Evolving Complexity Management on "The *LEGO* Batman Movie"

Jens Jebens
Animal Logic
Jens.Jebens@al.com.au

Damien Gray
Animal Logic
Damien.Gray@al.com.au

Simon Bull
Animal Logic
Simon.Bull@al.com.au

Aidan Sarsfield
Animal Logic
aidan.sarsfield@al.com.au

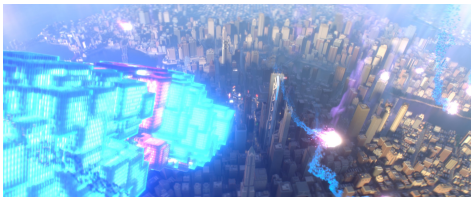

Figure 1: Complexity of Gotham City, Detailed textures for Bane, and Truck ©Warner Bros Inc., The LEGO Corporation. All rights reserved.

ABSTRACT

The demand for asset complexity has increased by several orders of magnitude since *The LEGO Movie*. This has resulted in the need for the team at Animal Logic to further develop their proprietary render and shading pipeline, while significantly optimising nearly all aspects of asset creation. Animal Logic's already extensive library of *LEGO* bricks was expanded considerably, and centralised for use across multiple shows and multiple locations. Continued development of asset creation tools, and significant increases in pipeline automation ensured increased review cycles, greater consistency and minimal duplication of effort.

CCS CONCEPTS

•Computing methodologies →Animation;

KEYWORDS

LEGO, complexity, city, sets, rendering

ACM Reference format:
Jens Jebens, Damien Gray, Simon Bull, and Aidan Sarsfield. 2017. Evolving Complexity Management on "The *LEGO* Batman Movie". In *Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3084363.3085059

SIGGRAPH '17 Talks, Los Angeles, CA, USA
 978-1-4503-5008-2/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3084363.3085059

1 MODELLING: FROM *LEGO* BLOCKS TO CITY BLOCKS

To meet the increased requirements for new bricks, Animal Logic's proprietary renderer *Glimpse* was more deeply integrated into the brick modelling toolset. As an interactive renderer within Maya, it vastly improved the speed, accuracy, and efficiency of the brick modelling process,

With significantly more bricks in play, the challenge was to achieve the production designer's brief of making Gotham City as much a character of the movie as Batman or Joker. The entire environment pipeline was re-oriented towards achieving massive scalability, whilst simultaneously improving the interface to be more familiar and user-friendly to the artists.

Animal Logic's proprietary environment tool *Scenery* was developed to use Maya as an interface for assembly and set dressing, while custom scene description files were exported and used to feed downstream DCCs. Further, using the interactive *Glimpse* viewport allowed artists to view rendered environments within Maya as they were being created, greatly tightening the feedback loop.

For a massive Gotham City, the team established a modular city grid structure that served as a framework for street and block placement. A procedural packing algorithm was developed to arrange buildings within each city block parametrically, tying lot size to building height based on real-world metrics. The end result captured the *texture* of a real city while ensuring that every city block was unique.

2 MAXIMISING REALISM WITH *ASH* AND *GLIMPSE*

Continuing on from *The LEGO Movie*, the team further extended a huge library of brick textures. These were then randomised and/or handpicked (per asset) to create a manifest for each asset's unique surfacing.

For all aspects of *LEGO* surfacing, from bricks to entire environments, *Glimpse's* fast and physically plausible shading made look development a fast and simple procedure. The efficiency of *Glimpse* and *Ash* contributed to a massive increase in the number of assets handled by the department. *Glimpse* also ensured consistent results from single asset to massive scenes (i.e. no scalability limits like shadow maps, shading rates etc.).

To further enhance the artists' user experience, and to simplify the creation of surfacing materials, the team built a proprietary shading graph system, *Ash* and editor UI. *Ash* is a first class component within *Glimpse* and provides not only interactive shader graph editing functionalities, but also access to *Glimpses'* shader assignment system and our material library.

Grunge and oxidisation effects were important visual elements on most assets. The team baked occlusion-like modulation textures into ptex files, but quickly hit the ceiling of ptex's limited multi-threading and mip mapping capabilities. As a result, per-face texturing capabilities were added to *Glimpse's* extremely efficient texturing format which converted ptex textures on the fly, allowing for even the biggest environments to be rendered with detailed grunge and dirt.

Another massive realism boost originated from the lights inside buildings and seemingly randomly closed blinds. We built a directable procedural placement tool to position simple room geometries behind building windows. The tool chose from a series of pre-defined sizes and tagged the geometry with appropriate user data. A specific incandescent library material read the data from the scene and shaded the rooms accordingly. *Glimpse's* shading parameter override system allowed downstream departments to hierarchaly change the shader to adjust hue and intensity on parts of the city, thus addressing art direction requests on the spot.

3 THE FUTURE OF COMPLEXITY MANAGEMENT

To further expand complexity management capabilities, the team will be exploring a shift towards *Pixar's USD* while experimenting with *USD* and *Glimpse* interoperability.