When should Game Developers use Vertex Shader Animations over Skeletal Animations?

COMP130 - Software Engineering

1703086

March 20, 2018

This essay will be comparing two popular animation techniques and reviewing how vertex animation could be used as a better creative and cheaper alternative to skeletal animation

1 Introduction

Animation is an integral part of video games, and without it, video games tend to be quite visually dull and boring. There are many different ways animation techniques that can be done in games these days.

This essay will be looking at 2 popular 3D animation techniques used in games development, and will be comparing the two to find out when and why each animation technique should be used in order to save development time for game developers.

This essay will also will review and look at cases in which vertex animation was used as a better and cheaper alternative to animating game objects that would normally be animated using skeletal animation.

2 Vertex shader VS Skeletal

2.1 Quick overview on what Vertex shader and Skeletal animations are

Skeletal animation is the process of animating a mesh that is skin weighted to an armature or bone rig, the process is most oftenly done in animation software such as Maya or blender. Skeltal animation is also a type of vertex shader, so it can work hand in hand with other shaders, but alone it is fairly limited.[10] Skeletal animation works by tweening/blending between different poses (keyframes) set at different times of the animation, the action of tweening/blending generates the other poses between the keyframes, filling in the animation with every single keyframed pose needed to create the illusion of animated movement.

Vertex shader animation is the process of animating a mesh with the use of applying a shader script. Shaders are a programmable part of the graphics pipeline and are created with the use of shading languages such as OpenGL's shading language GLSL or DirectX's High-Level Shader Language HLSL.[12] The way a vertex shader works is that it modifies the process of which it is rendered during the rendering pipeline it transforms each individual vertices of the mesh per rendered frame creating the illusion of animated movement.[9]

2.2 What is each animation technique best for?

Skeletal animation is most commonly used to animate characters and creatures as it allows for precise control over a character's different body parts and apparel, it is also used for animating very specific set of movements for an object, such as a robotic arm or a tree.[10]

Vertex shader animation is most commonly used for animating special effects for game objects or for animating the movement of environmental objects such as grass, leaves, bushes, flames, water...ect.[2][3][4][6] Vertex shaders can be very extensive in their use as they can be programmed in many different ways.

2.3 Comparing the two techniques

Comparing these two animation techniques comes down to convenience, Vertex shaders are very inconvenient for creating complex locomotion types of animation for characters as they aren't built to handle precise movement to transform specific vertex areas of a mesh, but they are very convenient when it comes to transforming all the vertices of a mesh at the same time, like creating a wave effect on a mesh plain to make it look like ocean waves or a flag in the wind.
As for skeletal animation it is most convenient for creating very specific character or object movement animations like locomotion cycles, fighting move sets, interactions...ect but skeletal animation is quite inconvenient for creating general purpose animations that can act on more than one mesh/gameobject as they are animated around a specific armature rig, it is also inconvenient for creating animations that affect all areas of the mesh as the mesh is specifically skin weighted to certain bones of the rig.

2.4 Skeletal animation used alongside Vertex shaders

Because Skeletal animation is just a form of a vertex shader, Skeletal animation and vertex shader animation can work great alongside to make very cool animated characters with interesting visual effects.

An example of this being done for a in-game character are these fictional creatures shown in this gif (https://i.imgur.com/U40e9Bm.gif), their main bodies are animated with skeletal animation but their cloud tails have a vertex shader applied to them to give them the continuous puffing up and down effect.

3 Cases in which vertex shader animations were a better solution to skeletal animations

3.1 Abzu

The game Abzu (a game about swimming around in the ocean with lots of fish) had some very creative and innovative ways of animating their fish in order to save performance all while keeping great animation quality.

One of the lead developers of Abzu did a GDC talk about their technical art challenges and how they solved them.[7] One of their key challenges was trying to render over 10,000 animated interactive fish on screen at the same time, the big issue they had was animating the fish with skeletal animation, each fish had their own skeletal rig with about 60 to 100 joints and updating each fish every frame with 10,000 fish on screen meant they had to update around 1,000,000 joints per frame which would be extremely costly for processing wise. So their solution was Static mesh instancing with the use of vertex shaders and blendshapes.

