LERP Animator 2

Introduction

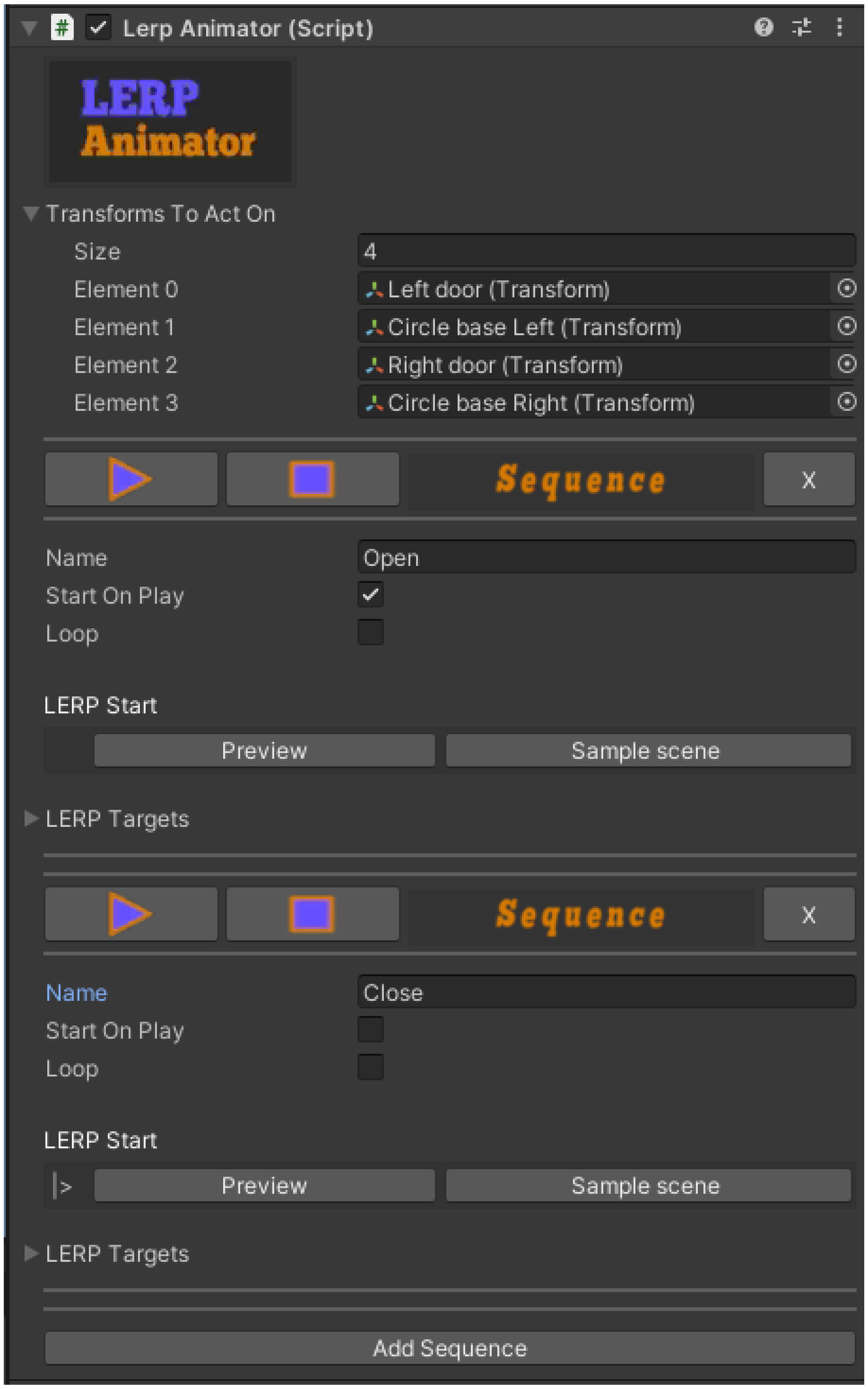
Many types of animations in games consist of linear interpolation between positions, rotations and scales, executed over a sequence of segments.

Game engines make interpolation easy, but it requires a bit of programming in relation to playback and data management to make it practical to do this for many transforms and segments in a single animatic.

The LERP Animator is a Unity component that lets you take an unlimited number of transforms and run them through an unlimited number of segments (LERP Targets) of interpolations, all controlled within a single custom inspector. Version two supports defining unlimited sequences in the component, and easily play them by name through code. This can replace the need for using third party software to rig and animate, then import them as assets, for these kinds of animations.

Other benefits of LERP Animator:

- It lets you preview segment states or play through the animatic live in the editor without starting game play.
- It makes it easy to set target position and scale data for interpolation, by sampling the data from the scene.
- Rotation is not as straightforward to deal with as position and scale, as you can't sample the euler angles or quaternion values of an object and expect it to interpolate past 180 degrees around an axis. Default rotation lerping will interpolate the shortest path between two rotations. LERP Animator solves this by letting you manually dial in a rotation offset from the previous state.
- For extra usefulness in game development, it can also fire any kind of events between segments.
- Very code friendly. The only need to code is if you want to trigger the animation any time other than when the game starts. This is easily done by getting a reference to the instance and calling the *PlaySequence()* function with the name of the desired sequence.



Size 4
Element 0 Left door (Transform)
Element 1 Circle base Left (Transform)
Element 2 Right door (Transform)
Element 3 Circle base Right (Transform)

Name Open
Start On Play ✓
Loop
LERP Start

Name Close
Start On Play
Loop
LERP Start

The interface

Transforms to act on

The transforms that can be manipulated by this component.

Playback mode

Start on play

Starts the specific sequence when you start the game in the scene.

