TSS

Plugin for Unity – quick and visual animating based on tweens.

Abilities

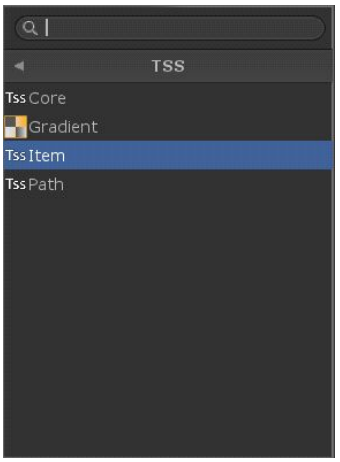
- Simple and friendly user interface for quick animating;
- Components support: Transform, RectTransform, `CanvasGroup`, `Image`, `RawImage`, `Text`, `TSSGradient`, `Button`, `Collider`, `AudioSource`, `VideoPlayer`, `Light`;
- Possible animating: position, rotation and scaling, color and transparence, image fill, text, number, volume of sound, light, action radius, characteristics of material;
- Timeline for quick animating and sequences editing;
- Adjustable inheritance (from forefather to offspring);
- 40 built-in eases, custom eases support;
- Various tweens for opening and closing objects;
- Path module sets objects movement trajectory;
- State machine module lets to group all the objects with tweens control their state;
- Adjustable updating mode and time scale ignoring for each object;
- Loops support;
- Control of components AudioSource and VideoPlayer;
- UnityEvent support;
- Save and load all object parameters to profile asset;
- Built-in examples with using of plugin abilities.

Glossary

- **Item** is a standard Unity object which has **TSSItem** component. An item includes states, sets and tweens. An item has 2 states: closed and opened, and 2 intermediate states: opening and closing;
- **Tween** is a function description for interpolation from closed state of any item parameter to opened one or backwards. A tween has a tween effect (the affected point), a tween ease (preset and custom), a mode (one ease for opening and closing or different eases) , a direction (tween is active only while its opening and/or its closing, while pushing the bottom);
- Tween **ease** is a curve, which is described with a formula or by user in the curve editor. The curves `Linear, Quad, Cubic, Quart, Quint, Sine, Expo, Circ, Elastic, Back, Bounce` in different variants `(In, Out, InOut, OutIn)` are used by default;
- **Tween effect** is a tween impact. So 'color' effect influence on the component with color attribute (Image, RawImage, Light .. etc);
- **Timeline** is a window, which permits visually redact time parameters of the item and their offsprings as timeout and time. Editing of opening and closing parameters are available;
- **Branch** is a cluster of the item and its offspring;
- **Profile** is an outside asset file, which contains all parameters of all item states, tweens and time configurations and extended configurations. Profile can be set on an item group and upgrade its parameters;
- **Core** is a standard Unity object which has TSSCore component. The core includes conditions and permits to switch between them. In one period of time only one condition in the core can be active;
- **Path** is an item which has **TSSPath** component. The path sets the way of item actions.

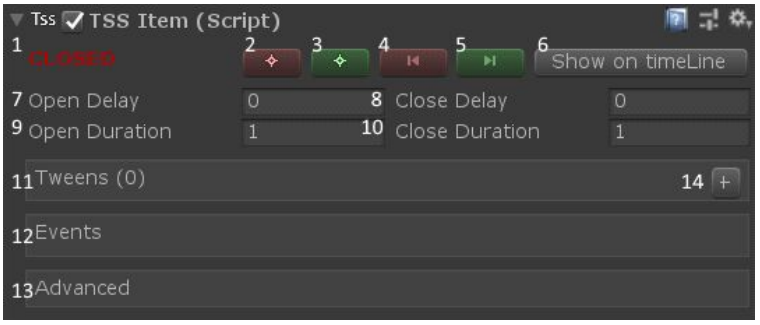
Getting started

Base component for work with plugin is **TSSItem**. You can add it through the standard window or menu (Component/TSS/Item):



(img. 1)

After the adding item looks like in the following way:



(img.2)

1. The states of item can be closed, opening, opened, closing (there is although a ‘slave’ state when the state of item is controlling by hand or with the help of timeline tool);
2. The button for the record of state of item as closed (primary state);
3. The button for the record of state of item as opened (end state);
4. Close the branch immediately (in the editor) or switch the item to the primary state (in runtime);
5. Open the branch immediately (in the editor) or switch the item to the end state (in runtime);
6. Open the timeline for the time parameters editing of this branch;
7. Time in seconds, before the branch starts the opening;
8. Time in seconds, while this certain item goes to the opened state;
9. Time in seconds, before the branch starts the closing;
10. Time in seconds, while this certain item goes to the closed state;
11. The tab is displaying tweens of this item;

