



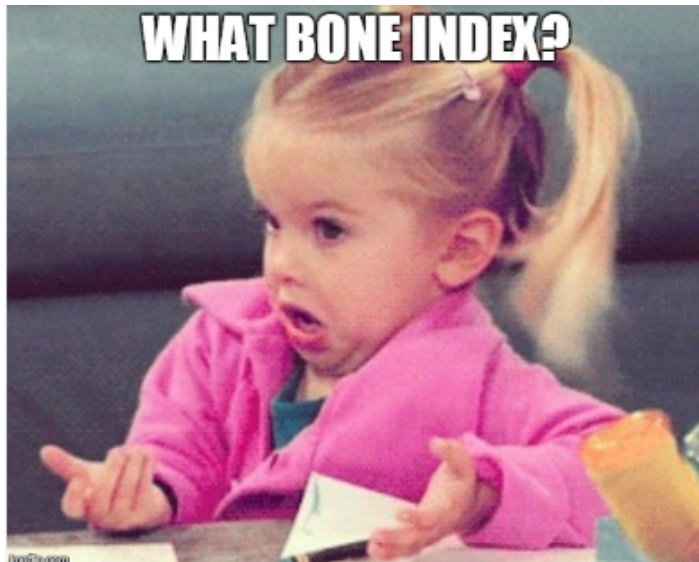
2016.4.29

骨架节点索引揭秘

功能 | 学习 | 美术

作者 Lina Halper

当工作在动画部分的代码上时，经常会看到骨骼索引（bone index）。由于实际上有几种不同类型的骨骼索引，因此刚开始看动画相关代码的时候会很困惑。正是这个原因，我想来给大家解释一下为何有这样几种不同的骨骼索引，以及它们之间有什么区别。



一共有三种主要的骨骼索引，在浏览动画相关代码是理解它们是十分必要的。它们是

1. 网格体骨骼索引（Mesh Bone Index）
2. 骨架骨骼索引（Skeleton Bone Index）
3. 压缩姿势骨骼索引（FCompactPoseBoneIndex）

网格体骨骼索引 (Mesh Bone Index) 是骨架网格体 (SkeletalMesh) 的一部分。其中包含所有关节，连接及依赖顺序，并且在渲染 SkeletalMesh 时会被使用。要查看这个网格体骨骼索引，只需要将导入的 SkeletalMesh 在 Persona 中打开，便能看到所有它所包含的关节信息。

可以通过网格体骨骼索引来访问 SkeletalMeshComponent 中的 SpaceBases 或 LocalAtoms。



既然说网格体骨骼索引是 SkeletalMesh 的一部分了，那么**骨架骨骼索引 (Skeleton Bone Index)** 又是什么呢？如果用过 Persona 的话，那么一定已经知道 USkeleton 这类资源。这类资源会在导入 SkeletalMesh 时被创建，或者在指定一个已存在的 SkeletalMesh 时被创建。USkeleton 是一个 USkeletalMesh 所有关节的集合。如果您不是很清楚为何骨架中包含多个网格体关节的话，请看一下 [Wes Bunn 的这个视频](#)，其中做了详细的解说。

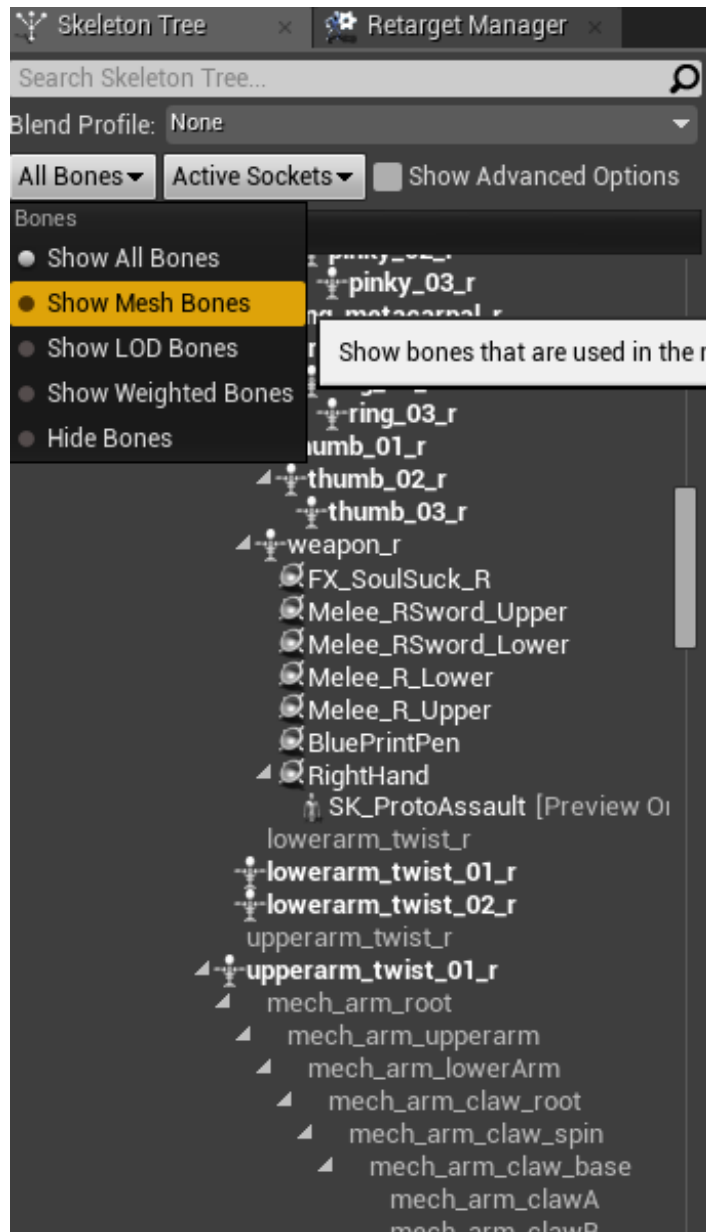
在虚幻引擎 4 中，USkeleton 资源为骨架的每个骨骼节点包含了一个参考姿势。骨架骨骼索引是这些骨骼列表的索引。在每个动画序列数据中，记录着内部的索引和骨架骨骼索引的关系。

