NTNU

TEAM NAME [TO CHANGE]
Team Number: 16

TDT4240 Software Architecture

GAME NAME [TO CHANGE]

Requirements
Android

*Cabral Cruz, Samuel (*496704*)*
*Claessens, Bart (*486346*)*
*Ihlen, Erling Hærnes (*765137*)*
*Trollebø, Jarle (*766901*)*

Primary Quality Attribute:
Modifiability and Usability
Seconday Quality Attribute(s):
Availability, Performance and Interoperability

Presented to
Alf Inge Wang

Department of Computer and Information Science
Norwegian University of Science and Technology
(NTNU), Trondheim
Spring 2018

Contents

List of Figures

List of Tables

Introduction

Requirement Phase

The first phase of this project is the requirements phase. The main objective of this phase is the creation of specification document which will serves as a guideline for the following phases. Different aspects of the project will be analyzed during this phase such as the goals, the scope and the market positioning of the desired product. A list of functional requirements and Quality Attributes (QAs) that should be fulfilled by the application will be established in order to precisely define the agreement between the different stakeholders. Moreover, a list of constraint and issues will be dressed so that their impact will be foreseen and the risk to face them minimized. Once all these information will have been gathered, it will be possible to identify the Architecturally Significant Requirements (ASRs) specific to the application. [Bass, 2013, Chapter 16]

Requirements Document

In this report, we first describe the goals and scope of the project followed by the game concept and the expected market positioning of the final product (Vision Statement). Next, a list of the functional requirements and the quality requirements will be presented in the chapters Functional Requirements and Quality Requirements respectively. The chapter COTS Components and Technical Constraints will discuss about the choices of the Commercial off-the-shell (COTS) components and their impact on the software architecture. The main issues and concerns of the project will also be available in the chapter Issues. Since the following document will be updated with the advancement of the project, a log of changes will be maintained (Changes). Take note that a glossary describing the technical expressions can be found at the end of the report.

Techniques and notations used in this document follow recommendations of Bass [2013] and Larman [2005].

Vision Statement

Project Goals

The goal of this project is to develop a multiplayer game called *[GAME NAME TO CHANGE]*. In this game, the users will incarnate a gladiator trying to make its way into a labyrinth to find and defeat its opponents to accumulate points. In addition to the multi player mode, the player will also have the opportunity to play the game on a single player mode where the goal will be to survive as long as possible by resisting to different waves of ennemies spawning. A tutorial mode will also be developed to make the introduction of the game and its different components much easier to the inexperienced users. A more in-depth description of the game concept is available in the section Game Concept.

Project Scope

The *[NAME OF THE GAME TO CHANGE]* will be developed in the context of the group project for the course TDT4240 Software Architecture. The project will be lead by a team of 4 developers that will work part-time for the next 3 months. The best case scenario would have been to find an artist to optimize the aesthetic part of the application.

The application should minimally works any Android platforms that have the necessary utilities to download and execute an Android Package (APK). Even if this project is mainly dedicated to an educational purpose and bounded to a learning context, the team members would like to officially publish the game and maintain it after the end of the semester.

Game Concept

As mentioned before, the concept of the game we are planning to develop is inspired from the vintage video game series called *Bomberman* first developed in 1983 by Hudson Soft. In those game, the user usually incarnate a robot that try to find its way out of the maze by destroying walls and find the key that will open the door to the next level. Multi player versions of this game has been developed since 1990, but for long time remained a side feature of the original single player game. Wikipedia contributors [2018]

Even if this game has been commercialized under new official console versions for more than 20 years, we still think that this concept can be pushed forward by the addition of new

