KONTRPİYE

TEST SPECIFICATION REPORT

LINUX EVOLUTION SOCCER

1394865 Ufuk Dallı
1448505 Hakan Çağlar
1502145 Berker Batur
1502202 Uğur Büyükköy

Contents

1.Introduction

1.1 Goals and Objectives

The purpose of this document is to define and outline the Test Specifications for Linux Evolution Soccer. It covers the general methods made use of in the tests conducted for LES. Since LES is a football game and it is supposed to be played for hours at average by the end users, each unit and the whole system needs further considerations rather than just testing single functionalities just after they are developed. Quality assurance, verification and validation, reliability estimation, a generic metric, and trade-off between time and quality imply the importance of testing. The goal is to end up with a robust, bug-free and high-performance LES after being finished all the testing process.

1.2 Statement of Scope

This document was prepared by the developer team of LES, Kontrpiye. It covers descriptions of the methods used in the testing process of LES, plan of the testing phase, procedures to be followed and techniques of keeping records during testing. During the development of project LES, each component of every module is to be tested by the developers right after it is “done”. However, since many modules including many components work interactive in LES, further separate testing of modules and testing of the whole system is obliged.

1.3 Major Constraints

1.3.1 Time

LES is designed to meet the needs for a cool native game of Linux users who might like to run it for hours. Optimization of all modules for better performance and any speed up possible is in the scope of the time constraint.

1.3.2 Data

Since LES is composed of many modules running concurrently and passing data to each other during the rendering minimizing the amount of data being transferred between modules is a goal in the sense of data constraint. For instance, the network module should be sending minimal data to the other participant for better performance of networking.

1.3.3 Hardware

LES is tested on different PCs in regards of different hardware configurations (in both Ubuntu and Windows Operating Systems) for performance and correctness constraints for the sake of portability.

2. Testing Specification

Testing specifications are Unit Testing, Integration Testing and Performance Testing.

2.1 Unit Testing

Since each module in LES is developed separately and independently from others in regards of Object-Oriented Design and tested after any development or a new task is implemented, testing of modules happens all the way during the development phase.

2.2 Integration Testing

After all the modules pass the unit testing of theirs, they get to be tested if they work correctly when they are running concurrently and communicating to each other. This specification is very important since different implementations coming from different modules may conflict while running together which is to be located during Integration Testing and to be recorded for fixing.

2.3 Performance Testing

Performance Testing aims to achieve reasonable response times to the user when he/she is running the game. Those tests are conducted within many different PCs with different configurations to observe the varying waiting times and video quality of LES. Until now, the performance tests have been satisfying on different hardware configurations via which they were done.

3. Testing Process

3.1 Modules Testing

Each module of LES is tested independently from others for the sake easier recognition of problems and after unit tests are completed some interrelated modules are tested with each other before the whole system integration such as Game Menu and Database integration or I/O, Graphics and Sound integration. At the end all of the modules are tested altogether within integration tests. The names of the major modules to be tested are:

i) Game Menu
ii) Game Engine
iii) Match Engine
iv) Sound
v) Graphics
vi) I/O
vii) Database
viii) Network

3.2 Fail/Pass Criteria

3.2.1 Game Menu

The game menu which is consisting of a toolbar and and regular push-down buttons is supposed to respond to the user quickly with fast switches from one menu to another. Moreover, the features that do not work and have links in the game menu should be disabled or return messages to the user informing that the feature is not up in the corresponding release. After the user proceeds to a match he/she should wait a reasonable amount of time as well.

3.2.2 Game Engine

Since Game Engine is the module that initializes all the other modules the interactions between all the other modules are to be tested if they happen correctly and quickly. So the Game Engine should run fastly and correctly when LES is first run by the user.

3.2.3 Match Engine

Match Engine communicates each module: Sound, Graphics, I/O, Database, Game Menu and Network. The AI implementation of the match is in this module as well and concurrency with all these interactions while running LES is the pass of this test. Since this module arranges those others, such as when an I/O event occurs and the user presses shooting button, Match Engine is supposed to immediately inform the Sound, Graphics and Network as well if it is a multiplayer game.

3.2.4 Sound

Sound module is in interaction with the Game Menu and Match Engine. This module has to make sure that when a request comes for a sound to be played/paused/stopped it is done immediately. It should also check the overlapping sounds/musics being played at a time and check if they can be played together. For instance, the music playing when surfing in the

Game Menu should be stopped when the match starts, the interval between the music stops and the pitch comes to the screen is also range of Sound test.

3.2.5 Graphics

Graphics test consists of quick response to the incoming requests from I/O module, choices of different resolutions in the Game Menu and collision detection during a match.

3.2.6 I/O

The I/O test is testing of input devices including the keyboard and the gamepad. The constraints are immediate response to a I/O event in the Match Engine.

3.2.7 Database

Database module makes sure that the query to be executed is prepared correctly and performed in a very quick manner. The test on this module fails if irrelevant data is gathered from the DBMS.

3.2.8 Network

Network module is to be tested if it sends and receives the packets to/from the server/client machine correctly and of course again quickly. It should be fast enough that the client user should have the same game state at a moment as the server. It should be also correct obviously, otherwise the users could end up different game states which could crash LES after repeating couple of times which is not the intention at all.

4. Testing Tools

Firstly, the developers of LES conduct the tests on their personal computers which vary on hardware configurations. Since LES runs on both Linux and Windows tests are conducted on both of those operating systems to see if anything goes wrong in one while it is okay in the other. The Debug mode of Eclipse IDE, via which the development is conducted, helps a lot during test processes and bug fixing as well. After a release passes all the tests on a developer’s computer the whole LES package is deployed on some another computer for further tests and considerations.

5. Testing Resources and Staffing

Responsibility	**Staff**
Unit Tests	ALL
Integrations Tests	Uğur BÜYÜKKÖY Hakan ÇAĞLAR
Performance Tests	Ufuk DALLI Berker BATUR
Feature Tests	ALL

6. Testing Schedule

Task	Start Date	Due Date
Test Specification Report	X	26.04.2010
Integration Tests	X	09.05.2010
Performance Tests	10.05.2010	20.05.2010
Alpha and Beta Tests	21.05.2010	25.05.2010
Bug Tracking and Fixes	26.05.2010	10.06.2010

7. References

√ METU Ceng492 Spring 2010 Software Testing Lecture Slides [1]

√ IEEE Standard for Software Test Documentation [2]

[1] http://cow.ceng.metu.edu.tr/Courses/download_courseFile.php?id=2468

[2] http://standards.ieee.org/reading/ieee/std_public/description/se/829-1998_desc.html