

Perugia, ITALY - April 2007

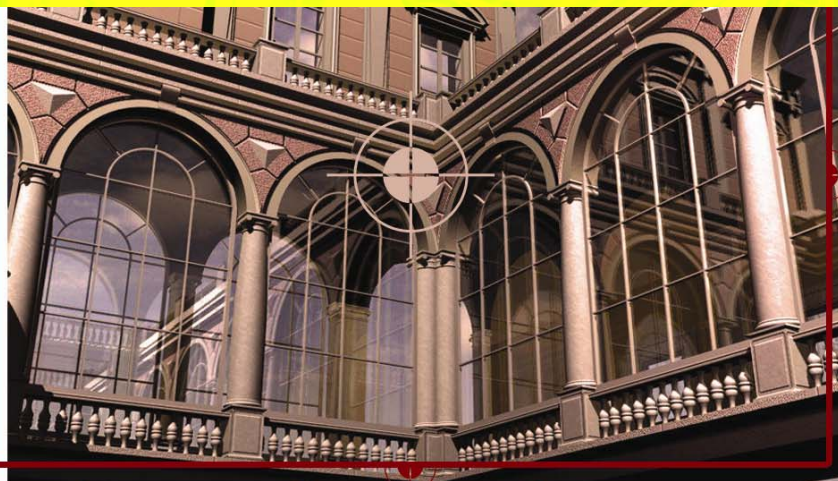
Web3D Games 2007

1st International Workshop

Editor:

Alessandro De Gloria, University of Genoa

Co-Editors:



Website: <http://www.web3d.org/web3d2007/workshops.html>

WEB3D GAMES 2007

1st International Workshop

Workshop Chairs:

Alessandro De Gloria, University of Genoa

Richard Bartle, University of Essex

Brunhild Bushoff, SAGASnet

Program Committee:

Giancarlo Bo, Giunti Labs, Italy

Craig Lindley, Gotland University, Sweden

Matija Marolt, University of Ljubljana, Slovenia

Mark H. Overmars, Utrecht, the Netherlands

Volker Paelke, Leibniz Universität Hannover, Germany

Matthias Rauterberger, Technical University of Eindhoven, the Netherlands

Stefanos Vrochidis ITI - Thessaloniki, Greece

Marc Price, BBC, U.K.

Table of Contents

A survey on Virtual World Creation Platforms	4
<i>Richard A. Bartle</i>	
Testing and evaluation of new platforms for delivery of distributed gaming and multimedia	7
<i>Audrius Jurgelionis</i>	
<i>Francesco Bellotti</i>	
<i>Peter Eisert</i>	
<i>Jukka-Pekka Laulajainen</i>	
A Web-Based 3D Gaming Style Multi-user Simulation Architecture	14
<i>David A. Ostrowski</i>	
PANORAMA: A Rich VRML Application Platform For Online Gaming	19
<i>Si Yin</i>	
<i>Anton Eliens</i>	
Road Rider: an interactive 3D game to promote road safety	23
<i>E. Baldi, F. Bellotti</i>	
<i>R. Berta</i>	
<i>M. Pellegrino</i>	
<i>L. Primavera</i>	
<i>A. De Gloria</i>	

A survey on Virtual World Creation Platforms

Richard A. Bartle
Dept. ESE
University of Essex
richard@mud.co.uk

ABSTRACT

Although they began as a form of entertainment, virtual worlds are increasingly attracting attention in educational and business circles. Individuals, universities and companies are looking to create their own virtual worlds.

However, creating virtual worlds from scratch is very expensive. Fortunately, there is a growing number of tools available for virtual world creation. Each of them offers a different set of features for virtual world development, with different attractions to different potential developers. Finding the right one is tricky, however, because they're not always easy to track down.

This paper lists the main virtual world development platforms available, and briefly outlines their strengths and weaknesses.

1. INTRODUCTION

Content in a virtual world is traditionally created in one of two ways: by developers; by users. All virtual worlds are created by a developer, of course, but the question is the degree to which users can create content within the virtual world. In developer-created worlds (which tend to be game-like, for example World of Warcraft [1]), users' ability to create is strictly limited; in user-created worlds (which tend to be social, for example Second Life [2]), it is open-ended.

Ultimately, this is a false dichotomy because at some point even user-created content becomes inviolate. If someone were to create a game-like sub-world within a social world, then that game-like world would be as limited as any other, despite its being wrapped in a free-form creation world. At some level, all content eventually becomes "owned".

If you are thinking of creating a virtual world, you therefore have the following options:

1. Write everything from first principles.
2. Use a commercial virtual world engine.
3. Use a virtual world software development kit.
4. Use an integrated creation and hosting solution.
5. Embed within a host virtual world.

Writing from first principles gives the most control, but involves a great deal of effort which costs a great deal of money. It is in attempting to avoid this expense that the other options have arisen.

The remainder of this paper outlines the various solutions currently available.

2. COMMERCIAL ENGINES

Although middleware such as this is not as expensive as writing something from scratch, it doesn't come cheap. These solutions are mainly aimed at commercial developers.

AliceServer [3] is an AI-driven engine, the technology for which allows for the creation of deeper ("more realistic") virtual worlds. Nicetech have signed up with the BBC to develop the Tronji virtual world. However, they prefer to work in partnership with developers, rather than simply selling middleware.

An established technology, albeit without any major hits to date, BigWorld [4] boasts that it is "fast becoming the industry standard". It comes with an integrated suite of development tools. There is, however, no academic solution available "yet".

The Chaos 3D [5] engine is fairly new, and although it is aimed squarely at commercial developers the team does seem at least amenable to the possibility of issuing non-commercial licences. Its main strengths are the tools it provides; the engine itself is primarily for graphics rather than for worlds.

Emergent [6] developed the well-respected (and well-used) Gamebryo 3D graphics engine, and is shortly to release a server engine which integrates this. Its technology is cross-platform (i.e. PC and consoles).

Simutronics is a company which has been developing textual virtual worlds for decades. Hero Engine [7] is its graphical offering, and it's impressive enough to have been adopted by BioWare for their upcoming virtual world. Its development tools are purported to be the best currently available. Again, it's strictly commercial – academics have to pay like everyone else.

Multiverse [8] was developed by the team that wrote the Netscape browser. Its business model is based on a policy of revenue sharing, which means that if you don't make any revenue (which is the case for educational worlds) it's effectively free. As a result, several university research projects are already using the Multiverse engine. There is more work to be done on the engine (customer service tools aren't great, for example), but it's benefiting from its connections with academia all the time.

Turbine [9] is the developer of a number of commercial virtual worlds, and their engine (if not their game

designs) is very well respected. Turbine are agreeable to licensing the engine to other developers, although at a cost. There are rumours that Turbine is looking to make its technology more available once its current projects (Dungeons and Dragons Online and The Lord of the Rings Online) are less of a draw on its resources.

3 SOFTWARE DEVELOPMENT KITS

For those who can't afford a commercial engine, a SDK is the next-best thing. It will lack some of the features, tools, integration and support of a fully-fledged system, but it is nevertheless sufficient to create a viable virtual world.

Although Fantasy Realm Online [10] is a 2½D world-creation system, and therefore (strictly speaking) shouldn't be in this report, it's a very nice-looking package. Its Ultima Online approach to graphics makes visual content much easier to create than fully-fledged 3D systems .

Irth Online [11] is an independently-developed virtual world, which in 2006 announced that it would be making its game engine and client available for people who wished to develop their own independent virtual worlds.

Kaneva [12] has a rather patchy approach, seeming to release as middleware the software it has written to create its own online games. It offers a software suite that includes all the major components needed to create a virtual world, but these are not fully integrated (and not as complete as other solutions). Nevertheless, it's free to use for educational purposes.

Minions of Mirth [13] is an indie virtual world that's up and running. It uses the Torque engine and is scripted in Python; players are encouraged to host their own servers with their own content – and they do.

A virtual world development kit based on the Torque Engine, MyDreamRPG [14] retails at under \$200. Finding out its capabilities is hard without actually paying the money, but it's getting good press among independent developers.

The NeL (“Nevrax Library”) [15] platform is the basis of the virtual world Saga of Ryzom. Most of the basic components for virtual world creation are available for free, but a full virtual world skeleton costs half a million Euros – or at least it would have done, had Nevrax not gone bankrupt in late 2006. Now, a number of organisations (developers, player groups, venture capitalists) are looking at the technology with a view to salvaging it.

Neverwinter Nights 2 [16] is a commercial role-playing game sold in game stores, which includes a development kit that can be used for making multi-player modules. Like its predecessor, Neverwinter Night, it has attracted attention from the modding

community, and a number of player-hosted modules are available. It's limited, however, in that it's set up only for a Dungeons & Dragons style world, with an associated ruleset.

Planeshift [17] is a free virtual world developed under an open source licence. You can take the source code and develop your own virtual world using it if you so desire – just so long as you assign copyright to the (non-profit) developers of Planeshift, Atomic Blue. The art, animation and audio assets for Planeshift are explicitly not included in the open source declaration: you need to create or buy in your own separately.

Project Darkstar [18] is Sun's attempt to create a virtual world toolkit. They're using Java, and it's open source. However, it's not available for use yet, and its capabilities are not clear.

For a one-off payment of \$100, you can download the RealmCrafter [19] engine and associated toolset tools. Very shortly afterwards, you can have the sample 3D virtual world up and running. The graphics quality is not state-of-the art, and you don't get the source code for the tools, but it's a great deal for small independents (and that would include academics).

As one of the longest-running open source virtual world development systems, WorldForge [20] has a lot of maturity. It's flexible, and has been used for a number of designs (although none are fully complete or played for fun – it's strongly developer-driven). Its tools aren't very comprehensive, however.

4. WEB 2.0 SOLUTIONS

Web 2.0 solutions have easy-to-use development tools online, and the results are hosted online. Although seen from the provider's perspective as user-created content, from the user's perspective they're individual and distinct.

The Areae technology [21] features a very user-friendly drag-and-drop creation system, however the company has only recently been formed and its offerings are still at pre-beta level. It promises to be a huge success, though.

Still in beta, Outback Online [22] is an attempt to create the virtual world equivalent of MySpace (and the 3D equivalent of Cyworld). Its big innovation is that it's peer-to-peer rather than client/server, which makes it far more flexible than Second Life and There.

5. WORLDS WITHIN WORLDS

If someone else has built a virtual world and contained within that world the tools to create objects and (crucially) functionality, then it's possible to use that virtual world to host a sub-world. Furthermore, it costs very little to do this, especially if other users of that world are happy to contribute to the project.

It's certainly possible to create a virtual world within Second Life [2], because people have done just that. The results are not entirely satisfactory, however, because of competition for resources with other users of the system. It has particular problems with flash crowds – large numbers of players appearing spontaneously in a usually unbusy area. Nevertheless, it's the market leader in social worlds, and there is a large pool of talent to draw upon. If you want, say, a medieval castle or a forest of thorns built for you, there are people in Second Life who will do it at very reasonable rates (perhaps even for free!).

