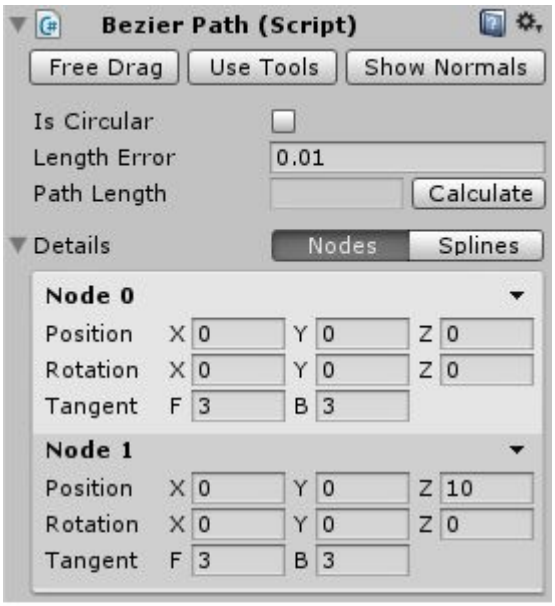
NGPT: Next-Gen Path & Tween

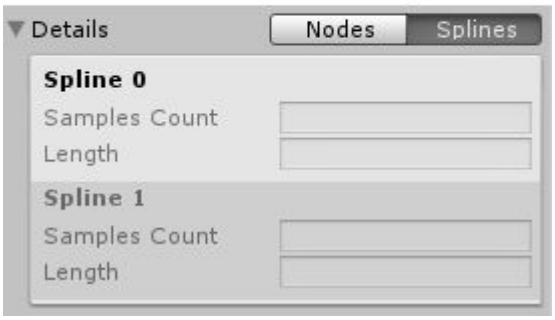
Thanks for buying *NGPT: Next-Gen Path & Tween* !

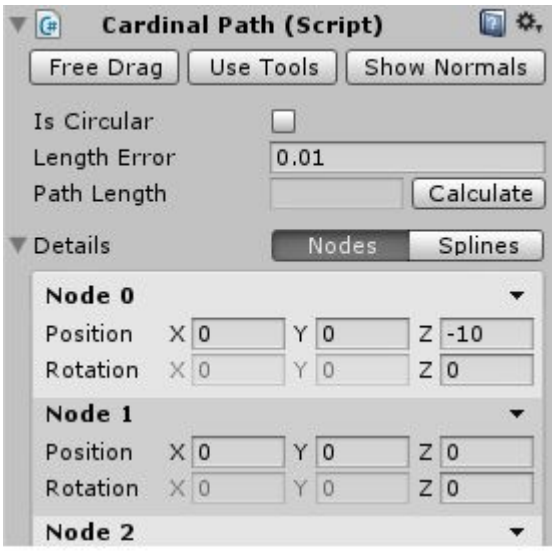
This package adds a **White Cat** group to unity **Component** menu. You can find some scenes in **Examples** folder for overview of these components, include sample scripts. This document will introduce how to use these components in unity editor and how to access them from scripts.

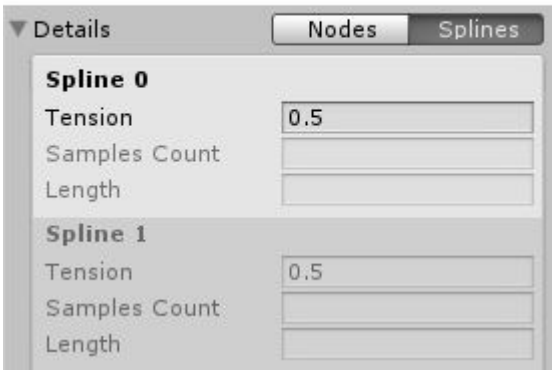
Path

A path include a list of nodes and a list of splines. There are two kind of path component: **Bezier Path** and **Cardinal Path**.









