Requirements Document
TDT4240 - Group A14

Framework:
Android

Quality Attributes:
Primary: Modifiability
Secondary: Useability

Group members:
Bremnes, Jan A. S.
Johanessen, Stig Tore
Hesselberg, Håkon
Kirø, Magnus L.
Randby, Simon
Tørresen, Håvard

March 6, 2013

Contents

Introduction

We are currently in the planning phase of the project. This project involves the creation of a multiplayer Android game. In this specific phase we need to iron out the requirements for the project and decide on the overall architecture of the game.

Battleships, but with bigger guns and faster ships! It will be like the classic Battleships, but with the possibility of upgrading ships and weapons and perhaps a larger playing field. It will be a multiplayer game played on one or more devices. This game will be a configurable reinvention of the classic game Battleships.

Functional requirements

- FR0: Player shall be able to place platforms with devices.
- FR1: Player shall be able to click.
- FR2: Player shall be able to click platforms.
- FR3: Player shall be able to click empty squares.
- FR4: Player shall be able to win
- FR5: Player shall be able to lose
- FR6: System shall support local multiplayer
- FR7: System shall play sounds
- FR8: App shall resize to the screen size of the device it is on.
- FR9: System may support network multiplayer (low priority)

Quality requirements

Scenarios for the most relevant quality attributes

Player places platforms containing devices: A player gets the game starting screen. He sees the platforms and the board. Because this is a classical game he immediately understands what to do and begins dragging platforms onto the board. Once he is done (has placed all the platforms on the board) he

presses the big "Done"-key and hands the device to the other player so that he may place platforms.

Player attacks and hits: The player taps on the board to shoot. He then gets feedback telling him his attack hit his opponent.

Player attacks and misses: Same as above, except that this time he misses and gets feedback that he has missed.

Player wins: The player sees a big splash screen stating that he has won. He then clicks on the "Done"-button, and hands the other player the phone with the "you lost" splash screen. The player gloats.

Player loses: The player has lost due to losing all his platforms. The opponent has seen his victory splash screen and hands out player the device with the "you lost" splash screen (just to make sure he understands).

Table of quality requirements with IDs

- QR0: The application shall not crash more than once every 1000 moves
- QR1: The application shall register clicks at least 75% of the time.
- QR2: The application shall correctly calculate the correct result of an action at least 99.9% of the time.
- QR3: The application shall scale and run correctly on at least 75% of tested devices running the tested version of android.
- QR4: The game shall recognize the correct player as the winner at least 99% of the time.
- QR5: MTBF shall be at least 1 hour.
- QR6: The application shall not brick any of the developers' phones or computers.
- QR7: The application shall not destroy the property of anyone who has influence on the developers' grades before the grades are final.

COTS components and technical constraints

We are constrained to using the Java programming language, running on the Dalvik VM. Also, we have to take into account variable hardware specifications e.g. processing power, memory, screen size and resolution and must take care to play along with the power-saving mode.

References

none

Issues

We are too awesome at procrastinating and derailing. Luckily, derailing the Fail Train is a win. We couldn't decide whether to focus on quality 1 or quality 2. We decided to focus on x because it seemed better. Also time.

Changes

none