我们使用这些来为正确的骨骼节点播放正确的动画数据，然而导致的后果是当骨架发生改变的时候，我们必须调整动画序列帧里的数据信息。这里会导致但修改骨架数据的时候，相应的动画数据会被标记为“Dirty”并需要重新保存。

最初的计划是只使用 USkeleton 来解压动画数据，并且控制解压后的数据和相应 SkeletalMeshComponent 能够对应正确。这粗看是个非常好的想法，然而实际上并不好，因为我们需要在 SkeletalControls 工作的时候就解压模型数据。比如，当使用 IK 来处理多个物体时，就意味着骨架的骨骼节点位置数据就没有意义了。

骨架的骨骼参考姿势的变化数据只有在动画重定向时使用。

如果在 Persona 中打开一个骨架是，可能看到如下信息：



默认情况下，会显示所有骨骼，这些是 USkeleton 中的骨骼。如果选择了“Show Mesh Bone”后，将只会显示当前预览的模型所具有的骨骼信息。

这里就能看到 USkeleton 的骨骼索引和 USkeletalMesh 的骨骼索引并不是一致的。那么 FCompactPoseBoneIndex 又是做什么的？

FCompactPoseBoneIndex 是一组只能用于 FCompactPose 的索引。FCompactPose 是运行时的骨骼设置组合。它是一组连续的列表，只包含当前模型 LOD 下的关节信息。当存在多个 LOD 时，可以对一些骨骼节点进行优化。在 FCompactPose 之前，使用过 branch，但是由于 branch 的不可预测性会带来性能上的问题。FCompactPoseBoneIndex 能让我们避免使用 Brancher 来遍历。

FCompactPoseBoneIndex 本质就是一个整型。它具有自己类型定义更多的是为了代码上的明确性。如果一个函数使用这个类型作为参数，或者返回这个类型，那么很明确的就能知道这里是骨骼索引在进行工作。同时也意味着不会错误的调用一个函数，在原本应该调用其他索引的地方调用了 compact 索引，也不会在应该调用 compact 索引的地方调用其他索引，编译器就会检查发现这个错误。

我们核心的动画代码都会用到 FCompactPose。在 API 上大家可以看到这三种类型的骨骼索引都会被用到。可以在 USkeleton 中找到 SkeletalMesh 和 Skeleton 互相转换的大部分 API (FSkeletonToMeshLinkup) 。

小结一下：

1. **网格体骨骼索引 (Mesh Bone Index)** - 网格体的骨骼索引，用于 SkeletalMeshComponent 的 SpaceBases 或 LocalAtoms 中。
2. **骨架骨骼索引 (Skeleton Bone Index)** - 骨架骨骼索引是所有 SkeletalMesh 的关节集合。用于动画数据跟踪索引。动画数据根据这个索引来更新数据。
3. **FCompactPoseBoneIndex** - 这是一个所有关节的子集，在 Mesh 的 LOD 上工作。比如 LOD 0 会和 MeshIndex 匹配，然而 LOD 1 或者 LOD 2 则只会是其中的一部分而不匹配所有 MeshIndex。

这里是一个从 CompactPoseBoneIndex 转换到 Mesh Bone 的示例。

```
// iterate through all compact pose
for (FCompactPoseBoneIndex BoneIndex : Output.Pose.ForEachBoneIndex())
{
    // ask mesh pose for the compact pose
    FMeshPoseBoneIndex MeshPoseBoneIndex =
    Output.Pose.GetBoneContainer().MakeMeshPoseIndex(BoneIndex);

    // assuming local mesh buffer is your buffer for skeletal mesh
    Output.Pose[BoneIndex] = LocalMeshBuffer[MeshPoseBoneIndex.GetInt()];
}
```

最后，其实 USkeleton 并不是表达它最佳的名称。[这个视频](#)应该能帮助大家了解 USkeleton 的更多细节。

希望这篇文章对大家理解骨骼索引的不同类型有帮助。

最近文章

2017年12月25日

Unreal Engine 4 的 光和影

2017年12月19日

独立团队的Xbox发布流程

2017年12月19日

4.19版本VR分辨率设
化

内容

排序: [最新最前](#) 最早最前 最多好评

请登录来进行评价



(<https://twitter.com/unrealengine>)



(<https://www.facebook.com/UnrealEngine>)



(<http://www.twitch.tv/unrealengine>)



(<http://instagram.com/UnrealEngine>)



(<http://www.youtube.com/unrealengine>)



(<https://www.unrealengine.com/rss>)



(<http://weibo.com/unrealengine>)

功能 (/features)

图标和品牌 (/branding)

开发路线图 (<https://trello.com/b/gHooNW9I/ue4-roadmap>)

教育 (/education)

教育合作伙伴 (/academic-partners)

资源 (/resources)

所获荣誉 (/awards)

新闻发布站 (<http://epic.gamespress.com/>)

EpicGames.com (<http://epicgames.com/>)

注册后便能获取虚幻引擎的新闻 ()

© 2004-2018, Epic Games, Inc. All rights reserved. Unreal and its logo are Epic's trademarks or registered trademarks in the US and elsewhere.

服务条款 (<http://epicgames.com/tos>) | 隐私声明 (<http://epicgames.com/privacypolicy>)



(<https://epicgames.com>)



