

Destroying the Pharaoh’s Army - Large-scale Dynamics in *Exodus*

Kai Wolter*
MPC

Mariano Blanc†
MPC

Francisco Gochez‡
MPC

Ruben D. Hernandez§
MPC

Abstract

For Ridley Scott’s *Exodus: Gods and Kings*, MPC was tasked with the challenge of simulating the Pharaoh’s army of hundreds of chariots, cavalry and soldiers. The Egyptian troops needed to crash into each other, tumble down the slope of a collapsing mountain and be swept away by a gigantic wave. Motion-captured chariots, horses and soldiers were combined with simulations of colliding teams of horses and breaking chariots. These all blended seamlessly into footage of practical effects and stunt performances.

While MPC has a long history of simulating crowd agents with our proprietary crowd software *ALICE* and physics simulation software *PAPI*, the complexity and number of set-ups required us to improve the quality of the simulations and streamline our dynamics pipeline.

1 Crowd Destruction

In one of the key shots, a convoy of chariots passes over a narrow path next to a steep slope. The ledge then collapses under the weight of the troops which tumble downward into the ravine. With a collapsing set the size of two football fields and 1800 agents on top, we decided to simulate the shots in layers. First, our FX department used *Kali*, our in-house FEM based destruction tool to crack and collapse the mountain. A cached tetrahedral representation of the ground was then released to the crowd department to simulate the agents tumbling and colliding with the crumbling ground.

Given the ground’s complex shape and high-velocity motion, using static or roto-animated collision objects as an approximation in the *ALICE* simulation was not an option. Also, updating the collision geometry at every Maya frame was not giving us the required temporal resolution to detect and resolve collisions robustly. Instead, we extended *PAPI* and the underlying physics engine *Bullet* to accurately interpolate not just rigid transforms but also *deforming collision shapes* at every internal sub-step of the solver. This greatly improved the quality of the *PAPI* simulation and also decoupled it from Maya’s time step—improving simulation time significantly.

Based on animated key frames or force thresholds, we needed to be able to dynamically disable constraints so objects, including harnesses, wheels and flag poles, would break off from chariots in a directed manner. To achieve believable motion, accurate physical properties were required for each (now independently) moving object. These included the type of collision shape (convex hull or concave surface), inertia, friction and mass.

*e-mail:kai-wo@moving-picture.com

†e-mail:mariano-b@moving-picture.com

‡e-mail:francisco-g@moving-picture.com

§e-mail:ruben-dh@moving-picture.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/10.1145/2775280.2792594>



Figure 1: *Convoy tumbling down a slope* ©2014 20th Century Fox. All rights reserved.

By *decoupling* the modelling and joint space from the physical center of mass used in the physics world, we were able to tweak the latter at any stage—avoiding expensive iterations on up-stream assets. Detaching arbitrary joint hierarchies of crowd agents to simulate anything from breaking wheels to detaching limbs throughout our pipeline is now a critical feature already in use on other projects.

2 Spatially Consistent Simulations

In crowd shots it is a common workflow to reposition, remove and add new agents based on the director’s or supervisor’s feedback. The goal is to improve specific areas in the shot whilst keeping approved portions of the simulation unaffected. Bullet and other physics libraries sacrifice this consistency in favour of performance. Hence artists are forced to record caches of parts of the simulation and mix them with new simulation passes. Depending on the nature of the iterative changes, this workflow quickly becomes tedious or unusable. We solved this problem by adapting *Bullet* so its broad-phase, near-phase and constraint resolution algorithms would execute in a consistent order *independent of spatial and quantitative changes* in the physics world. This allowed us to re-simulate a shot at any time and ensured that the approved motion of agents was accurately preserved until an affecting change was deliberately applied.

3 Dynamics Pipeline

In order to streamline the creation and validation of chariots, drivers, passengers, and horses hitched side-by-side, we implemented a workflow to let the artists set up, tweak and test each physical entity with *PAPI* natively in Maya. From here, all critical physical attributes were released into our asset management system. In the shot, *ALICE* then automatically recreated the rigid bodies and constraints accordingly to simulate large crowds with the exact same predefined physical properties.

Acknowledgements

To Adam Davis, Marco Carboni, Jo Plaete and Alan Stanzone in the crowd department, for their contribution to this development.