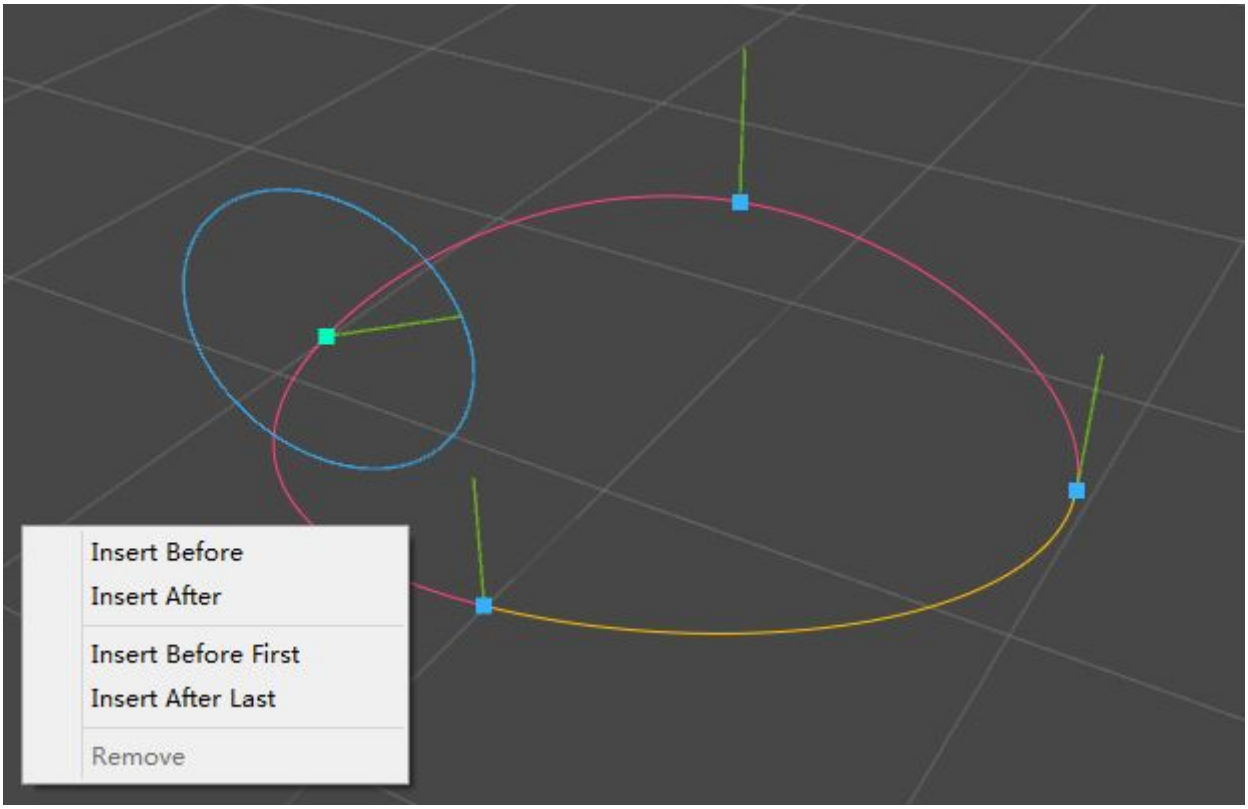
Properties

Free Drag	Enable this so you can free drag nodes in scene window.
Use Tools	Enable this so you can use move-tool, rotate-tool and scale-tool to edit nodes.
Show Normals	The forward direction of a node is always tangent of path, but upward direction is free to modify. The normal of a node is its upward direction. Enable this so you can check and edit normals of nodes.
Is Circular	If set to true the path will be head-tail like a circle.
Length Error	The error value in world unit of calculating path length.
Path Length	The total arc length of a path. You can click the “Calculate” button to get it.
Nodes	A node has position, rotation, and also has forward and backward tangent length if it’s a bezier node. All of these value are in local space.

Splines	A spline is a curve segment of a path. If you had click “Calculate” button you can see samples quantities and arc lengths of splines. Cardinal spline has extra tension value.

