– Developed by Dented Pixel

APIs

Classes Modules

Type to filter APIs

- LeanTween
- LeanTweenType
- LTBezierPath
- LTDescr
- LTRect

Games Developed by Dented Pixel

Monkeyshines – A swinging good time!

Princess Piano – Learn musical notation in this melodious adventure!

LeanTween Class

Show: ☑ Inherited ☐ Protected ☐ Private ☐ Deprecated

Defined in: `LeanTween.cs:880`

LeanTween is an efficient tweening engine for Unity3d

Index of All Methods | Optional Paramaters that can be passed

Optional Parameters are passed at the end of every method

Example:
LeanTween.moveX(gameObject, 1f, 1f).setEase(LeanTweenType.easeInQuad).setDelay(1f);

You can pass the optional parameters in any order, and chain on as many as you wish.
You can also pass parameters at a later time by saving a reference to what is returned.

Example:
LTDescr d = LeanTween.moveX(gameObject, 1f, 1f);
  ...later set some parameters
d.setOnComplete(onCompleteFunc).setEase(LeanTweenType.easeInOutBack);

Index Methods

Methods

`LeanTween.alpha` (ltRect:LTRect , to:float , time:float) LTDescr

Defined in `LeanTween.cs:1772`

Fade a GUI Object

Parameters:

- **`ltRect:LTRect`** LTRect
  LTRect that you wish to fade
- **`to:float`** Float
  the final alpha value (0-1)
- **`time:float`** Float
  The time with which to fade the object

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

Example:

LeanTween.alpha(ltRect, 1f, 1f) .setEase(LeanTweenType.easeInCirc);

`LeanTween.alpha` (gameObject:GameObject , to:float , time:float) LTDescr

Defined in `LeanTween.cs:1757`

Fade a gameobject's material to a certain alpha value. The material's shader needs to support alpha. Owl labs has some excellent efficient shaders.

Parameters:

- **`gameObject:GameObject`** GameObject
  Gameobject that you wish to fade
- **`to:float`** Float
  the final alpha value (0-1)
- **`time:float`** Float
  The time with which to fade the object

Returns:

LTDescr:

LTDescr:

LTDescr an object that distinguishes the tween

Example:

```
LeanTween.alpha(gameObject, 1f, 1f) .setDelay(1f);
```

`LeanTween.alphaVertex` (gameObject:GameObject , to:float , time:float) LTDescr

Defined in `LeanTween.cs:1788`

This works by tweening the vertex colors directly.

Vertex-based coloring is useful because you avoid making a copy of your object's material for each instance that needs a different color.

A shader that supports vertex colors is required for it to work (for example the shaders in Mobile/Particles/)

Parameters:

- **`gameObject:GameObject`** GameObject
  Gameobject that you wish to alpha
- **`to:float`** Float
  The alpha value you wish to tween to
- **`time:float`** Float
  The time with which to delay before calling the function

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

`LeanTween.cancel` (gameObject:GameObject)

Defined in `LeanTween.cs:1541`

Cancel all tweens that are currently targeting the gameObject

Parameters:

- **`gameObject:GameObject`** GameObject
  gameObject whose tweens you want to cancel

`LeanTween.cancel` (id:int)

Defined in `LeanTween.cs:1565`

Cancel all tweens that are currently targeting the gameObject

Parameters:

- **`id:int`** Int
  id of the tween you want to cancel

Example:

```
int tweenIDMove = LeanTween.move( gameObject, new Vector3(0f,1f,2f), 1f).id;
LeanTween.cancel( tweenIDMove );
```

`LeanTween.init` (maxSimultaneousTweens:int)

Defined in `LeanTween.cs:921`

This line is optional. Here you can specify the maximum number of tweens you will use (the default is 400). This must be called before any use of LeanTween is made for it to be effective.

Parameters:

- **`maxSimultaneousTweens:int`** Integer
  The maximum number of tweens you will use, make sure you don't go over this limit, otherwise the code will throw an error

Example:

```
LeanTween.init( 800 );
```

`LeanTween.isTweening` (ltRect:LTRect)

Defined in `LeanTween.cs:1673`

Test whether or not a tween is active on a LTRect

Parameters:

- **`ltRect:LTRect`** LTRect
  LTRect that you want to test if it is tweening

`LeanTween.isTweening` (gameObject:GameObject)

Defined in `LeanTween.cs:1658`

Test whether or not a tween is active on a GameObject

Parameters:

- **`gameObject:GameObject`** GameObject
  GameObject that you want to test if it is tweening

`LeanTween.move` (gameObject:GameObject , [] , time:float) LTDescr

Defined in `LeanTween.cs:1923`

Move a GameObject along a set of bezier curves

Parameters:

- **`gameObject:GameObject`** GameObject
  Gameobject that you wish to move

- **`[]`** Vector3 optional
  A set of points that define the curve(s) ex: Point1,Handle1,Handle2,Point2,...

