Demo: Combat State-Aware Interest Management for Online Games

Jing Yi Wang[1], Kaiwen Zhang[1,2,3], Hans-Arno Jacobsen[2]
[1]Technical University of Munich
[2]Middleware Systems Research Group
[3]University of Toronto

Abstract

Massively multiplayer online role-playing games (MMORPGs) allow thousands of players to interact with each other in a large-scale virtual environment. Interest management is an important technique used to raise the scalability of a game by limiting the amount of information transmitted to the players according to their relevance. In this paper, we focus on the problem of performing interest management during combats, which are highly interactive and fast-paced events. We have developed a combat state-aware interest management (CSAIM) system which can dynamically adjust update rates based on the current game context, thereby maximizing the utility in the trade-off between consistency and performance. We have implemented CSAIM in MMOnkey, our MMORPG research framework. Our interactive demo client visualizes the interest area adjusted based on players' actions.

CCS Concepts • **Computer systems organization → Client-server architectures**;

Keywords online games, interest management

ACM Reference Format:
Jing Yi Wang[1], Kaiwen Zhang[1,2,3], Hans-Arno Jacobsen[2]. 2017. Demo: Combat State-Aware Interest Management for Online Games. In *Proceedings of Middleware Posters and Demos '17: Proceedings of the Posters and Demos Session of the 18th International Middleware Conference, Las Vegas, NV, USA, December 11–15, 2017 (Middleware Posters and Demos '17),* 2 pages.
https://doi.org/10.1145/3155016.3155019

1 Introduction

Massively multiplayer online role-playing games (MMORPGs) inhabit thousands of players inside a persistent virtual world. The game state, including characters and other entities, is persisted throughout the game's lifetime, which lasts for multiple years. The authoritative state of the game world is stored on one or more servers. Players interact with the servers through actions and receive a replica state of the game world. Interest management is employed to determine which portion of the overall game state is replicated at each player client [2]. For instance, a player should only be informed about changes which are visible to the player.

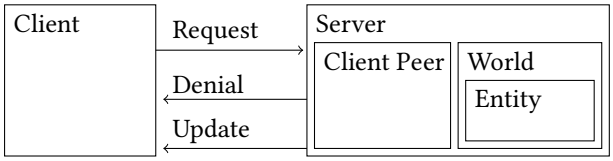

Figure 1. Simplified MMOnkey Architecture

The two primary objectives of the underlying MMORPG system are scalability and interactivity. Scalability refers to the ability of the game to support thousands of players concurrently with acceptable latency (e.g. > 100ms). Interactivity refers to the degree to which players can interact with the game world. Games that provide players with the ability to perform frequent and complex actions are usually more enjoyable. Thus, consistency becomes an issue, since the outcome of actions executed server-side must correspond to the expected state observed locally at the client.

In this demo, we focus on the problem of performing interest management in combat. A battle is a highly interactive scenario which occurs when players belonging to different teams repeatedly interact with one another (e.g. attacking with a weapon or a spell) in order to perform an objective (e.g., defeat the opposing force). Combat is the most interactive aspect of MMORPGs, since players are engaged in complex actions, which involve multiple players, in rapid succession. Latency is critical since players must be able to quickly react to the actions of others. In addition, consistency must be maintained to ensure players are correctly informed about the latest state of the battle.

Therefore, we argue that traditional interest management techniques must be extended to explicitly consider combat. Our proposed system, called Combat State-Aware Interest Management (CSAIM), dynamically adjusts the update rate for each player based on the context of the combat. This includes the profile of each player involved in a specific battle: the types of actions they have equipped. By leveraging battle-specific semantics, CSAIM provides the right balance

Middleware Posters and Demos '17, December 11–15, 2017, Las Vegas, NV, USA

ACM ISBN 978-1-4503-5201-7/17/12.
https://doi.org/10.1145/3155016.3155019

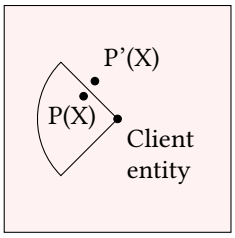

Figure 2. AOE: Inconsistency

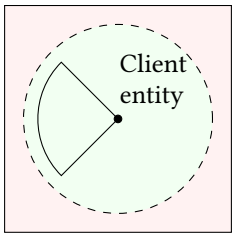

Figure 3. AOE Consistency Req.

