

Notes on Quadruped Rigging

a personal inquiry by Ian Thompson
MSc Computer Animation and Visual Effects

1. Introduction, Motivation and Aims

For our Masters project, myself and two animation students will develop a game. This game will use animals as a means of racing against other player and non-player competitors. As such, a game-compatible rig is required for each animal and rider. Focus in this study falls on the animal rig as a means to explore techniques and develop a useful skill.

I came to this inquiry with very minimal knowledge of rigging, understanding only the basics of inverse kinematics and the system of parented bone chains. I feel I have learnt a great deal and that I am now able to create usable rigs, but also that there is much more I can learn. This project has inspired me to pursue this investigation further and I hope that the rigs I produce with the knowledge I have gained will be beneficial to my future projects.

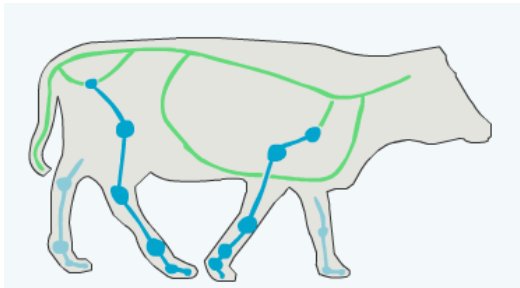
There will be two animals in the game (a cow and pig) which have quite distinct body types. The rig must be able to function in both situations. Working closely with the animators I was able to get feedback and make improvements throughout the development process, and looking at the motions of these animals was another useful reference. I had to allow for adjustments to be made once the rig had been set up, so that the same rig could be used across the different animal models.

Primary aims:

- To provide the desired functionality for the animator to produce realistic animal motion
- To be reusable and adjustable for multiple animal models
- To remain simple enough to be used in a game pipeline

2. Game Research & Animal Motion

While researching rigs I realised how fast game animation is progressing. Starting from sprite-based and vertex-based animation, then later onto weighted rigging systems and animation blending. I was lucky enough to attend a master class by Jo Plaete on the Integrated Creative Environment (ICE) in



XSI. The latter part of this talk mentioned the potential for complex custom rigging constraints created in an entirely visual environment. This certainly is the direction we seem to be heading with node based systems cropping up in various applications and SideFX's Houdini being built entirely upon this principle. Adding expression driven constraints when constructing a rig is now a common practice, but how much of this transfers to the game itself?

When looking at the tools used to develop games there is little difference to those used to produce animation for film. The main difference arises when we export the animation into a game engine. Here we are restricted by what can be achieved in real time and what must be optimised. In the past, all animation was *baked* onto the geometry itself and we were stuck with unblendable, per-vertex animation. Today we have more scope for calculations to be carried out by the engine (such as IK

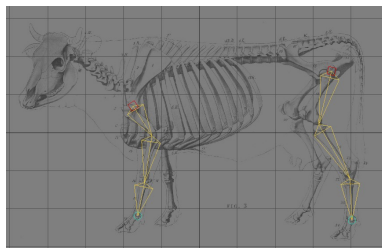
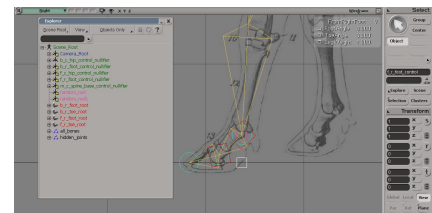
and ragdoll physics) and in the future we could see more simulation and custom constraints creeping in.

The other side to the research was to tie the rig to a specific purpose we had in mind for the game. It was to be a quadrupedal animal rig that could move at a variety of speeds. I watched videos of the chosen animals and observed their limits in the required gaits. I also asked our animator to produce some rough animation, both to test the rig for user-friendliness and to ensure it would stand up to the extremes imposed upon it. This was also an excellent opportunity to test the animation pipeline up to inclusion in the engine.

