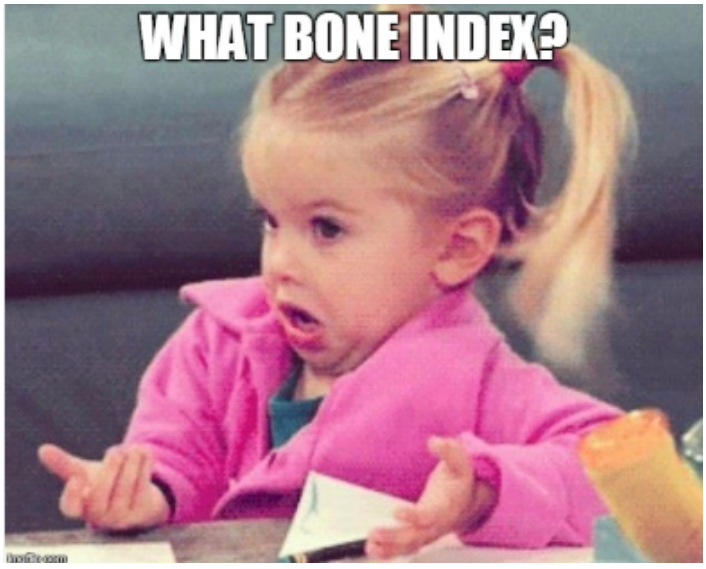
4.29.2016

Demystifying Bone Indices

Art | Features | Learning

By Lina Halper

When working on animation code, what you are often looking at is the bone index. As there are multiple different bone indices, it can be confusing to look through animation code for the first time. Because of this, I would like to explain why there are multiple bone indices and what the differences are between the bone indices.



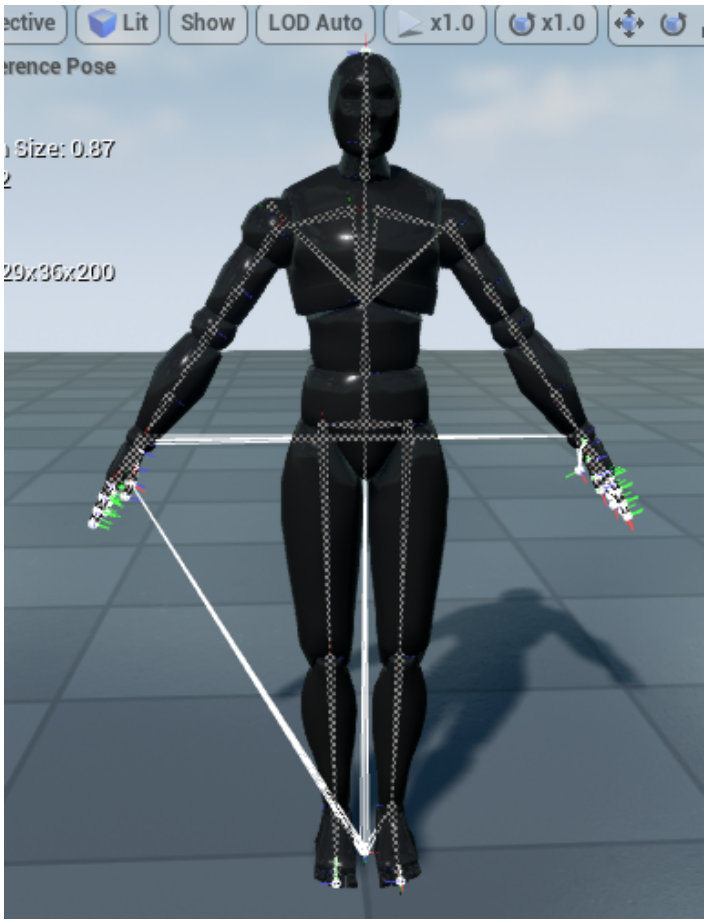
There are 3 major bone indices which are important when looking at animation code. These are

1. Mesh Bone Index
2. Skeleton Bone Index

3. FCompactPoseBoneIndex

The **Mesh Bone Index** is part of the SkeletalMesh. It contains all the mesh joints, the order of the hierarchy and will be used when rendering out the SkeletalMesh. To view the Mesh Bone Index, import the SkeletalMesh and open in Persona. There you can then see all of the joints that are contained.

