Peer-to-peer Infrastructures for Games

Tonio Triebel
University of Mannheim
triebel@uni-mannheim.de

Benjamin Guthier
University of Mannheim
bguthier@informatik.uni-mannheim.de

Richard Süselbeck
University of Mannheim
richard.sueselbeck@uni-mannheim.de

Gregor Schiele
University of Mannheim
gregor.schiele@uni-mannheim.de

Wolfgang Effelsberg
University of Mannheim
effelsberg@informatik.uni-mannheim.de

ABSTRACT

In this demo proposal we present Planet Π4, a Massively Multiplayer Online Game (MMOG) developed to evaluate and compare peer-to-peer-based MMOG systems with scalability in mind. The game requires low network latency and creates frequent game state updates. It has a modular architecture that can be adapted and extended with new functionality. Using this modular design we have developed different peer-to-peer infrastructures. Workshop participants will be able to play the game and compare the versions with each other.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

3D-Multimedia, Peer-to-Peer, Online Games

1. INTRODUCTION

Massively Multiplayer Online Games (MMOGs) such as World of Warcraft have recently made a significant commercial as well as cultural impact. These games are currently based on a client-server structure. This has several disadvantages, including high costs for server operation and bandwidth. Using a peer-to-peer architecture to run an MMOG as a fully distributed system can potentially solve these problems and has been the subject of research in the last few years. To achieve this goal a number of challenges such as scalability, consistency and security need to be addressed.

In this demo we describe a first-person shooter massively multiplayer online game (FPS-MMOG), called Planet Π4 after the name of our research group , "Praktische Informatik 4". Our primary design goal in developing Planet Π4 was to create a game suitable for evaluating research into peer-to-peer-based MMOGs. In the game players fly a spacecraft over the surface of the Planet Π4 (see Figure 1). The goal is to shoot at the opposing players and destroy their spacecraft.

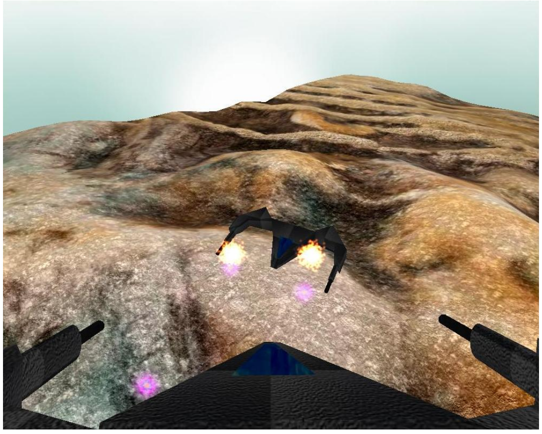

Figure 1: Screenshot of *Planet* Π4

This action-oriented design requires low network latency and results in frequent game state updates that are distributed among the peers. This allows us to test our algorithms under high system load. In addition we wanted to create a game that was easy to adapt to different peer-to-peer approaches. Therefore we designed Planet Π4 as a modular system that can be extended with new functionality. With this demo we plan to present Planet Π4 to the research community and to motivate other groups to use it for their evaluations.

2. GAME ARCHITECTURE

Planet Π4 is implemented in C++. The game is platform independent, currently runs on both Windows and Linux and supports cross-platform play. Its architecture consists of four different components (see Figure 2): 3D Engine, Game Logic, Group Selection and Networking.

3D Engine: The game's 3D environment is rendered using Irrlicht [1]. Irrlicht is an easy to use open source 3D-engine, that offers a full set of modern features, such as texture animation, parallax mapping, dynamic shadows and particle systems.

NOSSDAV '08 Braunschweig, Germany
Copyright 2008 ACM 978-1-60588-157-6/05/2008 ...$5.00.

