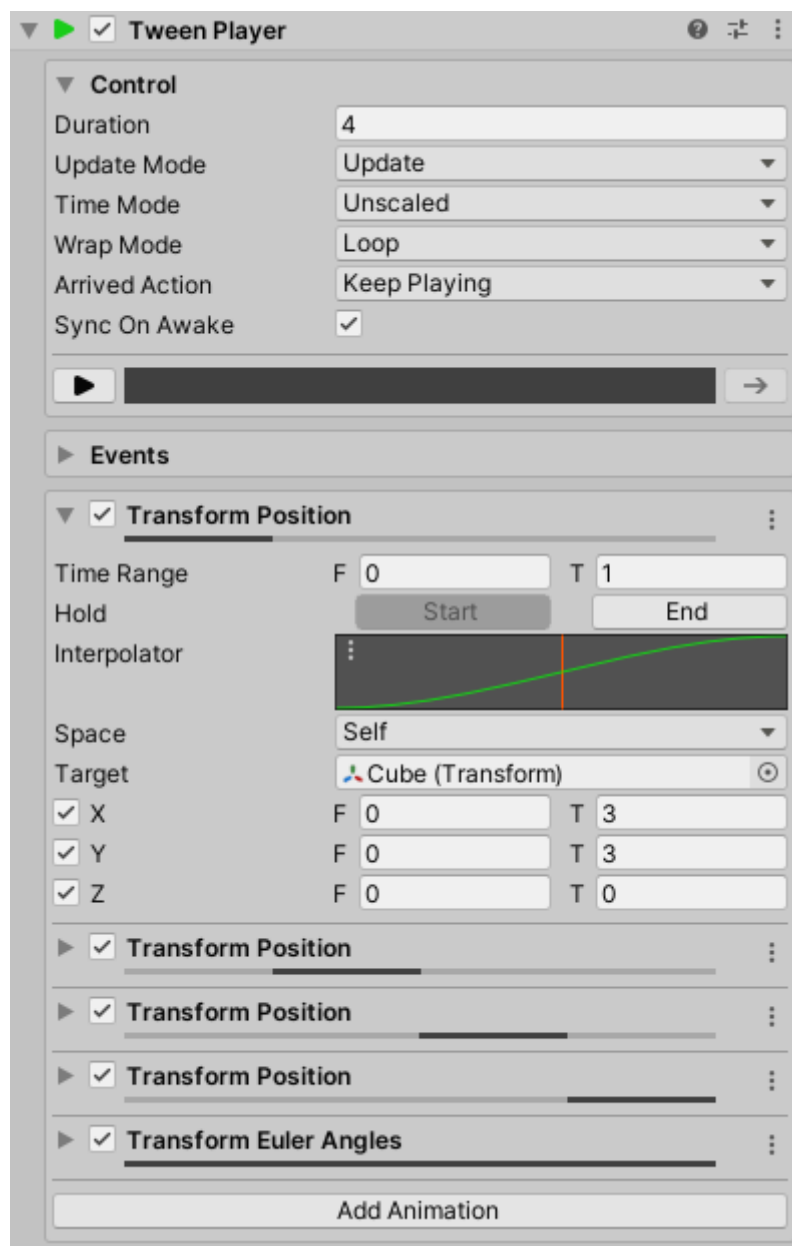# Tween Player

Tween Player is a simple & high performance & powerful Interpolation animation tool. Unlike other interpolation animation tools, **Tween Player** is more data-driven and extensible.

The main concept of this tool is a `TweenPlayer` component which can add multiple `TweenAnimation`. The menu path of `TweenPlayer` component is *Component > Miscellaneous > Tween Player*.

Inspector

`TweenPlayer` inspector includes three parts: **Control**, **Events** and **animation list**.

Control

There are some playback options in this part.

Duration: how long the playback will take, in seconds.

Update Mode: which updating phase the playback will be executed.

Time Mode: use normal game time or scaled time?

Wrap Mode: Clamp, Loop and Ping Pong.

Arrived Action: controls whether playback stops when it ends.

Sync On Awake: samples all animation states when this Tween Player awakes, it can avoid flashing caused by error initial states.

Events

There are two callbacks you can register: OnForwardArrived and OnBackArrived.

OnForwardArrived: called when playback gets to the end point.

OnBackArrived: called when playback gets to the starting point.

Animation List

Every animation has some common settings.

Time Range: the time range this animation works in.

Hold: controls the animation behavior out of time range. If *Start* is enabled, time before *Time Range* is considered the same as start of *Time Range*; if *End* is enabled, time after *Time Range* is considered the same as end of *Time Range*.

Interpolator: controls the interpolator curve in the *Time Range*. Every frame the animation calculates new state like this: *lerp(F, T, interpolator(t))*, *t* is normalized time in the *Time Range*.

How to···

···play or stop TweenPlayer at runtime?

Just set `TweenPlayer.enabled`.

···use Light2D animations?

Uncomment the first line of *URPAnimations.cs*.

···add custom animation?

It's simple, like this:

```csharp
[Serializable, TweenAnimation("Rendering/Light Color", "Light Color")]
public class TweenLightColor : TweenColor<Light>
{
    public override Color current
    {
        get => target ? target.color : Color.white;
        set { if (target) target.color = value; }
    }
}
```

If you run into any issues using my assets, please submit feedback here.