Tips

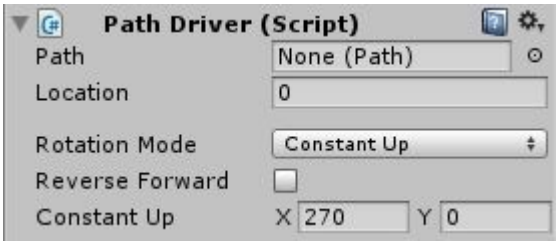
* If you had selected a node, you can press “F” key to focus it, or click mouse right button to show a popup menu to quickly insert or remove a node.



* You don’t have to click “Calculate” button manually, calculating method will be called automatically if it’s necessary. In order to avoid call calculating method at runtime, you’d better do it manually.

Path Driver

If you want to move a game object alone path, use path driver component.



Properties

Path	A reference of a path component. It can be a bezier path or a cardinal path.

Location	The arc length from path starting point. Modify this value can change world position of game object.
Rotation Mode	How to change rotation of game object when location has changed? There are 4 kind of rotation mode: ignore, constant up, slerp nodes and minimize delta. **Ignore :** Don't change rotation. **Constant Up :** Forward direction of rotation is tangent of path, referenced upward direction of rotation is a constant vector. **Slerp Nodes :** Forward direction of rotation is tangent of path, referenced upward direction of rotation is a interpolation result of two nearest nodes' normals. **Minimize Delta :** Forward direction of rotation is tangent of path, referenced upward direction of rotation is same with current value.
Reverse Forward	Only valid when rotation mode is not ignore. If set to true, the forward direction of rotation has reversed direction of derivative at specified location.
Constant Up	Only valid when rotation mode is constant up. It's referenced upward direction of rotation.

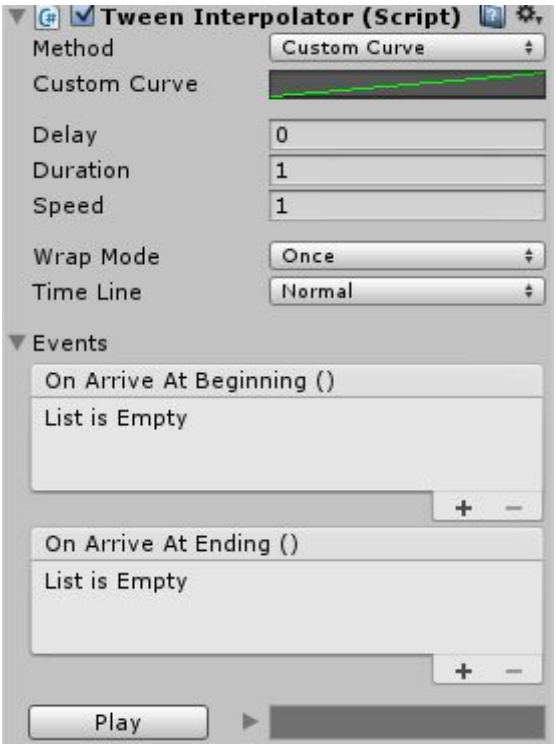
Tips

* If path is circular, location can be negative or a value greater than path total length.

* If rotation mode is constant up, the referenced upward direction should not be similar to any tangent of path.

Tween Interpolator

Any tween animation need an interpolator to drive, an interpolator can drive multiple animations at the same time.



Properties

Method	The interpolation method used for tween animations.
Custom Curve	Only valid when method is custom curve.
Delay	The time in seconds to wait before interpolating.
Duration	The duration time of process of interpolating.
Speed	Time scale, 1 means normal, 0 means pause.
Wrap Mode	**Once :** Stop playing when normalized time is becoming 0 or 1, component will be disabled automatically. **Loop :** Continue playing when normalized time is becoming 0 or 1, normalized time will jump to another endpoint. **Ping Pong :** Continue playing when normalized time is becoming 0 or 1, speed will be reversed. **Clamp Forever :** Continue playing when normalized time is becoming 0 or 1, but speed or normalized time will not change.
Time Line	There are two time lines in unity: normal time line and unscaled time line.
On Arrive At Beginning	Callbacks will be called when normalized time is becoming 0.
On Arrive At Ending	Callbacks will be called when normalized time is becoming 1.
Play	In edit mode, you can use “play” button to test tween animations; in play mode, “play” button state is same as enabled state.