- **`time:float`** Float
  The time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

Example:

Javascript:
LeanTween.move(gameObject, [Vector3(0,0,0),Vector3(1,0,0),Vector3(1,0,0),Vector3(1,0,1)], 2.0).setEase(LeanTween.easeOutQuad).setOrientToPath(true);

C#:
LeanTween.move(gameObject, new Vector3{Vector3(0f,0f,0f),Vector3(1f,0f,0f),Vector3(1f,0f,0f),Vector3(1f,0f,1f)}).setEase(LeanTween.easeOutQuad).setOrientTol

---

`LeanTween.move` (GameObject , vec:Vector3 , time:float) LTDescr

Defined in `LeanTween.cs:1819`

Move a GameObject to a certain location

Parameters:

- **`GameObject`** GameObject
  gameObject Gameobject that you wish to move

- **`vec:Vector3`** Vector3
  to The final positin with which to move to

- **`time:float`** Float
  time The time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

Example:

LeanTween.move(gameObject, new Vector3(0f,-3f,5f), 2.0f) .setEase(LeanTween.easeOutQuad);

LeanTween.move (gameObject:GameObject , [] , time:float) LTDescr

Defined in LeanTween.cs:1833

Move a GameObject along a set of bezier curves

Parameters:

- **gameObject:GameObject** GameObject
  Gameobject that you wish to move

- **[]** Vector3 optional
  A set of points that define the curve(s) ex: Point1,Handle1,Handle2,Point2,...

- **time:float** Float
  The time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

Example:

Javascript:
LeanTween.move(gameObject, [Vector3(0,0,0),Vector3(1,0,0),Vector3(1,0,0),Vector3(1,0,1)], 2.0) .setEase(LeanTween.easeOutQuad).setOrientToPath(true);

C#:
LeanTween.move(gameObject, new Vector3{Vector3(0f,0f,0f),Vector3(1f,0f,0f),Vector3(1f,0f,0f),Vector3(1f,0f,1f)}, 1.5f) .setEase(LeanTween.easeOutQuad).setOrientToPath(true);;

LeanTween.move (GUI) (ltRect:LTRect , vec:Vector2 , time:float) LTDescr

Defined in LeanTween.cs:1857

Move a GUI Element to a certain location

Parameters:

- **ltRect:LTRect** LTRect
  ltRect LTRect object that you wish to move

- **vec:Vector2** Vector2
  to The final position with which to move to (pixel coordinates)

- **time:float** Float
  time The time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

LeanTween.moveLocal (GameObject , Vector3 , float , Hashtable) LTDescr

Defined in LeanTween.cs:1909

Move a GameObject to a certain location relative to the parent transform.

Parameters:

- **GameObject** GameObject
  gameObject Gameobject that you wish to rotate

- **Vector3** Vector3
  to The final positin with which to move to

- **float** Float
  time The time to complete the tween in

- **Hashtable** Hashtable
  optional Hashtable where you can pass optional items.

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

`LeanTween.moveX` (gameObject:GameObject , to:float , time:float) LTDescr
Defined in `LeanTween.cs:1870`

Move a GameObject along the x-axis

Parameters:

- **`gameObject:GameObject`** GameObject
  gameObject Gameobject that you wish to move

- **`to:float`** Float
  to The final position with which to move to

- **`time:float`** Float
  time The time to complete the move in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

`LeanTween.moveY` (GameObject , float , float) LTDescr
Defined in `LeanTween.cs:1883`

Move a GameObject along the y-axis

Parameters:

- **`GameObject`** GameObject
  gameObject Gameobject that you wish to move

- **`float`** Float
  to The final position with which to move to

- **`float`** Float
  time The time to complete the move in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

`LeanTween.moveZ` (GameObject , float , float) LTDescr
Defined in `LeanTween.cs:1896`

Move a GameObject along the z-axis

Parameters:

- **`GameObject`** GameObject
  gameObject Gameobject that you wish to move

- **`float`** Float
  to The final position with which to move to

- **`float`** Float
  time The time to complete the move in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

`LeanTween.pause` (gameObject:GameObject)
Defined in `LeanTween.cs:1609`

Pause all tweens for a GameObject

Parameters:

