



NTNU – Trondheim
Norwegian University of
Science and Technology

TDT4240 - SOFTWARE ARCHITECTURE

Requirements Document

FOODFEUD

Group A6
Android:

Kjetil Aune
Annie Aasen
Mikal Bjerga
Nikola Radenkovic
Jonathan Bruschi Nielsen
Trapnes

PRIMARY FOCUS ATTRIBUTE:
MODIFIABILITY

SECONDARY FOCUS ATTRIBUTE:
TESTABILITY

February 27, 2015

Contents

1	Introduction	3
2	Functional Requirements	4
2.1	Functional requirements	4
2.2	Non-functional requirements	4
3	Quality Requirements	5
4	COTS - Components and Technical Constraints	6
5	Issues	7
6	Changes	8

1 Introduction

This document describes the architecture of our “TANK” artillery strategy game for Android developed by Annie Aasen, Mikal Bjerga, Nikola Radenkovic, Jonathan Brusch Nielsen Trapnes and Kjetil Aune. The game takes inspiration from the Worms series. It’s a turn-based multiplayer game where the objective is to hit and destroy the enemy tank with different types of ammo. When the round is over, players can visit the store where they can buy new weapons and tank upgrades. We have decided to use different types of food as ammo.

2 Functional Requirements

[1]

2.1 Functional requirements

- Playable as an offline multiplayer game

2.2 Non-functional requirements

- Support rapid design changes

3 Quality Requirements

Stakeholder	Concerns
Users (Anyone running the program)	Is the game easy and intuitive to use? Is the game fun to play?
Developers (Group A6)	Will it be possible to develop and test the application in 9 weeks? Is the game easily modifiable? Is our program easy to test?
Evaluators (ATAM-groups and course staff)	Does the application meet all the functional requirements? Does the application meet all non-functional requirements? Is the implementation easy to understand (e. g. intuitive variable names and explained through comments) and well-documented? Is the documentation complete and straightforward?

4 COTS - Components and Technical Constraints

We have chosen to use a logic view, development view and process view, as explained in *The “4+1” View Model of Software Architecture* [2]. We have opted not to use the physical view and scenarios from the same paper, as we have deemed them to be unnecessary. The physical view is not needed as we will only use a single device, running a single process, which does not require any documentation to understand. Using the scenario view was discussed, but abandoned, due to the simplicity of our game. If we choose to modify it into a more complex game, e. g. with multiple game modes or different ways of playing the game (e.g online), a scenario view would have to be added to the documentation.

5 Issues

In this chapter, you should discuss the results you have obtained from your implementation. These can be correctness results, i.e whether the implementation behaved as expected, or numerical results that express runtime or energy measurements.

6 Changes

In this chapter, you should discuss the results you have obtained from your implementation. These can be correctness results, i.e whether the implementation behaved as expected, or numerical results that express runtime or energy measurements.

Bibliography

- [1] D. Adams. *The Hitchhiker's Guide to the Galaxy*. San Val, 1995.
- [2] Krutchen. *The "4+1" View Model of Software Architecture*. 1995.