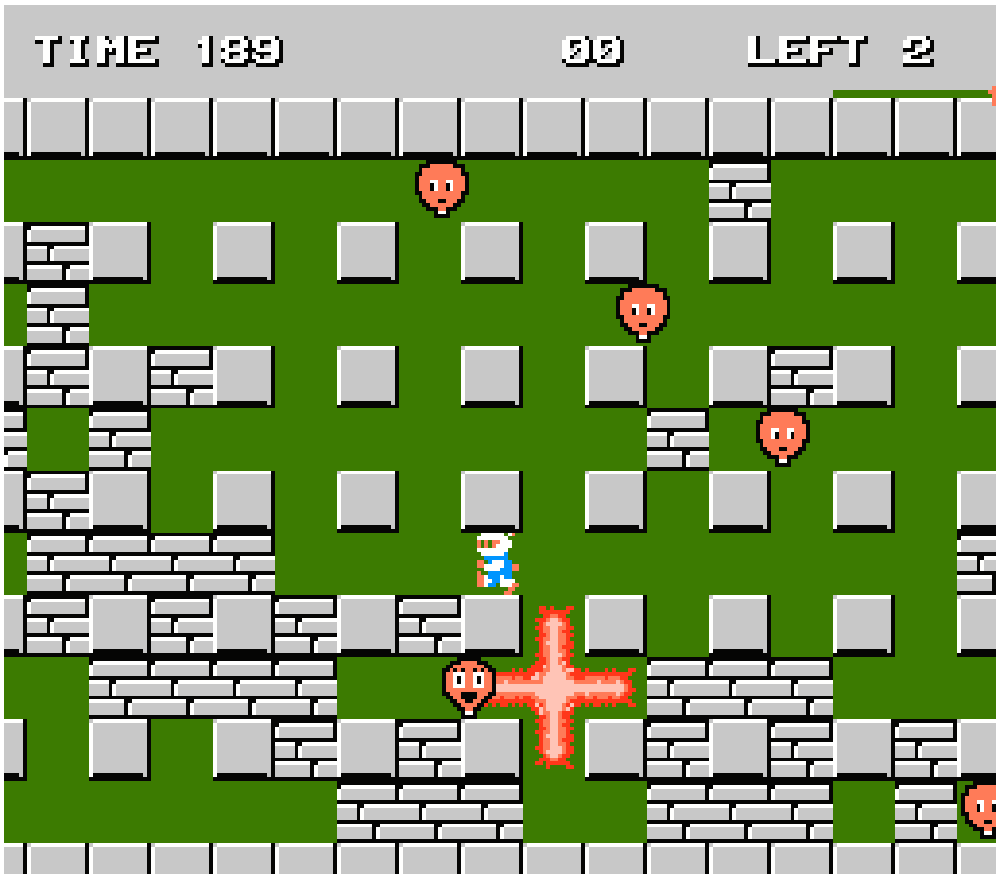

Figure 1: Gameplay of the original Bomberman on NES (1983)

characters, items, maps, abilities, etc. and by creating a mobile version out of it. Despite this long success story, *Bomberman* series did not see any new version released from 2010 to 2016. Its latest version at the moment is called *Super Bomberman R* and is focusing on the mutliplayer gameplay on a really similar way that we are about to do.

In our versions of the game, all the basic elements of the game will be conserved.

- 2D maze where the character can only move up, down, right or left
- Mix of destructible and indestructible walls
- Items dropping off the destroyed walls

The new elements added will mostly be related to the special abilities of the each character. Some examples of those abilities are defusing bombs, building walls, summoning Non-Player Characters (NPCs) and kicking bombs. Great care will be needed to balance the cool down of those special abilities to avoid some characters to be over-powered comparatively to the others. Moreover, we are planning to use bigger maps and smaller camera view that will allow us to use more detailed graphics and to add additional difficulty to the game since a

given user will not necessarily know what is happening on the rest of the map.

We are expecting the gameplay to be fast and to be flexible enough to account for many strategies. Users will be rushed to find the opponents to avoid that they build too much in force. For this part, we are planing to build a skill upgrading system that will introduce some variations depending on the character. Those variations will create some characters that will be stronger in early fights and some others that will only reach their full capacities around the end of the match. Also, the point system will be created in such a way that the only real way to make points will be to tear down the other users whereas breaking walls will only account for small amount of points. These dynamics will necessarily push the opponents to engage the fight faster.

Another interesting aspect of the game will be the accumulation of badges according to the way the users are playing the game. Based on the statistics, the different badges will be unlocked. Down here is a short list of achievements that is currently considered to be included in the game.

Collateral Dommage
Kill 3 adversaries at the same time

Kamikaze
Eliminate everybody on the map (including you) and win the match

Excavator
Destroy 1000 walls

Veteran
Play 100 Matches

Furthermore, all the users of the application will be ranked according to their overall statistics. This overall score will have to take several parameters into consideration to favor active users and distinguish skilled users at the same time.

Due to the simplicity of the gameplay, some efforts will be necessary in order to attract modern users by using smooth graphics and intuitive User Interface (UI). Some mobile versions of this game have already been created. The Figure 2 show one of these that we found particularly nice in terms of the graphical interface. We expect ending up with something that looks somewhat similar to this.

Figure 2: Bomberman mobile in action by Fejer

Market Positioning

Business Opportunities

Nowadays, mobile applications are not seen as something really extraordinary anymore. Their easy accessibility by the end-users joint to the rapidity of their development and deployment have lead to the creation of countless number of them, especially games. It is harder and harder to find new game concept that will be completely different as something that already exists somewhere else. Nevertheless, some games are still able to stand out themselves in this ocean of pastime.

Most of these outstanding games have somewhat the following set of features:

- Easy to learn and use
- Nice graphics and animations
- Can be played for 1 minute or hours
- Have some real challenges involved

- Constantly requests the attention of the user (fast gameplay)

With our concept, we think that most of these elements have been reunited, but having those is not necessarily a direct way to success and the only way to know it will be when the game will have been published.

On the other hand, creating mobile games is still considered as an open market comparatively to console games development where only few giant companies have sufficient money and staff to try their chance.

