Tween

Tweening is the process to define a starting positon and an end position, and let it transition from one to the other over the course of a specifed duration.

For exmaple, opening a door can be easily achieved defining it's starting position as its current position and its end point as the same as its starting one, plus 2 units up in the Y axis. Once you specify the duration, the door will slide upwards when the tweening is activated.

The Tweening library has been created with Game Creator in mind, but can also be leveraged to be used in other scripts. Use the `Tween.To(...)` static method to create a new transition.

The `To(gameObject, input)` has two parameters: The *Game Object* that recieves the tweening, and an instance of a `TweenInput` class, which configures the animation.

Following the example from above, let's say we want to slide a "door" object 2 units up in the air. We can define the `TweenInput` class instance like this:

```
Vector3 valueSource = door.position;
Vector3 valueTarget = door.position + Vector3(0,2,0);
float duration = 5f;

ITweenInput tween = new TweenInput<Vector3>(
    valueSource,
    valueTarget,
    duration,
    (a, b, t) => door.position = Vector3.Lerp(a, b, t),
    Tween.GetHash(typeof(Transform), "transform"),
    Easing.Type.QuadInOut
);
```

Transition Type

In this example we use a Vector3 transition, but it accepts any value type, like numbers, colors, quaternions, ... It's up to the *updateCall* to interpolate between the initial and final value.

Let's break down each of these parameters in order:

```
TweenInput<Vector3>(
    Vector3 start,
```

```
    Vector3 end,
    float duration
    Update updateCall,
    int hash,
    Easing.Type easing
);
```

- **start:** A value indicating the starting position
- **end:** A value indicating the end position
- **duration:** The amount of time it takes to complete the transition
- **updateCall:** A method called every frame while the transition occurs. Contains 3 parameters: The starting value, the end value and the completion ratio between 0 and 1.
- **hash:** An integer that uniquely identifies this transition. If another transition with the same id starts, it cancels the previous one.
- **easing:** An optional easing function. If none is provided, it will use a linear function.