Final Fantasy XV: Pulse and Traction of Characters

Noriyuki Imamura* Youji Shirakami Kousuke Namiki
Prasert Prasertvithyakarn Takanori Yokoyama Youichiro Miyake
Square Enix Co., Ltd.

Figure 1: *A battle scene from Final Fantasy XV: the player character (left) and AI-driven characters (the rest).*

Keywords: game production, AI, animation, workflow, state machine, behavior tree, motion blending

Concepts: •**Computing methodologies** → *Artificial intelligence; Procedural animation;*

1 Introduction

The mixture of reality and fantasy is the main expression of *Final Fantasy XV*, which is a title from the classical long-lasting role-playing game series. We embrace the words, *pulse and traction*, as one of the key concepts of the development. How we control motions of characters to make them plausible and fascinating is a challenging work in the area of AI and animation to achieve the goal.

We present our approach and solutions for controlling a player character and AI-driven characters including companions of the protagonist, pedestrians in towns, and various unique monsters. Each team makes those characters work to achieve its individual mission, therefore the requirements of the toolchain and necessary features differ from one another. For example, the companion characters are the closest characters to the player, so they are designed to take as many variations of behavior as possible and their quality should be high in all forms. On the other hand, because the team in charge of monsters focuses on the abundance of species, we need to pay attention to reusability of the assets and effectiveness of the workflow for mass production of monsters. We developed these modules of the game engine of Final Fantasy XV to be flexible enough to fit those teams' demands.

2 Overview

From a workflow perspective, the skillsets involved in making character behaviors are: animation, game design, and programming. Animators should spend time and effort on the visual quality. Game designers have their strong intention of the gameplay directly leading into the user experience. Programmers are focused on the implementation in way that also allows for scalability and extensibility. From the beginning, we planned to build data-driven systems for the animators and the designers: AnimGraph and AI Graph (Figure 2).

Node-based graph systems in the animation area are now widely used in many game engines [Epic Games, Inc. ; NaturalMotion]. However, these graphs grow larger and larger as development progresses and gameplay-specific features stream into the graph. To avoid this common situation, we add another data-driven graph system in between AI and animation encapsulating the complexity of AnimGraphs. The abstract states of actions are managed in this level, which we call *Body Graph*. We designed it as a beneficial system for programmers to organize characters' behaviors and movements.

The beauty of this three layer architecture is that we can design tools and specifications on a user-centric basis because it is obvious to define the expected skills and knowledge of the users and tastes of usability for each system.

3 Graph Systems

We detail the system design of the three layers (Figure 3) in the talk. The topics are briefly described here.

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

SIGGRAPH '16, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4282-7/16/07
DOI: http://dx.doi.org/10.1145/2897839.2927449