Similar in philosophy to Second Life, There [23] lost out because of its more conservative attitude to content-creation. Although anyone can create anything for There, it has to go through an editing process to ensure that it's suitable; this slows down creation considerably (and makes it more expensive, too). However, There seems willing to do what Second Life currently isn't: licence its server technology to others. Thus, although embedding a virtual world within the main There world might be rather a painful bureaucratic exercise, writing it within a There you host yourself is a lot more attractive.

6. CONCLUSION

There are several virtual world development systems available at the moment, with more being announced all the time. They range in scope from simple drag-and-drop or menu-driven systems to fully-fledged commercial engines that will deliver a professional-quality experience. Some are open source, some are free or low-cost, and others charge a premium. Academics may or (usually) may not be able to get a special deal because of their non-profit standing.

The overall trend is that the cost of virtual world software development is coming down and the quality of the available tools and middleware is going up. Sadly, this is not yet the case for assets: art and animation remain expensive to create, and even large repositories price models beyond the price of independent developers. Thus, although virtual worlds will cost less to create in the medium term than they do at present, it won't be until the price of artwork falls that they could actually be regarded as easily affordable.

8. REFERENCES

- [1] Pardo, Rob; Kaplan, Jeff and Chilton, Tom: World of Warcraft. Blizzard Entertainment, 2004
- [2] Rosedale, Philip and Ondrejka, Cory: Second Life. Linden Labs, 2003.
- [3] <http://www.nicetech.co.uk/>
- [4] <http://www.bigworldtech.com>
- [5] <http://www.chaosgameengine.com/>
- [6] <http://www.emergent.net/>
- [7] <http://www.heroengine.com/>
- [8] <http://www.multiverse.net/>
- [9] <http://www.turbine.com/>

- [10] <http://www.fantasyrealmonline.com>
- [11] <http://www.irthonline.com/>
- [12] <http://www.kaneva.com/>
- [13] <http://www.prairiegames.com/>
- [14] <http://www.mydreamrpg.com/>
- [15] <http://www.nevrax.org/>
- [16] <http://www.atari.com/nwn2/>
- [17] <http://www.planeshift.it/>
- [18] <http://research.sun.com/projects/darkstar/>
- [19] <http://www.realmcrafter.com/>
- [20] <http://www.worldforge.org/>
- [21] <http://www.areae.net/>
- [22] <http://www.outbackonline.com/>
- [23] <http://www.there.com/>

Testing and evaluation of new platforms for delivery of distributed gaming and multimedia

Audrius Jurgelionis
University of Genoa
Biophysical and Electronic
Engineering Department
Via Opera Pia 11a, 16145 Genoa,
Italy
+39 010 3532780
jurge@elios.unige.it

Francesco Bellotti
University of Genoa
Biophysical and Electronic
Engineering Department
Via Opera Pia 11a, 16145 Genoa,
Italy
+39 010 3532780
franz@elios.unige.it

Peter Eisert
Fraunhofer Institute for
Telecommunications
Einsteinufer 37,
10587 Berlin, Germany
+49 30 31002 614
eisert@hhi.fraunhofer.de

Jukka-Pekka Laulajainen
VTT Technical Research Centre of
Finland
P.O. Box 1100, FI-90571 Oulu,
Finland
+358 20 722 111
**jukka-
pekka.laulajainen@vtt.fi**

ABSTRACT

This paper presents Games@Large - a new system for pervasive gaming and multimedia and describes its components, which are relevant to the testing and evaluation of the system design. Complex gaming and multimedia frameworks comprise multitude of elements: heterogeneous end user devices, wireless and wired network technologies, and various multimedia and gaming services. They enable an easy access to the offered services through the use of a versatile system architecture. Testing and evaluating the design of the system takes central stage in the project through an iterative process which has a great impact on overall design course and the final product in terms of assuring a proper Quality of Service and usability of the system. The analysis of system elements and their testing and evaluation methods will facilitate the development of the methodology for the purpose of verification of the different components and the entertainment system as an integrated whole. The subsequent step of this analysis is an adaptation of the metrics, used for video game and multimedia evaluation, which will develop a comprehensive testing and evaluation methodology for newly evolving entertainment systems.

Keywords

Testing, evaluation, usability, user experience, multimedia, video games, pervasive entertainment.

INTRODUCTION

The future home is an always-on connected digital home. By 2010, there will be over 420 million broadband households worldwide [2], [3]. With the standard set for super-high speed, always-on connection, the way people view entertainment has fundamentally changed and created a new standard for consumption. Consumers no longer expect their internet access to be only from a

desktop PC - now they want it through the TV in the living room or in the palm of their hand, inside the house and on the go.

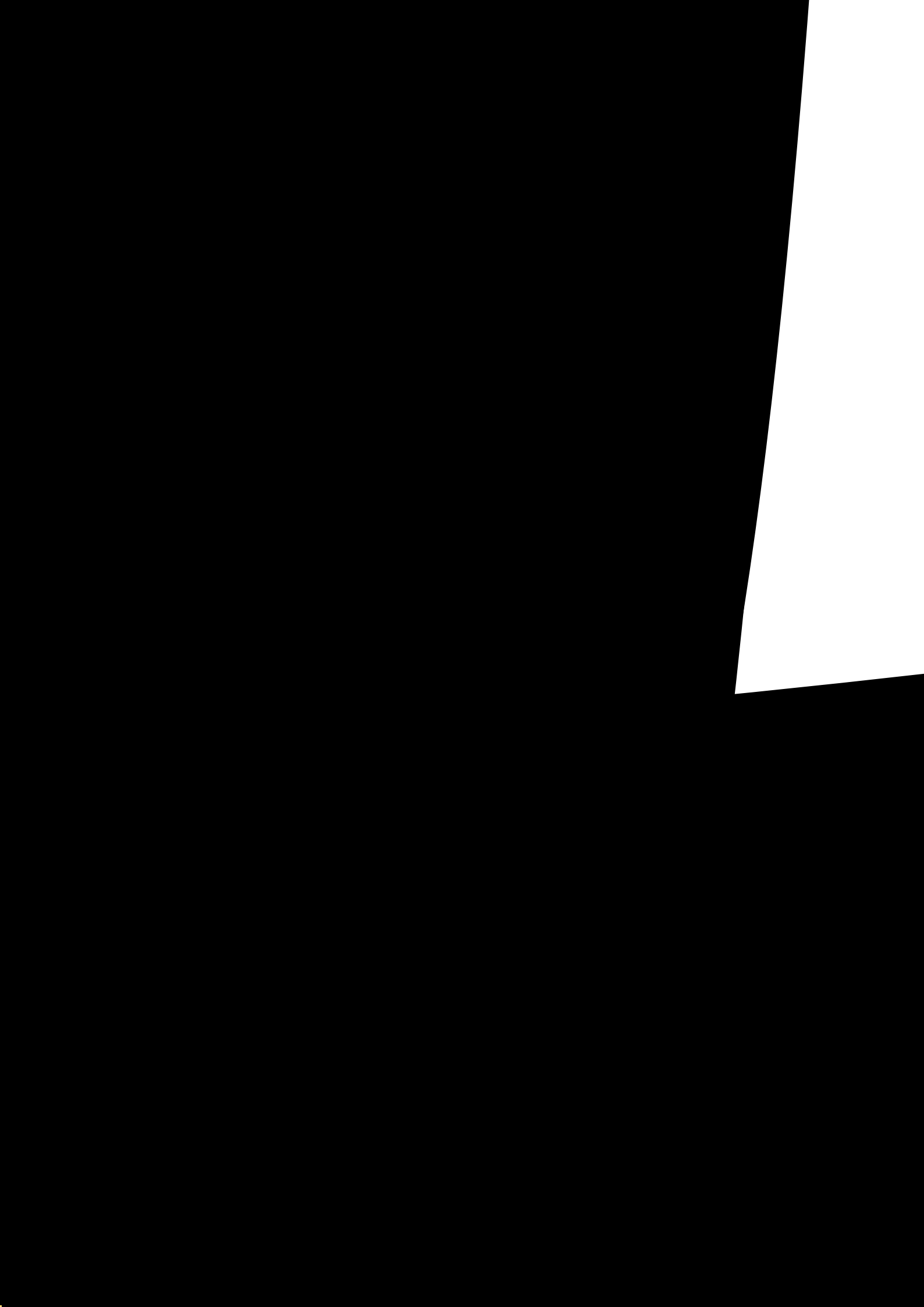
The presented scenario [4] bundles video gaming capabilities into consumer electronics devices, such as Set-Top Boxes (STBs), Digital Video Recorders (DVRs), home entertainment systems, TVs, handhelds and other devices that are not considered, today, as real gaming devices since they lack the necessary CPU power and graphical performance.

In this paper we present a new system for pervasive gaming and multimedia, which is being developed under the EU FP6 project, Games@Large (GaL) [1]. The paper is dedicated to the design testing concept elaboration, in order to base the approach for the development of evaluation and testing methodologies. Testing and verification process is part of the iterative, spiral-life workflow model (user-centred design and incremental improvement based on feedback from user and expert evaluation of prototypes).

In section 2 we present the new framework for pervasive gaming and multimedia, and QoS evaluation methods that are relevant to the design testing and evaluation. In section 3 we analyse relevant GaL system components, properties and possible testing methods, which are based on a literature review. Analysis information includes specifications for testing of different GaL elements, i.e. GUI, network impairment effects on user experience in video games, multimedia tests, etc.

SYSTEM FOR PERVASIVE GAMING AND MULTIMEDIA

The increasing number of broadband users, and the demand for service quality and diversity, especially in the entertainment area, drives to the development of new



multimedia services not only depends on network parameters, but also on higher layers' characteristics (coding and compression of the multimedia, recovery algorithms, nature of the content, etc...). In this sense, a final user might experience acceptable quality levels even in the presence of serious network problems. These observations show that rating the quality of new multimedia services (the Games@Large service suite) from the network side may no longer be effective. The user Perceived Quality of Service (PQoS) field addresses this problem, assessing the quality of a service as perceived by end users. This seems to be in fact the most coherent approach: after all, the user is the one who pays for the service, and QoS will reflect what he/she understands as such, independently of the state of the network that transports the service. [5]

The assessment of perceived quality in multimedia services can be achieved by two different kinds of methodologies, either subjective or objective [5].

The GaL testing is mainly based on subjective methods, as it implies the user-centred design process. Subjective methods define the most accurate metric as they present a direct connection with the user experience. This metric is suitable for evaluating and testing such a complex system, which comprises various elements and devices.

Objective non-intrusive methods can also be applied in parallel with subjective methods, as they are providing useful quantitative data.

Next chapter is dedicated to the analysis of the GaL system and its components in the context of the whole system testing and evaluation. The approach is based on subjective methods, although objective as well as functionality tests are also considered for some of the GaL components.

TESTING CONCEPT

This section describes the general concept of the GaL system testing. In order to better evaluate each system component and its importance in the whole system testing, relevant GaL framework properties are differentiated into 3 prime groups: system properties, user properties, and application properties. These groups reflect the key properties of different GaL elements, and show their impact to the system development and testing.