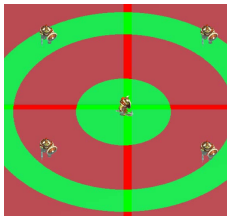

Figure 4. Target: Consistency Req.

between consistency and performance. We demonstrate our implementation using our MMORPG framework, which visualizes the impact of our solution in an interactive demo.

2 Background

Figure 1 depicts a reference architecture for MMORPGs. Clients send action requests to the server to modify entities in the game world. The server can deny or accept the request. On acceptance, the change is replicated to a subset of clients, as determined by their interest, which is reflected in the client peer stored at the server.

Interest management (IM) [1] is an important technique to reduce network load. IM filters messages and sends only relevant data to each player client. Every client peer (a server-side representation of the client) has an interest area of the game world [3]. The peer subscribes to the update channel of every entity intersecting the interest area. Upon receipt of a subscription, the entity sends a snapshot of its current state to the new subscriber. An entity whose state changes publishes a message to its update channel.

Entity position updates account for the majority of the load. They have to be sent at a high enough frequency to maintain accuracy of the client replica. Figure 2 shows a counter-example when position updates are infrequent. The player performs an Area of Effect (AOE) action on the marked cone area. The player sees X at position P(X), the last known position, and thus expects to hit X with the AOE action. However, X is actually at P'(X), outside of the AOE area. This position has not yet been replicated to the client and causes the player's attack to miss, which is not consistent with the player's view of the battle.

3 State-aware interest management

CSAIM dynamically adjusts the update rate depending on the battle state. An interest area in CSAIM has high frequency and low frequency areas, as depicted in Figure 3 and Figure 4. High frequency areas are painted in green. In order to smoothen the movement of low frequency, we employ dead reckoning methods to interpolate the positions [4]. Dead reckoning lacks accuracy compared to high frequency updates and should only be used for data of lesser importance.

CSAIM builds a table using the profile of involved players to determine consistency requirements. A table entry consists of: the minimum required update frequency, a distance interval, and the affected entity group (i.e., enemies or allies).

Players equip them self with different actions prior to the battle instance, thus allowing CSAIM to optimize itself for that situation. Figure 3 depicts the adjusted interest area, when a player is equipped with an AOE action. The AOE action forms a cone shape originating from the player with a certain radius. Therefore, the player requires high frequency updates in a circle of the same radius as the action in order to accurately determine which players will be affected.

Target actions only require the entity target to be in range, but not its exact position. This creates a ring-like high frequency area (see the outer ring in Figure 4). Frequency tables are adjusted during game play, which means that if an action is on cool down then the corresponding table entry is ignored until the action is recharged.

4 Demo Implementation

CSAIM is implemented on our MMORPG research framework MMOnkey. In our interactive demo client, the player can equip itself with different combat actions with different consistency requirements. Frequency tables used by the server can be visualized on the client. Figure 4 is a screenshot of the client equipped with a low range AOE action and a high range target action.

5 Acknowledgements

Supported by Alexander von Humboldt Foundation.

References

[1] Jean-Sébastien Boulanger, Jörg Kienzle, and Clark Verbrugge. 2006. Comparing Interest Management Algorithms for Massively Multiplayer Games. In *NetGames '06*.

[2] Graham Morgan, Fengyun Lu, and Kier Storey. 2005. Interest Management Middleware for Networked Games. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games (I3D '05)*. ACM, New York, NY, USA, 57–64. https://doi.org/10.1145/1053427.1053436

[3] Katherine L. Morse, Lubomir Bic, and Michael B. Dillencourt. 2000. Interest Management in Large-Scale Virtual Environments. *Presence* 9, 1 (2000), 52–68. https://doi.org/10.1162/105474600566619

[4] Curtiss Murphy. 2011. Believable dead reckoning for networked games. In *Game Engine Gems*. Vol. 2.