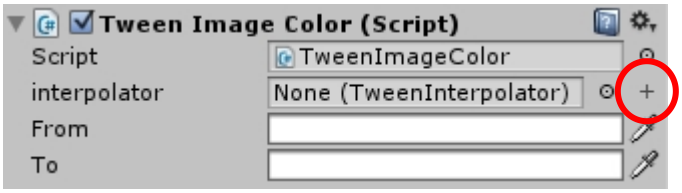
Tips

* When you stop playing tween interpolator in editor, all modified properties will be restored.

Tween Animations

Tween Base

Any kind of tween animation component relies on tween interpolator, but they need not belong to one single game object.

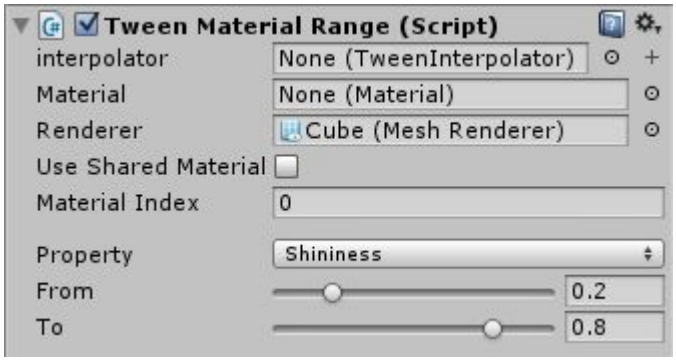


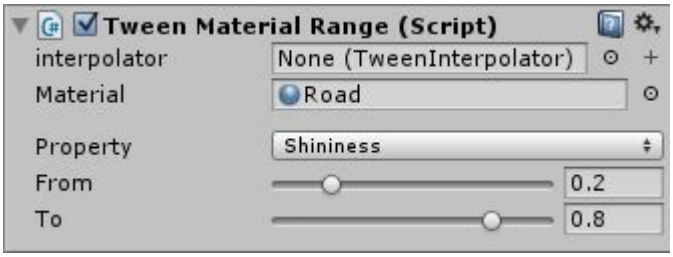
Click the “+” button can quickly add a tween interpolator to the same game object and reference it.

Almost every tween animation has a “From” state and a “To” state. Every frame the tween animation calculate and set current state base on “From” state, “To” state and result of interpolation.

Tween Material Property

Tween material property is subclass of tween base.





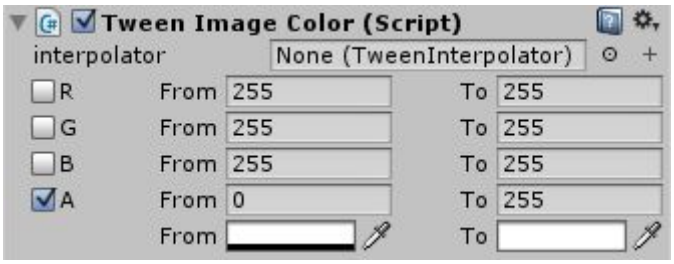
Material	You can reference a material directly in tween material components(Right Picture), if it's null you can also select a material from a renderer.
Renderer	The referenced renderer component.
Use Shared Material	Modify shared or independent material?
Material Index	The index of material in materials array.
Property	Name of property to modify.

Tips

* Every context menu of tween animations has "Set 'From' to current" "Set 'To' to current" "Set current to 'From'" "Set current to 'To'" options.

* OnRecord and OnRestore method are use to record and restore states in edit mode, you can call it manually at runtime for simplifying realization.

* Struct-properties can be controlled as multi-single properties in tween components.