• System properties

System properties or technical properties can be categorized in 3 classes: software testing, hardware testing, and system testing.

Software testing

Software in the GaL system consists of modules (video streaming, pre-rendering, and GUI modules) made by different partners. Before integrating the modules each should be tested separately. After the integration testing methods for these modules are described in application properties section.

Hardware testing

During the project also hardware is produced and must be tested. This hardware consists of Enhanced Set-Top

Boxes and Wi-Fi network equipment. Other hardware used does not have to be tested as such, but tests should be made to conclude whether the used hardware is suitable for GaL. This list includes local processing servers, storage servers, handheld devices, and network components like routers.

The WiFi Quality Assurance team will perform a series of well-defined tests to verify the design on software and hardware level, and provide a product able to get Wi-Fi Alliance's certificate, as well as pass all regulatory domain tests for certification (i.e. FCC).

System testing

The individual software and hardware components will be tested separately as described above. In addition to testing individual components, system-level testing for the whole GaL system is foreseen. The aims of system testing include finding out problems and properties of the entire system rather than of its separate components. From the user's point of view, the overall system performance is crucial, as the user is not using its components one by one but as a single service. Thus, the user's perceived quality of the system can be assessed with the results of system testing.

There are many technical properties that affect the system performance. The properties can be categorized in user-related, server-related, end-device-related, and network-related classes. The most important properties to be considered are listed below.

User related:

- Number of concurrent game users
- Users using other services through the same network
- User's performance related to delay of controls and graphics output (Recommended/acceptable input delay, recommended/acceptable graphics delay)

Server related:

- Processing capacity: The ability of the Local Processing Server (LPS) to execute and serve multiple game sessions (available CPU power)
- The ability of the Local Storage Server (LSS) storage to supply requests from multiple LPS servers to run multiple game sessions in the network (available disk space, available data transfer speed)
- Other programs running on the same computer (required disk space, required CPU power)
- 3D graphics streaming properties: The ability of the protocol to serve games and the compatibility of the protocol with graphics libraries (OpenGL, DirectX)
- Used video codec (coding efficiency, computational complexity, encoder delay)
- Used transport protocol

End device related:

The Games@Large software is designed to work on several different devices like Enhanced Set-Top Boxes and handheld devices. All possible combinations should be tested to see that the GaL framework actually works as

planned. At least the following parameters should be changeable while testing client devices:

- Processing capacity (by changing devices or simulating lower capacity)
- Clients: The ability of the end device to process the graphics/video input and controls input. The delay should be minimized to achieve minimum delay and maximal user experience
- Screen size, form factor (240 x 320, 1080p, etc), may influence the user experience for each device

Network related:

As multimedia applications develop, communications systems are expected to make provision for two of their aspects, namely Quality of Service (QoS) support and group communications services [6].

This section describes general communication characteristics independent of any layer in communication architecture.

- *Available bandwidth (downlink/uplink)*

Playing 3D graphics or video streaming based games sets a requirement for bandwidth in the order of several megabits per second depending on target resolution and frame rate for video and scene complexity for graphics streaming. If the available bandwidth for gaming is not sufficient, the game experience will be deteriorated. In the case of connection-oriented protocol, congestion in either direction causes problems to both game stream and control channel. If the protocol is connectionless, downlink bandwidth is critical for game streaming and uplink for control channel.

- *Delay (downlink/uplink)*

Games are very interactive and require minimal delay from the network. Both downlink and uplink delays should be minimized to enable the fast response to user commands [13]. This significantly differs from other multimedia applications like IPTV or music streaming, where delays can be tolerated to some extent. Minimizing the delay for the transmission of video and graphics is one of the most challenging aspects in Games@Large.

- *Jitter (downlink/uplink)*

Unlike the case of traditional video streaming, there is no possibility for using long jitter buffering in the case of game applications. Thus, jitter should be minimized. Too high jitter degrades the game experience as the response time of the game varies.

- *Packet loss (downlink/uplink)*

The effect of packet loss to user experience depends on the 3D streaming protocol. At least some of the 3D data are such that loss cannot be accepted and lost parts have to be recovered by retransmissions or forward error correction. In the case of video streaming-based gaming, some packet losses may be accepted if proper error concealment or resiliency techniques are used.

- *QoS enabled/disabled*

If the network supports QoS, the game sessions can be assigned higher priority in the network. This enables the smooth operation of game applications regardless of the other traffic. If QoS is not supported by the network, the game traffic competes with the other traffic with the same priority. This means that the game experience is affected if there is too much other traffic in the same network.

- *Wireless properties*

Presence of RF interference. RF interference may deteriorate the network performance, which in turn, degrades the user experience of the game user.

- Distance from the access point.

If the game users reside too far away from the access point or there are obstacles between the stations and the access point, the signal strength may decrease. This causes the usable bitrate to decrease, which affects the maximum possible number of users.

• User and stakeholder characteristics and impact

GaL targets at four basic scenarios: house, internet cafe, hotel, elderly house. User demographics plays an important role in the whole testing process. It should be noticed that stakeholder or environmental properties, and their implications to user demographics have to be considered when selecting test participants. In early project stages a detailed analysis and predefinition of possible choices of test participants has to be performed. When performing tests the test participants have to be chosen according to the specifications for each environment. This characterizes the users that should be selected and defines the criteria for the selection. With respect to appropriate use cases at different GaL focus areas, it will facilitate testing process procedures and enable to obtain useful and objective test results.

Environmental analysis is performed in terms of contextual analysis, which is differentiated in three categories of characteristics that bear direct relevance to gaming and game experience: the social, physical, and service context.

Finally, stakeholder properties, such as infrastructure capabilities, user demographics differences, offered services and requirements to the system will none the less impact testing and evaluation. And in particular, in test participants' selection, i.e. elderly houses have a different distribution of the age of users and different requirements to the GaL system, than hotels or internet cafes.

• Application properties and components

The description below presents Games@Large application components and relevant properties. They represent GUI, games, video, and other services and interfaces that users are interacting with. Therefore, testing those components from the user's point of view (subjective method) implies testing the whole system and its performance. As it was motioned previously, objective tests of separate system components are useful for obtaining quantitative evaluation.

Literature analysis on possible testing methods which can be applied in GaL testing process is conducted in order to base the development of GaL testing methodology.

Graphical user interface

According to given specifications, the GUI will be a web-based user interface, which will be transferred to various end devices, such as TVs, EHDs and laptops.

Functionality testing enables evaluating how the application and the user interact. This includes how the application handles keyboard/gamepad and mouse input and how it displays screen text, images, buttons, menus, dialog boxes, icons, toolbars and more. Functionality testing is commonly done by human testers, or by automatic tools. Given their increased importance, testing GUIs for correctness can enhance the entire system's safety, robustness, and usability. GUI designers during the development process will test its functionality. In the GUI testing, it is important to evaluate how it meets its written specifications.

An iterative user-centered design (UCD) design process is used during the development of a graphical user interface (GUI) for the GaL system. The fourth phase in an iterative process is the evaluation phase during which feedback about the interface is obtained through the functionality and usability testing [7].

Usability testing is defined as: "In System Testing, testing which attempts to find any human-factor problems". A better description is "testing the software from a user's point of view". Essentially it means testing software to prove/ensure that it is 'user-friendly', as distinct from testing the functionality of the software. In practical terms it includes ergonomic considerations, screen design, standardization etc [7].

Multimedia

GaL may work simultaneously with a variety of other multimedia services such as: Video on Demand, VoIP, IPTV, and DVR.

These services will not be developed in the project, but may be integrated into the same network. The testing process is likely to focus on the effects of games on multimedia applications, when both are running concurrently and consumed by the one or several end-users simultaneously.

Testing processes are intended to cover several types of multimedia applications, which will be available in different target environments, in order to comprehensively verify the current system design.

Testing should use different types of multimedia clips, representing different classes of motion contents. Performance should be evaluated in terms of infrastructure utilization, loss rates, number of simultaneous customers served and end-user perceived quality [8].

An additional assessment of GaL performance in terms of network-related and device-related parameters will be done. Subjective tests will be performed in order to determine the real end-users' perceptual assessment of the

quality of multimedia clips streamed using the GaL infrastructure.

Since the perceived quality of any multimedia sequence is highly influenced by the associated sound (e.g. audio and video must be appropriately synchronized), all clips should be streamed with their soundtracks.

The verification tests used to assess the performance of different coding standards can be slightly modified and adapted to the GaL testing purposes. Here we will not only consider the encoding efficiency, but rather we will investigate overall system performance, overall delay and user perceived quality, when multiple processes are running on the server (multiple game applications and/or multimedia applications are running concurrently). Certainly, a number of concurrent end users (processes) and application types should be varied. Network simulation tests will be performed in order to investigate and evaluate the influence of network parameters and to obtain data for quantitative evaluation.

The methodology of the AVC compression performance assessing is given in [9], [10], [11].

Video Games

This subsection describes testing of different types of games which will be covered in the GaL design process. Each game type described below may have unique QoS requirements that should be categorized during the project. Multiple games that represent game genres and types will be selected and evaluated.

While designing a network-based entertainment system, it is important to understand how the degradation of network conditions affects the overall user performance.

Tests of first person shooter games (FPS) revealed latency to be a critical issue. While movements are not significantly affected by network conditions, precision shooting is highly affected by high latencies. Moreover, the game becomes less reactive and less enjoyable at higher latencies. While low values of packet loss (0%-4%) might be acceptable, it is important to keep the latency at low values (<100 ms) to ensure the quality of the service [12], [13], [14], [15]. However, it must be noted that all these studies have been focusing on traditional type of networked gaming where the game software runs on each player's computer and the network is used only to synchronize the players' contexts together. Thus, the results may not be exactly applicable to GaL type of gaming.

Further studies may also be performed on different kinds of games, such as strategy and racing games, where network performance decay may not alter the overall quality of the games as much as it does in FPS games.

Tests will be done while simulating a wireless network connection, in which latency and packet loss are not constant and usually higher than in regular cabled connections.

A subjective and objective evaluation of the effects of loss and latency on videogames in the GaL system is therefore

useful for making required improvements or modifications in the design [12], [13], [14], [15].

On the other hand, frame rates and resolutions rendered in a computer game directly impact the player performance, influencing both the overall game playability and the game's enjoyability. Insights into the effects of frame rates and resolutions can inform system designers in their development of new platform. [16], [17] present results of carefully designed user studies that measure the impact of frame rate and frame resolution on user performance in first person shooter games. Contrary to previous results for streaming video, frame rate has a marked impact on both player performance and game enjoyment while resolution has little impact on performance and some impact on enjoyment [17].

There have been numerous studies that examined the effects of frame rate and frame resolution on users passively watching streamed video. These studies have found that a decrease in frame resolution corresponds to a decrease in user satisfaction, while a decrease in frame rate does not decrease user satisfaction as much. However, watching video, even during a video-conference, does not have the same interaction requirements, in terms of the required response time, as do some other interactive media applications [17].

