Field update mechanism in project three

Overview

Project three uses a simplified custom mechanism for animating node properties. A special X3D node Link is introduced to bind together three components required for animation which are timers, interpolators, and fields to be updated. Specifically, each link stores the pointer to the field to be updated, and it references the timer and interpolator nodes. In each frame, before the rendering pass, each link updates the value of the field by the value produced by the interpolator. Each interpolator node has its key and value sequence as attributes read from the X3D file.

Each interpolator may be used several times: for instance, a pendulum orientation interpolator may be reused in several different clock gadgets present in the scene. All of these gadgets would go through the same sequence of orientations, although with different speeds and not synchronized with each other. All of these however may use the same interpolator but different timers that would drive the interpolator. Thus, for each gadget a separate link node will be used which will refer to the same interpolator node but specify different timers. The role of the timer is therefore to take a global current time and perform a timing transformation on it. Schematically, then a single update operation for a link *L* can be represented as:
**L.field_pointer = L.interpolator.Evaluate(L.timer.ConvertTime(t))*, where *t* is the current global time. Global current time runs in sync with the system clock, however it may paused and restarted by the application.

Timers

Timer nodes define time conversion from global time to a warped time used as a parameter to the interpolator. A timer has two attributes: period and shift.

Timers with positive periods are periodic, they produce values between zero and one. Therefore, in order to produce periodic motion, one should define interpolators whose key sequence is starts at zero and ends at one. To avoid abrupt transitions at the end of each cycle, the periodic interpolator should have the same starting and ending value in its value sequence.

A timer with a negative period attribute is non-periodic, it simply scales the input time by 1/(-period). This can be used with any interpolator with no restrictions on its key sequence (as long as it is increasing).

Please refer to X3Timer::ConvertTime function for more information.

Interpolators

Interpolators represent a function that maps its input time into a value. The parameters of the mapping are the key and value sequences specified in the corresponding attributes in the X3D file. Depending on the output value, Project three defines three interpolator classes:

- Scalar interpolators (return a single float value)
- Position interpolators (return a three-dimensional vector of floats, i.e. XVec3f)
- Orientation interpolators (return a rotation value, defined in rotation_t struct)

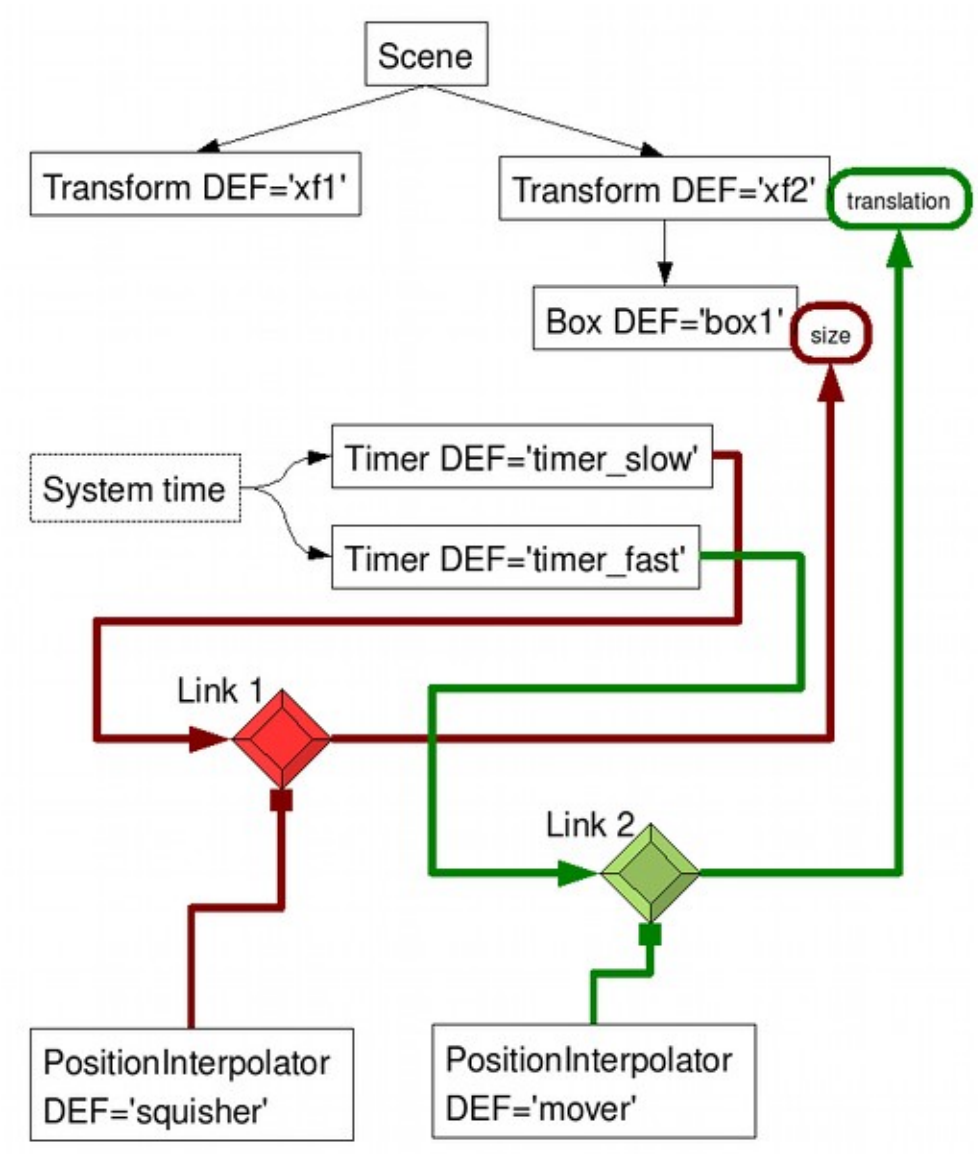
Depending of the value of the *smooth* flag, the interpolators implement piecewise linear or cubic Catmull-Rom spline interpolation. Orientation interpolators will not use smooth interpolation.

