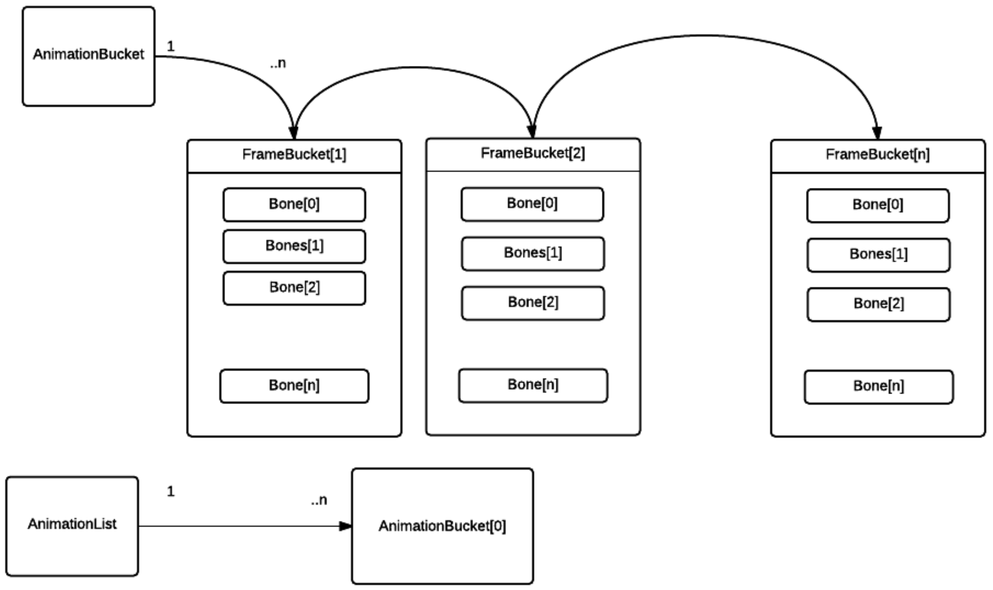
ANIMATION SYSTEM

Converter Overview:

Converter for animation has three structures to create file that used in Game Engine. Lowest layer is bone data, it contains required data for one. Second layer is timeframe data that contains list of bones and frame time as integer. List of bones may vary according to model. Upper level is animation data that contains timeframe list and total number of frame data. This structure have all needed data for just one animation. Because model may have more than one animation highest level is list of animation. Designing this file system is not hard but I struggle a lot when I try to implement this design. I just create this architecture then push data accordingly in already written functions that print output.

Hardest part of this standalone project is how to use SDK. Unfortunately, because lack of my knowledge of SDK most of my code is hard coded for teddy and humanoid, little tricks all over place. For example for empty bones data teddy bear does not give problem at all but in humanoid even data is empty we still need curved data from that frame rate. Eliminating useless data like effectors or meshes I am checking its type if its limb node then I push that data to the list. But I get another problem for humanoid hierarchy. For just one file that contains three humanoid animations insert mesh in to the list even I checked before. So I put another condition that returns if its type is Mesh.

This converter is separate from milestone one converter. I don’t use archiver at all. So, it does not have a feature to convert model data and animation data together.