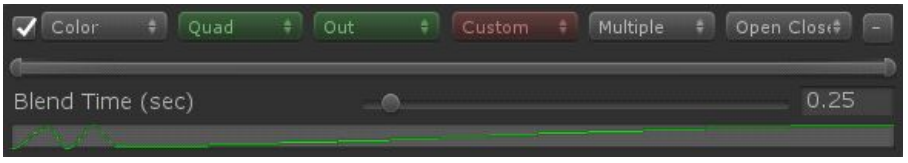
12. The tab of activities which activates by switching the states of item;
13. The tab of extended sets.

Click on the button [add new tween] (img.2, the button 14) to add a tween to the item:

(img.3)

1. If the tween is involved or not (it's possible to turn it off and try another tween with the same effect);
2. The tween effect. On which parameter is affected by tween;
3. The tween ease is a curve by which the parameter value is interpolated;
4. The variant of tween ease at the start and/or end;
5. The tween mode *Single* sets one ease for the opening and closing, *Multiple* lets to set 2 eases and time for which the interpolation will pass to the other ease, if the state of item is changed at the moment, when the item doesn't get the end or primary state;
6. The tween direction. The tween can come into action only by opening and/or closing, as well as tween can work only for click animation, if the item is bottom;
7. Remove the tween from the item;
8. Relative time offset. For example, the item with the opening time of 3 seconds can change the position for the first second, the rotation for the second one and the color of item will be changing from the first to the third seconds of opening. This slider lets to set the series of tweens operation;
9. Add one more tween.

The ease can be described with one of given curves or with custom curve with the help of curve editor. The tween can have only 1 custom ease, it's important, if the tween mode is set on <Multiple>. So, on the following example, the item by the opening will smoothly change its color, and there will be a flicker by the closing at the end of animation:



(img.4)

General sequence for the animation setting (for example, the position):

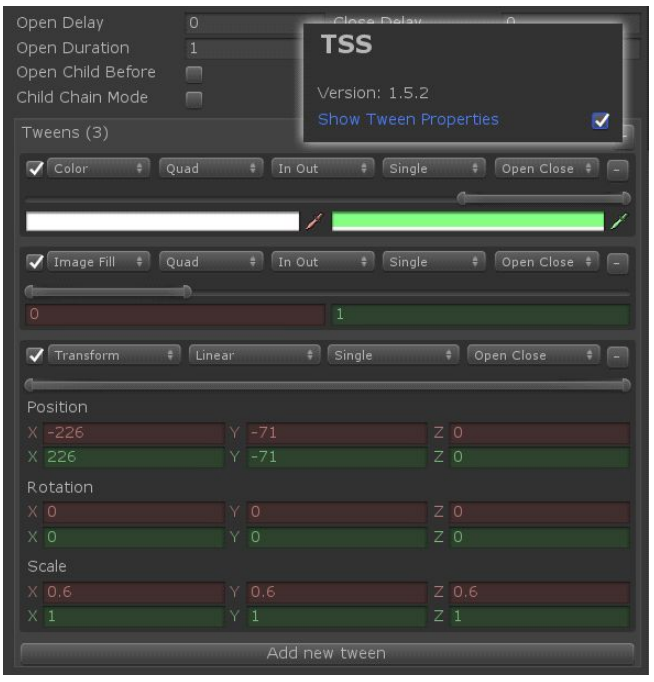
- Add the component (img.1);
- Add the tween (img.2, the button 14);

- Put the object to the primary position and click the button [record the primary state] (img.2, the button 2);
- Put the object to the end state and click the button <record the end state> (img.2, the button 3)
- Open the timeline for the fine animation tune (img.2, the button 6) – it isn't necessary;
- Adjust the tween (img.3).

Now to add a new tween, for example, color, you need to:

- Close the branch (img.2, the button 4);
- Add the tween (img.2, the button 14);
- Set the effect *Color* to the given tween (img.3, the parameter 2);
- Change the color of the object in which it's going to be colored in the closed state;
- Record the primary state (img.2, the bottom 2);
- Open the branch (img.2, the bottom 5)
- Record the end state (img.2, the bottom 3);
- Open the timeline for the fine animation tune (img.2, the bottom 6) – it isn't necessary;
- Adjust the tween (img.3).

