Animation Targets, and Clearing Targets

Clearing a target can be very handy on objects with very simple behaviours. I recently had a requirement where an object would perform an animation every time another object collided with it.

As per other tutorials, my code called *orxObject_SetTargetAnim* and away it animated. However, because the object was not under any kind of input control, there was no opportunity to clear the target, and therefore, the object animated over and over.

This is where the *cleartarget* symbol becomes useful.

Let's work through an example to see how it works.

See below the following spritesheet of a spring loaded bumper.



The job of this bumper is to spring up and knock away any objects that collide with it. After it has sprung up, it needs to compress down again and stay compressed until the next collision.

First, make the object, graphics, and animations (body collision information left out):

```
[BumperObject]
Graphic         = BumperGraphic ; Only used to supply a default size for all
frames. If not supplied, the first animation frame will supply the size.
AnimationSet = BumperAnimationSetAll
Position     = (100, 100, -0.1)
```

```
[BumperGraphic]
Texture     = bumper.png
Pivot       = top left
TextureSize = (114, 117, 0)

[BumperAnimationSetAll]
Texture                = bumper.png
TextureOrigin          = (0,0,0)
FrameSize              = (114, 117, 0)
KeyDuration            = 0.05
BumperCompressAnim     = 20
BumperCompressedAnim   = 1
BumperSpringUpAnim     = 20
StartAnim              = BumperCompressedAnim
BumperCompressedAnim-> = BumperCompressedAnim # BumperSpringUpAnim
BumperSpringUpAnim->   = BumperCompressAnim
BumperCompressAnim->   = BumperCompressedAnim

[BumperCompressedAnim]
TextureOrigin = (456, 351, 0)

[BumperSpringUpAnim]
Direction   = left # up
```

So that completes the animation setup.

In code, the bumper is defined at the top of the code:

```
orxOBJECT *bumperObject;
```

and created in the Init() function with:

```
bumperObject = orxObject_CreateFromConfig("BumperObject");
```

We'll use a key to set the animation target, but provide no way to clear the animation target in code. Create this following key map:

```
[MainInput]
KEY_ESCAPE = Quit
KEY_SPACE  = Activate
```

Add the following to the Run() function to start the animation target:

```
  if (orxInput_IsActive("Activate") == orxTRUE &&
orxInput_HasNewStatus("Activate") == orxTRUE)
  {
      orxObject_SetTargetAnim(bumperObject, "BumperSpringUpAnim");
```

```
  }
```

Compile and Run.

The bumper target is being set as default to *BumperSpringUpAnim* which means that the animation graph system will always seek it constantly. *BumperSpringUpAnim* will always remain the target, and therefore be sought, until found again, or otherwise cleared.

In code, if the object was under keyboard control, you could issue the following on keyup to clear a target from an object:

```
orxObject_SetTargetAnim(bumperObject, orxNULL);
```

But our object is not cleared from code. We want it to clear after our target animation has completed.

Note that when the object is first initialised, “BumperCompressedAnim” is the animation that will play. This is because it is specified in the `StartAnim` property.

So in the present state, setting the target to “BumperSpringUpAnim”, the animation graph will take the object through the following chain of animations trying to get to “BumperSpringUpAnim”:

1. BumperCompressedAnim
2. BumperSpringUpAnim
3. BumperCompressAnim
4. BumperCompressedAnim
5. BumperSpringUpAnim
6. BumperCompressAnim
7. BumperCompressedAnim
8. BumperSpringUpAnim
9. BumperCompressAnim
10. etc etc etc..

See how it is caught in a loop?

The secret is to change the list of animations linked to `BumperCompressedAnim` by adding the cleartarget marker. By changing this line:

```
BumperCompressedAnim-> = BumperCompressedAnim # BumperSpringUpAnim
```

```
BumperCompressedAnim-> = BumperCompressedAnim # !BumperSpringUpAnim
```

Notice the `!` symbol? In short, this means, once the `BumperSpringUpAnim` plays, clear the target.

That little `!` is the whole the point to this article.

The object will no longer have a target, and *BumperCompressedAnim* will be the last animation to play.

That is until the object is given the animation target of `BumperSpringUpAnim` again.

This is a very handy symbol for objects that animate through once.

From:
https://www.orx-project.org/wiki/ - **Orx Learning**

Permanent link:
https://www.orx-project.org/wiki/en/tutorials/animation/animation_clearing_targets

Last update: **2025/09/30 17:26 (14 months ago)**