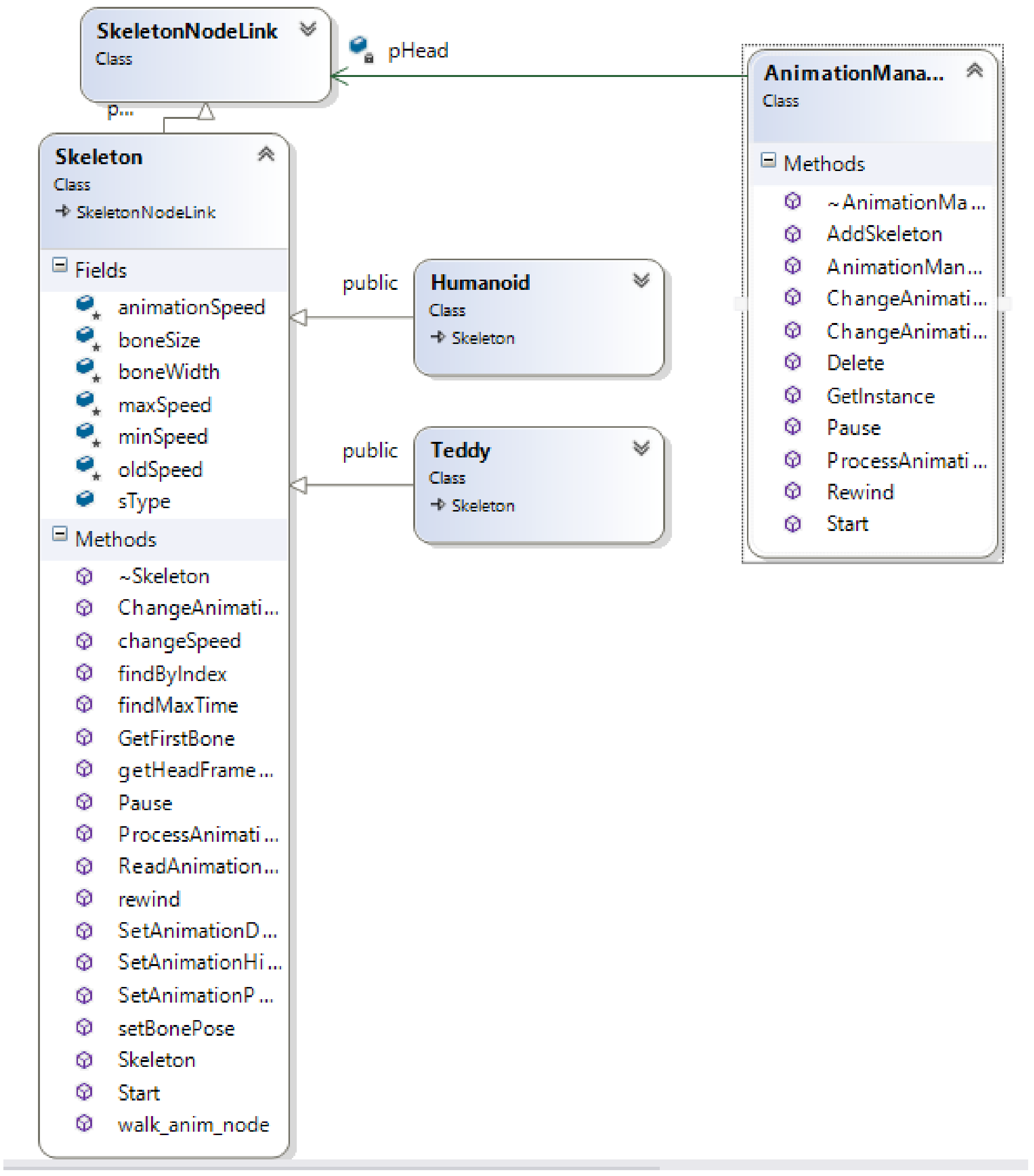
Game Engine Overview:

After successfully load correct animation data and see some animation on screen first thing to I do in game engine put anim2bone and anim file function to class. Also I created own version of classes that using in animation. So if I mess up I can still go back to beginning. I created teddy class same that contains necessary functions like setting hierarchy, setting animation data … most of Anim2bone functions. Next job was remove all global data. I put all global data to teddy class. SetAnimationBone function that process animation and creating skeleton look for our bones now doesn't need first bone as parameter. Getting rid of extern FrameBucket little bit trickier. Because now Teddy object contain class that has process Animation function, now process animation take framebucket head as a parameter it also take bone number as parameter so I can use that function to process animation of two or more Skeletons. FrameBucket Extern in GameObject is harder to rid of. Now every GameObject also have pointer to Skeleton object. Skeleton create bone object and set its Skeleton to himself.Another way to solve this problem set framebucket in root and all other children use that bucket to get bone data.

After completing this task I created another animation model class for humanoid. And these two classes nearly identical. Implement strategy pattern may reduce too many repeated code. Only differences are bone number and bonewidth, I develop this way because   it gives easy to debug. Its much more easy to see teddy or humanoid broken and where. These two classes inherit from Skeleton class. Skeleton class is just interface that have mostly pure virtual functions to use in GameAnimationManager. Skeleton class also inherit SkeletonList class to create list of Skeleton. Problem I encounter after I finished writing these classes and hope to see something on screen was object are not move at all. If only one object created it works flawlessly but not for two. I solved this problem use own Time object for tcurrent that using in AnimationProcess and setBone. Because every Skeleton has animating different frame rate. Just use one is set Max time for looping is works for one but other not for another.

Lastly I create animation manager for removing global objects and ability to play with animation(Backward, SlowDown, Speed up..).

This engine based on Proof's engine. It is my prototype before writing for my engine. Unfortunately I couldn't find time to implement to my engine.


SkeletonNodeLink
Class
pHead
p...
AnimationMana...
Class
Methods
~AnimationMa...
AddSkeleton
AnimationMan...
ChangeAnimati...
ChangeAnimati...
Delete
GetInstance
Pause
ProcessAnimati...
Rewind
Start
Skeleton
Class
SkeletonNodeLink
Fields
animationSpeed
boneSize
boneWidth
maxSpeed
minSpeed
oldSpeed
sType
Methods
~Skeleton
ChangeAnimati...
changeSpeed
findByIndex
findMaxTime
GetFirstBone
getHeadFrame...
Pause
ProcessAnimati...
ReadAnimation...
rewind
SetAnimationD...
SetAnimationHi...
SetAnimationP...
setBonePose
Skeleton
Start
walk_anim_node
public
Humanoid
Class
Skeleton
public
Teddy
Class
Skeleton