There is a way which helps to simplify the repeatable hopping between states. It's essential to activate the visualization of states values. For this you need to put the parameter *Show Tween Properties* to the active position in the window Edit/Preferences/TSS (img.5).

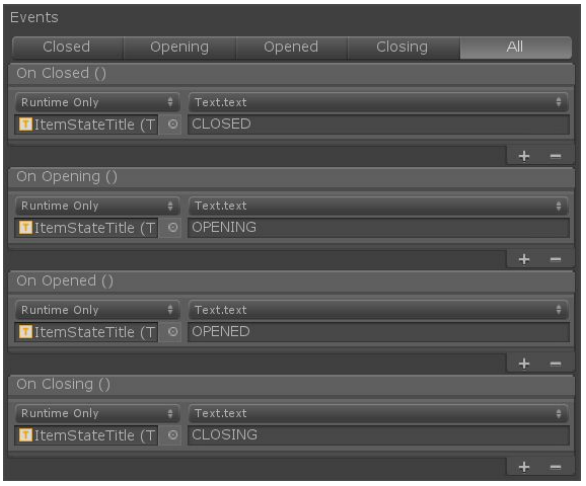


(img.5)

Events

An item has 4 events (img.2, the tab 12)

- *Closed* works out when the item entirely closed. The item is closed if all its parameters came back to the primary state and although all its offsprings are closed, if it has any of them;
- *Opening* responses when the item goes to the opening state;
- *Opened* is similar to *Closed* but works out only with the entirely opened item and its offsprings;
- *Closing* responses when the item goes to the closing state.



(img.6)

Advanced tab

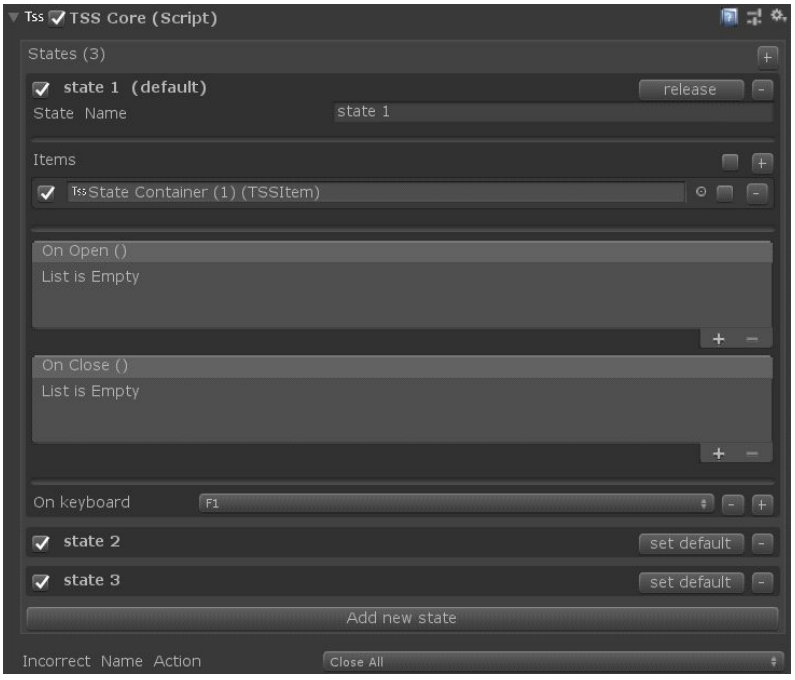
The tab *Advanced* (img.2, the tab 13)

(img.7)

1. The profile is connected to the item;
2. The activation mode by the command *Open()* means to open the item and all its offsprings. *Open Branch* is used by default;
3. The activation mode by the command *Close()* means to close the item and all its offsprings. *Close Branch* is used by default;
4. The activation mode by initialization. By the awaking the item immediately goes to the closed state. *Close Branch Immediately* is used by default.
5. The quantity of repetitive executions. The item can repeat an animation expansion the prescribed amount of times. By the value -1 the item will endless play the animation expansion till the getting the command *Close()*;
6. The item ignores an existence of the offsprings (the offspring branch interrupts on this item);
7. The item ignores an existence of the forefather and becomes independent;
8. The mode of the rotation interpolation: *Quaternion* is a search of the least significant arc, a turn isn't wider than 180 degrees. *Euler* is more robust interpolation, but lets to set some rotations by closing. *Path* is an object equation along the pathway. While the equation along the pathway the parameters *Mask* (what arcs are to line up), *Normal* (up to what local arc is it to line up. In the 3D space it's better to use *Up*, in the 2D space it's rather to use *Back* which moves away very quick object turnovers) are available;
9. The mode of material: *Instanced* means that an instance of the pointed material is fixed to render while the initialization, the item affects only to its own material copy. <Direct> means that the render gets the direct link to the material while the initialization, so the item can globally affect to the material and all objects with this material will although change;
10. The control of the connected interactive components is to switch off the interactivity by closed state and to switch it on by opened state;
11. The control of the impenetrability of the components is to switch on by opened state and to switch off by closed state;
12. The control of the connected component *AudioSource*;
13. The control of the connected component *VideoPlayer*;
14. The mode of the item updating: Update, LateUpdate, FixedUpdate;
15. If the time scaling influence on the item.

