***Multimedia Systems –***

**C-WG3#1**

Project Plan

**PURPOSE OF THIS DOCUMENT**

This document contains everything concerning the implementation phase of the Cloud Gaming project and further development ideas.

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision # | Date | Author(s) | Comments |
| 0.1 | 26/11/09 | Lauri Majamaa | First version |
| 0.2 | 29/11/09 | Lauri Majamaa | Some text and notes added |
| 0.3 | 01/12/09 | Pasi Keski-Korsu | Chapter 2 written |

**ABBREVIATIONS**

|  |  |
| --- | --- |
| Scrum | Iterative incremental framework for agile software development |
| LibVLC | Video decoding and playing framework |
|  |  |

**REFERENCES**

**TABLE OF CONTENTS**

1INTRODUCTION 4

2Final Analysis 5

3Project outcome 6

4Improvements for the future 7

# INTRODUCTION

This document is a final report for our Cloud gaming project done for Multimedia Systems course in University of Oulu fall 2009. Our group consisted of five members, Lauri Majamaa, Antti Lampela, Tuomas Vähänen, Pasi Keski-Korsu and Antti Väyrynen. The project was done as an agile software project, meaning the team was called a Scrum and development cycle was divided into sprints. Scrum model defines the software development iterative incremental. Before a sprint cycles a backlog or project plan is made. The backlog contains initial requirement specification, which determines project deadlines and what is to be implemented before each deadline. During the implementation phase we went through one sprint, lasting one month, from 1.11.2009 to 1.12.2009. The Scrum method could be only partially used to our project because the short development period and amount of implementation needed, and therefore only partial gain from the method was achieved.

The aim of the project was to create a solution for playing games with cloud architecture. The game is run on a server instead of your own personal computer, and its video is streamed to your computer over the net. Your game controls will be also forwarded from client to server in the process. The playing experience is aimed to match what you would have if the game was run locally. Cloud architecture makes possible to play power consuming games on a relatively powerless computer, so investing in state-of-the-art hardware anymore, because video rendering is done by the server. Client platform only has to be able to decode the video stream and have a fast enough connection to receive it.

# Final Analysis

**Problems**

**At first, we determined that our problems will be both computers' processor speed. We were partly right. Server computer needs a lot processor capacity to run the game and process the video capturing, compression and stream. With a two-year-old laptop, which is designed for playing, server program could not run properly. We also noticed that latency between client video and actual game on server is a big problem. We could not use buffer which was big enough to run the game smoothly. Video quality was quite bad as well. Better quality takes more network bandwidth.**

**Solutions**

Problem with server computer's capacity is easily fixed: buy better hardware. In cloud gaming, idea is to have one efficient computer which can run multiple processes that cannot be possible to run on client computer. Latency between client and server could be fixed by optimizing buffer. That can be carry out by analyzing LibVLC switches thoroughly. Picture quality is a difficult task to improve because network bandwidth is always a limit. Still, problem could be solved by making different codec choices and find an efficient way to encode and decode video stream.

# Project outcome

Here is listed the overall project outcome that is delivered on the project DVD.

• Project plan – MMS\_Project\_plan.pdf

• Final Report – MMS\_Final\_Report.pdf

• Project web-site

• Requirement Specification – MMS\_Requirement\_Specification.pdf

• Design Specification – MMS\_Design\_Specification.pdf

• Executable binaries for client on Windows and Linux platforms, server scripts and source code

# Improvements for the future

Possible future developments for the project software could include porting the client to mobile devices, for example the Maemo platform used in Nokia’s newest smartphone. Better optimization would be needed to reduce stream bandwidth to a limit capable of transmitting through wireless networks. This means having higher compression in the video stream and reduction of stream receiving buffer. Lower buffer puts higher demands for the network connection stability. When there are 25 frames per second arriving to the client, even if few frames get lost on the way, the video stream turns in to a stuttering collection of images.

Software

* Portability
* Buffer optimization
* Better codec
* More powerful server platform
* Audio included

Development

* Strict documentation cycle
* More code-aware people on the team
* More scrum-like development, not normal cowboy coding.
* Equal work amount for all project members

Possible future developments could also allow the game to access many servers at once to receive more calculation power if needed by the game. This and heavy increase of network bandwidth is needed for cloud gaming to become a serious choice for personal gamer.