Software Architecture

TDT4240

Requirements Document

Group 17 : "XNA videogame focus on modifiability"

Marius Greve Hagen

David Rozas Domingo

Maria Fernandez-Rodriguez

Jarle Lindseth

1 Index

2 Introduction

This document deals with the requirements of the project "SuperPang videogame". It summarizes the main goals of the project. It starts with a brief description of the functionality of the system and an enumeration of the basic and extended functional requirements.

Next, a discussion of the most important quality requirements is presented in Quality Requirements, as well as some scenarios reflecting some of the possible use cases.

In COTS section, we talk about the use of XNA, XQUEST and AMS.Profile, and the constraints we have found.

Finally, a References section, in which we list our main sources, and the list of changes can be found.

3 Functional Requirements

3.1 Description of the main logic and functional requirements

We have decided to design a videogame similar to the classic Super Pang in which you have to bust all the balls around different stages.

Screenshot of the original game

The videogame starts displaying a menu, where the user can choose between start a new game, see the highest scores or see the credits.

If the user choose start a game, the user goes to the first stage with an initial fixed number of lives (currently 2). The player has to destroy all the balls present in the current stage without being reached by them. The balls have different sizes, and they are divided into two smaller balls when they are reached, until the balls are so small that in case of being reached disappear.

In each stage we have two possible ends:

➢ The player has destroyed all the balls, advancing to the next stage.

➢ The player is reached by one ball, losing one life. If the player has more lives, he will play in the same stage with the initial conditions (initial number of balls, the balls have the initial size, ...). If not, the game finishes.

If the player is able to finish all the stages, then a congratulations message is display.

During the game the player receives bonus points for many actions:

➢ Bonus points for destroying a ball. The smaller the ball is, the more points he receives.

4

➢ Bonus points for finishing a stage. The player receives more points if is able to finish it with more lives.

➢ Bonus points for finishing the game.

➢ (...)

Once the player has finished the game or has lost all his lives, the program checks if the current score of the player is between the best five scores, which are permanently stored in a configuration file. If the player has been able to break the current scores, the videogame displays a menu where the user can introduce three letters (old arcade-style), and his score will be stored. Then the videogame displays the updated list of scores to the user.

If the user is not able to break the current high scores, then is redirected to the main menu.

In the main menu the user can also see the high scores or visit the credits, where our names are displayed.

3.2   Description of the main functional requirements

For the functional requirements the following notation is used:

➢ BFR N: Basic Fuctional Requirement N

➢ XFR M: Extended Fuctional Requirement M

We have thought that the functional requirements needed for a simple version of the game are the following:

➢ BFR 1: One player which can move horizontally is displayed.

➢ BFR 2: The player is able to shoot a "laser gun" pressing the space.

➢ BFR 3: There is one stage, with one ball bouncing.

➢ BFR 4: There is only one kind of ball, which disappears once it has been reached by one of the "laser shots".

Due to our main quality requirement consists of being easily modified, we have anticipated the following possible extensions:

➢ XFR 1: Addition of three different sizes of balls. When a ball of size three is reached, then it is divided in two of size two, when a ball of size two is reached, then it is divided in two balls of size one. If a ball of size one is reached, then it disappears.

➢ XFR 2: Addition of new levels with different number and sizes of balls. This has

to be easily extendable, reading from a XML file.

- ➢ XFR 3: Addition of score, which is increasing due to different events (reaching a ball, accomplish one level, finish the game, etc.)
- ➢ XFR 4: Addition of high scores displaying component, where the best five scores are displayed. This information has to be stored permanently in a XML file.
- ➢ XFR 5: Addition of input name of the user tool in old arcade style (three letters). It will be displayed if the high scores have been broken.
- ➢ XFR 6: Addition of a menu where the user can choose between start a new game, see the current high scores or see the credits.
- ➢ XFR 7: Addition of double shoot feature. This is done when the player reached a fixed amount of points.
- ➢ XFR 8: Addition of different kind of items which allow the user to get extra points, extra lives, etc.
- ➢ XFR 9: Addition of two players.

4 Quality Requirements

Besides modifiability, we think that for our project testability and usability could be really important.

Following, some examples of possible scenarios for this quality attributes are shown.

4.1 Modifiability

Scenario 1 [M1]: "Addition of double shooting capability when a certain score is reached"

Portion of Scenario	Possible Values
Source	Developer
Stimulus	Add new functionality
Artifact	Code
Environment	Design time
Response	Makes modifications without affecting other functionality
Response measure	Less than one day

Scenario 2 [M2]: "Addition of balls of different sizes which are divided in smaller balls"

Portion of Scenario	Possible Values
Source	Developer
Stimulus	Add new functionality
Artifact	Code
Environment	Design time
Response	Makes modifications without affecting other functionality
Response measure	Less than one day

Scenario 3 [M3]: "Addition of new levels, which are stored in XML format"

Portion of Scenario	Possible Values
Source	Developer
Stimulus	Add new functionality
Artifact	Code
Environment	Design time
Response	Makes modifications without affecting other functionality
Response measure	Approximately one day

Scenario 4 [M4]: "Addition of high scores stored permanently in XML format"