(**Note!**) Since a transform can be added to several LERP Animators, no action is taken to ensure that transforms are in any particular start states when the game starts or when starting the sequence through code. This is left up to the user. Press *Preview* on *Start states* to leave them ready to be animated.

Loop

Will loop the sequence in both game play mode and while in editor.

LERP Start

The starting point of the animation sequence. Important things to remember:

Adding a transform to the "Transforms to act on" array automatically saves its data to LERP Start and any existing LERP Targets.

LERP Start contains the starting point for position, rotation and scale. Since rotation offsets are applied from the start states, saving a new rotation start state by pressing "Sample scene" after adding rotation offsets in targets, will produce different rotation destinations than originally shown. This is not an issue for positions and scales.

- ***Preview*** applies the stored start states to the transforms.
- ***Sample from scene*** stores the positions, rotations and scales of the transforms.

LERP Targets

- ***Play*** plays the sequence in editor from that specific segment and runs to the end of the whole sequence.
- ***Name*** makes it easier to keep track of the different LERP targets.
- ***Events*** will make events fire before or after lerping for that segment. Events are only fired during gameplay.
- ***Duration*** is the time in seconds you want the lerping to last.
- ***Pause after*** will apply a pause after lerping is done for that segment.
- ***Curve*** will set the animation curve for that segment. Important to note:

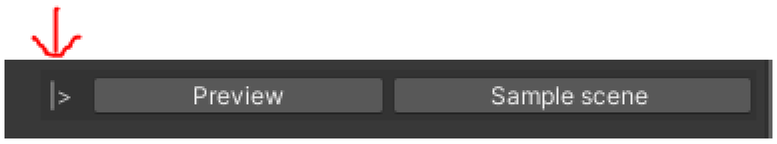
LERP Animator lerps without clamping, so the curve can go above 1 or below 0 along the way in the y axis to make some interesting effects.

The curve must end on 1 in the x-axis. Only set duration via the Duration property.

- **Rotation offsets** lets you dial in rotation offsets for each transform and axis for that LERP Target.

- ***Preview*** applies the stored LERP Target state to the transforms.
- ***Sample from scene*** stores the positions and scales of the transforms.

Last selected state indicator



The *Last selected state indicator* shows you which state you last pressed *Preview* or *Sample scene* on. This makes it easier to keep track when going between the inspector and the objects in the scene you are manipulating.

Workflow

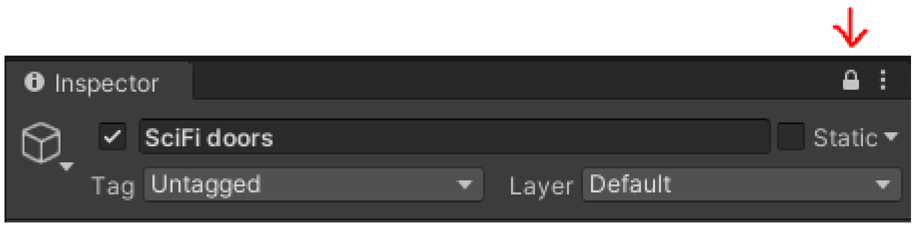
When adding a new transform to the list of transforms to act on, transform data is sampled into start states (all data), and any existing targets (position and scale).

When adding a new segment, data is also auto sampled into that segment.

This suggests that an efficient work flow would be to:

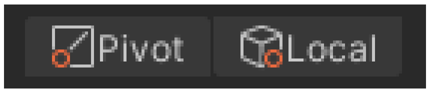
1. Set up all the transforms in the desired start states
2. Add transforms to the list on a LerpAnimator instance
3. Alternate between moving the transforms around <---> adding a segment and dial in desired rotation offsets

(**Pro tip!**) Pressing the “Lock inspector” button will keep the LERP Animator inspector showing, regardless of which game objects are selected. This makes it easier to deal with child objects.



(Note!) Adding a LERP Animator component to a game object will NOT automatically add the transform of that object to the list. This is because it is intended to support many transforms. In that case it can be desirable to have it on a top level object with manipulated transforms as children, and not necessarily animate the object the LERP Animator is attached to.

(**Note!**) LERP Animator works with local transform data. If you change the hierarchy of your chosen transform after adding it to a LERP Animator, pressing preview on start states or targets will not yield the same result as before. It is recommended to be done setting up transform hierarchies before adding transforms to LERP Animator.



The position tool handle must be set to “Pivot” when dealing with child objects to get the gizmo in the center of that object.

The rotation tool handle must be set to “Local” to be able to correctly gauge which axis rotation offsets will happen around when dialing in offsets in any LERP targets.

FAQ

Nothing is animating!

Did you actually add the transform to the TransformsToActOn list? Adding the Lerp Animator component won’t add the transform for that object to the list by default. Since Lerp Animator is designed to also work with any number of transforms simultaneously it may be preferred to have a top level object with animated transforms as children, which typically will not warrant the top level object to be added.

How do I trigger the animation through code?

Get a reference to the particular *LerpAnimator* component, and call the public function *PlaySequence()*, passing in the sequence name, to play the whole sequence.

Can I have the same transform present in multiple LerpAnimator's?

Yes you can, but it's up to you to make sure they aren't running at the same time during playback.

How do I copy data from one segment to another?

Press *Preview* on the *LERP Target* you want to copy from, then press *Sample scene* on the *LERP Target* to copy to.

How do I edit existing LERP Targets?

Press *Preview* on the LERP Target, manipulate your transforms, then go back and press *Sample scene* on the target.

How do I stop a running animation running in the editor?

Press *Preview* on LERP Start or any LERP Target, or the top level *Play* or *Stop* buttons.

Support contact

Please send support requests to spheroidgamestudio@gmail.com

Video tutorial

Here's a tutorial video on how to use the LERP Animator:
https://youtu.be/ndURS6a4MlE