|  |  |
| --- | --- |
| **翻以前字数** | **369** |
| **翻译后字数** | **778** |
| **目录** | **Unity Manual/Animation/Animator Controllers/Animation Blend Shapes** |
| **链接** | <https://docs.unity3d.com/540/Documentation/Manual/BlendShapes.html> |

**Animation Blend Shapes**

**动画混合形状**

**Preparing the Artwork**

**准备美术素材**

Once you have your Blend Shapes setup in Maya:

如果您在Maya中设置了混合形状：

* Export your selection to fbx ensuring the animation box is checked and blend Shapes under deformed models is checked.
* 导出您选择的fbx时确保动画和变形模型下的混合形状的复选框被选中。
* Import your fbx file into Unity (assets->import new assets->[name of file].fbx).
* 导入fbx到Unity中（Assets->Import New Assets->【文件名】.fbx）。
* Drag the asset into the hierarchy window. If you select your object in the hierarchy and look in the inspector, you will see your Blend Shapes are listed under the SkinnedMeshRenderer component. Here you can adjust the influence of the blend shape to the default shape, 0 means the blend shape has no influence and 100 means the blend shape has full influence.
* 把资源拖拽到层级视图窗口中。如果您在层级视图窗口中选择了这个物体，在展示面板中您将看到在SkinnedMeshRenderer组件下混合形状的列表。您可以调整影响混合形状的默认形状，0意味着混合形状不影响默认形状，100意味着混合形状完全影响默认形状。

**Create Animations In Unity**

**在Unity中创建动画**

It is also possible to use the Animation window in Unity to create a blend animation, here are the steps:

也可以使用Unity中的动画窗口创建混合动画，按照以下步骤：

* Open the Animation window under Window->Animation.
* 通过Window->Animation打开动画窗口。
* On the left of the window click ‘Add Curve’ and add a Blend Shape which will be under Skinned Mesh Renderer.
* 在窗口左侧点击“Add Curve”并在Skinned Mesh Renderer下添加混合形状。

From here you can manipulate the keyframes and Blend Weights to create the required animation.

在这里您可以操作关键帧和调整混合比重创建所需要的动画。

Once you are finished editing your animation you can click play in the editor window or the animation window to preview your animation.

当您完成修改动画，通过点击编辑器窗口的播放按钮或者在动画窗口里预览您的动画。

**Scripting Access**

**脚本访问**

It’s also possible to set the blend weights through code using functions like GetBlendShapeWeight and SetBlendShapeWeight.

也可以通过代码使用GetBlendShapeWeight和SetBlendShapeWeight的函数来设置混合比重。

You can also check how many blend shapes a Mesh has on it by accessing the blendShapeCount variable along with other useful functions.

您也可以通过访问blendShape计数变量以及其他有用的函数来检查网格对其形状有多少混合。

Here is an example of code which blends a default shape into two other Blend Shapes over time when attached to a gameobject that has 3 or more blend shapes:

当游戏对象附有3个或者3个以上的混合形状时，将默认形状与其他两个混合形状融合在一起的代码示例。

//Using C#

using UnityEngine;

using System.Collections;

public class BlendShapeExample : MonoBehaviour

{

int blendShapeCount;

SkinnedMeshRenderer skinnedMeshRenderer;

Mesh skinnedMesh;

float blendOne = 0f;

float blendTwo = 0f;

float blendSpeed = 1f;

bool blendOneFinished = false;

void Awake ()

{

skinnedMeshRenderer = GetComponent<SkinnedMeshRenderer> ();

skinnedMesh = GetComponent<SkinnedMeshRenderer> ().sharedMesh;

}

void Start ()

{

blendShapeCount = skinnedMesh.blendShapeCount;

}

void Update ()

{

if (blendShapeCount > 2) {

if (blendOne < 100f) {

skinnedMeshRenderer.SetBlendShapeWeight (0, blendOne);

blendOne += blendSpeed;

} else {

blendOneFinished = true;

}

if (blendOneFinished == true && blendTwo < 100f) {

skinnedMeshRenderer.SetBlendShapeWeight (1, blendTwo);

blendTwo += blendSpeed;

}

}

}

}