Core component

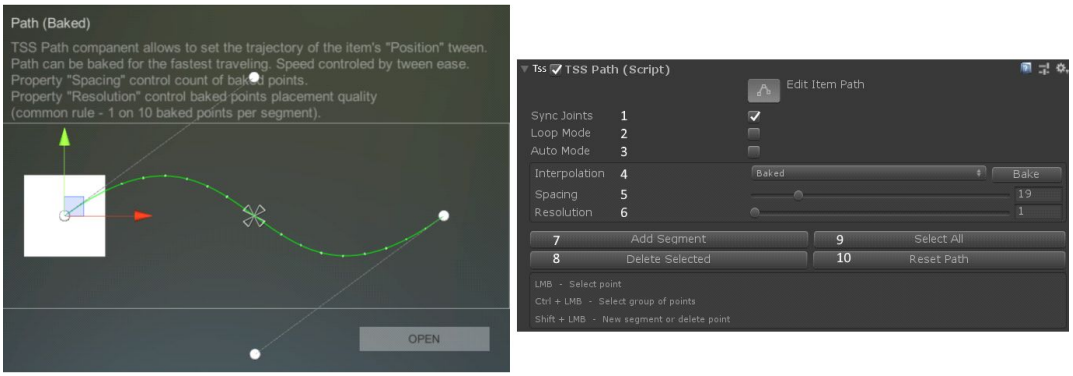
The core includes states and lets to switch between them. At one moment of the time there can be active only one state in the care. The state includes items (or activators) in container format. By a state choosing all the connected to it items become a command *Open()*, and all the items of the previous state become the command *Close()*. The core can rewrite the activation modes for each item in the container and although can rewrite the activation modes for all items in the container. The state can be chosen by the push on the button on the keyboard. The state has 2 events. Although the behavior of the core can be set by the activation of the incorrect state. The state can be identified as *Default* and is chosen by the initialization.



(img.8)

Path component

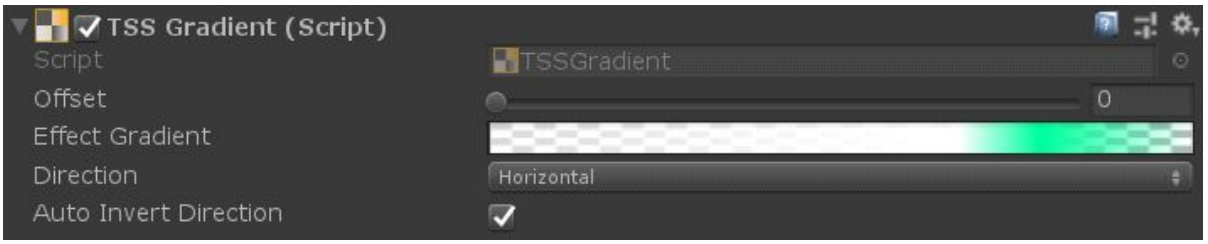
With the help of the component *TSSPath* it's possible to set the pathway of the item movement during its state changing. Without this component the item linearly (from point A to point B) changes its position and tween positions only set a speed of movement between these two points. The path can set the item path so as in 2D and as in 3D space and although it allows to equalize the item along the path (the tab Advanced/Rotation Mode). The path can inherit from another path. The path can be rotated by the rotation of its parent object.



