



FlowEnt

[Website](#) [Support](#)

FlowEnt is a tweening library based on the fluent paradigm meant to provide a framework for tweening in a simple or complex fashion but also allowing its users to create their own "motions" based on their needs.

The library is revolving around the concept of reusability and extensibility.

Table of contents

Installation

Usage

[Tweens](#)

[Options](#)

[Events](#)

[Motions](#)

[Flows](#)

Tools

[Settings](#)

[Inspector](#)

Demo



1. Installation

FlowEnt requires no special configuration and can be used straight after installation.

2. Usage

2.1. Tweens

Tweens are the core of animations and they have 3 parts: Options, Events, and Motions.

2.1.1. Options

The tween options are used to create a tween and set specific parameters for the tween.

2.1.2. Events

The events are used to register callbacks for when specific, tween-related events happen.

2.1.3. Motions

Motions are the essence of the animations. They are used to register the animation using the tween parameters

Here's an example of a tween:

```
Tween tween = new Tween(10f)
    .SetLoopType(LoopType.PingPong)
    .SetLoopCount(4)
    .SetDelay(4f)
    .OnStarting(() => Debug.Log("Before start."))
    .OnCompleted(() => Debug.Log("After completion!"))
    .For(transform1)
        .Move(Vector3.one)
    .For(transform2)
        .Move(Vector3.one * 2)
    .For(transform3)
        .Rotate(Vector3.up * 360)
        .Scale(Vector3.one * 3)
    .Start();
```



2.2. Flows

The concept of flows exists because tweening is never enough and we sometimes need to link multiple tweens together to create an animation.

You can create an animation by queueing up tweens one after another in a flow, but also you can thread those queues, all within one flow.

Here is an example of a flow:

```
Flow flow = new Flow()
    .Queue(new Tween(2f).For(transform1).MoveX(10f))
    .Queue(new Tween(3f).For(transform1).MoveY(10f))
    .At(1f, new Tween(2f).For(transform2).MoveTo(new Vector3(10f, 10f,
0f)))
    .Start();
```

You can also queue other flows inside a flow:

```
Flow innerFlow = new Flow()
    .Queue(new Tween(2f).For(transform).MoveY(10f));

Flow flow = new Flow()
    .Queue(new Tween(2f).For(transform).MoveX(10f))
    .Queue(innerFlow)
    .Start();
```

3. Tools

3.1. Settings

- FlowEnt Debug - you can enable this Compilation Symbol in order to enable debugging for your app. Debugging will show you the stack trace for there an animation was created, if that animation fails to update

3.2. Inspector

You can use the FlowEnt inspector to view detailed information about how your animations are running and where they were created.

You can also control the animation but play/pause and even the timescale!

4. Demo

In the Demo folder(FlowEnt/Demo) you will find a demo which contains numerous examples of library usage with different cases and approaches.