The diversity of game hardware in GaL results in the same game being played at different frame rates and frame resolutions. A quantitative understanding of the effects of frame rate and resolution on game playability is therefore critical for GaL platform developers, to enable better targeting of platform improvements to aspects of the display that matter [17].

Finally, simultaneously running games and multimedia applications will impact each other's performance and definitely the Perceived Quality of Service. Thus the effects of multimedia on the video games should be considered and evaluated in the testing process.

Concluding this section we also have to mention controllability issues, since heterogeneous control devices and units are used in the GaL system. Special attention should be paid to gamepad control capabilities, as keyboard controls are mapped onto the gamepad. This issue has to be taken into account while considering usability issues in GaL testing process.

CONCLUSIONS

This paper has introduced a new system and its components for pervasive gaming and multimedia. The aim of the paper was to identify system elements and their parameters, which are relevant to the testing and evaluation of the design.

The GaL system and prospective entertainment systems consists of multiple elements and is intended to serve a number of different users in terms of cultural belonging and environmental dependence. Therefore testing and evaluating of such systems is a complicated task. For the purpose of verification of the different components and the entertainment system as an integrated whole, the GaL framework and its services were differentiated into 3 general groups (system, user and application properties).

The differentiation and analysis of the GaL elements, and their parameters and testing methods will facilitate the development of testing methodology intended for comprehensive system design verification. The testing process provides qualitative and quantitative results, which are evaluated and supplied to the system designers, thus advising them and facilitating the correction of the errors and improvement of the design.

ACKNOWLEDGMENTS

This article was composed within the process of testing and evaluation methodologies development for EU FP6 project, Games@Large.

REFERENCES

- [1] <http://www.gamesatlarge.eu>
- [2] http://www.findarticles.com/p/articles/mi_m0EIN/is_2006_Sept_26/ai_n16837715
- [3] http://www.emarketer.com/Report.aspx?bband_world_jun06&src=report_summary_reportsell
- [4] Y. Tzruya, A. Shani, F. Bellotti, A. Jurgelionis, Games@Large - a new platform for ubiquitous gaming, BroadBand Europe 2006, Geneva, Switzerland, November 2006
- [5] P. Casas, D. Guerra, I. Irigaray, User Perceived Quality of Service in Multimedia Networks: a Software Implementation, Joint Research Group of the Electrical Engineering and Mathematics and Statistics Departments, 2006
- [6] L. Mathy, C. Edwards, D. Hutchison, Principles of QoS in group communications, Journal Article, Telecommunication Systems, Springer Netherlands, 2004, pp. 59-84
- [7] Software Quality Assurance & Usability Testing, <http://members.tripod.com/bazman/usability.html?button6=Article+on+Usability+Testing>
- [8] G. M. Muntean, P. Perry, and L. Murphy, Subjective Assessment of the Quality-Oriented Adaptive Scheme, IEEE Transactions on Broadcasting, vol. 51, no. 3, Sept. 2005
- [9] T. Oelbaum, V. Baroncini, T. K. Tan, C. Fenimore, Subjective quality assessment of the emerging AVC/H.264 video coding standard, International Broadcast Conference (IBC), Amsterdam, Netherlands, Sept. 2004
- [10] C. Fenimore, V. Baroncini, T. Oelbaum, Tobias, T. K. Tan., Subjective testing methodology in MPEG video verification, Applications of Digital Image Processing XXVII. 2004, pp. 503-511
- [11] International organization for standardization, ISO/IEC JTC 1/SC 29/WG 11, Coding of moving pictures and audio, Report of The Formal Verification Tests on AVC (ISO/IEC 14496-10 | ITU-T Rec. H.264), JVT, Test and Video Group, 2003, Waikoloa Janice C. Redish, Joseph S. Dumas,

A Practical Guide to Usability Testing, 1999 Intellect Books

- [12] M. Claypool, The effect of latency on user performance in Real-Time Strategy games, Int. Journal of Computer and Telecommunications Networking, pp. 52-70, 2005
- [13] T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, E. Agu, M. Claypool, The effect of loss and latency on User Performance in Unreal Tournament 2003, Proc. ACM Network and System Support for Games Workshop (NetGames), Portland, USA, Sept 2004
- [14] T. Lang, User Experience while playing Halo with network delay or loss, Technical Report CAIA-TR-031205A, Centre for Advanced Internet Architectures, Swinburne University of Technology, Dec 2003
- [15] T. Henderson, The effects of relative delay in networked games, Department of Computer Science University College London, February 2003
- [16] T. Connor, A. Fiske and R. Kennedy, The Impact of Frame Rate and Resolution on Player Movement in First-Person Shooters, Interactive Qualifying project, Worcester Polytechnic Institute, 2006K
- [17] M. Claypool, K. Claypool and F. Damaa, The Effects of Frame Rate and Resolution on Users Playing First Person Shooter Games, Proc. ACM/SPIE Multimedia Computing and Networking (MMCN), San Jose, USA, Jan. 2006

A Web-Based 3D Gaming Style Multi-user Simulation Architecture

David A. Ostrowski
Ford Research and Advanced Engineering
1 313 322 3642
dostrows@ford.com

ABSTRACT

Infrastructure of the World Wide Web has matured to support real-time delivery of web-based 3D content [1]. This status supports the implementation of multi-user applications including gaming, simulation, training and scientific visualization. Original networking implementations relying on custom APIs (C, C++, Java) have demonstrated limitations, among them being lack of scalability, flexibility, compatibility (custom clients), lengthy development cycles and complexity in implementation [2][3]. These limitations support the need for a lightweight, convenient, browser-based architecture to view and manipulate 3D content. In this paper, we present a quick prototyping approach for the development of a multi-user environment through the employment of scripting languages (Javascript, Python), 3D Markup (VRML, X3D) and Ajaxian techniques (XHR, DHTML, FRAMES). This work supports the trend towards XML (AJAX, Web Services) as the primary supporting technology for data representation and networking on the web.

1. INTRODUCTION

3D environments have existed on the WWW for over a decade. High performance real-time 3D content was originally provided through the means of graphic APIs (OpenGL, Qt, Open Inventor) implemented as custom clients [4]. Later, the use of VRML, Java3D and X3D supported a movement to browser-based standards. Each of these approaches demonstrated constraints in their utilization and development when implementing real-time, multi-user interaction. Custom clients typically necessitate a software installation in order to support their use. This can restrict the number of users due to problems ranging from incompatibilities to inconveniences of software installation. Although VRML/ X3D standards have supported browser-based applications, there has not been a convenient means of two way communication between the browser and the server. Java-based architectures have provided a two-way, browser-based means of communication, but can present lengthy development cycles and limitations of accessibility within the context of an HTML page.

Our focus is to support the movement towards an efficient environment that enables the development of multi-user applications, providing functionality with the highest level of compatibility. Section Two provides motivation of this approach as well as identifying necessary background technologies. Section Three

provides an overview of our proposed architecture. Section Four provides a comparison with competing software architectures and applications. Section Five presents a case study implementation of this technology in an industrial simulation application. Section Six provides the conclusion.

2. BACKGROUND

Web applications have begun to demonstrate increased utility which includes higher use of participation, scalability, collective intelligence and remixable data sources. This trend is characterized as the next-generation web or web 2.0 [5]. The outcome of this trend is a higher reliance on served browser-based applications. Our goal is to contribute to this trend by adding utility to the browser-based standard.

2.1 Ajax

Due to its architecture neutral approach, XML has evolved to be a standard technology of web-based development. One of the most popular trends in web applications is enriching client functionality through the means of Asynchronous Javascript and XML (AJAX) [6]. AJAX is a combination of existing technologies rather than a completely new innovation. The primary component is the XMLHttpRequest (XHR) object that provides for the ability to perform asynchronous communication with a server via Hypertext Transfer Protocol (HTTP). Used in coordination with either Dynamic Hypertext Markup Language (DHTML) or use of HTML FRAMES, the XHR request object allows a high level of performance thus avoiding the call-and-response mechanism of earlier web applications. Among the commercial applications of these technologies are the widely used Google Maps and Google Suggest [7][8].

2.2 Integration of 3D and Ajax

Replacing the integration of DHTML functionality with the X3D event model, applications have applied the use of X3D markup technologies in the context of an AJAX-based application[9]. This approach has identified a trend of scripting the 3D scene hierarchy in the same fashion as the Document Object Model (DOM) and handling the events through Javascript. By relying on AJAX-based network calls, a user interacts with backend tiers via a 3D scene in asynchronous fashion previously provided by employment of HTML 'DIV' tags. This technique has supported a much higher level of interactivity and efficiency in web-based 3D

applications. This industry development has initiated a consortium effort to the definition of 3D functionality in conjunction with AJAX-based technologies[10]. Initial interest in this consortium effort is demonstrated through the employment of gaming architectures to this standard [11][12].

2.3 Further Integration of XML-Based Technology

Before the creation of the WWW, the term client/server computing referred to a custom client who initiated requests to a central server. This concept evolved to what was referred to as a three-tier architecture where client requests were handled by a backend database. As additional functionality was added between multiple servers it was necessary to derive an architecture-independent method of calling methods or procedures. The technology that was accepted as a standard was termed as a Web Service [13]. Two major categories of Web Services are XML Remote Procedure Calls (XMLRPC) and Simple Object Access Protocol (SOAP). XMLRPC is a standard for the delivery of XML-defined remote procedure calls between independent architectures. SOAP is known as a more sophisticated implementation supporting more complicated data representations. An implementation of web services includes definition of the REST architecture in which complete web service commands are developed in the context of an HTTP protocol. This approach is not a defined standard but rather an implementation style representing the trend of integration between AJAX techniques and web services [14].

The purpose of building web services is to be able to move to an "application-centric" web, where applications are called by other applications which form the premises of what is defined as the Semantic Web [15]. Industrial simulation such as Semantic eWorkcell has demonstrated the incorporation of 3D web-based interfaces to backend web services via AJAX-based technology [16][17].

2.4 Web-Based Architectures

Movement to the web-based application model, while providing many advantages has an identified inefficiency of HTTP being a request/ response protocol. [18] Here, the client must initiate the communication to receive a reply. While allowing for necessary security advantages, this makes it difficult to maintain efficient two-way communication between the client and server.

This limitation has been addressed through a number of frequent polling mechanisms that employ server-side functionality. Server information can be continually read by utilizing the HTML-based Refresh command [19]. By employing the 'Refresh', a server side process will automatically reload within a given time period. XHR requests can also be managed using system calls (timers) within CGI scripts to also support frequent polling of a resource. Server to client communication can also be handled via 'server-push' designs. An

example of 'server-push' is that of No-Parsed-Headers or NPH. This approach supports the server attaching output directly to the browsers input by means of bypassing a server script buffering mechanism [20]. Another technique termed as 'slow load' maintains a process that does not complete until a specified event has occurred. Upon completion, the event is reloaded through the utilization of the 'onload' HTML command [21].

