Online Game Quality Assessment

C00164522 Luca Venturelli

1 Introduction

Much work has been done at examining, at many different layers , features and metrics to quantify and compensate as an effort to improve player reaction in the online environment. The focus on distributed interactive applications and the adaptation of the application at both client and server has changed the way games are designed and developed for online purposes. In order to maintain a large number of online players it is becoming essential for the game service providers to estimate game player perception and performance for their games. On the other hand, advanced information about game playing conditions would allow game players to select different online games, different networks and ultimately different tariffs. Traditionally the end-user perception for online games has been measured using subjective game quality assessment only. Although they claim to estimate a user perception, the existing models are based on network components which are limited in fully predicting the user satisfaction. Research carried out by US Entertainment Software Association (ESA) revealed that the percent of online game players is as high as 62% from total game players [1]. The statistics indicate that the average game player has been playing games for 12 years. Adult gamers have been playing for an average of 14 years; males average 16 years of game play, females average 12 years. The average game player is 30 years old. the 15% of most frequent game players pay to play online games. These statistics demonstrate a high level of online game players and a relevant amount of people that pay to play online games.

1.1 Overview

As follows an overview of the chapters of this Manual:

- In section 2 we will analyse the most important metrics proposed showing the problems and the lacks they have.
- In section 3 we will then analyse some of the most famous open source game engines and the game Doom3. We'll explain the reason why we choosed Ogre3D and Torque 3D for the implementation of our metric and Doom3 for the test and tuning of the metric's parameters.
- In section 4 we will propose a new metric that takes advantage of the existing metrics and add some new parameter to perceptually estimate the game quality.
- In section 5 we will analyse the results obtained.

2 Existing Metrics

The network layer is one of the most important aspect to consider while approaching a quality metric for online games. In terms of network awareness user perception in terms of player tolerance is crucial. In [2] the author reveals that there are two possible approaches for discovering player tolerance to network disruption. The first is to build a controlled lab environment in which to test small groups of players under selected conditions and secondly to monitor player behaviour on public servers over thousands of games. As follow an analysis of the metrics proposed to evaluate the game quality and the aspects they cover:

2.1 Quake3 G-model

Quake3 G-model is a model proposed by Ubicom [3] that introduces a new benchmark, OPScore, or the Online Playability Score, to describe the effects of network impairments on the playability of online games. The authors focus their attention on latency, the average amount of time necessary to transmit information about a players actions, and jitter, the variance of latency. This technique uses measurements of traffic in a realistic home network environment to forecast the playability of online games. The model defines an impairment factor R given by:

$$R = (WL * L + WJ * J)(1 + E) \tag{1}$$

where WL is set to 1 for the test, K is the Average frags per minute, WL is the Latency weighting factor, equal to 1 by definition, E is the Packet loss, as a percentage of bytes lost and R is the Impairment factor in ms.

2.2 Quake IV G-Model

A successive approach proposed by Wattimena [4] uses a similar model named the Quake IV G-Model to predict the perceived quality of a First-Person Shooter. The authors conducted a number of subjective experiments to quantify the impact of network parameters on the perceived quality of a FPS game. The model proposed enables to predict a gamers quality rating based on measured ping and jitter values, and it shows a very high correlation with the subjective data. The final score is given by the following mapping function:

$$MOS\_model = -0.00000587X3 + 0.00139X2 - 0.114X + 4.37 \quad (2)$$

where: X - the network impairment is defined as:

$$X = 0.104 * pingAverage + jitterAverage. \quad (3)$$

WJ that is the Jitter weighting factor is calculated with the following formula

$$WJ = (\Delta K/\Delta J)/(\Delta K/\Delta L) \quad (4)$$

The experiments done demonstrate that ping and jitter have a significant negative effect on both the subjective (MOS) and objective gaming quality (kills), while packet loss goes unnoticed for values up to 40%. Especially the introduction of jitter in the network has a large negative effect on the perceived quality of the players. The model has been developed for the game Quake IV and was not tested on other games and platforms.

