

# A Modular Crowd (n)Cloth System for *Exodus: Gods and Kings*

Clair Bellens\*  
MPC

Marco D'Ambros†  
MPC

## Abstract

For *Exodus: Gods and Kings*, we were required to find a way to deal with simulating the cloth of crowds with up to 200,000 agents.



Figure 1: ©2014 20th Century Fox. All rights reserved.

We decided to tackle this problem by performing three major changes to our crowd cloth pipeline: using simulated cloth models as a starting point for our simulations, using a modular cloth setup system and switching from our own solver to nCloth.

## 1 Preparing the Models

Previously our cloth models were sculpted by the modeling department. Given the amount of assets to parse and the tight deadlines, this often resulted in uneven topology and start shapes that only approximated a physically based look, which became obvious when not simulated. For *Exodus* we had to parse on average crowds of several ten thousands of agents and processing each agent to get the desired result wasn't an option.

The solution we chose for this problem was to start using *Marvelous Designer*, a 3D clothing design software, which allowed for the draping and simulation of garments. The advantage of this was that the topology of the models stayed even, while giving the feeling of nicely draped cloth. This improvement meant we avoided simulating large parts of the crowds in many shots, as the render skinning of these new models worked well enough.

## 2 Building the Setups

Keeping in mind our experience with previous crowd shows, such as *World War Z*, and the specific requirements for *Exodus*, we came up with the idea of developing a modular cloth system working on a mesh level that would allow us to create a cloth setup library.

\*e-mail:clair-b@moving-picture.com

†e-mail:dambros.marco@gmail.com

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

Copyright is held by the owner/author(s).

SIGGRAPH 2015 Talks, August 09 – 13, 2015, Los Angeles, CA.

ACM 978-1-4503-3636-9/15/08.

<http://dx.doi.org/10.1145/2775280.2792574>

This, opposed to having one big setup containing all the data for each unique variation of an archetype (for example: male Hebrew), allowed us to split the work between different cloth artists and thus reduced the working hours. The system recreates the agent, with all the variations given from the crowd data, building it “on the fly” when simulating on the farm or locally.

Under the hood the biggest change was in the way we stored our setups/data: instead of saving a Maya scene, we exported the data to script files on a per mesh basis. This limited the artist to using only nodes that were supported by the system, but, to compensate for this, the tool also supports the use of custom setup scenes.

In most shows, there are often model updates throughout the production. To make these updates flow through the pipeline more easily, we gave the artist the option to store the data for the constraints, pins and such on a per vertex or on a uv basis. Saving out the data on a uv basis meant that, instead of having to rebuild (parts of) the setup the artist just had to load in the old setup, do a check and fix it if needed. Another benefit was that we could deal with variations in the body (e.g. thin vs. fat) with the same setup giving us a flexible parallel workflow essential to our pipeline.

With that major change done, it was a small step to support LODs in a more efficient way by adding a pre simulation subdivision node that could increase the resolution of meshes for more accurate simulation, or reduce it for background agents. This data was saved out in the caches and at render time that information could be parsed and the appropriate actions performed.

Another layer of complexity was how we should deal with the idea of “partner agents”, in the case of *Exodus* we needed to simulate multiple agents together in order to achieve the required result (horses + chariots + drivers). While building our setups we loaded in the agent with all its partners and gave the artist the opportunity to tag certain meshes of the other agents as collision objects, so that when present in the shot they would be taken into account.

An advantage with how we structured our new crowd setups and switching from our solver to nCloth, is that they are compatible with the hero cloth setups and the other way around. If in some cases we needed to extract agents from a crowd (for example when the client wanted a custom animation for a particular agent) and create a hero version of them. Instead spending time to recreate the setup with a different solver, we could reuse the existing setup and load it in on the hero model.

Instead of incorporating features like ground collisions, wind directions and such in the setups, we kept them separate and introduced them on a shot by shot basis. This meant that we could have custom “additions” to the general cloth setup without needing to alter and rerelease it.

Another change that had a significant impact on our workflow was the creation of custom nodes to allow us to read out data from the crowd animation caches (other than the animation data itself). This was mainly used to get the speed and direction of the agent, and as a result, we could deal more accurately with wind settings, without needing to do custom setups on a shot basis.