

Requirements

Avalanche Rush

Group 09

Members:

Nico Hannoun
Kacper Multan
Alicja Jonczyk
Arianna Moser
Margaux Xu
Gagan Gupta
Nicolò Costa

Chosen COTS:

Android Device
Android Studio
LibGDX
Firebase Database
Google Play Games Services

Primary quality attribute: Modifiability

Secondary quality attribute: Usability

1 Introduction

1.1 Description of the project and this phase

The project involves creating a multiplayer game or fun app for Android with the following features:

1. Multiplayer Functionality
2. Online Functionality
3. Server Component
4. Complexity

The primary goal is to create a modifiable application, allowing for easy modification and addition of features.

The app must exhibit at least one secondary quality attribute. The chosen quality attributes must be reflected in the quality requirements, in the architectural design (tactics and patterns), the implementation, and in the final tests of the application.

The first phase is about addressing the background for this project, and the architecturally significant functional requirements. The intention of this document is to help the development team to determine how the app will be structured at the highest level.

1.2 Description of the game concept

Avalanche rush is an endless runner game, where the player has to flee from an avalanche while avoiding obstacles. It is a game similar to the *Subway Surfer's* game which we have inspired from but with a multiplayer mode option. In multiplayer mode, the aim is to be the last one still in the race. Players can push each other to get the power-ups. The game ends when one of the players hits an obstacle.

2 Requirements

2.1 Functional Requirements

Functional requirements are the requirements that the user can see directly in the final product, they define the main functions of the system; for our project we decided to prioritize them in this way.

Priority:

High: Mandatory

Medium: Should be in game

Low: Optional

ID	Name	Description	Priority
1	Handle touch	The application must be able to take	High

	input - player control	input from the user through touch control. This is the main way to control the character.	
2	Single player/ Multiplayer mode	The game should be available for a single player as well as for 2 players. Two players is the maximum number of players allowed.	High
2.1	Invite player	The user should be able to invite another player to the game. In our game it is optional because our idea is to assign each lobby a specific id and use that to play in multiplayer mode	Optional
3	Create game	The user must be able to create a new game. While creating a new game, they can choose whether it is going to be a single or multiplayer game.	High
4	Environment interaction	Environment interaction will allow the player to interact with different power ups, obstacles and the other player.	High
5	Pause	The user should be able to pause the game in single player mode.	Medium
6	Sound	The game should be able to play different sounds when the player interacts with the environment.	Low

2.2 Quality Requirements

Quality requirements are supposed to define the capabilities of the system, specified to a certain scenario. In addition, it displays the various parts of the system being affected, how it is affected and how it responds to said scenarios.

The following tables describe concrete scenarios for the two quality attributes (Modifiability and Usability) that we are going to include in the app.

2.2.1 Modifiability

The Modifiability of a system refers to how easily developers can incorporate or expand its functionality. This is typically assessed by the time required to integrate such enhancements.

M1 Add new power-ups

Source of stimulus	Developer
Stimulus	Add power-ups to the game
Artefact	The application
Environment	Run-Time
Response	Deployed power-ups occurring in the application
Response measure	40 minutes

M2 Add new skins of characters and skis to play with

Source of Stimulus	Developer
Stimulus	Add new skins to the game
Artefact	Graphical resources
Environment	Run-Time
Response	Deployed new skins in the application
Response Measure	20 minutes

M3 Add new and different types of obstacles

Source of Stimulus	Developer
Stimulus	Add new obstacles to the game
Artefact	The application
Environment	Run-Time
Response	Deployed obstacles occurring in the application
Response Measure	40 minutes

2.2.2 Usability

- Single Player
- Multiplayer
 - Join a Room
 - Create a Room

Usability examines how users engage with a system, assessing the ease or difficulty of task completion and whether desired outcomes are achieved. Evaluation typically involves measuring task completion time or the frequency of errors encountered throughout the process.

U1 Let an opponent join a lobby

Source of stimulus	Player which wants to play multiplayer game
Stimulus	Generate a lobby id
Artefact	The user interface
Environment	Run-Time
Response	The opponent uses the code
Response measure	Generate the id and wait for the opponent to join takes 40 seconds

U2 Simple interface to understand

Source of stimulus	Players
Stimulus	Players interact with the game interface, including menus, buttons, and navigation elements
Artefact	The user interface
Environment	Run-time
Response	The player's ease of navigation and understanding of the interface elements.
Response measure	Error rate of the players

U3 Game rules are simple for new players

Source of stimulus	New players who are encountering the game for the first time
Stimulus	The player starts the game and is presented with the game environment and its elements
Artefact	The user interface
Environment	Run-time
Response	The player's understanding and interpretation of the game rules
Response measure	Time to understand (expected less than 5 minutes)

3 COTS

- **Android Device:** The application will be developed for Android devices, accommodating a wide range of screen resolutions and hardware specifications across different models.
- **Android Studio:** The application will be created using Android Studio, which is Google's designated IDE for Android development. For our project we chose Java as the programming language.
- **libGDX:** The application will be built on the libGDX Framework, which is an efficient cross-platform game development framework.
- **Firebase Database:** The application will be using FireBase RealTime DataBase for the global leaderboard and creating a global user database.
- **Google Play Games Services**
- **Networks**
- **Constraints for Architecture due to these COTS.**

4 Issues

We had no issues that are suitable for this section.

5 Changes

This section lists change history for this document.

Date	Change history	Comments
February 26, 2024	First released version	None

6 References

- Subway Surfers: Play Subway Surfers Online for Free on PC & Mobile, [online] <https://subway-surfers.org/>, last accessed 20.02.2024.
- General knowledge about Norway, acquired in the first month being in Norway
- Example of final delivery group 11 (Blackboard project section)
- Example of a final project delivery (group 2) (Blackboard project section)
- Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice – Third edition", Addison Wesley, September 2012

7 Individual Contribution

For this first phase of the project, where it was necessary to collect everyone's ideas and opinions, it was not possible to specifically define everyone's contribution.

Content	Contributors
1 Introduction	Everyone
2 Requirements	Everyone
3 COTS	Everyone