2.3 QoE

A recent paper [5] presents a new metric to measure the quality of experience (QoE). In this paper is presented an analysis of the causal relationship among network delay, system inconsistency and QoE. The QoE is divided into responsiveness, precision and fairness. They create a metric to qualify the view inconsistency. A function is built to map from the objective system

inconsistency to the subjective QoE property score. The input of this mapping function considers only the relative inconsistency between views that takes into account any built in time compensation algorithms, because the same network delay can produce different inconsistencies in different games and the same absolute inconsistency value can lead to different impacts in different scenarios.

2.4 Conclusion

The approaches described above have at least one of the following weaknesses:

- Although they claim to estimate a user perception, the parameters used are based on network components which are limited in fully predicting the user satisfaction.
- Implementing limited subjective testing for a short period of time with a small user pool
- While a high level of correlation between the subjective and the proposed models is shown, this could be justified only for one defined game in restricted testing conditions.
- None of the model presented presents a parameter to measure the game client and I/O devices except from the MOS model.

These are the reason why we propose a new metric that takes

3 Game Platforms (Engine)

For this project we decided to use an open source game engine to implement and test our metric.

- The principals factors that we used to evaluate the games engine are:
- The compatibility with the operating systems.
- The availability of documentation
- The programming language used
- How the networking module is integrated in the game engine.

- The availability of the engines source code
- Games using the platform
- The code licence

As follow an analysis of some of the most famous open source game engines and a review of the game Doom 3.

3.1 Panda 3D

This platform is available for Windows, Mac OS X and linux and provides also an sdk with full access to the source code. The programming languages available are C++ and Python. It supports the versions of python starting from the 2.4. All the core functionality are written in C++. Panda3D contains support for networked games. This includes both a low-level stream based API, and a higher level distributed object API. This system has a Manual not complete and with a lot of references to the forum for the parts still not covered. The main features of this library are Shader Generation, easy import of 3d models, a support of the audio libraries OpenAL, FMOD and Miles, different options for Physics simulation, a simple AI library, debugging tools [6]. This platform provides also powerful performance monitoring and optimization tools to identify cpu and gpu usage, the cpu time use is decomposed into more than 250 categories, and allows to implement new cpu-usage categories. This system is used to deliver also several commercial games Toontown Online and pirates of the Caribbean online. The last version of the software is 1.8.0 released on February 2012. Panda3d is released under the revised BSD license.

3.2 OGRE

This platform is available for Windows, Mac OS X and linux and provides an sdk with full access to the source code. The programming language used is C++. Ogre is not a game engine but it can be used to make games. For features like sound, networking, ai, collision, physics it has to be integrated with other libraries [7]. Ogre provides a very integration friendly API and let the developer choose the other libraries, if needed. In this way there are no inbuilt constraints. The principle of ogre is of collaboration and integration with other libraries, rather than assimilation of them, a standard tenet of

component-based design. There is a complete API reference for OGRE. An official Manual and The Ogre Wiki are available for the developers. Every feature that goes into OGRE is considered thoroughly and slotted into the overall design as elegantly as possible and is always fully documented, meaning that the features which are there always feel part of a cohesive whole. Ogre uses a flexible class hierarchy allowing the design of plugins. This system is used to develop several open source games games as Awakened,Rigs of Rods, StuntRally, Summoning Wars, Trinity Reign, Walabers Trampoline, WorldForge Ember 3D client. There are also some games with proprietary licenses as Ankh, Ankh: Heart of Osiris, Dramagame, Earth Eternal, Garshasp: The Monster Slayer, Garshasp: Secret of the Dragon, Jack Keane, Motorm4x, Next Life, Nimbus, Pacific Storm, Raindrop, Roblox, Snakeworlds, The Dead Linger (TDL), Torchlight, Torchlight II, Venetica, Zero Gear, Zombie Driver. Ogre 1.8, the last official version, has been released on may 2012. OGRE 3D is released under the MIT license.

3.3 Delta 3D

Delta3D is developed and tested on Windows XP using Microsoft Visual Studio and Linux using gcc. Delta3D is a widely used and well-supported open source game and simulation engine. The programming language used is C++. Delta 3d Provides the simulation core library to create networked games. There is a complete api Documentation available. Delta3D is a fully-featured game engine appropriate for a wide variety of uses including training, education, visualization, and entertainment. Its modular design integrates other well-known Open Source projects such as Open Scene Graph, Open Dynamics Engine, Character Animation Library, and OpenAL Delta3D integrates the modules together. Delta3D renders using OpenGL and imports a whole list of diverse file formats (.flt, .3ds, .obj, etc.) [8]. The last version of the software is 2.5.0 released on November 2010 under the LGPL license.

