IMPLEMENTATION PHASE

1. INTRODUCTION

These are the instructions for the implementation phase of the Game project. The focus of this phase is to design, implement and test the game itself. The documentation to be delivered from this phase should focus on the test results when it comes to evaluating the game application towards the requirements and discussing the relation between the implemented game and the architectural documentation. The implementation should be delivered both with source code and class files (where applicable), in addition to updates of the previous deliveries (Requirement, Architecture design, and ATAM (the ATAM report written by your group)). It is important that the game is easy to install and run.

2. DELIVERIES

The deliverables are:

- Implementation document (See below)
- Application source code and class files (Zipped)
- Architecture document (Updated)
- Requirements document (Updated)
- ATAM document (If any changes are made)

About the change list of each document: They should not only list what was changed and how, but importantly also why things were changed.

3. IMPLEMENTATION DOCUMENT TEMPLATE

The Implementation document should contain the following:

- Front page, that includes:
  - Group name/ Name of game
  - Group members
  - Chosen COTS
  - Primary quality attribute chosen
  - Secondary quality attribute(s) chosen
- Introduction
  - Description of the project and this phase (requirement phase)
  - Description of the game concept (sufficiently described and explained and with illustration or screenshots from a similar game)
  - Structure of the document
- Design details / Implementation details
  - More detailed description of how the game was designed and implemented, including complete class-diagram, description of the implemented classes, etc.
- User Manual
  - Description of how to install, compile, and run the game
  - Screenshots from the game explaining how to play it

- Test report
    - This section should contain test reports for both functional- and quality requirements (quality scenarios)
    - The reports must include requirement ID (e.g. F1, F2, A2, A3…), description of requirement, who performed the test, when it was performed, duration of the test, evaluation (failure/success), and comment (discussion/comment about the result).
    - Quality requirement tests must additionally include stimuli, expected response measure, and observed response measure.
    - See figures below for example for reporting functional and quality requirement tests.
- Relationship with architecture
    - This should list the inconsistencies between your architecture and the implementation.

      Give the reasons for these inconsistencies. Discuss whether they could have been discovered at an earlier point, for instance during the ATAM evaluation.
- Problems, issues, and points learned
    - In addition to listing problems and issues with the document or implementation process, this is also the place to reflect upon the project and discuss what you would have done differently if you were to start again from scratch.

4. TEST EXAMPLES

F1: Hitting the space button should shoot a projectile	
Executor:	Kari Nordmann
Date:	15.03.2016
Time used:	2 minutes
Evaluation:	Success
Comment:	Hitting the space button fired a bullet, the game felt very responsive and did not "miss" any inputs
M1: Modify enemy movement	
Executor:	Kari Nordmann
Date:	17.03.2016
Environment:	Design time
Stimuli:	Change movement of enemy NPCs
Expected response measure:	20 minutes
Observed response measure:	50 minutes
Evaluation:	Failure
Comment:	Lack of coherence made changes more difficult and several modules had to be changed.