(img. 9)

1. The mode allows to control the position of the opposite joint automaticly or move the joints together with baked point. It gives the possibility to make a broken line;
2. It closes the path;
3. It smoothes the path automatically with little changes of the neighbor baked points position. It has the parameter of smoothing by the activation and it's possible to make the path entirely from straight lines;
4. The interpolation mode:
    a) *Baked* means that the amount of points which are placed along the path is estimated on the stage of the path editing. The movement of the item forward the path goes along these points path from one to another. Because the position of these points is estimated only once this mode has a work speed advantage, although the movement along this path is linear and the tween only influences on the movement speed;
    b) *Dynamic* means that the movement along the path exactly happens on the curve of the path, each sheet is estimated by the item location. This mode has advantage that allows to change the position of baked points in process and although to connect the position of these baked points to another objects (for this it's necessary to choose the baked point while the path editing and a field with the link to the object what it's possible to connect the chosen points to will be available). But this mode is much more "expensive" from the productivity side and can move along the path in an uneven manner with a delay at the baked points;
5. It sets the distance between temporary points on the baked path and entirely influences on the quantity of temporary points;
6. It sets the quality of the temporary points arrangement. By the value 1 each segment divides into 10 imaginary parts in which disruptions will be put temporary points;
7. To add a new point (segment);
8. To remove selected points;
9. To select all points;
10. To reset the path to the base state.

Gradient component

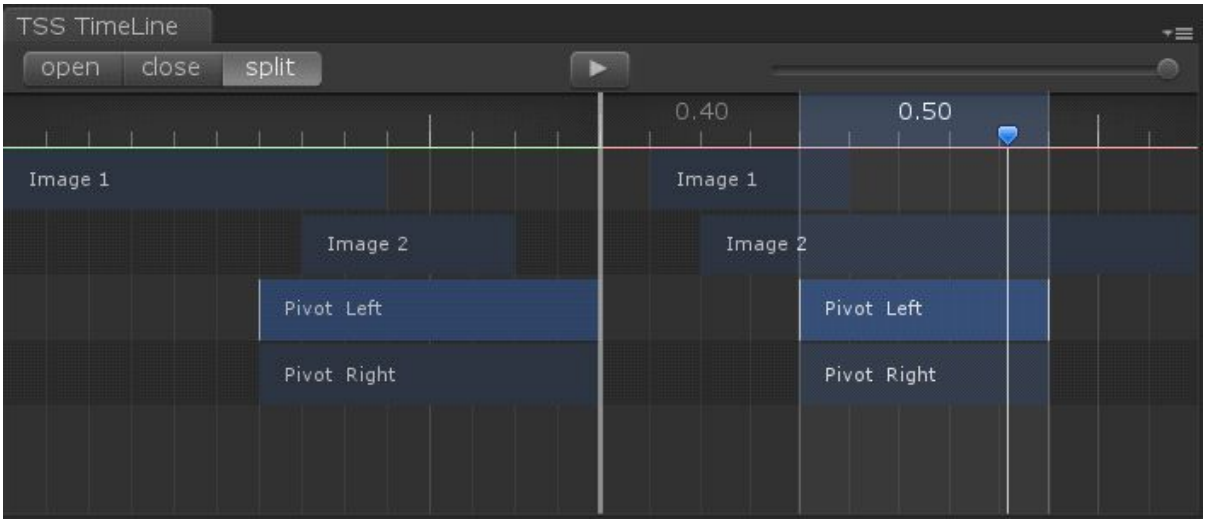


(img. 10)

TSSGradient is small extension of UI, which allows to change the vertex colors of UI elements, so as *Image* and *Text* gradient. *TSSItem* remembers the gradient state and can change its value by opening and closing (*Offset* parameter). The gradient can be invert automatically if it's entirely opened or closed what removes the "playback" effect.

TimeLine tool

The tool *TimeLine* allows to edit time parameters of item and its offsprings as timeout and time visually. It's possible to edit parameters for opening and closing.



(img. 11)

To open the TimeLine window, click on [Show On TimeLine] button on any item (img. 2, the button 6) or open it through the menu (Window -> TSS -> TimeLine) and after that choose the item in the hierarchy window.

The TimeLine allows to control the time of the opening, closing or both directions at the same time (img. 11)

The click on the item highlights it in hierarchy and reflects the animation duration and the timeout. The offsprings are although highlighted.

It's possible to set a fine tune of the animation moving the slide box.

The button *play/pause* starts or ends the presenting in the editor mode (Comment: in the editor mode the transparence of UI elements is estimated in another way, sometimes the transparence wouldn't be upgraded if there is no transformation).

