STUDY PLAN

This study plan proposes a research project focused on the theme of massively multiplayer online games (MMOGs). Although this kind of games is usually deployed on a client-server infrastructure, with a large and expensive central server, it is possible to use a network with a decentralized architecture. Therefore, ways of providing such decentralized support should be investigated, reducing the maintenance cost of these games, while still providing the basic requirements of MMOGs. This document presents the research plan to be followed by the applicant, Carlos Eduardo Bezerra, during his stay at the University of Lugano (USI). This work is part of the applicant's PhD thesis.

Context

In the past few years, thanks to widespread Internet access, traditional computer games have evolved into multiplayer online games. Initially involving a few players, these environments have grown quickly. Faster, cheaper, and more reliable Internet connections have started a new trend in Massively Multiplayer Online Games (MMOG), with potentially tens of thousans of simultaneous players. Moreover, studies show that massively multiplayer games will keep growing at a considerable rate.

The main reason for the success of these games is the kind of interaction they allow. Usually, such games contain a virtual environment where numerous players may interact in many different ways, both collaboratively and individually. Each player interacts with the virtual environment—or game world—by controlling an entity called an *avatar*, a representation of the player in the virtual environment. The avatar performs the actions requested by the player, interfering in the game world and potentially changing its state and the state of other avatars. Additionally, the game state should be durable, i.e., the changes resulting from a player's actions should persist, no matter if the player keeps playing continuously or disconnects from the game for a while.

Designing and implementing an MMOG poses a number of scientific challenges. In the traditional central server model, each player sends his (or her) desired actions to a single server, which processes them, calculating their outcome, represented as new states for the world and for some avatars. These new states are then broadcast to other players, so they can perceive the changes. The server also acts as an arbiter, who decides the actions performed by non-playable characters (e.g., an object), whose state is also sent to the players.

It is easy to observe that the central server eventually becomes a scalability and an availability bottleneck. Receiving and treating actions from thousands of players, and broadcasting the resulting state of each of these actions to every player will eventually saturate the server's processing power and, specially, its bandwidth. To address this prob-

lem, some systems replace the single server by a cluster of nodes connected by a high throughput and low-latency local network [1]. Obviously, such an infrastructure is highly expensive, preventing small game-developing companies or independent groups with low financial resources to enter the MMOG market.

The purpose of this research is to develop an innovative solution to MMOGs based on a decentralized approach. More specifically, it is intended to create a middleware which uses a decentralized overlay network such as those used for resource sharing (e.g., Pastry [2], Tapestry [3], Koord [4]) to create a geographically distributed system composed of low-cost volunteer nodes. This system will then act as a geographically distributed server to which the players connect via a gateway. Obviously, this model brings several questions, which were trivial to solve when using the traditional server architecture, but become much more complex in a geographically distributed scenario, such as: state consistency among the server nodes; coordination of entry, exit and crash of some of them, in order to keep the service available as much as possible; distributed persistence; optimal management of the scarce resources of the server system; and so on.

Studying in Switzerland

Prof. Fernando Pedone from the University of Lugano (USI) heads a research group working on the area of distributed systems and distributed data management systems. His work has addressed issues related to reliability, as well as replication in such systems by using group communication abstractions. These research topics are closely related to the MMOG study plan outlined here, for the intended MMOG server system has reliability as one of its critical questions to solve. Also, the distributed state persistence and consistency fit on the definition of replicated distributed data management system, only adding some requirements specific to MMOGs (such as timeliness on the retrieval and storage of parts of the game state).

In addition to developing the ideas that will eventually become part of the applicant's PhD thesis, spending one year in Switzerland will strengthen the collaboration between researchers from USI and UFRGS, the Brazilian institution where the applicant is currently doing his PhD. The close interaction between researchers from both institutions, working on related areas, will provide a good environment to the conception of new ideas regarding the research project presented here. Prof. Pedone has already started discussing with the applicant the ideas in this project. After a few personal meetings and several exchanges via e-mail, many details of the research plan have been detailed. The granting of a scholarship would be a final step to execute this project.

Work schedule

To achieve the goals outlined in the research project, a schedule has been designed, considering a nine-month academic year. This period has been chosen to fit the length of the scholarship. The tasks to be performed in Switzerland are:

1. Detailed definition of the services to be provided by the MMOG decentralized support middleware, such as:

   (a) Synchronization of the virtual environment simulation among the server nodes;

   (b) State storage and retrieval;

(c) Management of the entrance and exit of nodes in the server system; and

(d) Fault-tolerance, by allowing the system to keep working correctly even upon unexpected disconnection or crash of some servers.

2. Research and development of innovative solutions to deliver these services, considering the needs of MMOGs:

   (a) Developing algorithms or techniques to manage the consistency of the game world simulation among the server nodes of the system;

   (b) Building a distribute persitence scheme to store and retrieve the persistent state of the game, using replication and providing reliability, while minimizing the delay of each operation;

   (c) Designing algorithms to reduce the inter-server communication overhead, preventing state-related network messages from overloading the server system; and

   (d) Creating an abstraction for these services so that the game developer who uses the intended middleware will not have to consider questions regarding the distribution of the server.

3. Creation of a prototype implementing these services.

4. Validation of the proposed middleware by simulations.

Also, besides attending a local language course, publications are expected. The work schedule for the period in Switzerland is intended to be as follows:

Task	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May
Language course	X	X	X						
1	X								
2 (a)		X	X						
2 (b)				X	X				
2 (c and d)						X	X		
3						X	X	X	
4									X
Writing of papers			X	X	X	X	X	X	X

REFERENCES

[1] Wu-chang Feng. What's Next for Networked Games? *Sixth annual Workshop on Network and Systems Support for Games (NetGames)*, 2007.

[2] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, 2001.

[3] BY Zhao, L. Huang, J. Stribling, SC Rhea, AD Joseph, and JD Kubiatowicz. Tapestry: a resilient global-scale overlay for service deployment. *Selected Areas in Communications, IEEE Journal on*, 22(1):41–53, 2004.

[4] M.F. Kaashoek and D.R. Karger. Koorde: A Simple Degree-Optimal Distributed Hash Table. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 98–107, 2003.