Link nodes

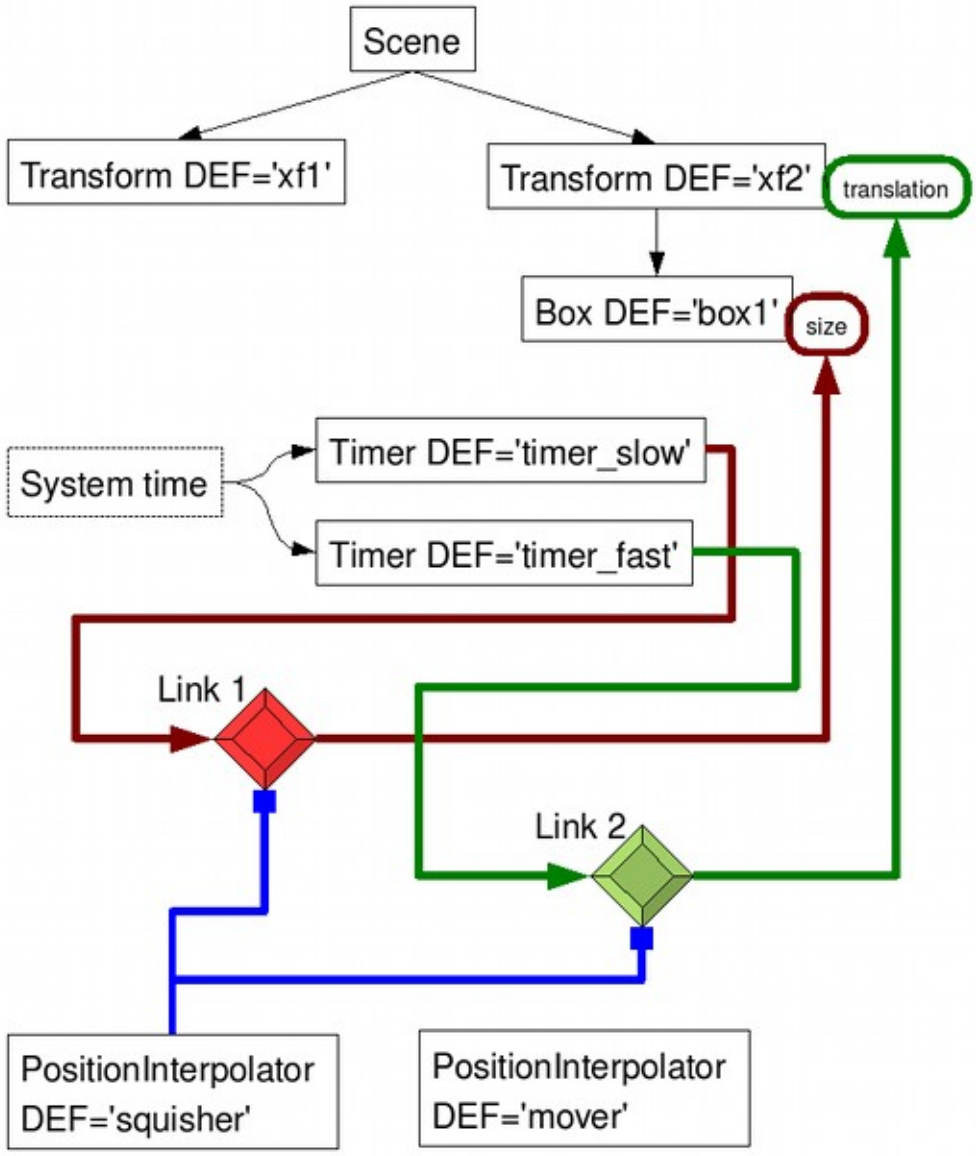
Link node is a non-standard node introduced in project 3 to define update relations. A link node can appear anywhere within the scene, and has four required attributes:

- TIMER refers to a timer's name defined earlier in the file
- INTERPOLATOR refers to an interpolator's node name defined earlier in the file
- TO_NODE refers to a node whose field needs to be updated in every frame by the output of the interpolator. Again the attribute should contain a name of such node defined earlier in the file. In order for a node to accept values into one of its fields, it needs to process the field's name in its GetFieldPointer method (for an example see X3Transform::GetFieldPointer function).
- TO_FIELD refers to a field's name within the updatable node declared in TO_NODE attribute.

The diagram below shows an example arrangement with two links using two different timers, two interpolators, and two updated fields.



In some situations, we can reuse interpolators or timers. For example, in the following diagram, both links use the same interpolator but different timers. Therefore, the same sequence of 3D vector values will be placed into two updated fields, possibly with some delays or perhaps the values in the translation field of the transformation node will change more quickly (since it is using 'timer_fast' timer which may indicate a shorter period value).



In a similar fashion, the same timer can drive two different interpolators.