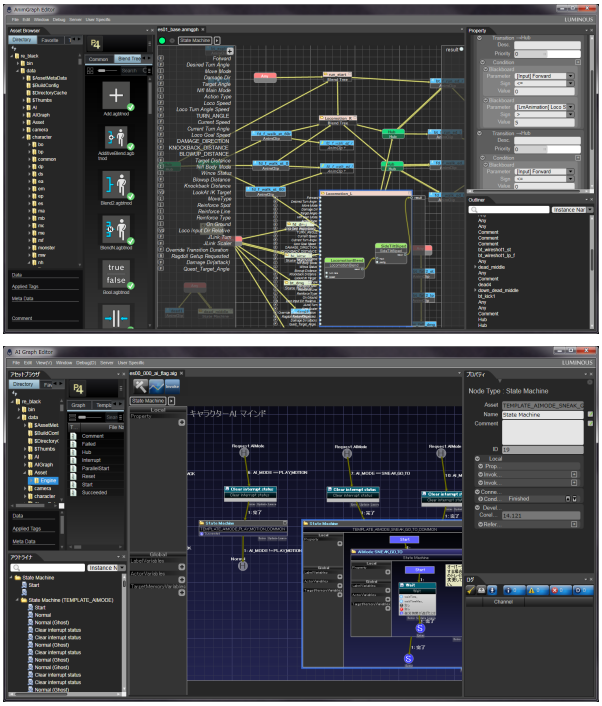

Figure 2: *AnimGraph Editor (top) and AI Graph Editor (bottom)*

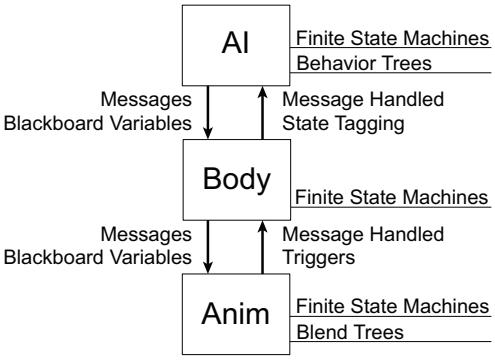

Figure 3: *Interaction among three layers*

AI Graph The users of AI Graph Editor can make graphs composed with hierarchical finite state machines (HFSMs) [Millington and Funge 2009] and behavior trees [Isla 2005]. You can seamlessly assemble those two systems with this novel hybrid architecture to exploit the advantages of each system according to the situations.

AI Graph supports the unique concepts of *parallel node* and *interrupt node*. Sometimes, natural behavior and thought can be observed as doing multiple things at once: for example, waving a hand during walking, and thinking about the next target while attacking another enemy. We can intuitively implement these behaviors with the nodes.

Body Graph Body Graphs are usually created with HFSMs. Some of the states in the graphs are driven with procedural systems which require composite character controls such as swimming on the ocean, sliding on a steep slope, mounting onto an animal for transportation, etc. Programmers can implement these in a modular and reusable manner coordinated with programs.

AnimGraph The animation system provides a lot of advanced features which contribute to motions adaptive to the environment and help avoid the users feeling a sense of Deja Vu. Various inverse kinematics techniques, physics based animation, secondary bone systems, and parts blending are implemented in the game engine.

Generally speaking, it is strenuous for animators to understand the full functionality of those systems. Moreover, there is the risk of introducing complexity into AnimGraphs due to the complexity of controlling the status and the parameters of the multiple systems concurrently. We added a feature to deal with these issues.

4 Use Case

In this section, we present a use case of a monster in a battle.

First, to increase the reality of monsters, the visual ability of AI is simulated. The target of the enemy is selected from a weighted list of targets based on the visual information. How we select the target is parameterized according to each monster type. This target selection system is registered by placing a node onto the AI Graph of the monster. This target is generally used to decide next attack target.

After the decision is done, the AI finds a way to the target. Using a point query system [Jack 2013] and a navigation system [Buckland 2004], the AI determines the optimal path leading to the situation. Calculated speed and turning ratio are written into the blackboard variables as desired speed and rotation. The AnimGraph in turn uses those values as inputs for motion blending.

The three graph systems have a templating system, so the same graph can be used as a module of the character. Specific logic of the template can be overridden with a similar system of inheritance as often seen in programming languages.

5 Conclusion

The characteristics of the various characters that appear in Final Fantasy XV are designed with these AI and animation systems. The effective toolchain and concept of three data-driven systems; AI Graph, Body Graph, and AnimGraph help to realize the fascinating user experience.

References

BUCKLAND, M. 2004. Practical path planning. In *Programming Game AI by Example*. Jones & Bartlett Learning, ch. 8, 333–377.

ISLA, D., 2005. Managing complexity in the halo2 ai. Game Developr's Conference 2005.

JACK, M. 2013. Tactical position selection: An architecture and query language. In *Game AI Pro: Collected Wisdom of Game AI Professionals*. CRC Press, ch. 26, 337–359.

MILLINGTON, I., AND FUNGE, J. 2009. State machines. In *Artificial intelligence for games*. CRC Press, ch. 5.3, 309–333.

NATURALMOTION. Morpheme. http://www.naturalmotion.com/middleware/morpheme/.

EPIC GAMES, INC. Animation blueprints, unreal engine 4. https://docs.unrealengine.com/latest/INT/Engine/Animation/AnimBlueprints/.