- **`gameObject:GameObject`** GameObject
  GameObject whose tweens you want to pause

`LeanTween.resume` (gameObject:GameObject)

Defined in `LeanTween.cs:1644`

Resume all the tweens on a GameObject

Parameters:

- **`gameObject:GameObject`** GameObject
  GameObject whose tweens you want to resume

`LeanTween.resume` (id:int)

Defined in `LeanTween.cs:1630`

Resume a specific tween

Parameters:

- **`id:int`** Int
  Id of the tween you want to resume ex: int id = LeanTween.MoveX(gameObject, 5, 1.0).id;

`LeanTween.rotate` (ltRect:LTRect , to:float , time:float , optional:Array) LTDescr

Defined in `LeanTween.cs:1974`

Rotate a GUI element (using an LTRect object), to a value that is in degrees

Parameters:

- **`ltRect:LTRect`** LTRect
  LTRect that you wish to rotate

- **`to:float`** Float
  The final rotation with which to rotate to

- **`time:float`** Float
  The time to complete the tween in

- **`optional:Array`** Array
  Object Array where you can pass optional items.

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

Example:

```
if(GUI.Button(buttonRect.rect, "Rotate"))
LeanTween.rotate( buttonRect4, 150.0f, 1.0f).setEase(LeanTween.easeOutElastic);
GUI.matrix = Matrix4x4.identity;
```

`LeanTween.rotate` (GameObject , Vector3 , float) LTDescr

Defined in `LeanTween.cs:1959`

Rotate a GameObject, to values are in passed in degrees

Parameters:

- **`GameObject`** GameObject
  gameObject Gameobject that you wish to rotate

- **`Vector3`** Vector3
  to The final rotation with which to rotate to

- **`float`** Float
  time The time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

Example:

```
LeanTween.rotate(cube, new Vector3(180f,30f,0f), 1.5f);
```

`LeanTween.rotateAround` (gameObject:GameObject , vec:Vector3 , degrees:float , time:float) LTDescr

Defined in `LeanTween.cs:2044`

Rotate a GameObject around a certain Axis (the best method to use when you want to rotate beyond 180 degrees)

Parameters:

- **`gameObject:GameObject`** GameObject
  Gameobject that you wish to rotate

- **`vec:Vector3`** Vector3
  axis in which to rotate around ex: Vector3.up

- **`degrees:float`** Float
  the degrees in which to rotate

- **`time:float`** Float
  time The time to complete the rotation in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

`LeanTween.rotateLocal` (gameObject:GameObject , to:Vector3 , time:float) LTDescr

Defined in `LeanTween.cs:1992`

Rotate a GameObject in the objects local space (on the transforms localEulerAngles object)

Parameters:

- **`gameObject:GameObject`** GameObject
  Gameobject that you wish to rotate

- **`to:Vector3`** Vector3
  The final rotation with which to rotate to

- **`time:float`** Float
  The time to complete the rotation in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

`LeanTween.rotateX` (GameObject , to:float , time:float) LTDescr

Defined in `LeanTween.cs:2005`

Rotate a GameObject only on the X axis

Parameters:

- **`GameObject`** GameObject
  Gameobject that you wish to rotate

- **`to:float`** Float
  The final x-axis rotation with which to rotate

- **`time:float`** Float
  The time to complete the rotation in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

`LeanTween.rotateY` (GameObject , to:float , time:float) LTDescr

Defined in `LeanTween.cs:2018`

Rotate a GameObject only on the Y axis

Parameters:

- **`GameObject`** GameObject
  Gameobject that you wish to rotate

- **`to:float`** Float

- **to:float** Float
  The final y-axis rotation with which to rotate

- **time:float** Float
  The time to complete the rotation in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

LeanTween.rotateZ (GameObject , to:float , time:float) LTDescr
Defined in LeanTween.cs:2031

Rotate a GameObject only on the Z axis

Parameters:

- **GameObject** GameObject
  Gameobject that you wish to rotate

- **to:float** Float
  The final z-axis rotation with which to rotate

- **time:float** Float
  The time to complete the rotation in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

LeanTween.scale (gameObject:GameObject , vec:Vector3 , time:float) LTDescr
Defined in LeanTween.cs:2058

Scale a GameObject to a certain size

Parameters:

- **gameObject:GameObject** GameObject
  gameObject Gameobject that you wish to scale

- **vec:Vector3** Vector3
  to The size with which to tween to

- **time:float** Float
  time The time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

LeanTween.scale (GUI) (LTRect , Vector2 , float) LTDescr
Defined in LeanTween.cs:2071