2.5 Trends in Multi-User, Gaming-Style Applications

Since the explosion of the multiplayer gaming industry, the corresponding technology has influenced applications including simulation, education and training [22]. Traditionally, developers have relied on C or C++ exclusively for gaming development [23]. While providing a tight coupling with operating systems, this approach is slowly changing towards development in languages that are more portable and provide for faster development cycles (Java, Javascript, VB, Python, Lua) [23][24][25][26][27].

The existence of thousands of game architectures implemented in Java has emphasized our theme that compatibility and ease of development is representative of next-generation gaming-style applications. Finding parallels along with the success of architectures such as Hibernate, Spring, Zope and Ruby on Rails, the author feels most current trends support lighter, more portable technologies [28][29][30]. This trend currently supports movement to the AJAX-3D paradigm. This is evidenced by games being implemented in these technologies, server-push technologies embraced by vendors as well as advances in patented AJAX peer-to-peer applications [31]. Along with increased performance of the WWW these issues point towards a rapid development model for next generation gaming architecture.

3 OVERVIEW OF ARCHITECTURE

Our software design relies on the expansion of server-push through the incorporation of web service technologies. This architecture uses two servers (web-service based) in combination with an initial server (CGI). One server is designated for the purpose of maintaining a global data set. In order to update the global data set, an XHR request is sent to the server via the browser-based client. The initial server is used as a bridge to the update the data server through the means of a web service request. Here, the movement of a 3D object, the client can update the server via XHR (Javascript). The data server in turn, sends an update request to an event server which has been established by each individual client. This server which has been initiated by the client exists as a hidden FRAME. This design expands on the slow-load technique in that the hidden frame does not continually poll the server, but waits for the network call – acting as a server itself.

3.1 Detailed examination

The initialization process of our application consists of retrieving an HTML page containing the FRAMESET directive. The frames included in this page will exist as two CGI scripts. The first script will generate all VRML and handle all AJAX-based communication. The second frame will initialize a second CGI process which will exist as a hidden frame. This process will communicate via Javascript to the first CGI reloading itself through employment of the Javascript 'location' method. Each client will maintain a separate backend script for obtaining data updates from the central real-time data server.

4. COMPARISON

AJAX vendors have embraced applications of server-push technologies primarily in support of financial applications and e-commerce. The frameworks surveyed offer high performance server-push capability [32][33].

Disadvantages of most of these applications include the close integration with Java technology (adding complexity) along with limited portability across servers. AJAX push technologies have found acceptance across a number of web based games starting from board based games such as WEBOGGLE and working towards 3D interactive games including StrikeFighter[34][35]. While some of the architectures surveyed at this writing maintain similar qualities to our application, direct comparisons are difficult to perform. Advantages of our approach include support of web services, a portable solution that could be implemented across a number of architectures and a small program size completely implemented in scripting languages.

4.1 Competition to AJAX

Direct competition to AJAX based technology has included the application of FLASH development as well as Java Applets [36][37]. The Macromedia FLASH environment has been used to replicate existing games in the market. Utilized in conjunction with FLASH remoting, it can be utilized in the same fashion as XHR calls. One drawback of FLASH implementations in relationship to our goals is in the implementation of open standards. Also, with the support of FLASH based implementations true 3D effects are not supported.

Java Applets can be run behind a browser or described in a "faceless" mode [24]. In this approach, an Applet is interfaced with Javascript bypassing SWING or JFRAME based graphical input/output (unlike earlier descriptions of Applet to Servlet communication in this paper). This can perform similar benefits by maintaining a complete socket connection between the client and server. Disadvantages of this approach are that most implementations are limited by browser-compatibility.

4.2 Alternatives

Alternatives to our approach include utilizing the "faceless" Applet approach using scripting constructs to allow for a lighter developmental (non-bloat) approach. The author sees potential in this area by Applets being

implemented in JVM-based scripting languages (such as Jython, J-ruby, Groovy) allowing for Java server capabilities along with benefits of rapid development through utilization of scripting languages. Open source implementations of the FLASH web remoting also present interesting alternatives to our approach providing remote scripting capability in the context of an open architecture [38].

5. CASE STUDY

Semantic eWorkcell is an application developed at Ford Motor for the purpose of factory floor simulation. [39][40][41] Utilizing mod_python scripts on an Apache Web Server, this system presents an AJAX-enabled, VRML representation. In this version, a mode is provided that enables multiple clients to simultaneously interact with the same scene, sharing the same representation. Through interaction with the VRML-based scene utilizing Parallelgraphics Cortona browser, users can re-arrange parts of a scene which in turn can be viewed by other users. Through the employment of SAJAX-based XHR requests, each client can send vector information to a data server via Python xmlrpclib library. This XHR request will update the data server providing a global representation of movable objects in the scene. Upon an update, the data sever in turn performs a client request to a third Python script that exists as a hidden frame. Then, through employment of the DOM hierarchy, it pushes an event down to the scene. Each client maintains its own event server which can in turn, respond to events, pushing them through the frameset hierarchy.

The process can be viewed in detailed steps (figure 2.). Before the scenes are downloaded, the data server is initialized independently (step 2.0). At the point of requesting a scene (step 1,2), two processes are generated from the initial HTTP GET request. The first process maintains the scene interaction as a Python script via SAJAX library and second exists as a hidden frame (3). After a client interacts with the 3D scene an independent XHR request is performed to the server passing necessary vector information of operators, containers and vehicle coordinates (4). At this point, the scene-based CGI will update the data server (5). The data server (now acting as a client) will call the event server(s) (6) which will update their corresponding client via the top pointer utilized in DOM hierarchy (7).

Figure three provides a higher level diagram of data paths with three clients in our architecture. Note that each client is identified by 'C' (for client) and 'E' (for individual event server) which each interact in fashion described by figure 2 with the shared scene and data servers (marked 'D').

6. CONCLUSION

An approach to the development of real-time multi-user architecture is presented. A prototype system is implemented demonstrating an interactive 3D simulation between multiple clients. This system

supports a fast development cycle through the leveraging of scripting technologies predicated on the web 2.0 design. Through the employment of Javascript and XML based technologies this represents a highly compatible web-enabled 3D solution for multi-user interaction. Through the support of a multiple user environment, we intend to model scenarios involving multiple workstations end-to-end. Through the provision of a multiple user environment, users can interact among each other working towards complete line balancing by performing adjustments to workstations within a shared interface. Through this technology we intend to leverage current gaming architectures and the AJAX-3D paradigm to present novel solutions of optimization to our manufacturing environment.

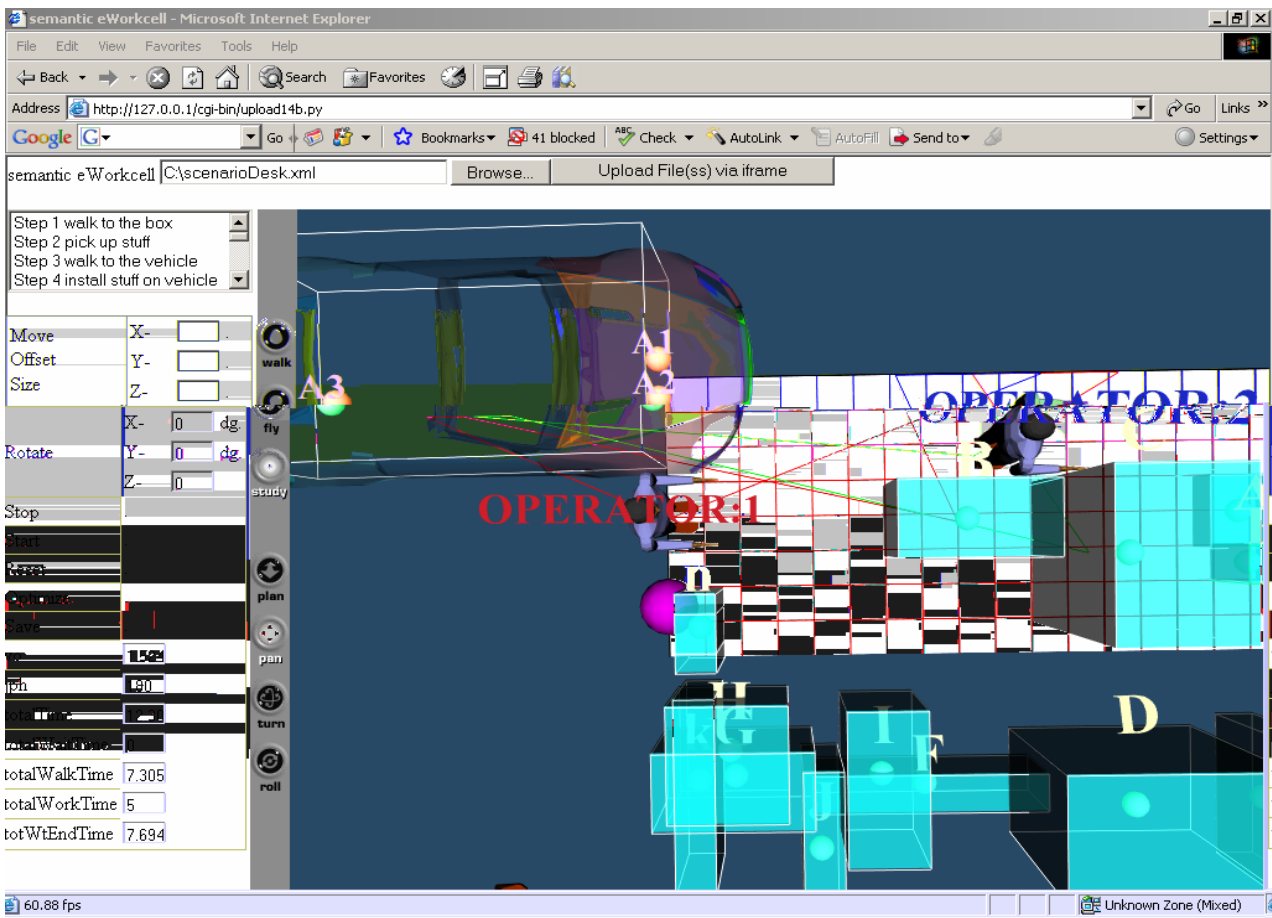


Figure 1 Semantic eWorkcell

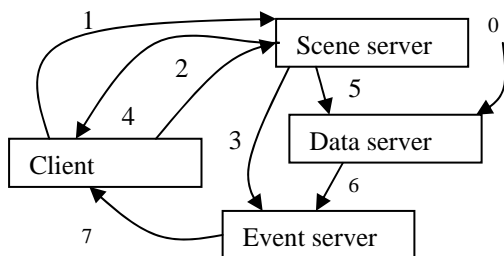


Figure 2 Data Path Representation

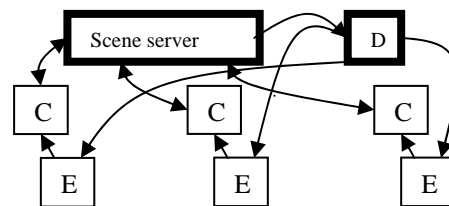


Figure 3. Multiple Client Implementation

7. ACKNOWLEDGEMENTS

Thanks to Rita Turkowski, Executive Director of Web3D Consortium and Tony Parisi, CEO of Media Machines for valuable input.