WhiteCat.Path

int nodesCount { get; }

The total nodes quantity of path.

int splinesCount { get; }

The total splines quantity of path.

float pathTotalLength { get; }

The total arc length of path.

float GetPathLength (int splineIndex)

Get the arc length from path starting point to specified spline ending point.

splineIndex : The specified spline index in list.

float GetPathLength (int splineIndex, float splineTime)

Get the arc length from path starting point to specified spline point.

splineIndex : The specified spline index in list.

splineTime : The specified spline time parameter, at range [0, 1].

void InsertNode (int nodeIndex)

Insert a new node to nodes list of path.

nodeIndex : The index of new node to insert.

void RemoveNode (int nodeIndex)

Remove a node from nodes list of path.

nodeIndex : The index of node to remove.

Vector3 GetNodeLocalPosition (int nodeIndex)

Get local position of a node.

nodeIndex : The index of node to get local position.

void SetNodeLocalPosition (int nodeIndex, Vector3 localPosition)

Set local position of a node.

nodeIndex : The index of node to set local position.

localPosition : New local position to set.

Vector3 GetNodePosition (int nodeIndex)

Get world position of a node.

nodeIndex : The index of node to get world position.

void SetNodePosition (int nodeIndex, Vector3 position)

Set world position of a node.

nodeIndex : The index of node to set world position.

position : New world position to set.

Quaternion GetNodeLocalRotation (int nodeIndex)

Get local rotation of a node.

nodeIndex : The index of node to get local rotation.

void SetNodeLocalRotation (int nodeIndex, Quaternion localRotation)

Set local rotation of a node.

nodeIndex : The index of node to set local rotation.

localRotation : New local rotation to set.

Quaternion GetNodeRotation (int nodeIndex)

Get world rotation of a node.

nodeIndex : The index of node to get world rotation.

void SetNodeRotation (int nodeIndex, Quaternion rotation)

Set world rotation of a node.

nodeIndex : The index of node to set world rotation.

localRotation : New world rotation to set.

Vector3 GetSplinePoint (int splineIndex, float splineTime)

Get coordinate at specified point of spline.

splineIndex : The specified spline index in list.

splineTime : The specified spline time parameter, at range [0, 1].

Vector3 GetSplineDerivative (int splineIndex, float splineTime)

Get derivative at specified point of spline.

splineIndex : The specified spline index in list.

splineTime : The specified spline time parameter, at range [0, 1].

Quaternion GetSplineRotation (int splineIndex, float splineTime, Vector3 upwards, bool reverseForward = false)

Get rotation at specified point of spline. The forward direction of returned rotation is parallel to tangent at specified point.

splineIndex : The specified spline index in list.

splineTime : The specified spline time parameter, at range [0, 1].

upwards : The referenced upward direction of returned rotation.

reverseForward : If set to true, the forward direction of returned rotation has reversed direction of derivative at specified point.

Quaternion GetSplineRotation (int splineIndex, float splineTime, bool reverseForward = false)

Get rotation at specified point of spline. The forward direction of returned rotation is parallel to tangent at specified

point, the upward direction of returned rotation depend on normals of starting node and end node of spline.

splineIndex : The specified spline index in list.

splineTime : The specified spline time parameter, at range [0, 1].

reverseForward : If set to true, the forward direction of returned rotation has reversed direction of derivative at specified point.

void GetPathPositionAtPathLength (float pathLength, ref int splineIndex, ref float splineTime)

Get spline index and time at specified arc length of path.

pathLength : specified arc length of path.

splineIndex : The inputted value is a suggested value, the outputted value is index of spline at specified arc length of path. A negative inputted value means no suggestion.

splineTime : The outputted value is time parameter of spline at specified arc length of path.

WhiteCat.TweenInterpolator

bool isPlaying { get; set; }

Is the interpolator updating continuously? It's same as enabled in play mode.

float normalizedTime { get, set }

Progress of interpolating, at range [0, 1].

void ReverseSpeed ()

Reverse the sign of current speed.

void Replay ()

Reset internal time and enable the component.