

# Tweenimations

## New Animation System

### Basic Structure

1. Animations consist of 'animation frames', or 'key frames' (same concept, different name).
2. An animation frame contains a number of renderable elements.
3. Each element must be renderable, and may have multiple fields that can be manipulated (e.g. location, rotation).

### Frame to Frame

Think of a frame as a **snapshot** of an animation, at a particular point in time. The frame knows each element that exists at that point in the animation, and the properties about those elements necessary to properly render. Some basic properties include the position and rotation of the renderable elements.

### Behind the Scenes (Tweens)

Tweens are the underlying tool used for the new animation system. In short, tweens manage data between a starting and ending point in time.

Let's look at a simple example (pseudo code):

```
start = 10;
end = 150;
durationInSeconds = 6.0;
tween = new Tween(start, end, durationInSeconds);

tween.getValueAtTime(3.0);
// => returns 80
// This is the half way point between 10 and 150,
// as 3.0 seconds is half of 6.0 seconds.
```

Data in each from can be of any type, and it is up to the user to determine how that data is interpolated from one point to another.

There is the possibility of automatically interpolating numbers and arrays of numbers. This may be preferable to forcing users to implement their own interpolation function.

The best of both worlds, would to be automatic interpolation of numbers and arrays of numbers, as well as providing callback functionality to the end user. They would have the ability to intercept the automatic interpolation, as well as implement whatever code they require/desire.

The difficult part about this, is that it is the code built into the engine which loads their sprite assets. How will arbitrary variables (not predefined by the engine) by handled and passed to the object when read?

## **New approach thought**

Animations will only consist of images, which (in JSON) have their own preset properties:

1. Location
2. Rotation
3. Scale
4. Alpha

There may be others added in the future, but these are the defaults. Users will have the ability to hook into the animation system with a callback, so providing their own animation properties is not necessary, and this greatly simplifies the animation system.