Portion of Scenario	Possible Values
Source	Developer
Stimulus	Add new functionality
Artifact	Code
Environment	Design time
Response	Makes modifications without affecting other functionality
Response measure	Approximately one day

Scenario 5 [M5]: "Insertion of new high score and username in old-arcade style"

Portion of Scenario	Possible Values
Source	Developer
Stimulus	Add new functionality
Artifact	Code
Environment	Design time
Response	Makes modifications without affecting other functionality
Response measure	Approximately one day

Scenario 6 [M6]: "Add a menu to provide access to a new functionality"

Portion of Scenario	Possible Values
Source	Developer
Stimulus	Add new functionality
Artifact	Code
Environment	Design time
Response	Makes modifications without affecting other functionality
Response measure	Approximately one day

Scenario 7 [M7]: "Addition of two players feature"

Portion of Scenario	Possible Values
Source	Developer
Stimulus	Add new functionality
Artifact	System that interoperates with target system
Environment	Design time
Response	Locates places in architecture to be modified
Response measure	- Around 2 days

4.2 Testability

Scenario 1 [T1]: "System test"

Portion of Scenario	Possible Values
Source	System tester
Stimulus	Subsystem and classes
Artifact	Piece of code
Environment	Development-time
Response	Choose, capture and analyze state and computer values. A general preparation of the test is also needed.
Response measure	Probability of failure less than 1%

Scenario 2 [T2]: "End-user testing"

Portion of Scenario	Possible Values
Source	End user
Stimulus	System delivered
Artifact	Complete application or prototypes
Environment	Runtime
Response	- Prepare customized tests which measure the level of satisfaction of the common user through different questions. - These questions have a scale (1-5), being 5 the maximum level of satisfaction.
Response measure	The satisfaction level arithmetic media should be equal or bigger than 3.

4.3 Usability

Scenario 1 [U1]: "Customizable control keys"

Portion of Scenario	Possible Values
Source	End user
Stimulus	Feature which allows the user to choose his preferred control keys
Artifact	System
Environment	At runtime
Response	Adapt the system (customizability)
Response measure	Feel comfortable

5 COTS Components and Technical Constraints:

The Software Engineering Institute (SEI) defines COTS product as one that is:

- ➢ sold, leased, or licensed to the general public
- ➢ offered by a vendor trying to profit from it
- ➢ supported and evolved by the vendor, who retains the intellectual property rights
- ➢ available in multiple, identical copies
- ➢ used without modification of the internals

In addition, SEI also offers a definition of a COTS Based Application (CBA) as a system for which at least 30% of the end-user functionality is provided by COTS products, and at least 10% of the development effort is devoted to COTS considerations.

We consider that our video game development is categorized as a CBA based on XNA Framework, since most of the functionality depends on it.

XNA Game Studio 2.0 is a set of tools based on supported versions of Microsoft Visual Studio tools that allow students and hobbyists to build games for both Microsoft Windows and Xbox 360. XNA Game Studio also includes the XNA Framework, which is a set of managed libraries based on the Microsoft .NET Framework 2.0 that are designed for game development.

We are going to use XQUEST, as well, a small and lightweight 2D game library / game template that contains convenient game components, helper classes, and other classes that can be used in your XNA game projects. We do not consider that our application is a CBA respect to XQUEST. It neither has supposed any technical constraint for our project, although it could create some technical constraints if we would have had more ambitious goals, as expanding a 2-D game to 3-D.

For the components related to XML, we are going to use the library AMS.Profile by Alvaro Mendez, which is under CPOL (Code Project Open License, see reference). This library provides a very easy way to read and write XML files, and we consider it is not going to suppose any kind of technical constraint.

6 References

-http://msdn2.microsoft.com/en-us/library/bb200104.aspx accessed 24. February 2008

-http://www.sei.cmu.edu/news-at-sei/columns/the_cots_spot/1999/December/cots-dec99.htm, accessed 24. February 2008

-"Software Architecture in Practice, Second Edition", Len Bass, Paul Clements, Rick Kazman (Tapir).

-http://www.crazymonkeygames.com/Super-Pang.html accesed 24 February 2008

-http://www.codeproject.com/info/Licenses.aspx , accesed on 7th April 2008.

-http://www.codeproject.com/KB/cs/readwritexmlini.aspx , accesed on 7th April 2008

7 Changes

- ➢ 3rd April 2008: Extension of the introduction, explaining briefly which are the sections and subsections added.
- ➢ 3rd April 2008: Functional requirements section is divided into to subsections: description of the logic of the game and functional requirements.
- ➢ 3rd April 2008: Extension of the description of the logic of the game.
- ➢ 4th April 2008: Deleted old scenarios M2 and M3
- ➢ 4th April 2008: New description in the old scenarios: M1 and T2
- ➢ 4th April 2008: Added scenarios M2-M7
- ➢ 7th April 2008: Extension of the COTS and Technical constraints section.
- ➢ 7th April 2008: References to articles used in COTS section added in References sections.
- ➢ 7th April 2008: Extension of the introduction, explaining the libraries which we are going to use, and which are explained in detail in COTS section.