Flexible Pipeline for Crowd Production

Mungo Pay
Double Negative
mungo@dneg.com

Damien Maupu
Double Negative
dmu@dneg.com

Martin Pražák
Double Negative
map@dneg.com

Figure 1: Stills from 'The Young Pope' courtesy of Wildside and 'EXODUS: GODS AND KINGS' courtesy of Twentieth Century Fox. All rights reserved. Crowd shots realized in two different variants of the crowd production pipeline.

ABSTRACT

The complexity of crowd shots can vary greatly, from simple vignetting tasks that add life to an environment, to large and complex battle sequences involving thousands of characters. For this reason, a "one size fits all" crowd solution might not be optimal, both in terms of design and usability, but also allocation of crew. In this talk we present a suite of tools, developed across multiple platforms, each optimised for specific crowd tasks. These are underpinned by a data interchange library to allow for modification at any stage of the pipeline.

CCS CONCEPTS

•**Computing methodologies** →**Animation;** *Procedural animation;* Mesh geometry models;

KEYWORDS

Crowd Simulation, Animation

ACM Reference format:
Mungo Pay, Damien Maupu, and Martin Pražák. 2017. Flexible Pipeline for Crowd Production. In *Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3084363.3085042

1 THE TOOLS SUITE

When deciding how to implement a crowd solution for a VFX studio, there are many factors that must be considered: ease of use, speed, extensibility, licence cost, development overhead, pipeline integration to name but a few. For many years, our flagship crowd tool was a standalone, in-house developed simulation system called Riot. Over the past year we have focussed on diversifying our toolset so as to be more adaptable to the kinds of crowd shots we may have to deliver. This has included the development of plugins for both SideFX's Houdini and Autodesk's Maya to best leverage the core functionality of each tool. Another desire was to be able to have tools that could fit more efficiently into our pipeline. By writing these crowd solutions as plugins to our default 3D authoring software, integration with other aspects of our pipeline such as hair, cloth, rigging and FX immediately becomes much more streamlined. In this section we detail the pros and cons of each solution.

1.1 Riot

Designed as a standalone application from the ground up, we are in full control of all aspects of the system, which means it can be heavily optimised for our needs. It can handle hundreds of thousands of agents in a scene before becoming prohibitively slow. The turnaround on new behavioural plugins can be fast because of direct communication between the RND development team, and Riot's users. However, while Riot is extremely powerful, and has delivered multiple shows, there is an overhead on RND resources. Responsibility for all infrastructure, UI elements, pipeline integration and support falls on the crowd development team. Extensibility is also limited as it requires any new functionality to be added by RND programmers, rather than technical artists. From a staffing perspective, it is a separate tool for which training is required.

1.2 Houdini

Houdini is a third-party, particle-based framework that allows artists to procedurally create 3D scenes using a node graph interface, and is our primary tool for all FX work. By leveraging the crowd tools already provided in Houdini, the development of very powerful additional tools could be expedited. This, along with a

SIGGRAPH '17 Talks, Los Angeles, CA, USA
© 2017 Copyright held by the owner/author(s). 978-1-4503-5008-2/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3084363.3085042

close collaboration with senior FX TDs, ensured that the tools were designed to suit Houdini best practices. In addition to this, any pipeline, UI and support work was ameliorated by other development teams. Financial cost is the largest downside to using Houdini; there is the licensing cost, and while the toolset is easy to pick up, it can require either some additional training or the allocation of more technical artists.

1.3 Maya

Maya is a more artistically driven 3D package than Houdini, allowing direct viewport interaction for 3D content creation, and is our primary tool for modelling, rigging, layout and animation. A suite of Maya-based crowd solutions allows for our tools to be used by a broader pool of artists, who can quickly create scenes without the need for extra training. This toolset does not have the power of either Houdini or Riot, as it has been designed to be used for simple placement and duplication of loopable animation clips (vignettes) coupled with basic trajectory alteration. For simple layout of crowd characters, this allows for shots to be constructed extremely quickly with minimal training. It has the added benefit of a more seamless interaction with our Maya-centric rigging framework. This allows users to promote crowd agents to hero rigs for when animation cleanup is required. While, like Houdini, there is a licence cost for Maya, this is nullified due to different licensing schemes. The major downside of our current Maya toolset is its simplicity, only allowing for more basic crowd caches to be produced. However, this is a design decision, as more complex crowd scenes should be created in either Houdini or Riot.