8. References

- [1] Bender, Michael, Klein, Ralf, Disch, Andreas, Ebert, Achim, A Functional Framework for Web-Based Information Visualization System, *IEEE Transactions on Visualization and Computer Systems*, Vol 6. No 1. Jan-March 2000
- [2] Astrachan, Owen L., Bernstein, David, English Andrew, Koh Benjamin, Development Issues for a networked, Object-Orientated Gaming Architecture (NOOGA) Teaching Tool, 32nd ASEE/IEEE Frontiers in Education Conference, November 6-9, Boston, MA 2002
- [3] Holmberg, Nathan, Wunsch, Burkhard, Tempero Ewan, A Framework for Interactive Web-Based Visualization, *Seventh Annual Australasian User Interface Conference AUIC2006*, Hobart, Australia
- [4] Marcus, A., Feng, L., Maletic, J.I., 3D Representations for Software Visualization, *Proceedings of the ACM Symposium on Software Visualization (SoftVis'03)*, San Diego, CA, June 11-13, 2003, pp. 27-36.
- [5] Oreilly, Tim, What is Web 2.0, <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>
- [6] Crane, David, Pascarella, Eric, James, Darren, *Ajax in Action*, Manning pb., October 2005
- [7] Google Maps, <http://maps.google.com>
- [8] Google Suggest, <http://www.google.com/webhp?complete=1&hl=en>
- [9] Parisi, Tony, <http://www.ajax3d.org/whitepaper>.
- [10] AJAX3D, <http://www.ajax3d.org>
- [11] Rosenthal, Larry, <http://www.ajax3d.org/content/strikefighter/larry.html>
- [12] Media Machines Gallery <http://www.mediamachines.com/browse.php>
- [13] Cerami, Eric, "Web Service Essentials", Oreilly Pub. Feb 2002
- [14] Gross, Christian, "Ajax and REST Recipes", APress pub., Springer-Verlag, 2006
- [15] Geroimenko, Vladimir, Chen, Chaomei Ed, Visualizing the Semantic Web, *Springer Pub.* 2005
- [16] Ostrowski, David, *A Web-Based Interactive 3D Visualization Architecture*, IEEE Visualization - Poster, Baltimore, MD, November 2006
- [17] Ostrowski, David, A., Semantic eWorkcell, *Ford Technical Journal*, December 2006
- [18] Brenna, Lars, Johansen, Dag, "Configuring Push-Based Web Services", *Proceedings of the International Conference on Next Generation Web Service Practices, NWeSP '05*
- [19] Musciano, Chuck, Kennedy, Bill, "HTML The Definitive Guide", Oreilly, 1998
- [20] Stein, Lincoln, "Official Guide To Programming With CGI-PM", Wiley, 2002
- [21] Nesbitt, Bryce, The Slow Load Tehnique/Revese AJAX – Simulating Server Push in a Standard Web Browser, http://www.obviously.com/tech_tips/slow_load_technique
- [22] Jain, Sanjay, McLean, Charles R., Integrated Simulation and Gaming Architecture for Incident Management Training, *Proceedings of the 2005 Winter Simulation Conference*, 2005
- [23] Davison, Andrew, *Killer Game Programming in Java*, Oreilly, 2005
- [24] Jaworski, James, *Matering Javascript*, Sybex pub., 1999
- [25] Harbour, Jonathan S., *Visual Basic Game Programming with Direct X*, Premier Press 2002
- [26] Riley, Sean, *Game Programming with Python*, Charles River Media 2004
- [27] Gutschmidt, Tom, *Game Programming with Python, Lua and Ruby*, Premier Press, 2004
- [28] Tate, Bruce A., Gehtland, Justin, Better, Faster, Lighter Java, Oreilly, 2004
- [29] Richter, Stephan, *A Zope 3 Developers Handbook*, Pearson education, March 2005
- [30] Raymond, Scott, *Ajax on Rails*, Oreilly pub. 2006
- [31] Zhang, Chenglin, *Browser-to-Browser, dom-based, peer-to-peer communication with Delta Synchronization*, US Patent Office # 6865599, Mach 8, 2005
- [32] www.cosmethods.com
- [33] www.lightstreamer.com
- [34] <http://weboggle.shackworks.com/>
- [35] <http://www.ajax3d.org/content/strikefighter/index.html>
- [36] <http://www.macromedia.com>
- [37] <http://www.sunw.com>
- [38] http://www.osflash.org/open_source_flash_projects
- [39] Gusikhin, Oleg, Klampfl, Erica, Rossi Giuseppe, Aguwa, Celetine, Coffman, Gene, Martinak, *Interactive Planning and Optimization of Workcell Layouts in an Automotive Assembly Line Environment*, *Ford Technical Journal*, Vol 5, No. 2, March 15 2002
- [40] Optimization of Workcell Layouts in a Mixed-Model Assembly Line Environment. E. Klampfl, O. Gusikhin and G. Rossi, *International Journal of Flexible Manufacturing Systems* 17, 277-299 (2006).
- [41] Ostrowski, David A., A lightweight Framework for Web-Based, 3D, Information Visualization, Intl. Conf. on Enterprise Inf. Systems and Technologies, EIWSST 2007, (to be presented June 2007)

PANORAMA: A Rich VRML Application Platform For Online Gaming

Si Yin, Anton Eliens
Department of Computer Science
Vrije Universiteit Amsterdam

De Boelelaan 1081/1083
1081 HV Amsterdam the Netherlands

E-mail: {syin, eliens}@few.vu.nl

ABSTRACT

Traditional VRML games are normally designed in single user mode, lacking in communications with other sources. Though there already exists solutions like communication servers to provide such service, such hosts are regularly expensive and not general enough for developers to expand. In this paper, we present PANORAMA, a platform which makes use of the AJAX technique, to demonstrate a new design model which can fully support multi-users and incorporate with other network applications. Also in PANORAMA we have experienced the GWT framework to build a friendly user interface as a communication layer between normal web context and 3D VRML world.

Keywords: VRML, AJAX, X3D, Web Games, GWT.

1 INTRODUCTION

The emerging of VRML dates back to 1997 when VRML2 was released and used by some personal homepages and sites such as "CyberTown" which offered 3D chat rooms. With the booming power of nowadays GPU, VRML games are becoming more and more powerful and realistic, especially by supporting the hardware Shader programs [Bitmanagement, 2007]. Another standard X3D, the successor of VRML format, features extensions like NURBs, Humanoid Animation and using an XML syntax, which makes third party applications easily incorporate with. So why can't VRML/X3D application and games be popularized or any 3D gadgets came out at current stage? We conclude that there are mainly two reasons as follows.

- The first reason is the *compatibility conflict* between different VRML viewers}. According to the list from the Web 3D consortium [Web3d, 2006], there are more than ten VRML players exists and about four of them are widely used. The truth is that most of the players can't view others' VRML world well, especially when involving special non-standard features. We think such dilemma is more than a technical problem and will not be discussed in the rest of the paper.
- The other reason is the *lack of communication interface*. Current VRML/X3D standards [VRML,

2001] don't have the facility to support multi-users or network connections except simple URL indication. One solution, provided by the Blaxxun Company in their communication server product [Blaxxun, 2005], is expensive and not general enough to expand. The other solution named Distributed Logic Programming (DLP) [6], which supports sockets connections, focuses on intelligence research with less care on the performance. The internal Java support is also another option but it gives too much workload on the client.

In this paper, we propose PANORAMA as a project focusing on social awareness and interactions [Social, 2006], resulting an example of how to give a general communication interface to fully support multi-users and cooperation with other network applications.

In previous work, we have experienced several kinds of game designs using different techniques such as DLP and Shader programs. In this work, we prove that it's a great opportunity to bring real web online games through the PANORAMA platform. The paper starts with an overview of the system design followed by the implementation description in Section 3. Section 4 explains the possible future improvements and we end with a summary in Section 5.

2 SYSTEM DESIGN

The original idea of PANORAMA is to explore different approaches to show self-reflections in certain community and social network. It should be intelligent, interactive and visually good looking to mediate non-work related information and enhance awareness within the work environment. We build it on the web due to the ubiquity of the browser as a client, also called thin client architecture which provides better maintainability and developing environment.

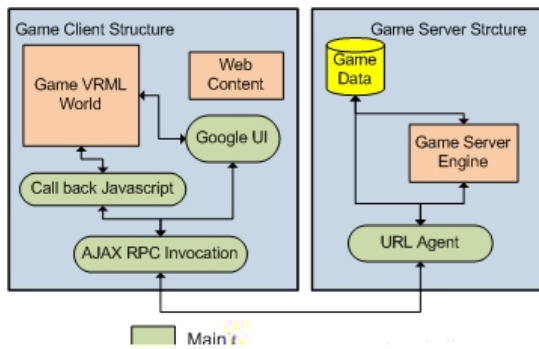


Figure 1: A bird's eye view of PANORAMA.

The structure of PANORAMA is more like a normal three-tier web application which contains game server, the central database and the user client shown in Figure 1. We can see that VRML game is only part of the whole system and the main flow process is controlled by the AJAX script used. When a game event occurs like getting the current rank list or updating some data on the server, the VRML world will trigger the callback JavaScript which then invokes HTTP request through a unique URL indication. When the game server receives such request, it extracts the information from the HTTP header and processes it in the server engine.

There may be several problems in such design like *security trust* (how to protect user data from modifying) and *synchronization issue* (when several clients update and read from the database). We discuss respective solutions in Section 4.



Figure 2: My naughty tiger game in PANORAMA

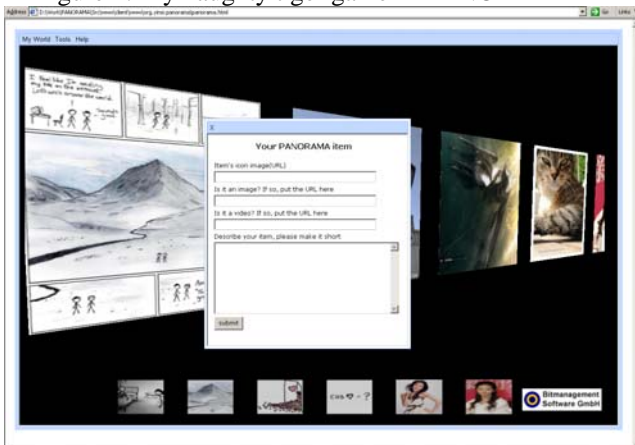


Figure 3: Upload a new photo into your gallery

We have built one game and one application within such platform. The game shown in Figure 2 is named 'My naughty tigers' that player places boxes to drive the jumping tiger into the smoke. Multi-players can rank their scores after each winning. The other application is called 'My Photo Gallery' in Figure 3 which gives a gallery walking view to enjoy pictures and also can upload new ones without refreshing the page.

