SOFTWARE ARCHITECTURE PROJECT INFO (REQUIREMENT & ARCHITECTURE PHASE)

This document contains info about the main task of the software architecture project, as well as templates for the first delivery. As you should be acquainted with the COTS (Android and iPhone SDKs) by now, we will not go into details about that.

1. The Goal of the Project

The task is to make a functioning multiplayer game for Android or iPhone, based on your own game concept. The game must be designed according to a specified software architecture (presented in this course). The game must offer a multiplayer feature, exactly how this feature is implemented is up to you (hot seat, real time, turn based, asynchrony, etc.). It does not have to be a networked game, but as the complexity will increase – it will improve the chances of getting a higher grade.

2. Group Specific Tasks

The group should develop a game with focus on one primary and at least one secondary quality attribute:

- Primary quality attribute:
  - **Modifiability:** The software architecture and implementation should be easy to change or configure in order to add or modify functionality (e.g. add/remove players, change size of board, functionality that affects the architecture)
- Secondary quality attribute:
  - Testability
  - Usability
  - Performance
  - Availability: Only applicable if you create a networked game

The choice of the quality attributes must be reflected in the quality requirements, in the architectural design, the implementation, and in the final tests of the application.

3. Your Solution

The perfect implementation is not the ultimate goal of this project, but the quality of the implementation counts together with all the documents in the final grade (do not forget that this is a software architecture course, not a game design course). Also remember that the implementation and the architectural description should reflect each other. Most likely you have to change/modify/update the architecture based on what is being implemented. The final code and implementation have to be well documented and easy to compile and run. Too much hassle (and time) in order to get your system to run will count negative towards your final result.

4. About the Documents

The document should be written in English and the only acceptable format is PDF. The project will be graded on the final delivery, but documentation with large holes during the project can subtract from the final score.

4.1 Requirements Document

In this document you will list the requirements for your game. Use the following structure:

Front page, that includes:

- Project/Game name
- Group name
- 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

Functional Requirements

- A complete list of the functional requirements you have to fulfill in order complete the task. Each requirement must have a unique ID, e.g. FR1, FR2, FR3 etc. Can also be decomposed into sub-requirements such as FR3.1, FR3.2.

Quality Requirements

- Write scenarios for the most relevant quality attributes (modifiability, testability, usability, availability)
- Use scenarios (text or table) of the type used in Chapter 4 of the textbook.
- Make the quality requirements measurable/testable with values that later can be checked/validated. Make estimates for the result of the later test.
- Tables are recommended to specify quality requirements
- Quality requirements must also have a unique identifier; such has A1 , A2... for availability, M1, M2… for modifiability.

COTS Components and Technical Constraints:

- Describe the constraints on the architecture due to your choice of COTS. This part should specify the interfaces to COTS you must use and how the choice of COTS (e.g. Android or iOS platform) will restrict/constrain your architecture.
- State any other constraints (on the architecture) relevant for your project.

Issues

- Optional point of issues you faced working with this project and this document

Changes

- Describe the changes carried out with this document from first draft until the final delivery, including all improvements based on feedback from course staff and others.

References

- List of references to books, articles, web-pages, and documents used.

4.2 Architectural Description Document

This is your main document and contains the architectural description for the project. It should contain (based on recommendation from IEEE 1471):

Front page, that includes:

- Group name and name of Game
- Group members
- Chosen COTS
- Primary quality attribute chosen
- Secondary quality attribute(s) chosen

Introduction

- Description of the project and this phase (architectural phase)
- Description of the game concept (sufficiently described and explained) with illustrations
- Structure of the document

Architectural Drivers / Architectural Significant Requirements (ASRs)

- The main drivers (functional, quality, and business requirements) that most affects the system architecture

Stakeholders and Concerns

- The stakeholders of the system and their concerns related to the software architecture

Selection of Architectural Views (Viewpoint)

- A list of the views (viewpoints) that you will use in the architectural documentation, their purpose, target audience (which stakeholder you are addressing), and what notation will be used for each view (use a table for this purpose).
- You have to choose at least one logical view, one process view, and one development view (or sufficiently similar alternatives). If your game has an online component, you should also provide a physical view.
- Candidate views can be found in TextBook or 4+1 paper
- Specify the notation used (e.g. UML class diagram, UML sequence diagram etc.).

Architectural Tactics

- Describe tactics for the architecture to meet the quality requirements

Architectural and Design Patterns

- Selection of architectural- and design patterns used in your architecture

Views

- Document architecture using your selected views (at least logical, process, and development) through diagrams and supporting text. If your game has an online component, you should also provide a physical view.
- Remember that the view should be useful and practical, e.g. the development view should help assign programming tasks to group members and allow for simultaneous development of multiple parts of the system.

Consistency among architectural views:

- Describe inconsistencies among the views in your architecture

Architectural Rationale

- Explain why you have chosen the architecture you have (the motivation for your choices). This part should explain/discuss why your selected architecture will work and how/why it will fulfill all the requirements.

Issues

- Optional point of issues you faced working with this project and this document

Changes

- Describe the changes carried out with this document from first draft until the final delivery, including all improvements based on feedback from course staff and others.

References

- List of references to books, articles, web-pages, and documents used.

4.3 Code Skeleton (Optional, but recommended)

We recommend that you implement and deliver a code skeleton for your architecture if you find the time to do so. A code skeleton is the main structure of your system with no or little functionality. It is important to start the implementation as early as possible, and to ensure that it is possible to implement the architecture you have designed, within the constraints given.

4.4 Templates for References

Here are some examples of how to write proper references to books, articles, and web - pages

Book:

- Len Bass, Paul Clements, Rick Kazman, “Software Architecture in Practice – Third edition”, Addison Wesley, September 2012

Article:

- Phillipe B. Kruchten, “The 4+1 View Model Of architecture”, IEEE Software Magazine, Volume 12, Issue 6, 1995.

Web page:

- Microsoft, “Xbox Live Indie Games – xbox live indie games development”; Web: http://www.xna.com/ Accessed 19. February 2015

Tips: If you want to find scientific articles related to the topic, you can use google scholar. Google scholar can also provide you with the right way of referring to a book or an article.