3. Rig Production in XSI

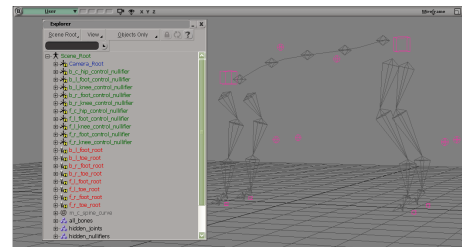
Following tutorials by Saf Efstathiou here at the NCCA and the Digital Tutors' DVD, *Character Setup in XSI*, I was able to learn the principles of rigging in a relatively short space of time. I also picked up a few tricks which I used to my advantage.

It was important for me to understand where all the data was stored for each operation, so where automated tasks were performed (such as mirroring the rig) I looked at the underlying operations and mimicked the process.



XSI stores transformations in several places allowing the user to hide unnecessary information. However when it comes to using the rig outside the XSI environment, the behaviour of these hidden transformations could be troublesome. One particular trick suggested by Saf was to use Nulls to keep any hidden transforms. This was utilised so that the parenting system (common to all modern rigging systems)

could be used instead of the XSI-specific Default Pose construct. The animation curves are then relative to this Null making it easier for the animator, and better prepares the rig for export into the game engine pipeline. To be able to solve the problems I faced, having all the information on the surface was critical. Also establishing good conventions was a key part of the rig production process. Naming conventions alongside XSI's ability to colour-code and shape-code every null object made handling the large hierarchy much easier. Groups were added to provide additional control and allowed different portions of the rig to be grouped for specific tasks. Groups were also used to hide or block certain parts of the rig from selection.



Overall, XSI has a plethora of useful tools and organises its data in a very convenient fashion. Where things are a little more unusual (for example, in chained up vectors) the documentation provides useful information on how this affects the data. There were of course some problems, particularly with XSI's FBX support but I have recently found an excellent workaround which has led to a complete, functioning rig in the engine we are using for our game.

4. References

Books, Articles and Papers

Baran, I. and Popovic, J. 2007. *Automatic rigging and animation of 3D characters*. ACM

Trans. Graph. 26, 3 (Jul. 2007), 72. DOI= <http://doi.acm.org/10.1145/1276377.1276467> [Accessed: 3 May 2009]

Kavan, L. and Zára, J. 2005. *Spherical blend skinning: a real-time deformation of articulated models*. In Proceedings of the 2005 Symposium on interactive 3D Graphics and Games (Washington, District of Columbia, April 03 - 06, 2005). I3D '05. ACM, New York, NY, 9-16. DOI= <http://doi.acm.org/10.1145/1053427.1053429> [Accessed: 6 May 2009]

Smith, J. and White, J. 2006. BlockParty: modular rigging encoded in a geometric volume. In ACM SIGGRAPH 2006 Sketches (Boston, Massachusetts, July 30 - August 03, 2006). SIGGRAPH '06. ACM, New York, NY, 115. DOI= <http://doi.acm.org/10.1145/1179849.1179993> [Accessed: 7 May 2009]

Pilgrim, S. J., Aguado, A., Mitchell, K., and Steed, A. 2006. Progressive skinning for video game character animations. In ACM SIGGRAPH 2006 Sketches (Boston, Massachusetts, July 30 - August 03, 2006). SIGGRAPH '06. ACM, New York, NY, 114. DOI= <http://doi.acm.org/10.1145/1179849.1179992> [Accessed: 7 May 2009]

Bloomenthal, J., Lim, C., 1999. Skeletal Methods of Shape Manipulation. In Shape Modeling Int'l '99, 1999 (invited paper). Available from: <http://www.unchainedgeometry.com/jbloom/papers.html> [Accessed: 14 May 2009]

Software, Video and Images

Digital Tutors., Character Setup in XSI [video]. Available from: <http://www.digitaltutors.com/store/product.php?productid=3537>

Efstathiou, S., 2008. Rigging Legs and Spine Tutorials [video]. (Unpublished)

Anon. 2000s. Cows exit from cattleshed on farm, Mondsee, Austria [video]. Available from: <http://download.bbcmotiongallery.com/quicktime/01/01/23/10123-34.mov> [Accessed: 8 May 2009]

Anon. 2000s. Family of feral pigs wander through Lakefield National Park, Queensland [video]. Available from: http://download.bbcmotiongallery.com/quicktime/00/65/64/6564-2_70.mov [Accessed: 8 May 2009]