Scale a GUI Element to a certain width and height

Parameters:

- **LTRect** LTRect
  ltRect LTRect object that you wish to move

- **Vector2** Vector2
  to The final width and height to scale to (pixel based)

- **float** Float
  time The time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

Example:

Example Javascript:

```
var bRect:LTRect = new LTRect( 0, 0, 100, 50 );
LeanTween.scale( bRect, Vector2(bRect.rect.width, bRect.rect.height) * 1.3, 0.25
).setEase(LeanTweenType.easeOutBounce);
function OnGUI(){
  if(GUI.Button(bRect.rect, "Scale")){ }
}
```

Example C#:

```
LTRect bRect = new LTRect( 0f, 0f, 100f, 50f );
LeanTween.scale( bRect, new Vector2(150f,75f), 0.25f ).setEase(LeanTweenType.easeOutBounce);
void OnGUI(){
  if(GUI.Button(bRect.rect, "Scale")){ }
}
```

`LeanTween.scaleX` (`gameObject:GameObject` , `scaleTo:float` , `time:float`) LTDescr

Defined in `LeanTween.cs:2098`

Scale a GameObject to a certain size along the x-axis only

Parameters:

- **`gameObject:GameObject`** GameObject
  Gameobject that you wish to scale

- **`scaleTo:float`** Float
  the size with which to scale to

- **`time:float`** Float
  the time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

`LeanTween.scaleY` (`gameObject:GameObject` , `scaleTo:float` , `time:float`) LTDescr

Defined in `LeanTween.cs:2111`

Scale a GameObject to a certain size along the y-axis only

Parameters:

- **`gameObject:GameObject`** GameObject
  Gameobject that you wish to scale

- **`scaleTo:float`** Float
  the size with which to scale to

- **`time:float`** Float
  the time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

`LeanTween.scaleZ` (`gameObject:GameObject` , `scaleTo:float` , `time:float`) LTDescr

Defined in `LeanTween.cs:2124`

Scale a GameObject to a certain size along the z-axis only

Parameters:

- **`gameObject:GameObject`** GameObject
  Gameobject that you wish to scale

- **`scaleTo:float`** Float
  the size with which to scale to

- **`time:float`** Float
  the time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

`LeanTween.value (float)` (`GameObject` , `callOnUpdate:Action<float>` , `float` , `float` , `float`) LTDescr

Defined in `LeanTween.cs:2137`

Tween any particular value, it does not need to be tied to any particular type or GameObject

Parameters:

- **`GameObject`** GameObject
  gameObject GameObject with which to tie the tweening with. This is only used when you need to cancel this tween, it does not actually perform any operations on this gameObject

- **`callOnUpdate:Action<float>`** Action
  The function that is called on every Update frame, this function needs to accept a float value ex: function updateValue(float val){ }

- **`float`** Float
  from The original value to start the tween from

- **`float`** Float
  to The value to end the tween on

- **`float`** Float
  time The time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

`LeanTween.value (float,object)` (`gameObject:GameObject` , `callOnUpdate:Action<float,object>` , `from:Vector3` , `to:Vector3` , `time:float`) LTDescr

Defined in `LeanTween.cs:2167`

Tween any particular value (float)

Parameters:

- **`gameObject:GameObject`** GameObject
  Gameobject that you wish to attach the tween to

- **`callOnUpdate:Action<float,object>`** Action
  The function that is called on every Update frame, this function needs to accept a float value ex: function updateValue(Vector3 val, object obj){ }

- **`from:Vector3`** Float
  The original value to start the tween from

- **`to:Vector3`** Vector3
  The final Vector3 with which to tween to

- **`time:float`** Float
  The time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween

`LeanTween.value (Vector3)` (`gameObject:GameObject` , `callOnUpdate:Action<Vector3>` , `from:Vector3` , `to:Vector3` , `time:float`) LTDescr

Defined in `LeanTween.cs:2152`

Tween any particular value (Vector3), this could be used to tween an arbitrary value like a material color

Parameters:

- **`gameObject:GameObject`** GameObject
  Gameobject that you wish to attach the tween to

- **`callOnUpdate:Action<Vector3>`** Action
  The function that is called on every Update frame, this function needs to accept a float value ex: function updateValue(Vector3 val){ }

- **`from:Vector3`** Float
  The original value to start the tween from

- **to:Vector3** Vector3
  The final Vector3 with which to tween to

- **time:float** Float
  The time to complete the tween in

Returns:

LTDescr:

LTDescr an object that distinguishes the tween