

# Welcome to UITween System

This system is only used to animate Unity UI elements with the new integrated UI System (4.6 >)

## A- Editor Setup

Add the “*EasyTween*” class to a GameObject in the scene and set a reference to the UI RectTransform Component in the Editor.

The “*Animation Parts*” class in the editor is responsible for animation properties such :

- 1- Object State : The initial and current Object State (UI element)
- 2- Save State: Reset the Object to its initial Object State
- 3- Atomic Animation: Doesn't blend the Exit animation with the Callbacks Event (the exit animation is usually blended by 10% of the duration)
- 4- End State: What will happen with the object at the end of the Animation
- 5- Call Callback: When and which Callback group will be called
  - a- There are 2 Groups of calls
    - 1- **START\_INITIAL / START\_EXIT** triggers the callback at the very start of the Entry animation (Callback Object Intro) or Exit animation (Callback Object Exit)
    - 2- **END\_OF\_INTRO / END\_OF\_EXIT** triggers the callback at the end of the Entry animation (Callback Object Intro) or Exit animation (Callback Object Exit)
  - b- Every Combination is made with this 2 main group of calls
- 6- Intro Events: Events to trigger at the start or end of the Initial Animation
- 7- Exit Events: Events to trigger at the start or end of the Exit Animation
- 8- Animation Duration (sec): in seconds.

The Editor Class Expands the Object class by adding these features:

- 1- Fade In & Out: On Intro Animations the object fades in, and out on Exit
- 2- Position Animations: Set the Start Move, End Move and types of Easing for each iteration from Start- > End & End -> Start
- 3- Rotation Animations: Set the Start Rotation, End Rotation and types of Easing for each iteration from Start- > End & End -> Start
- 4- Scale Animations: Set the Start Scale, End Scale and types of Easing for each iteration from Start- > End & End -> Start

## B- Dynamic Changes

You can set the “animationParts” class properties dynamically by adding a new “animationParts” class to the [EasyTween](#) object Instance.

Here is the standart class contructor.

```
AnimationParts animationParts = new AnimationParts(  
    AnimationParts.State.CLOSE,           // Objetc State  
    false,                               // Save State  
    false,                               // Atomic Animation  
    AnimationParts.EndTweenClose.Deactivate, // End State  
    AnimationParts.CallbackCall.NOTHING,   // Call Callback  
    new UnityEvent(),                     // Intro Events  
    new UnityEvent());                   // Exit Events
```

In the [EasyTween](#) class you can use the following method to change animation properties:

*+ SetAnimationProperties(animationParts: AnimationParts): void*

For Position / Rotation / Scale / Fade changes, the following methods are available in the [EasyTween](#) object class:

+ [SetAnimationPosition](#)(StartAnchoredPos: [Vector2](#), EndAnchoredPos: [Vector2](#), EntryTween : [AnimationCurve](#), ExitTween : [AnimationCurve](#)): void

+ [SetAnimationScale](#) (StartAnchoredScale: [Vector3](#), EndAnchoredScale: [Vector3](#), EntryTween : [AnimationCurve](#), ExitTween : [AnimationCurve](#)): void

+ [SetAnimationPosition](#)(StartAnchoredEulerAng: [Vector3](#), EndAnchoredEulerAng: [Vector3](#), EntryTween : [AnimationCurve](#), ExitTween : [AnimationCurve](#)): void

+ [SetFade](#)(fade: FadeOverride): void // For fade Override

+ [SetFade](#)(): void