2 AN INTERCHANGE LIBRARY

All of our crowd tools are built on top of an in-house developed core crowd library (mob), which provides functionality for skeletal deformation of linearly skinned geometry. A mob actor consists of a lightweight rig and a geometry set which defines its skinning. Costume variation (or alternate selection) is also handled using the mob library, so geometries can be exchanged to create diverse scenes with no overhead. It also provides a caching format, which stores keyframed poses of our simplified rigs. The caches can be generated in any way that makes sense for the application, usually being created from blended animation files (HTR or Houdini's clip format) applied to our lightweight rigs. The caches are exported as a single file.

Geometric level of detail (LOD) is natively handled in the mob library. This means that in each application, users can specify the per-actor LOD that suits the type of scene they are trying to work with, and the mesh geometries will be automatically replaced with more lightweight versions produced by the modelling department.

We have provided integration into all of our lighting and rendering tools including Clarisse, Katana and Mantra. This means that lighting TDs can make edits to the crowd like changing alternates or making slight alterations to positions, without the need go back to the crowd TDs.

By using this interchange library as the underlying framework for all of our crowd tools, and providing import/export functionality in each, we are able to create, alter and export crowd caches consistently in all of our applications. We have also added the crowd cache as a recognised type to Asset Packages (APKGs), our hierarchical scene description format. This allows shows to block out crowd at the layout stage, so that experienced crowd TDs can work on them later on in the pipeline. In addition to this, the variety of tools allows simpler shots to be allocated to more junior artists, which in many circumstances can get finaled without the need for an experienced crowd TD to interact with them.

3 PIPELINE EXAMPLES

To highlight the flexibility for shows of having these different tools, some use cases are now described:

3.1 Full Houdini Crowd Shows

This is currently the most common setup for crowd shows, used on *Fantastic Beasts and Where to Find Them* and *Assassin's Creed*. Experienced Crowd TDs use Houdini to create crowd caches and add them to an APKG, so that their work can be passed down the pipeline for lighting and rendering in Clarisse. Minimal edits can be done in lighting, but more often, edits would be completed back in Houdini.

3.2 Riot Crowds Instantiated in APKGs

As used on *Star Trek Beyond*, crowd caches were simulated using Riot and instanced in APKG nodes with time offsets applied to avoid twinning. Once again, minimal edits could be applied in Clarisse.

3.3 Clarisse Instancing of Riot Crowds

This method was recently used in the TV series *The Young Pope*, where large crowds populated St. Peter's Square in Vatican city. Single crowd characters were simulated in Riot, cached out, and then instanced in Clarisse using a point scatterer. This solution was possible because the crowds were spatially stationary and the ground flat, making simple individual placement possible.

3.4 Maya and Houdini for Cache Creation

For wide angle shots, blocking crowds are added to APKGs in the layout department using Maya. For more complex shots, these will be replaced by a Houdini-generated crowd cache by experienced crowd TDs. For simple shots however, many will not need any extra work, and when they do, they can be cleaned up by junior or mid-level artists. They would then be rendered in Clarisse. This workflow is being used on shows currently in production.

4 CONCLUSION

By diversifying our toolset, we are able to be much more efficient in the turnaround of crowd shots. This is due to our ability to accommodate different requirements and be less prescriptive in the solutions we provide. By moving to more scriptable DCCs, we also allow for more technical artists to be able to create their own solutions to problems when a new requirement arises. It also allows artists to use the software that they are most comfortable with, and when an artist is comfortable, the work they produce will always be of a better quality.