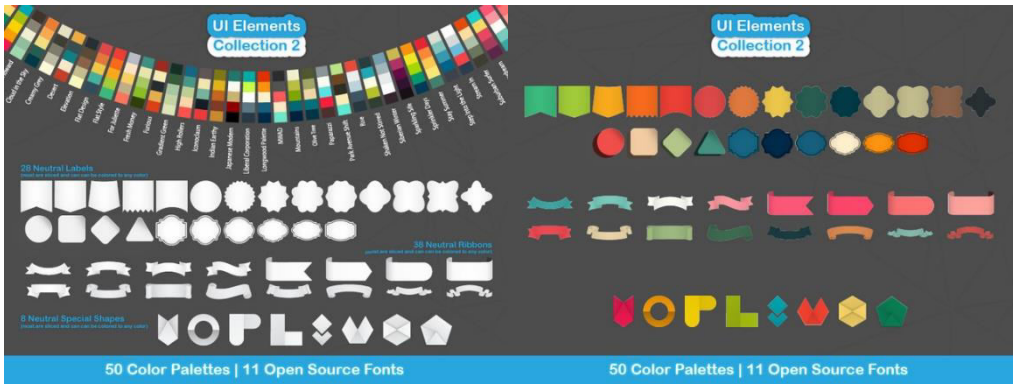


- UI Kit - Basic Design - http://u3d.as/fyv



- UI Elements - Collection 1 - http://u3d.as/g4y

- UI Elements - Collection 2 - http://u3d.as/ghU



- and others - https://goo.gl/kEADpX