You would use the Mesh Bone Index to access the SpaceBases or LocalAtoms in the SkeletalMeshComponent.



If the Mesh Bone Index is part of the SkeletalMesh then what is the **Skeleton Bone Index**? If you have used Persona, you'll already know about USkeleton assets. These are either created when you import a SkeletalMesh or assigned using an existing one. The USkeleton asset is a union of all the joints of a USkeletalMesh. If you are unsure as to why

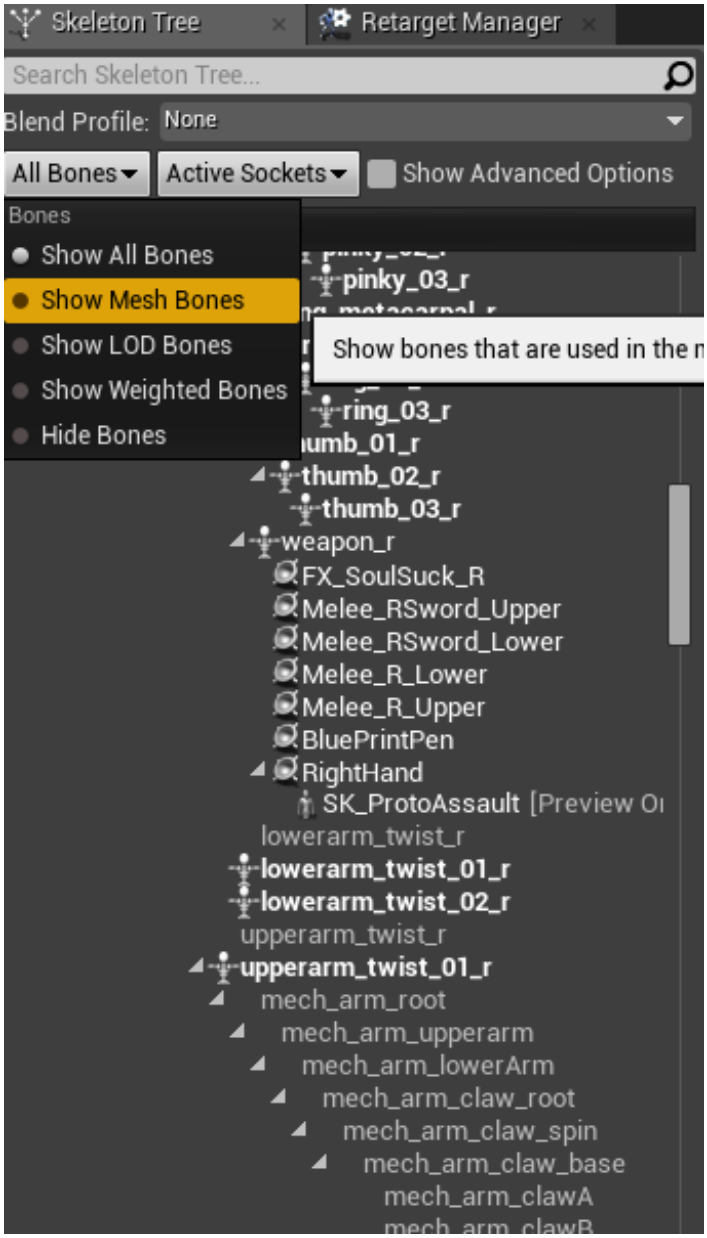
skeletons contain multiple meshes' joints, please check out this video from Wes Bunn who explains this is more detail.

