

Environment Workflow of Final Fantasy XV

Hiromitsu Sasaki*
Business Division 2, Square Enix Co., Ltd.
Junichi Miyajima‡
Business Division 2, Square Enix Co., Ltd.

Norihito Ueno†
Business Division 2, Square Enix Co., Ltd.
Eitaro Iwabuchi§
Advanced Technology Division, Square Enix Co., Ltd.



Figure 1: A World of Final Fantasy XV (open world / town / dungeon).

Abstract

In this session, we will be covering how the immense and unique world of Final Fantasy XV was created. We will be introducing the technologies and the development environment that we worked with engineers to create, and which allows users to experience the world imagined by the artists as if it were real. We will talk about a number of the supporting tools that make large-scale development like this possible, and introduce our workflow using those tools. Final Fantasy XV is set in an immense open world with day and night cycles and weather effects that change in real time. The seamlessly connected regions have gigantic cities that reflect their individual cultures, and the caves that stretch out endlessly underground conceal terrors that adventurers would never face on the surface.

Keywords: realtime rendering, open world, asset management

Concepts: ●General and reference → ; *Surveys and overviews*;
●Computing methodologies → Computer graphics; Graphics systems and interfaces; ●Applied computing → Computer games;

1 Data

This is an introduction of the data structure. The terrain is created based on a heightmap. Towns, dungeons and various other assets are created with polygon meshes. For roads, assets are placed according to curve data, and trees use data that has been optimized for mass placement. In order to construct a high-quality open world that is based on physics-based rendering, the assets were created

with LOD in mind, and optimized shadow generation and convex collision data were used to find a balance between performance and quality.



Figure 2: wire frame and final rendering image

2 Game engine & tools

Here we will introduce the Final Fantasy XV game engine, development environment and toolset.

2.1 Level Editor

The Level Editor is the core game development tool that allows us to manage and edit all of the information about the game. It has a variety of innovations that make it possible for hundreds of team members to simultaneously perform level design, asset creation, and lighting. As environment artists, we also use the Level Editor for everything from scene construction work to making adjustments to the quality of the final look.

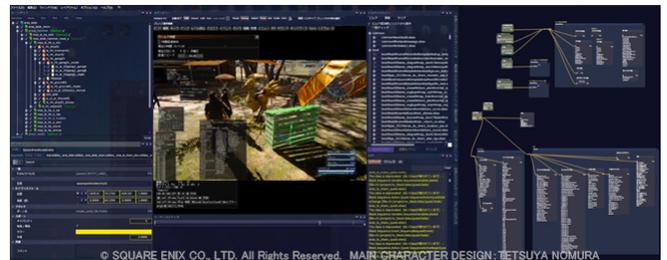


Figure 3: level editor

*e-mail:hiromits@square-enix.com

†e-mail:ueno@square-enix.com

‡e-mail:miyajima@square-enix.com

§e-mail:iwabeita@square-enix.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 profit or 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(s). © 2016 Copyright held by the owner/author(s).

SIGGRAPH '16, July 24-28, 2016, Anaheim, CA,

ISBN: 978-1-4503-4282-7/16/07

DOI: <http://dx.doi.org/10.1145/2897839.2927444>

2.2 Asset browser

The larger the world you are creating, the more important the issue of managing this immense amount of data becomes. Managing the huge number of assets we were creating required a proprietary asset browser. The asset browser has a proprietary filtering feature and proprietary menus for each file type, and we embedded metadata into the files that would allow us to check the data inside. This made it possible for us to quickly and flexibly access our assets, improving the efficiency of our work.

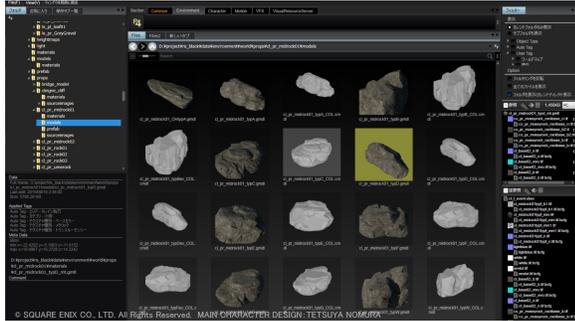


Figure 4: asset browser

2.3 Final Fantasy XV data pipeline with Maya

While we would normally use the Level Editor to construct scenes, it was also designed to make it possible to use Maya to perform layout work. When performing layout work of complex scenes requiring a high degree of accuracy, many artists wanted to use Maya, with which they are very familiar, and then use the Level Editor for the final quality check. To make this possible, we needed a mechanism that would enable us to edit in both directions between the game engine and Maya. To solve this problem, we used an xml-based data format so that either side could make layout or parameter changes.

2.4 Viewport 2.0 customization

We used Maya for the creation of assets such as characters and environments. To enable us to check things like lighting and post-processing using the same environment as the runtime, we customized Maya Viewport 2.0. This allowed the artists to check the final quality while creating assets without the need for conversion, and also made it possible to perform quality control for the large-scale outsourcing of asset creation.

2.5 Collision workflow

For collisions, we created the models in Maya and then exported them with IDs, which are identifiers used in the game. We used a data flow as shown below that would function on the actual console.

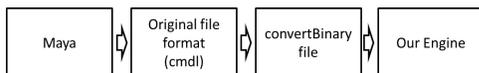


Figure 5: collision workflow

There are three types of IDs: Layer, Attribute, and Combination. The applications of each type are shown below, and each type was used according to the situation.

Collision ID Type	purpose	number
Layer	High speed, but a little	32
Attribute	Low speed, but many	many
Combination	Combination with layer	32

Figure 6: ID types

3 World Tools and workflow

Here we will introduce a representative workflow, along with World Tools.

3.1 Continent design: An artist's clay model

The world of Final Fantasy XV was born from the imaginations of artists, and the continent on which it takes place was no exception. It was created via a clay model made by the art director



Figure 7: clay model (sample)

3.2 Heightmap creation: Capture and optimization

We used PhotoScan to digitize that model, then brought it into Maya and optimized it. After that, we used World Machine to export it as a heightmap.

3.3 Continent data creation: World Tools

We imported the heightmap into Level Editor and used the cell manager to perform area settings. After that, the sculpt tool was used to adjust the terrain, the curve tool to create roads, and the foliage tool to place trees. The collisions and attributes necessary for gameplay were also set using Level Editor.



Figure 8: world tools

3.4 Environmental settings, lighting and post-processing

Level Editor was used to perform settings for wind, weather, environmental noises, VFX, and so on. Lighting was based on sun and sky settings, as well as the placement of local lights. Post-processing for each situation was then performed to bring it up to the final level of quality.