

Requirements

**LARGE AND HEAVILY ARMORED WARSHIPS
FOR ANDROID**

Primary quality attribute: Modifiability

Secondary quality attribute: Usability

Trond Kjetil Bremnes, Marius Glittum, Alexander Perry
Even Stene, Elisabeth Solheim, Trond Klakken

February 27, 2012

Chapter 1

Introduction

This document contains the requirements for an Android game called 'Large and Heavily Armored Warships or LaHAW', to be created by Trond Kjetil Bremnes, Marius Glittum, Alexander Perry, Even Stene, Elisabeth Solheim and Trond Klakken. The purpose of the creation of the game is to train ourselves in use of *Commercial Of The Shelf* (COTS) software, as well as architectural design.

1.1 The game

LARGE AND HEAVILY ARMORED WARSHIPS (LAHAW) is based on the popular Battleships[2] game, a game where the objective of the game is to sink enemy ships, while concealing the positions of one's own ships. The main game screen is based on an ocean space with multiple tiles (like a grid), where each tile represents a coordinate on the player's ocean space. A ship can span several tiles, in the x and y direction (or it could use just one tile).

Ship name	No. of occupied tiles	No. of ships per player
Aircraft	5	1
Battleship	4	2
Submarine	3	2
Destroyer	3	1
Patrol boat	2	2

Chapter 2

Functional Requirements

- FR1** – Change difficulty/size of ocean space
The user should be able to choose between different difficulty levels. When a level is chosen, the game creates the grid with a given size.
- FR2** – Set/change player name
The player should be able to change their name
- FR3** – Game over
The game ends if a player gets all of their ships destroyed. The winner is the player with any ships left.
- FR4** – Place ones ships at the start of the game
The player should be able to place his/her ships at the start of the game and only then. During gameplay the ships must be stationary. He/she can choose to use any of the ships given at start, as seen in the table in chapter 1.1
- FR5** – Play audio
Play audio when a ship is hit and when the player misses.
- FR6** – Fire on enemy ships
A players must be able to fire a shoot when it is his/her turn on one of the tiles that have not allready been fired at.
- FR7** – Register hits on friendly ships
A player should be able to observe if the enemy player hits the other players ship and thus knowing the "score".

Chapter 3

Quality Requirements

3.1 Modifiability

M1 Change difficulty

The user is able to choose between three difficulty levels: easy; medium; and hard. Ocean space is determined by these, where easy gives the largest ocean space, and hard gives the smallest ocean space. If the ocean space is small, the probability to get hit increases and vice versa.

Portion of scenario	Values
Source	Developer
Stimulus	Add difficulty opportunity
Artifact	A change on target system
Environment	Design time
Response	Opportunity to change ocean space
Response measure	10 hours to modify

M2 Set player color

The user should be able to change the color of their ships, i.e. their player color.

Portion of scenario	Values
Source	Developer
Stimulus	Add opportunity to change your player color
Artifact	A change on target system
Environment	Design time
Response	Opportunity to change ship colors
Response measure	10 hours to modify

M3 Set player name

The user should be able to set their player name before they start to play.

Portion of scenario	Values
Source	Developer
Stimulus	Add opportunity to set your player name
Artifact	A change on target system
Environment	Design time
Response	Opportunity to set player name
Response measure	5 hours to modify

3.2 Usability

According to Bass, Clements and Kazman [1], *usability* is ...

U1 Placing the Large and Heavily Armored Warships

The user should be able to see each of the Large and Heavily Armored Warships being placed on the ocean.

Portion of scenario	Values
Source	End User
Stimulus	Change position of the ships
Artifact	System
Environment	Game play
Response	Move ships
Response measure	The user is able to move each ship one time

U2 Player name and color saved

The player name and color will be saved at start for later use in the game.

Portion of scenario	Values
Source	End User
Stimulus	Store player name and color
Artifact	System
Environment	At configure time
Response	The player name will appear at the game play screen, and the ships will be painted in the predefined player color
Response measure	Name and color updates when the player starts the game

U3 Dialog boxes confirming the users actions

The user should be prompted to confirm their actions for severe¹ decisions.

Portion of scenario	Values
Source	End User
Stimulus	User regret their decision
Artifact	System
Environment	Run-time
Response	Dialog box with confirmation button(s) (i.e. "OK")
Response measure	Goes to either the next step in the process, or the previous step

¹A decision with great potential consequence

U4 Actions should be no more than three clicks away

To increase efficiency

Portion of scenario	Values
Source	End User
Stimulus	User wants to use the system efficiently
Artifact	System
Environment	Run-time
Response	Support efficient use
Response measure	Reduce task time

U5 The game should show game tips where appropriate

The user gets well-founded recommendations or tips during game play, in order to increase confidence

Portion of scenario	Values
Source	End User
Stimulus	User is uncertain on how the game is played, or what their next move should be
Artifact	System
Environment	Run-time
Response	A message box with tips
Response measure	User gets more confidence in their next move

Chapter 4

COTS, Components and Technical Constraints

Commercial of the Shelf (COTS) software is defined as software or components that can be readily purchased[1]

Android is an operating system developed for devices such as tablets and mobile phones, with various screen sizes and hardware specifications. There is a few technical constraints linked to the Android platform, but far less than on the iOS platform (the largest competing mobile operating system).

4.1 Native Android game

The game runs native on the Android platform, and only this platform.

4.2 Touch screen

The game will be tailored for Android devices with a touch screen as the main source of user input. The graphical user interface must be adapted accordingly.

4.3 Version segmentation

Devices using the Android platform runs several different versions, but most of them run version 2.2 and up.[3] The game will run on devices with Android version 2.2 and up.

4.4 Programming language

As the Android SDK[4] uses Java, the game will be coded in this language.

4.5 AndEngine

To aid in the rapid development of LaHAW, the free and open source *AndEngine* game engine is to be used. This will have an impact on how the game will be

written.

Bibliography

- [1] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2 edition, 2003.
- [2] Boardgame geek. Battleship, February 2012.
- [3] Google. Android platform versions, February 2012.
- [4] Google. Android software development kit, February 2012.