3 SYSTEM IMPLEMENTATION

Before we dump into the specific implementation of our PANORAMA platform, it's better to introduce the developing platform at first. We have tested the system on both Firefox and Internet Explorer browser with the latest Bitmanagement Contact 7.0 VRML plug-in installed. The server agent is built using PHP programming with MySQL as the database. There are mainly four new components in PANORAMA compared with regular VRML applications and each of them is explained as below.

3.1 GOOGLE WEB USER INTERFACE

Google Web User Interface is a HTML interface programmed using Java and generated by Google Web Toolkit (GWT) [GWT, 2007]. It acts as glue between VRML games and user operations. Functions like 'Login', 'Upload', and 'Chat' can all be implemented by using such interface like in Figure 3. For internal game usage, it can also set up the 'Setting' window which complements the weakness on the user interface in VRML. The communication between VRML and HTML page is through the VRML Script EAI and in List 1 we demonstrate how it works in GWT.

List 1: Set event value into VRML node in GWT

```
public static native void setEventIn(node, field, value) /*-\ {
    document.Contact3D.setNodeEventIn(node, field, value);
-*/;
```

3.2 CALLBACK FUNCTION IN JAVASCRIPT

Callback function is usually invoked when VRML game needs to communicate with the outside world. In List 2, we give a sample code to describe how this works both in VRML and HTML.

List 2: Callback function load by VRML and HTML

```
DEF Script Script {
    field MFString url ["OnEvent(par)"]
    url "javascript:
        function value_changed(v, t) {
            # load the current URL in default
            Browser.loadURL(url, v);
        }"
}
and the HTML definition
<script language=Javascript>
    function onEvent(par) {
```

```
    alert(par);
  }
</script>
```

In the above source code, the URL loaded by VRML is only the function name embedded in the current HTML file. The parameter passed is a **MFString** type value with each sub-string matching the list of the function definition. When event occurs, the JavaScript function will be invoked and makes the request.

3.3 AJAX REQUEST AND URL AGENT

Making an AJAX request is widely used and standardized in nowadays web sites. The procedure works like that the script first checks the version of the client browser and then instantiates a **XMLHTTP** object and finally makes the connection to the server agent. A **setTimeout()** callback function can be set up since games always need to update the scene frequently. More details about the exact code and tutorials can be found on the Mozilla developing center [Mozilla, 2007]. At the server side when the agent receives a request, it should first identify which HTTP method the client uses (**POST** or **GET**) and then extract the variable values from the header file and process the request. This can be easily handled by reading the **_GET** or **_POST** variables in PHP.

4 FUTURE IMPROVEMENT

The initial result shows that the current PANORAMA is promising but still there is a lot of work left. We consider adding the following features in the next release version.

1. *Authorization mechanism.* The server should maintain authorization function otherwise one can easily change other's data through the HTTP header request. Since it's nothing to do with VRML, normal web solution is sustainable.
2. *Synchronization issues.* Upon the time the paper is written, we haven't experienced large scale multi-user scenarios on PANORAMA and such problem didn't occur at this moment. But we think it's a web developing problem rather than a complex game engine design and PHP can easily handle that.
3. Modification in GWT for better VRML support. Again, we discovery that the GWT is a well developed framework which greatly saves time on debugging web applications and building UI. Since it uses an open source license, it's possible to build a derived package that specially serves for the VRML game development.

5 COMPARISON WITH OTHER PLATFORMS

Inside the web game market, there are already several other platforms exists (e.g. Flash 3D, Quick Time VR, etc.). Those products are dominated by big companies and have been developed for years. But we believe

PANORAMA as a VRML platform can still compete with others with the following advantages:

1. *Comparing with other commercial platforms,* PANORAMA can be totally built upon open source software. For example, the server can use Apache while the VRML client has Flux [Flux, 2007] and the GWT library is already Apache licensed. Such environment gives developers more spaces to investigate and expand.
2. *The out performance of VRML.* By fully supporting the DirectX9, VRML players gain the best performance in 3D graphics rendering, resulting in high quality game images. The VRML/X3D are also open standards file format with a rich set of components featuring visualization and entertainment.
3. *Royalty free Development.* Commercial platforms need you to buy the authoring tools to create content while VRML applications can be totally developed by text editors. Some respect authoring tool like Flux Studio and VmlPad are free to download and use.

The disadvantage of PANORAMA can be concluded into unifying issues. Apart from the weak compatibility between different players, the support of browser is also a problem. Almost all the VRML players need user to install the plug-in manually, without any notification by the browser. But we think this dilemma can be gradually conquered when the application is getting widely used.

6 CONCLUSIONS

Traditional VRML games are mostly played in single user mode lacking communication with others. With the popularity of AJAX technique, it's possible to build multi-users games in VRML through the EAI interface. In this paper, we are proposing PANORAMA, an early bird work which uses AJAX for communication and GWT for building UI, resulting in a hybrid platform that is general enough to adapt to games and other applications. We evaluate small games and discuss several pitfalls which can be fixed in the next release. Also we recognize that building a framework based on the GWT is meaningful and surely can boost the popularity of web 3D games.

7 ACKNOWLEDGMENTS

The PANORAMA project is funded as the multimedia project in Vrije Universiteit Amsterdam. We also would like to thanks Marek for his suggestions on the visual design of the application.

References

[Blaxxun, 2005] Blaxxun Product. Blaxxun Communication Server.
<http://www.blaxxuntechnologies.com>.

[Bitmanagement, 2007] Bitmanagement developer website. [http:// www.bitmanagement.com/developer/](http://www.bitmanagement.com/developer/).
[GWT, 2007] Google Web Toolkit. <http://code.google.com/webtoolkit>.
[Social, 2006] PANORAMA system: Being social. Vrije Universiteit Amsterdam. <http://www.few.vu.nl/~dvyas/resources.html>.
[DLP, 2005] Distributed Logic Programming. Vrije Universiteit Amsterdam. <http://dlp.cs.vu.nl>.
[Mozilla, 2007] Mozilla Developer Center. AJAX: Getting Started. <http://developer.mozilla.org/en/docs>.
[Web3d, 2006] Web 3D Consortium. VRML viewer, browsers and plug-ins. <http://www.web3d.org/x3d/vrml/tools>.
[VRML, 2001] ISO/ ISO/IEC14772-1:1997andISO/IEC14772-2:2004. Virtual Reality Modeling Language (VRML).
[Flux., 2007] Media Machines. Flux Studio. <http://www.mediaamachines.com>.

Road Rider: an interactive 3D game to promote road safety

E. Baldi, F. Bellotti, R. Berta, M. Pellegrino, L. Primavera, A. De Gloria

DIBE – Department of Electronics and Biophysical Engineering
University of Genoa

Via Opera Pia 11/a
16145 Genova Italy

Tel: +39-010-3532795
Fax: +39-010-3532795

E-mail: {baldi.enrico, franz, berta, pellegrino, primavera}@elios.unige.it

ABSTRACT

Serious Games can join their capability to engage and entertain with a potential to train users in a target application field. In this context, we have developed Road Rider, a 3D videogame aimed at promoting road safety among middle and high school students. The game looks similar to commercial videogames, but all the important game situations and reward/penalty score mechanisms are tied to road safety. This implies that users should learn about road safety seamlessly, as players learn “tricks” in usual videogames. The project has actively involved end-users to focus on usability issues and steer the design process according to the user needs. In this paper we present quantitative data analysis and understanding gained from an early user test, with 56 students coming from three different schools. Results show a good students’ game acceptance, in terms of usability, ease of use, enjoyability, playability and graphic quality.

1. INTRODUCTION

Research work argues that computer games are an engaging medium for learning since they can stimulate cognitive processes as reading explicit and implicit information, deductive and inductive reasoning, problem solving, and making inferences from information displayed across a number of screens [Pillay, 1999]. Examining the brain’s electrical activity, [Kahana, 1999] have shown from a neuroscientific point of view that computer games engage spatial learning. In this context, a specific game typology, the Serious Games [Rieber, 1996], was developed in order to exploit the potential of games in supporting acquisition of knowledge and skills. Serious Games originally dedicated to the training of critical workforce, such as the military [Bonk and Dennen, 2005], are currently used to develop specific skills that a user could gain only through direct experience or a computer-based simulation. A serious game shows the same likelihood level of a virtual reality simulation. Furthermore it provides the added-value of a game framework, which emotively engages users, and proposes concrete problem-solving cases in a

consequential, compelling narrative structure (not as isolated steps of a tutorial). This is particularly useful to test the ability of a user in long-lasting, highly-realistic situations.

In a typical Serious Game, the pedagogical objectives are not directly apparent. They have to remain in the background of the story, leaving entertainment in prominence. According to constructivist approach [Von Glaserfeld, 1990], learning depends on the active engagement of the subject that learns and on his ability to construct knowledge and understanding on the basis of interaction with the environment. Thus, Serious Games exploit all the appeal of a pleasant videogame (they are not to be perceived by players as “educational” games), while also providing educational value. A major difficulty of Serious Game design consists in effectively embedding the learning aspects as seamless mechanisms (not as add-ons) in a meaningful and compelling whole. This is inline with the constructivistic principles that tend to orient learners towards meaning-making over simple content-collecting. Users learn by dealing with contextualized problem sets and situations.

Road safety is a promising field of application for serious games. The high number of accidents and fatalities that happen daily all over Europe [eSafety, 2007] are often due to the lack of experience that drivers and pedestrian have in dealing with dramatic events (e.g. a car ahead which suddenly brakes, a pedestrian who suddenly crosses the road, a very slippery road) and their lack of respect of road rules. However, awareness of the importance of a correct behavior is still very limited among the citizens. For instance, as reported in detail by table 1, in Italy road incidents due to subjective causes (incorrect road behavior) are 95.5% of total incidents [ISTAT, 2003].

Incident cause	Percentage
Distraction	15.9%
Lack of “safety distance”	15.0%
Disruption of car speed limits	12.1%
Irregular steer	6.8%

Table 1 Main irregular behavior of Italian drivers [ISTAT, 2003]

This paper presents the first developments of the Road Rider Project. Road Rider is a game, mainly targeted to middle and high-school students, which aims at providing information about road safety.. First, we discuss the game philosophy and plot. Then we introduce the technological framework. Finally, we present early results from an end-user test.

2. THE ROAD RIDER GAME

The main purpose of Road Rider Game Project is to promote awareness and an effective use of road rules among the general public and the youngsters in particular.

Road Rider is an interactive first person 3D game, where the player controls a virtual character whose task consists in reaching the site of a rock concert. During her/his trip, the user walks around a city (that is a 3D reconstruction of a portion of the Genoa city center - figure 1). The game plot consists of a number of “missions”. Missions involve finding a car, getting money to buy a ticket for the concert, driving the car to visit friends who live in different cities to another, and finally reaching the destination site. Every mission features an increasing level of difficulty. In order to enhance the player’s engagement, the game plot is dynamic. The missions are not be predefined and do not follow a fixed sequence. Conditions and events change in every mission.

