

Intelligent Interactive Narrative Simulator

FRAMEWORK FOR MASSIVELY MULTIPLAYER ONLINE ROLE-PLAYING GAMES BASED ON INTERACTIVE DIGITAL STORYTELLING AND COMPLEX SYSTEMS

Eduardo Ridao Pérez

Dragon Corp. Games S.L.

SUMMARY

The video game industry has developed around recognized computer technologies, such as digital imaging, sound and simulation. However, the application of certain gaming concepts to the digital platform has resulted in a vague approximation. This is the case of role-playing video games, and more specifically MMORPGs, which result in a product that is very far from the original board game format. This differentiation has been justified in many ways by the authors of the different titles. In this paper we demonstrate that the difference between tabletop role-playing games and role-playing video games is due to defects in the analysis of game systems, and to a random use of the available technologies, analyzing the role-playing games themselves from a computer application point of view, and proposing an integral framework for the simulation of the dynamics of a role-playing game.

SNI2, *Intelligent Interactive Narrative Simulator*, is a system for the development of Massive Multiplayer Online *True* Role Play Games (MMOTRPG). Integrating interactive digital storytelling and state-of-the-art game engines, through a unique system, for an experience much closer to traditional role-playing games.

FEATURES

An MMOTRPG is a system that simulates the interaction between a role-playing player and the scene in which their character finds themselves, behaving like the game master in all aspects.

Any MMOTRPG developed with the SNI2 engine presents solutions to MMORPG problems, as well as offering a unique role-playing experience.

SNI2 has a number of implicit technical features and requirements, described below.

CLOUD ARCHITECTURE

SNI2 is a distributed system, so it can be run on server farms, or on cloud services.

The individual scenes are processed independently, distributed by the system. Each compute node supports a varying load, depending on its capabilities, and the system is kept consistent by coordinating the different nodes.

The data that describes the game universe is contained in the model by distributed services to facilitate access and persistence of the data.

The SNI2 server is a set of services that, in a distributed way, carry out the processes that allow the development of the role-playing game, manage the state and persistence of the world, and perform the narrative processes that define the interactions of the players.

PERSISTENCE AND CONCURRENCE

Using SNI2, the actions of a single player directly affect the overall state of the virtual world where the game takes place.

The storylines are built from the interactions of the player characters with the narrative contexts. The scale of each context, and the involvement of the characters, will be foreseen by the authors, being able to generate global contexts, divided into smaller contexts, with the same background.

The narrative contexts influence the scenes of each player character through very different areas.

Each player's actions in a scene affect the narrative contexts that scene is related to. Indirectly, global contexts are affected by the actions of hundreds of thousands of players.

SNI2 thus solves the problem of persistence and concurrency of MMORPGs, and opens up a new dimension in the history of virtual worlds.

VIRTUAL WORLD

Unlike in any MMO game, there is no linear management of the game world: the server maintains the virtual world as a structure of planes at different scales, while the client system instantiates a custom scene for each player, and includes all the necessary elements to achieve a coherent representation with the virtual world.

User interactions always occur at the scene level, and global processes are simulated in narrative nodes.

In this way, virtual worlds of unprecedented scope can be simulated, because with a limited number of scenes to the number of connected players, any place in the game world can be represented without overloading the servers with empty areas.

NARRATIVE INTELLIGENCE

The challenges presented by the MMOTRPG include all issues related to computational narrative, narrative theory, story generation systems, computer systems that exploit human narrative intelligence, interactive digital storytelling systems, and the agents that use or are used in narrative.

Narrative theory is a field as close to literature and philosophy as it is to computer systems engineering, since it is in the context of narration that the system communicates with the user.

Systems have been developed around the concept of a dramatic director with scene-level controller capabilities to leave characters autonomy in their low-level behavior.

DECOUPLED SIMULATION

In an MMOTRPG, the client, the server (and the servers with each other) do not share hardware, nor do they perform the same functions.

The client's behavior is to represent, as faithfully as possible, the scene of the character that the player controls. To do this, it runs independent simulation systems, such as a physics engine, inverse kinematics, audio engine, and 3D graphics engine. Moderated by the set of supervisors who act as an interactive narrator for the user.

The server maintains consistency at the scene level, for each player, and relates each scene to the global contexts, resorting to content generation dynamically, modifying each of the contexts according to the actions of each character in each scene. Scenes that share scenarios and characters are handled allowing the concurrence of contexts between several players.

The server no longer handles play areas, but is instead a "scene server", creating each scene tailored to the story of each player's character, largely controlling the interaction of said story with the plot contexts, which can be shared by other players' stories.

The client, on the other hand, supports the graphic and physical simulation load, executing a simulation that results in an animated visual representation, consistent with the narrative context of the scene that runs on the server. The scenes are considered as shared instances, taking into account the positions that the characters occupy in the different places, solely because of their importance in the narrative processes.

The actions of the characters are not synchronized at a simulation level between server and client, as they execute different processes to synchronize at the narration level: for each action of a character in a scene, the server draws a timeline that begins at the current moment and ends at the end of the proposed action.

The client, in turn, translates these timelines and the results of the actions into a graphical representation with the relevant animations and visual effects.

UNREAL ENGINE

Unreal Engine is a game engine for PC and consoles created by the company Epic Games. Initially implemented as FPS, it has also been used in other genres such as RPGs and RTS. It is compatible with common video game platforms, PC, Mac, and video game consoles.

Unreal Engine 5 offers additional tools that are of great help to designers and has an extensible architecture that allows you to extend its behavior and functions through modules that can be packaged as installable plugins. A plugin, specially designed to integrate the overall system architecture, with Unreal Engine 5's proprietary architecture, enables Emergent Narrative.

The communication between the graphical client interface and the system is defined by links that the authors establish between Unreal blueprints and narrative contexts that are handled by the server.

An additional module, only used in the developer environment, allows the use of Unreal Engine's visual editing tools for the development of server logic.

Unreal Engine offers a professional-quality finish at no cost, with the proviso that, if this project is officially marketed, Epic Games would get 5% of the product's profits, quarterly, when revenue exceeds the first \$3000.