Creating the Flying Armadas in *Guardians of the Galaxy*

Rob Pieké* MPC
Lucy Bailey† MPC
Kai Wolter‡ MPC
Jo Plaete§ MPC

Abstract

For *Guardians of the Galaxy*, we were required to create an epic space battle spanning hundreds of shots, with tens of thousands of ships dogfighting.

Figure 1: *Entering the battle - ©2014 Marvel. All rights reserved.*

As characters unto themselves, significant effort was spent in designing unique flight characteristics for each type of spaceship. This required improvements to MPC's technology to allow for quick iterations and compelling results for the bulk of the ships via our crowd tools, while streamlining the process for fine-tuning the motion of any ships which didn't match the director's vision.

The vast increase of geometric complexity for these "crowd agents", compared to MPC's previous shows, forced us to rethink our rendering pipeline, developing new tools and workflows.

1 Animation and Crowd Simulation

While MPC has a long history of simulating battles between armies, this project provided new challenges requiring us to extend and fundamentally change the use of ALICE, our internal crowd system. Unlike human-based armies which generally remain upright, spacecraft can twist and turn to obtain arbitrary directions. Impressed by the capabilities of Autodesk Softimage's ICE framework, we bridged it with ALICE, allowing such motions to be quickly orchestrated and transparently blended with existing ALICE operator graphs, requiring almost no further changes to our pipeline.

In stark contrast to previous shows where we used motion clips to drive positional data, we took the opposite approach with this project, building a custom flying framework using ICE as a base. Starting with a generic boid simulation, similar to that described by [Reynolds 1987], we layered in more exotic behaviors to emulate specific flight characteristics. Two examples of this were dogfighting behaviors that had ships trying to get into favorable shooting positions, and flight path generators which would take an overall direction and create a series of organic paths for ships to follow. These behaviors allowed us to generate thousands of frames of animation without much need for artist intervention.

Working with the constraints of our flight models, our simulation framework ensured that we got proper banking and trajectory smoothing, effectively for free. Lastly, as the behaviors were controllable by the TDs in the crowd department, we were able to efficiently react to art-direction and feedback from the director.

In cases where shots were extended, or pre-roll was required for other departments, it was important that we were able to generate new simulation caches without a significant amount of artist time. We improved our crowd-motion extrapolation tools to better leverage existing data about spacecraft's velocities, accelerations, etc, and generate new caches with compelling motion automatically.

Continuing work from other recent shows, the bridge between animation and crowds was strengthened, allowing both departments to contribute to the motion of the ships based on the fidelity required. To ensure that work was done in context, we improved our preview capabilities to allow each department to see their work seamlessly integrated with that of the other.

2 Lighting and Rendering

To combat geometric complexity, we streamlined the use of multi-resolution assets. Choosing a level-of-detail for models was previously a decision made early in the pipeline by the layout team, but on this show we introduced mechanisms for the lighting department to override this on-the-fly, without the need for new caches from the animation department. This was accomplished with a combination of up-front geometry and skeleton caching, as well as animation rigs which could seamlessly bind to different ship models.

Similarly, we introduced level-of-detail for *animation*. While our caches always contain rich data about the nuanced motion of the ships and their components, we provided the lighting department with controls to decide how much of the data was visually-relevant in the context of each shot. While foreground ships might visually benefit from subtle bending and flexing, background ships could be processed drastically faster by only using data for the rigid rotation of the wings and weapons. This also opened up opportunities for render-time instancing, further reducing our memory requirements.

While most changes were improvements to our existing crowd rendering pipeline, described by [Haddon and Griffiths 2006], we also benefited from the adoption of Katana and its internal workflows.

References

HADDON, J., AND GRIFFITHS, D. 2006. A system for crowd rendering. In *ACM SIGGRAPH 2006 Sketches*, ACM, New York, NY, USA, SIGGRAPH '06.

REYNOLDS, C. W. 1987. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '87, 25–34.

*e-mail:rob-p@moving-picture.com
†e-mail:lucy-b@moving-picture.com
‡e-mail:kai-wo@moving-picture.com
§email:jo-p@moving-picture.com

SIGGRAPH 2014, August 10 – 14, 2014, Vancouver, British Columbia, Canada.

ACM 978-1-4503-2960-6/14/08