In Unreal Engine 4, USkeleton assets contain a reference pose (USkeleton.ReferenceSkeleton) for every bone that a skeleton has. The Skeleton Bone Index is the index into this list of bones. Inside of each UAnimSequence the mapping between the internal track indices and the matching Skeleton Bone Index is stored. We use this during our evaluation to apply the right animation to the corresponding bone, but as a consequence of this, we have to update this data on animations when the skeleton changes. This can cause the animation to be marked as 'dirty' and need to be resaved when a change to the Skeleton is made.

The original plan was to use USkeleton only to extract animation data and evaluate which could then be retargeted to the appropriate SkeletalMeshComponent. This sounds like a wonderful idea but didn't quite work out because we need exact mesh data when working with SkeletalControls.

The Skeleton's ref pose bone transform is only used for retargeting with Animation Scaled.

If you open Skeleton window in Persona, you'll see this:



By default, it displays all the bones, which are the bones in the USkeleton. If you change to “Show Mesh Bones”, it will only show the bones that belong to the current preview mesh.

So now you can see USkeleton bone index does not match with USkeletalMesh bone index, but where does the FCompactPoseBoneIndex fit in?

FCompactPoseBoneIndex is an index that can only be used on an FCompactPose. FCompactPose is the runtime bone set that will be used during the evaluation. It is a contiguous list that contains only joints that matter for the current mesh LOD. When you have multiple LODs, you can only have a subset of joints for optimization. Prior to FCompactPose, branches were used, however, branches caused performance issues

because of branch mispredictions. FCompactPoseBoneIndex allows us to iterate without branches.

FCompactPoseBoneIndex is ultimately just an integer. The reason it is its own type is for code clarity. If a function takes or returns a FCompactPoseBoneIndex then you know exactly what space that bone index works in. It also means that you can't accidently call a function intending to use a compact index with a mesh or skeleton index (and vice versa) as the compiler will reject it.

All of our core animation code uses FCompactPose. You can go between all of these 3 bone indices using APIs. Most conversion APIs between SkeletalMesh and Skeleton are found in USkeleton. (FSkeletonToMeshLinkup)

To sum up:

1. **Mesh Bone Index** - SkeletalMesh Bone Index used in SpaceBases or LocalAtoms of SkeletalMeshComponent.
2. **Skeleton Bone Index** - Skeleton Bone Index contains the union of joints of all SkeletalMeshes. Essentially, this is an animation track index. Animations reference this index for track data.
3. **FCompactPoseBoneIndex** - This is used for the subset of all joints that are part of the Meshes LOD support - i.e. LOD 0 will match with MeshIndex, but once it goes LOD 1 or LOD2, it can only have subset, thus no longer matches with MeshIndex.

This is one of the examples of converting to Mesh bone from CompactPoseBoneIndex:

```
// 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 skeletalmesh
    Output.Pose[BoneIndex] = LocalMeshBuffer[MeshPoseBoneIndex.GetInt()];
}
```

As a final note, USkeleton isn't the most indicative name for what it is. This video should help you to understand what a USkeleton is in more detail.

Hopefully this has helped you to understand what the different types of bone indices are and how you can use them.

RECENT POSTS

Jan 1, 2018

Getting Started with Unreal Multiplayer in C++

Dec 30, 2017

Unreal Engine Developers Featured in IndieDB's Top 100 of 2017

Dec 30, 2017

Celebrating an Unr

COMMENTS

SORTED BY: NEWEST OLDEST HIGHEST RATED

LOGIN TO COMMENT


(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)

Features (/features)

Logo & Branding (/branding)

Roadmap (https://trello.com/b/gHooNW9I/ue4-roadmap)

Education (/education)

Academic Partners (/academic-partners)

More Resources (/resources)

Awards (/awards)

Press Site (http://epic.gamespress.com/)

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

SIGN UP FOR UNREAL ENGINE NEWS ()