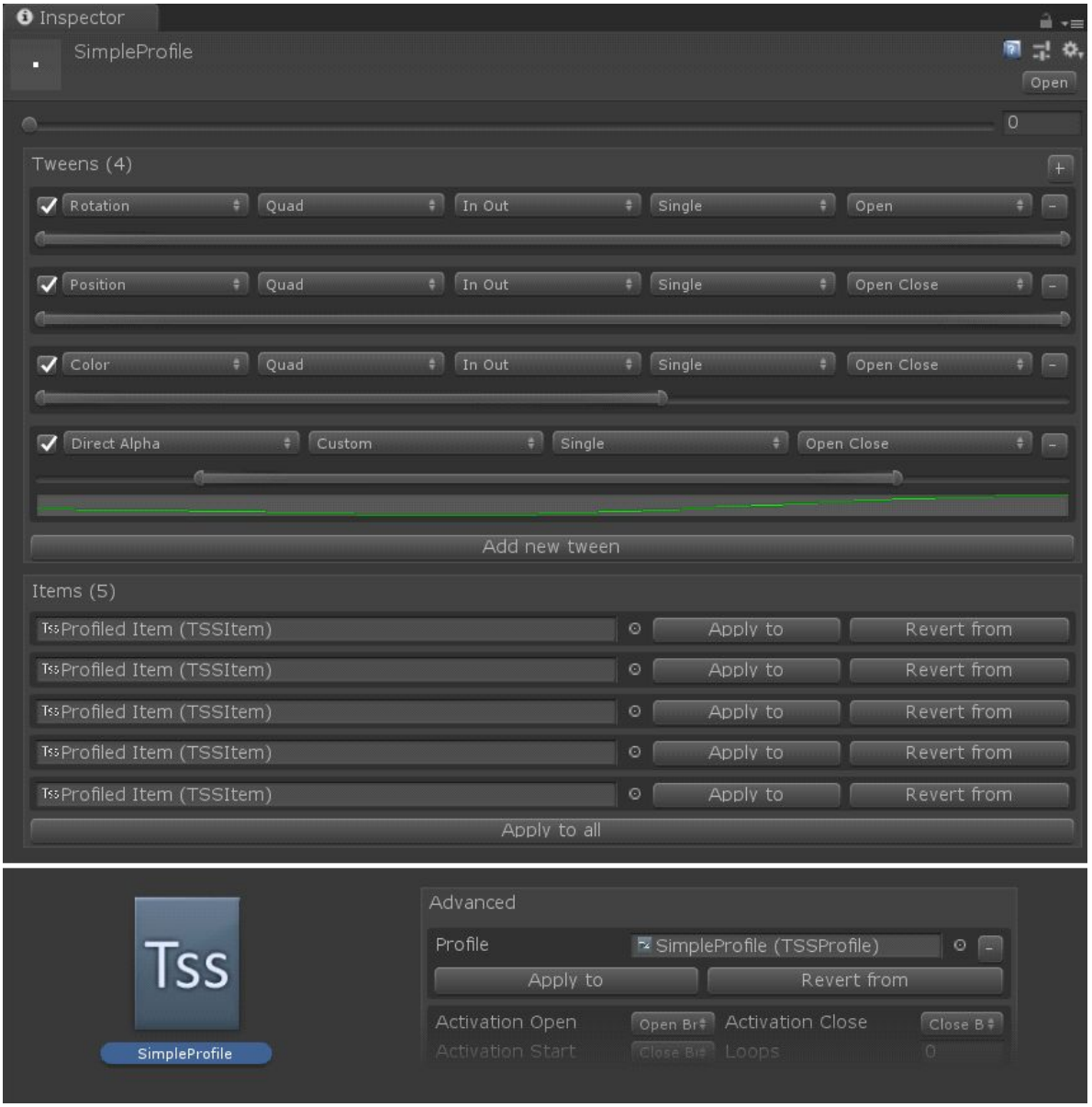
The left and right boards of the item allow to change the timeout and duration (To change the animation duration of all item offsprings in the chain mode it's necessary to push <Shift> button).

Open Child Before and Close Child Before allow to play the animation of offsprings and after the parents' one.

Profile

The profile allows to save all parameters and states of the item in an outside asset-file on HDD and to use it thereafter (and although in other projects) and although to control the item group with similar animation if it's important to change the animation. The profile interface allows to edit all the parameters and tweens and after to apply the changes to any item in a scene or to all of them at the same time. Although it's possible to upgrade the profile from the item interface (the tab Advanced -> Profile). It only upgrades profile but no other items with the same profile. To upgrade the item group it’s only possible in the profile interface.

The profile saves the following major build of the plugin and only works with this build.



(img. 12)