So instead of using skeletal animation to animate swim cycles for the fish they wrote vertex shader scripts that allowed them to animate the fish using a mix of yaw rotations, sin waves and pans to transform the position of all the vertices of the mesh and distort it to look as if the fish is swimming. Because they animated the fish without bones means they were able to successfully render over 10,000 fish on screen without losing performance or frame-rate.

Another great technique they used for more specific animations was blendshapes (also known as morph target animation) which is the process of taking a mesh in two different poses, and then lerping their vertex positions over delta-time to perform the animation, this is essentially what skeletal animation is, just without the bones/joints and skin weighting. Using this technique they could create animations such as bite animations for predator fish, by using the fish's mesh in a open mouth positions and a closed mouth position they could blend between the two poses creating the blendshape animation.
They managed to use this for crab walk cycles and seagull flying cycles all without the use of bones in engine to save performance.

4 Conclusion

In conclusion, both vertex animation and skeletal animation have their strengths and weaknesses depending on the types of animations that need to be performed, so they should be used accordingly. Skeletal animations will most oftenly be the best used for specific or complex character animations, vertex animations will most oftenly be best used for animating environmental game objects or for animating special or simple effects for meshes.

But a note to be made is that skeletal animation can be quite costly performance wise, if you wanted to populate a game with many animated gameobjects or characters, skeletal animation might not be the way to do it, as the game engine would need to update a lot of joints, so coming up with creative ways of using vertex shaders (but not limited to vertex shaders) would be a good idea as they don't require bones.

5 Show references

[1][2][3][4][5][6][7][8][9][10][11][12][13]

References

[1] L. Lingxin, "Research on vertex deformation algorithms of character animation model," pp. 797–800, June 2015.

[2] S. Guerraz, F. Perbet, D. Raulo, F. Faure, and M. P. Cani, "A procedural approach to animate interactive natural sceneries," pp. 73–78, May 2003.

[3] X. Li and M. Xu, "Water simulation based on hlsl," pp. 1066–1069, Nov 2009.

[4] F. Perbet and M.-P. Cani, "Animating prairies in real-time," pp. 103–110, 2001. [Online]. Available: http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/364338.364375

[5] J. Bloomenthal, "Medial-based vertex deformation," pp. 147–151, 2002. [Online]. Available: http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/545261.545285

[6] D. Hinsinger, F. Neyret, and M.-P. Cani, “Interactive animation of ocean waves,” pp. 161–166, 2002. [Online]. Available: http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/545261.545288

[7] M. Nava, “Creating the art of abzu,” [Online]. Available: https://www.youtube.com/watch?v=l9NX06mvp2E, [Accessed: 17-Mar-2017].

[8] R. the ClassicGameJunkie, “How did they do that - crash bandicoot’s cartoony animation,” [Online]. Available: https://www.youtube.com/watch?v=ZrCdnTbMGec, [Accessed: 17-Mar-2017].

[9] J. de Vries, “Shaders,” [Online]. Available: https://learnopengl.com/Getting-started/Shaders, [Accessed: 17-Mar-2017].

[10] S. Cheng, “Human skeleton system animation,” [Online]. Available: https://bib.irb.hr/datoteka/890911.Final_0036473606_56.pdf, [Accessed: 17-Mar-2017].

[11] B. Knowles and O. Fryazinov, “Increasing realism of animated grass in real-time game environments,” pp. 48:1–48:1, 2015. [Online]. Available: http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/2787626.2787660

[12] N/A, “Vertex shaders,” [Online]. Available: http://www.nvidia.co.uk/object/feature_vertexshader.html, [Accessed: 17-Mar-2017].

[13] J. Lindquist, “The inner workings of fortnite’s shader-based procedural animations,” [Online]. Available: https://www.youtube.com/watch?v=7Fl3so0Z5Tc, [Accessed: 17-Mar-2017].