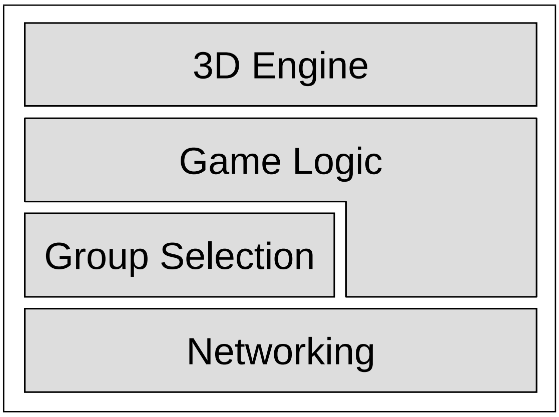

Figure 2: Game architecture

Game Logic: The game logic component is responsible for computing and maintaining the state of the game world. Based on the player's input, it calculates the movement of the spacecraft and projectiles. It provides collision detection and offers a damage model for the spacecraft. The game logic generates the events that are transmitted via the network.

Group Selection: Many research approaches for peer-to-peer-based MMOGs are based on locality of interest, i.e., players are only interested in events that are occurring in their vicinity. This can be exploited to find solutions for a variety of research challenges. For example, network traffic can be reduced by sending update messages only to the group of peers who are affected by them, thus increasing scalability, while consistency only needs to be maintained between groups of players that are interacting with each other. The group selection component is responsible for creating and maintaining these groups of players.

Networking: The networking component makes use of the groups provided by the group selection component, by delivering the updates that have been generated by the game logic component only to those peers for whom they are relevant. In addition it supports direct point-to-point communication.

We use this modular architecture to develop versions of Planet Π4 that are based on different peer-to-peer systems. To do so, we adapt the group selection and the networking component. These versions are described in the next section.

3. VERSIONS OF THE GAME

So far we have adapted Planet Π4 to three different systems and are working on an additional version.

3.1 IP-Broadcast

Our first version uses a very basic approach, based on IP-Broadcast. All peers are members of the same group. Therefore every event is delivered to all players. Group communication is realized by broadcasting to all other players in the same subnet. This does not scale well, but is sufficient as a proof-of-concept.

3.2 Skype

The second version is based on Skype. Skype is a popular peer-to-peer-based voice-over-IP application, which offers an API that allows delevopers to use its infrastructure. A key advantage of this infrastructure is its ability to transparently deal with firewalls and NAT. In addition, it offers voice and text communication for free. The game world is divided into several separate areas and the players are assigned to groups based on their location. We use the Skype feature of public chats to create the groups. An early version of this implementation is described in [4]

3.3 SpoVNet

The third existing version is based on SpoVNet [3]. SpoVNet (Spontaneous Virtual Networks) is a cooperative scientific project funded by the Förderprogramm Informationstechnik Baden-Württemberg (BW-FIT). Group selection is realized by a distributed event service, while group communication is based on an application layer multicast service.

3.4 Peers@Play

We also plan to use Planet Π4 for evaluation in Peers@Play. The Peers@Play project [2] investigates how peer-to-peer technology can be used to create distributed interactive world models. The Peers@Play middleware provides a group selection mechanism based on area-of-interest-management, and a peer-to-peer network that can deal with firewalls and NAT.

4. DEMO PLAN

We plan to demonstrate our game by setting up two game instances (SpoVNet, Skype) at the workshop site that will allow participants to play the game. In addition we will have several remote instances running, located at our laboratory in Mannheim. This allows us to demonstrate the behavior of the two versions in a realistic scenario with heterogeneous network latency and bandwidth. We will display actual traffic data for each individual implementation and compare the results.

5. CONCLUSION

In this proposal we have presented Planet Π4, an FPS-MMOG based on a peer-to-peer architecture. We implemented different versions of Planet Π4 that make use of the game's modular architecture. We plan to make it available to the community as an open source project.

6. REFERENCES

[1] http://irrlicht.sourceforge.net/.
[2] http://www.peers-at-play.org.
[3] http://www.spovnet.de.
[4] T. Triebel, B. Guthier, and W. Effelsberg. Skype4Games. In *Proc. of the 6th Annual Workshop on Network and Systems Support for Games: Netgames 2007*, Melbourne, Australia, 09 2007.