3.4 Irrlicht Engine

The Irrlicht Engine is a high performance open source and cross platform 3D engine for creating realtime 3D applications, developed using C++. The engine runs under linux windows and Mac OS X. Irrlicht provides a wiki still under construction and tutorials for the installation of the engine. The networking module is not built in but can be easily implemented using the

libraries suggested. Its main targets are to be easy to use, extremely fast, extensible and crash safe. [9] The engine is a quite flexible graphics engine, and it is possible to write lots of different applications with it. Irrlicht is not a game engine but it provides some libraries that can be easily integrated in the game engine. See [10] for a list of available games that use Irrlicht engine. The last version of the engine has been released under zlib license on November 2012.

3.5 Torque 3D

This platform is available for Windows XP SP3/Vista/Windows 7, it provides also an sdk with full access to the source code. The programming language used is C++. The networking module is integrated in the engine. It handles all the elements of a game that run in real time on the computer. The project is well documented. Torque 3D features PhysX support, modern shader features, and an advanced deferred lighting model. [11] Much of the game play logic, camera controls, and user interface can be written in TorqueScript. It is a powerful and flexible scripting language with syntax similar to C++. The key benefit of TorqueScript is that the developer do not need to know well a language like C++. Developer do not have to recompile the executable to see changes in the game. By modifying a script, save, and then run the developer can see the changes in the game from the Toolbox. Torque 3D allows developers to add functionality, increase optimization, and learn how everything works. Torque 3D is open source since September 2012 and has been released under the MIT license. A full list of games that use Torque 3D is available at [12].

3.6 Doom 3

Doom 3 is a first person shooting game developed by Id Software. The source code of this game has been released on 2011 and available at [13]. Doom 3 uses id-Tech 4 as game engine. The code is written in C++. It is well commented and the code is clean and easy to read thanks to the OOP encapsulation. [14]

3.7 Conclusion

The language we are going to use to implement the plug-in is C++. The plug-in will be integrated into Torque 3D engine and Ogre3D. Torque 3D was one of the best commercial game engine platform. A lot of games have been developed using this platform and the possibility to have full access to the code is very interesting. This engine has a very good documentation, It is compatible with all the most important operation systems and has been released under the MIT license that guarantees the reuse of the code in proprietary software. Ogre has been released with the same license. It is one of the most important open source platform for 3D development and, thanks to his modularity, is very easy to develop a plug-in for it. It doesnt provide a network framework but there are a lot of plug-ins that implement this function in Ogre. Its community is the largest and most active between the engine analysed. There is also the possibility to integrate the plug-in directly into a game. As follows the description of one of the most famous open source first person shooting game chosen for our tests.

4 What is new?

The metrics introduced in section 2 consider just the network factor without considering other perceptual parameters that can affect the quality of a game. The tests made to prove the metric proposed are based on one single game and have been tested for a very short period of time and with a small user pool. None of the metrics presented defines a parameter to measure the game client and the I/O devices used. To overcome this problem a new game Assessment Metric for online gamer, able to estimate the players overall perception of games quality assessment introducing the end-user experience/knowledge, is proposed here. The new proposed model overcomes the weaknesses of the existing models by:

- Combining different parameters such as the user-based experience/knowledge factors and distortions introduced by user equipment.
- Moving from a game-specific model to a wide range of existing online games (including console based games) implementing plug-ins for different game engines.