These situations and conditions are similar to those of state of the art commercial videogames. This makes this activity be perceived pleasantly by the user as a game. However, there are two main aspects that differentiate Road Rider: the road settings and the score mechanisms. All the important game situations are tied to road safety (road-signs, vehicles, roads, cross-roads, pedestrians, etc.). And this is true also for the score rules. Score is a fundamental element of the game, since it provides the main motivation for a user to improve her/his performance. So, the criteria according to which points are assigned are very important because they define what layers’ operations, actions and behaviours are positive (i.e. rewarded by the system by increasing the score) and what are not relevant or even negative (i.e. causing score penalties). In Road Rider, the system penalizes hazardous behaviors (e.g. crossing a road without a crosswalk, failure to give priorities to pedestrians in a zebra-crossing, exceeding speed limits, driving with switched-off lights, violating one-ways, parking off-place etc) and rewards safe road-behaviors (e.g. successful completion of a mission meeting all the legal and safety constraints, the ability to avoid or cope with dangerous situations). The score has a fundamental implicit impact on the player’s behaviour, since it spurs her/him to automatically perform (and this requires implicit learning) those operations that allow her/him to compete better. Putting it in gamers’ words, players learn the game’s tricks in order to successfully advance in the game and in the scoreboard. A major idea behind Road Rider is that all the tricks are associated with road safety issues. This implies that users should learn about

road safety as players learn “tricks” in usual videogames.

Affection to a game is a key-factor to accomplish its educational objectives. In usual games, players learn tricks only if they play the game several times, trying to overcome difficulties in order to reach the end of the game. In the same way, Road Rider’s tricks can be learned only if users play the game several times. For this reason, as anticipated, particular attention has been paid in the definition of the missions, that should be perceived as engaging, ever new activities, that can be performed several times without annoying nor boring the player.



Figure 1 Snapshot from Road Rider, player is a pedestrian involved in a “mission” in the city center.



Figure 2 Snapshot from Road Rider, player is a driver involved in exurban “mission”.

The user can choose and customize her/his game avatar (i.e. the character that represents her/him in the game) from a pre-defined set. There are different characters (e.g. a musician, a voyager, etc), for all of which a short description of the personality is provided. Different kinds of personalities are represented, in order to give players the possibility of choosing the character they feel most similar or most interesting to them. The

relation between the player and the character is seen as an important factor for the educational experience [Aylett et al., 2006]. In general, the creation of an empathic relation between the player and the character could help a player achieve the engagement and the suspension of disbelief that can effectively draw a player into a game's world. Moreover, in the educational field the relation player-character may symbolize the typical teacher-learner relation, thus making the player feel responsible for the behaviour (and thus the learning) of her/his education.

Finally, in the future planned implementations of Road Rider, the engine will record the user activity to maintain a user profile, that will be used to adapt each game's session to the current user's abilities in order to optimize the user's training effects by challenging her/him more frequently with situations that feature issues, items and cases that are more difficult for her/him to learn. For instance, certain driving scenarios will be repeated in subsequent games' sessions, so that the user may become familiar with situations that are more difficult for her to manage (e.g. encountering a crossroads).

3. THE GAME ENGINE

The game has been realized with the Torque Game Engine, a high quality/price ratio free source game engine [Garage Games, 2007]. This tool provides a valid support to realize:

- a realistic 3D visualization, which is quite complex to achieve in a networked environment;
- an effective narrative, which is important in order to engage the user in interesting and compelling situations;
- a realistic simulation, which is important to increase the likelihood that the user may transfer in the real-life techniques and skills learned through simulation.

The last feature is particularly difficult to achieve, since professional car simulators, for example, involve very complex hardware equipments to realize real-life-like situations. While here we want to use a tool which works with low-cost standard hardware, in order to reach a wide audience. This requires also the study of different interaction modalities corresponding to the various hardware devices available at the user's site. In the current release, a user can play the game through mouse and keyboard, but also through low cost steering-wheel and pedals mock-ups, which are available on the market for game consoles.

4. EARLY FIELD EVALUATION

In proper educational applications assessment and testing are crucial to determine if the students have understood the material and are able to appropriately recall and use the learned material. Since serious games, differently from usual games, clearly aim at providing specific knowledge and skills, it is necessary to carefully evaluate their impact on the target audience.

Real evaluation of a serious game is difficult and costly. Evaluation requires analysis of a large distribution of real users in a real context of use for long time periods. In order to evaluate continuous interaction with the game, test-people should use a fully operational game for a long period of time with evaluation of learning at the end of the trial and a follow up after a proper period of time (typically 6 months).

Widely accepted, detailed procedures for test and evaluation of Serious Games are not yet established in the scientific community [Squire, 2003]. Nevertheless, it is important to perform early tests to verify end-user acceptance and overall usefulness of the game, and to receive feedback to inform future design.

In order to tackle this issue, we have adopted a user-centric design methodology [Carroll, 1997], involving end-user students at all stages of the game development (participatory design [Beck, 1993]): from the user requirements analysis to the final evaluation of the game. This methodology, which helps us focus on end-users and consequently steer the design process, supports the evaluation of early concepts, prototypes and final releases. Not only have we asked users opinions on the game experience, but have actively involved them in the design and decision-making processes. The work was articulated in a cycle of design-implementation-evaluation of prototypes to adjust the design in order to meet the expectations coming from the end-users and the pedagogical purposes.

This section provides an account of the results coming from user test sessions that we carried out in collaboration with three different schools (middle and high schools) at an early stage of game development.

We have organized this first workshop with young people in order to collect feedbacks from students and evaluate overall user acceptance.

The workshop consisted in an introduction to the Road Rider Game Project (without mentioning the educational objectives) and a 10 to 15 minute game trial, followed by a think-aloud session. Then, each user was asked to assess the game through Likert scales on 5 dimensions: perceived usability, ease of use, enjoyability, "playability" and graphic quality. The total number of respondents was 56 students (all the test students accepted to respond), 17 from middle schools and 39 from high school (19 technical high school, and 20 scientific high school). In table 1 we summarize the results. The respondents assessed the tool as usable, easy to use and enjoyable. However, test results highlight some differences between the groups, in particular concerning enjoyability and graphic quality. These two features are evaluated better from middle school students than by high school students. In the think aloud session we have found out that our Road Rider design better meets the expectations of younger students, because they are less demanding in terms of graphics quality. Moreover older students consider the game trials as not sufficiently engaging. They suggest that trials should be more tricky, like commercial state of art games. We will address these criticisms to improve the future releases, in particular by

implementing more difficult trials, with improved graphics quality. The game will also adapt the difficulty level according to the player abilities.

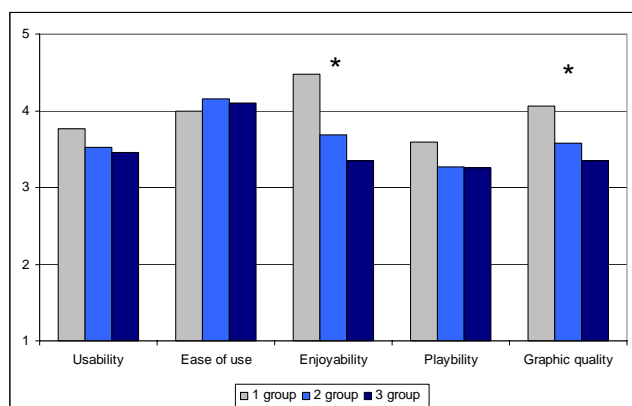


Table 1 Average results (considering 3 groups of users: group1 = middle school students, group2 = technical high school students, group3=scientific high schools students), from students' evaluation on 5 dimensions assessed with Likert scales. Asterisks highlight a significant difference in evaluation of enjoyability and graphic quality among groups according to Anova test.

5. CONCLUSIONS AND FUTURE WORKS

The main idea behind the Road Rider project is that players may learn road safety topics as they successfully and pleasantly learn “tricks” in usual videogames. The assumption and long-term expectation of the project is that safer road behaviors, learned as tricks in order to better perform in a videogame’s virtual world reconstruction, should be put in practice from the users also in their real life activities. But this is still an open issue, that has to be verified in depth in extensive end-user tests. Do players use in their real life the “tricks” they have learned in the virtual environment of the game? The answer requires an appropriate verification methodology and a large deployment of Road Rider on schools and a continuous monitoring over time about users behaviour. In this view, we have defined, in collaboration with Regione Liguria, a deployment plan to make the game available to several Liguria schools. In the next year we will have a large community of players and the possibility to realize a very wide user test campaign on this topic.

At present, we have realized early field tests to assess usability to steer future design. The results show a good users' acceptance, in terms of usability, enjoyability, playability, ease of use and graphics quality. In the future we will also explore collaborative and competitive aspects that will be allowed by the multiplayer version of the game.

6. ACKNOWLEDGMENTS

The Road Rider Project has been partially funded by Regione Liguria, “Assessorato alle Infrastrutture, Trasporti, Porti, Logistica”.

References

- [Pillay, 1999] H. Pillay, J. Brownlee, and L. Wilss, “Cognition and Recreational Computer Games: Implications for Educational Technology”, *Journal of Research on Computing in Education*, Vol. 32 No. 1, pp. 203-215, 1999.
- [Rieber, 1996] L. P. Rieber, “Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games“ *Educational Technology Research & Development*, 1996, Vol. 44, No. 2, pp. 43-58.
- [Von Glaserfeld, 1990] E. von Glasersfeld. “Constructivist Views on the Teaching and Learning of Mathematics”. *Journal for Research in Mathematics Education. Monograph*, Vol. 4, pp. 19-29+195-210, 1990.
- [ISTAT, 2003] ISTAT Web Site, http://www.istat.it/dati/catalogo/20051013_00/
- [Kahana, 1999] Kahana, M., Sekuler, R., Caplan, J., Kirschen, M. & Madsen J., “Human theta oscillations exhibit task dependence during virtual maze navigation”. *Nature*, 399, 781-784, 1999
- [Bonk and Dennen, 2005] CJ Bonk, VP Dennen, *Massive Multiplayer Online Gaming: A Research Framework for Military Training and Education*, Indiana University at Bloomington, 2005
- [Zyda, 2005] Zyda Michael, “From Visual Simulation to Virtual Reality to Games”, *IEEE Computer*, September 2005, pp. 25-32
- [Shawn, 2003] Green Shawn C., Daphne Bavelier, “Action video game modifies visual selective attending”, *Nature*, Vol. 423, 29 May 2003
- [Prensky, 2001] Prensky M., “*Digital Game-Based Learning*”. McGraw-Hill, New York, 2001
- [Aylett et al., 2006] Ruth Aylett, Sandy Louchart, Joao Dias, Ana Paiva, Marco Vala, Sarah Woods, Lynne Hall, "Unscripted Narrative for Affectively Driven Characters," *IEEE Computer Graphics and Applications*, vol. 26, no. 3, pp. 42-52, May/Jun, 2006.
- [eSafety, 2007] eSafety Web Site, http://ec.europa.eu/information_society/activities/esafety/index_en.htm
- [Garage Games, 2007] Garage Games Web Site, www.garagegames.com
- [Squire, 2003] K Squire “Video games in education” *International Journal of Intelligent Simulations and Gaming*, 2003