# Animation Timeline JSON Data

Copyright (c) 2020. Ascendant Design and Training, LLC.  
Last updated by Daniel Patterson, Tuesday, October 20, 2020

The Scaffold animation timeline JSON data is used to drive timed animations of any kind. It has a consistent structure that makes it quick and easy to use for generating timed scenes.

The structure of this timeline is slightly different than the standard timelines you may have already become accustomed to. Where other timeline expressions have only a values-per-keyframe model, this model separates the concepts of what it is from when it happens, which both yields freedom along the temporal axis and reduces the number of expressions required to produce the expected animation.

Although full use of a values-per-keyframe approach is completely valid in this dataset, the more preferable workflow in the Scaffold animation system involves splitting your animation work into two concisely single-focused tasks; setting the marks and timing the transitions.

## Scenes, Marks, and Characters

In this dataset, a character is any sprite, including anything from a human actor to a moveable prop. The scene is the definition of what is happening in a certain context, and closely represents one layer or one slide of the physical file being driven. Most often, a specific set of characters will only be found in one of many scenes, but it is also common for a single character to be featured in two or more scenes. Within the scene, the mark is a definition of the places or conditions within each of the characters are statically found during a certain focus of that scene. Generally, follow these steps to define all of the marks for the characters in the scene.

* Define one or more scenes.
* Within each scene, define one or more marks.
* Set each sprite's coordinates and properties for that mark.

## Keyframes

This dataset handles keyframes in a very similar way to other animation timelines you might have worked with before. For most practical intents and purposes, you will select a frame number, then set one or more properties of one or more objects at that frame number. As the runtime frame counter indexes through each of those frames, milestone values are reached and passed. When linear and shaped-based interpolations are used, calculated intermediate values are calculated for a property of a character for which a keyframe doesn't exist on the current frame and for which the previous and next keyframe values are different.

Managing the entire timeline in one approach is traditionally more difficult because you have to continually maintain both temporal and physical senses simultaneously. In other words, using only keyframes, you constantly have to keep track of where you are in time, and what you want to happen at that exact moment.

## Data Model

As mentioned above, this dataset is described in JSON format, which is a human readable, embeddable structure of object and array definitions. The hierarchical pattern of the data is described in this section.

* **Animation Timeline Dataset**. *Object*. The entire dataset is an object that can be embedded or used alone.
  + **Frames**. *Array*. The collection of keyframe definitions.
    - **Frame**. *Object*. Individual keyframe.
      * **Description**. *String*. Brief description of this frame.
      * **Elements**. *Array*. Collection of definitions at this keyframe.
        + **Element**. *Object*. A single frame element definition.
        + **Description**. *String*. Brief description of this element.
        + **ElementID**. *String*. The unique identification of the element to animate.
        + **ElementName**. *String*. Ignored unless **ElementType = SceneMark**, this value identifies the name of the scene and mark, in dotted notation. For example, if the scene name is "FirstScene" and the mark is "Start", **ElementName** will be "FirstScene.Start".
        + **ElementType**. *String*. The type of element being referenced. Valid choices are found in the **ElementType Values** table.
        + **InterpolationCount**. *Number*. Count of frames to which the interpolation will be applied if a measured interpolation type is selected.
        + **InterpolationType**. *String*. Name of the interpolation type used for defining the type of animation action occurring between two keyframes. Possible values for this property are listed in the **InterpolationType Values** table.
        + **Properties**. *Array*. Collection of properties being set for this keyframe. Each object in this collection is a single property structured as "Name": Value, where "Name" is the name of the element's property to be set, and Value is the value to be assumed at this frame, expressed in the appropriate formatting method for the type of data.
      * **FrameIndex**. *Number*. Absolute index of this frame in the timeline.
  + **Scenes**. *Array*. The collection of scene definitions.
    - **Scene**. *Object*. Individual scene definition.
      * **Description**. *String*. Brief description of the purpose of the scene.
      * **Marks**. *Array*. Collection of marks for this scene.
        + **Description**. *String*. Brief description of this mark.
        + **MarkName**. *String*. Name of the mark.
        + **MarkSettings**. *Array*. Collection of settings that establish this mark.

**MarkSetting**. *Object*. Individual setting for mark.

**CharacterName**. *String*. Name of the SVG character element.

**Description**. *String*. Brief description of the setting.

**Interpolations**. *Array*. Collection of interpolation definitions for properties of this character on this mark.

**Interpolation**. *Object*. Individual interpolation definition for this character at this mark.

**Description**. *String*. Brief description of this interpolation task.

**InterpolationCount**. *Number*. Count of frames to which the interpolation will be applied if a measured interpolation type is selected.

**InterpolationProperty**. *String*. Name of the property to which this interpolation is applied.

**InterpolationType**. *String*. Type of interpolation applied. See the table at **InterpolationType Values** for available types.

**Properties**. *Array*. Collection of properties to which this mark is being set. Each object in this collection is a single property structured as "Name": Value, where "Name" is the name of the element's property to be set, and Value is the value to be assumed at this frame, expressed in the appropriate formatting method for the type of data.

* + - * **SceneName**. *String*. Name of the scene.

### ElementType Values

The types of element that can be keyframed.

|  |  |
| --- | --- |
| **Name** | **Description** |
| **Character** | The ElementID property refers to the ID property of a character object within the shapes of the file. |
| **Layer** | The ElementID property refers to the ID of a layer. |
| **SceneMark** | The element is a scene mark in this file. The local ElementID property refers to the ID property of a character object in the file, and the ElementName property refers to the name of the scene and mark in dotted name notation.  For example, if the scene name is "FirstScene" and the mark is "Start", ElementName will be "FirstScene.Start". |

### InterpolationType Values

Types of interpolation that can be established between two keyframes.

|  |  |
| --- | --- |
| **Name** | **Description** |
| **CountIn** | The change occurs gradually over the last **InterpolationCount** number of frames prior to the current frame, ending at the current frame. |
| **CountOut** | The change occurs gradually over the next **InterpolationCount** number of frames after the current frame, starting at the next frame. |
| **Immediate** | The entire change occurs at this frame. |
| **LinearIn** | The change between the previous and current keyframes is linear. If the previous keyframe also has an out interpolation, the linear portion of the interpolation extends either from the end of a measured timespan or the center of the total timespan, depending upon the previous type of interpolation selected. |
| **LinearOut** | The change between this and the next keyframe is linear. If the next keyframe also has an in interpolation, the linear portion of the interpolation extends either from the beginning of a measured timespan or the center of the total timespan, depending upon the next type of interpolation selected. |

### Properties

Several properties are available within scene and keyframe definitions. The supported properties are listed here.

To support use with measured interpolation types, the notions of .In and .Out suffixes are also supported on each of the listed properties. If defined, the .In variation of the property is compared when transitioning from the previous to current keyframe, and the .Out variation of the property is compared when transitioning from the current to next keyframes.

If the .In and .Out suffixes are not defined when a measured interpolation type is encountered, the entire change between the previous and current frames is made within the measurement window for incoming measurements, and the entire change between the current and next frames is made within the measurement window for outgoing measurements.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **Height** | Number | Height of the object. |
| **Opacity** | Number | The degree of opacity on the object. Values range from 0.0 to 1.0. |
| **Rotation** | Number | Rotation of the object, in degrees. |
| **Width** | Number | Width of the object. |
| **X** | Number | X-coordinate of the object. |
| **Y** | Number | Y-coordinate of the object. |