- Proposing objective tests and values to estimate the single parameters based on the empiric tests done
- Using a number of users for subjective testing in line with accepted test models used in telecommunications based scenarios recommended by ITU-T
- Expecting a level of correlation in excess of 95

The model proposed will take into consideration the following parameters:

1. The experience of the user will be based on the average of the players score in the game played.
2. Distortions introduced by game client equipment (graphic card, memory, processor) and I/O devices (mouse/joystick,screen, keyboard). Each component will be rated according to a ranking of the hardware used. The values obtained will be added together with a different weight (that will be estimated during the tests) and then it will be divided by 6(number of components considered).
3. Distortions introduced by the network
4. Distortions introduced by the game server

Using the above-mentioned parameters a Game Rating Factor (GRF) is proposed. The Game Rating Factor is inspired from an ITU-T recommended computational model (E-model)[15]. The model is used to assess the combined effects of variation in several parameters that may affect end-user perception of speech quality. The GRF can be described as follows: a maximum value that reflects the highest level of game quality will be reduced in proportion with the distortion caused by various impairment parameters.

$$GRF = GRFMAX - IGCD - IN - IGS + A \tag{5}$$

where GRFMAX is the maximum GameRatingFactor(90), IGCD is the impairment factor representing all impairments due to Game Client and I/O device, IN represents all the impairments due to network connection between the game server and game client, IGS is the impairment factor representing all impairments due to Game Server and A represents the end-user experience with online games. The GRF is then related with the Mean Opinion Score that is a value between 1 and 5 where 1 is bad and 5 is excellent.

5 Conclusions

The proposed model is based on the ITU-T E-model for telecommunication and can be used to measure the quality of the game player experience combining the network impairments with different objective measurable parameters. The model will be developed on the Doom 3 game and two plug-ins for Ogre and Torque will be developed. This model will be tested on a FPS game but we believe that the model will be general enough to be adapted easily for other gaming categories, and thanks to the plug-ins could be integrated in a wide range of games. The approach used to develop the metric is to merge the subjective experience with an objective evaluation of the different parameters adapting the weight of the parameters involved in the formula. In order to accurately predict and validate the effects of network impairments, other influences on a user's subjective experience will need to be isolated and studied. Going forward, this will be broken into four major tasks:

1. Isolation and subjective testing of external factors and their influence.
2. Subjective testing of isolated network impairments.
3. Measuring the effect of combined network impairments. Currently we have assumed that (like the E-model) combined network impairments will have an additive effect in reducing MOS. However this will need to be validated. Also consideration will be given to the non-additive effect of these parameters on user perception.
4. Games employ a number of different compensation techniques to mask network impairments from the user.

The completion of these four tasks will allow us to prove our model. Network impairments can then be passively or actively measured on any given network and the likely subjective effect on various games calculated.

References

[1] "2012 Essential Facts about the Computer and Video Game Industry", Entertainment Software Association, EF, May 2012

[2] G. Armitage, L. Stewart, "Thoughts on Emulating Jitter for User Experience Trials", Proceedings of NetGames 2004 Workshop, ACM SIGCOMM2004, Portland, Oregon, USA, August 2004.

[3] Ubicom White Paper, "OPScore or Online Playability Score: A Metric for Playability of Online Games with Network Impairments", 2005,

[4] A.F. Wattimena, R.E. Kooij, J.M. Van Vugt, O.K. Ahmed, "Predicting the perceived quality of a First Person Shooter: the Quake IV G-model", Proceedings of NetGames 2006, Singapore. October 2006

[5] Chen, P., El Zarki, M.: "Perceptual view inconsistency: an objective evaluation framework for online game quality of experience (QoE)". In: Proceedings of International ACM SIGCOMM Workshop on Network & System Support for Games (NETGAMES) (2011)

[6] Panda3D Features http://www.panda3d.org/manual/index.php/Features

[7] Ogre 3D About http://www.ogre3d.org/about

[8] Delta 3D Documentation http://www.delta3d.org/index.php?topic=docs

[9] Irrlicht FAQ http://irrlicht.sourceforge.net/faq/

[10] Irrlicht Powered Games http://www.irrlicht3d.org/wiki/index.php?n=Main.IrrlichtPowered

[11] Torque 3D documentation http://www.garagegames.com/products/torque-3d/documentation

[12] Torque 3D Games http://www.garagegames.com/torquepowered/3d-games

[13] Doom3 source code https://github.com/TTimo/doom3.gpl

[14] Doom3 code review http://fabiensanglard.net/doom3/index.php

[15] ITU-T Recommendation G. 107 "The E-model, a Computational Model for use in Transmission Planning", Geneva, March 2003.