

Managing thousands of assets for the *Prince of Persia* city of Alamut

Gregory Meeres-Young*
MPC

Hannes Ricklefs†
MPC

Robert Tovell‡
MPC

Abstract

The medieval Persian city of Alamut was constructed with over one hundred thousand assets. From pot to palace, it provided the stunning playground in which an adventurous prince struggled to protect the sands of time.

This colossal environment was subject to intense scrutiny as dizzying up-close action seamlessly transitioned to sweeping panoramic cityscape views.

Short schedules and the necessity to accommodate dynamic changes at any point in the production, precluded the use of the existing environments pipeline and forced a re-think. In its place, a pipeline enabling highly parallel workflow across multiple departments was created with an artist-friendly toolset.

This talk discusses the methodology and challenges involved in creating effective pipelines for the construction of large assets such as the greatening city of Alamut.



Figure 1: the city of Alamut. ©2010 Walt Disney Pictures.

1 The path to parallel success

The notion of multi-discipline parallel development was first effectively realised at MPC during the production of *Prince Caspian: The Chronicles of Narnia* [Butler et al. 2008]. This predominantly character-driven pipeline armed us with some robust production techniques and proprietary software. Packaging, Streams and Approvals were some of the key pipeline components refined to tackle the very different problem of one huge environment asset.

MPC's asset management system, *Packaging*, enables the definition of relationships between different assets whilst simultaneously keeping a record of any changes to these during the production time line. The environment package definitions used on previous films could only describe flat groupings of model instances. This proved insufficient as it could not intuitively represent the hierarchical nature of a city. *ModelHierarchyPkgs* were created to represent a hierarchy of multiple instances of model packages. This new package definition enabled props to be grouped with buildings, buildings to be grouped into Streets, and streets to be grouped into districts.

Distinct departments such as modelling/texturing, layout, and lighting/look-dev worked on different package types that existed

in independent streams. Isolating these department-specific assets proved an essential component of the parallel workflow paradigm. It provided a safe sandbox and diminished the risk of propagating broken assets between departments.

Syncing of streams propagates the work done between the isolated departments. Package differences are compared and resolved where appropriate. Not all changes get synced, an approval system acts as a gate keeper between department streams giving artists control over which assets they want to be made available to other departments.

2 Artists tools for asset creation

Effective and transparent artist creation/modification of the city was provided by a set of packaging tools integrated into Maya. *PackageEditor* and *TownPlanner* were developed specifically for the rapid layout, creation, extraction and exploration of *ModelHierarchyPkgs*. The layout department used *TownPlanner* to block out large sections of the city procedurally, sourcing model packages created by the modelling/texturing department. Hand layout refinements were performed with *PackageEditor*. Both tools generated *ModelHierarchyPkgs* that were picked up by the lighting and look-dev departments to be lit and shaded. An OpenGL view of packages representing large sections of the city was exposed in Maya's view port through a single node. This provided a number of advantages, such as light scene files, non destructive updating of scene assets and intuitive city interaction. As changes were frequent and time-pressured, lighters used *PackageEditor* to extract sections of the city letting them perform layout changes catering to shot-specific conditions such as modifying buildings, level of detail and positions.

A migration from implementing pipeline tools in *MEL* and *Lua* to *Python* proved far more versatile. Python operated nicely as a scripting glue between Maya and MPC's C++ 3D description library *Muggins*, and the use of *PyQt* for user interface design unified tool aesthetics for operation both in and outside of Maya.

3 Summary

The pipeline and tools developed for *Prince of Persia: The Sands of Time* empowered layout artists to become architects of great cityscapes whilst modellers created and refined the intricate buildings that lined the streets. Meanwhile, lighters were able to update their scenes of entire districts with the latest changes whilst still maintaining the control to manipulate the smallest pot that adorned the heaving market stalls.

References

BUTLER, G., LANGLANDS, A., AND RICKLEFS, H. 2008. A pipeline for 800+ shots. In *SIGGRAPH '08: ACM SIGGRAPH 2008 talks*, ACM, New York, NY, USA, 1–1.

*e-mail: greg-my@moving-picture.com

†e-mail: hannes-r@moving-picture.com

‡e-mail: robert-t@moving-picture.com