Overall, the team does not expect to get money out of this project, but, since they are obliged to develop this game, they will make their best to have a nice final product that **may** lead to some returns through advertisements.

Product Position Declaration

[NAME OF THE GAME TO CHANGE] is developed for a very diverse clientele of any age and culture wishing to have some good moments during their spare times independently of their location. The main aspect of the game that we think will make it different from the other games available or variants of the *Bomberman* concept is the real-time interaction between the different users. Instead of trying to create any kind of sophisticated Artificial Intelligence (AI), users will have to compete against each others. Another trilling aspect is the tracking of the user statistics that will allow them to unlock achievements badges and the ranking of the users. We will also develop the game in such a way that the addition of new features such as characters, items and maps will be easy to perform. This modifiability will allow us to keep the active users interested in the game and to have new things to exploit and discover along their play time. We must however keep in mind that the vast majority of our customers will probably be aged from 6 to 18 years old. Hence, the animations should not be to crude while at the same time remains funny. Finally, the game will provide a platform allowing the users to defeat their friends in a bombing skirmish, thus main purpose of this application is recreative.

Functional Requirements

In this chapter, an exhaustive list of all the function requirements will be established. For each requirement, an unique ID, a description and a .

Quality Requirements

COTS Components and Technical Constraints

In this chapter, we will describe the constraints imposed by our choice of COTS. Among the different COTS that will be used for the development of this application, we can find `LibGDX`, `Artemis-odb`, `Android`, and `Google Play Game Services`. For each of these, the rationale behind their selection, the specific interface that we have to implement and an analysis of the constraints they add to the project will be provided.

LibGDX

During early assignments in this course, we got to know the way `LibGDX` [Zechner] can have a stack of screens, and only render and display the top one. This will constrain how we develop the way we change screens, and how we should handle dumping all entities once we drop a screen. This is important to get right as the performance might vary if this is done poorly.

Artemis-odb

Since we want to develop our game using Entity-Component-System (ECS) pattern, we looked at different frameworks to make this easier. Frameworks such as `Artemis-odb` [Papari] will constrain our architecture so we have to make components for our entities, instead of inheriting from other classes. It will also constrain us on how we define components, so we make them compatible with this framework.

Android

One of the mayor challenges for developing a game for Android, is that the version of Android and Java might vary a lot based on what device is playing the game. We imagine we will probably just restrict the game to not work on older devices, to save ourselves some headache with trying to understand why something might not work on older devices. Another thing that will constraint how we make our game, is the difference in resolutions and hardware performance on each device. This means we will have to come up with a way to scale our UI and on-screen elements, instead of just rendering based on a set resolution. This also means we will have to follow strict rules on how to clear unused elements in the game world, so the memory of the device doesn't get filled up.

Google Play Game Services or other solutions

Since we want to have some service that can set up Peer-to-peer (P2P) connections for us, we realize our game architecture needs to include some form of manager or handler that will communicate with this service. Assuming we use some COTS like Google Play Game Services (GPGS) [Google], we will most likely have to follow some very strict rules that allow P2P setups via this service. These sort of interactions will happen when a user sets up a room, when a user want to join that room and when users are restricted to join certain rooms.

Issues

Before starting this project, we had a few ideas of what issues might arise during this project.

Graphics

We have no graphical experience within the group, and as we look at other mobile games, it becomes apparent that a simple graphical look is something we wish to achieve during this project. This is to make the game more appealing to the users.

peer-to-peer connection

Since we rely on peer-to-peer to transfer the data between devices, we might encounter problems with devices that are unstable. How we will we handle bad devices potentially crashing game sessions and ruining the experience for other players.

Changes

This section records the changes brought to the current document since its creation.

Date	Version	Description
2018/02/21	**1.0**	Initial draft

Glossary

ASR Requirement that will have a profound effect on the architecture—that is, the architecture might well be dramatically different in the absence of such a requirement.
[3, p.291].

Cool-Down The minimum length of time that the player needs to wait after using an ability or item before it can be used again.
[1, cool-down].

NPC Taken from the world of pen-and-ink role-playing games, an NPC is a character encountered in an RKJ who is not controlled by the user.
[2, p.38].

Bibliography

Dictionary definitions. URL http://www.yourdictionary.com/.

Lexicon A to Z: NPC (Nonplayer Character). *The Next Generation Magazine*, (15), Mar 1996.

Len Bass. *Software architecture in practice*. Addison-Wesley, Upper Saddle River, NJ, 2013. ISBN 978-0-321-81573-6.

Kenneth Fejer. Pixel art. URL http://www.kennethfejer.com/pixelart.html.

Google. Google play game services. URL https://developers.google.com/games/services/. [Online; accessed 19-February-2018].

Craig Larman. *Applying UML and patterns : an introduction to object-oriented analysis and design and iterative development*. Prentice Hall PTR, Upper Saddle River, N.J, 2005. ISBN 0131489062.

Adrian Papari. artemis-odb wiki. URL https://github.com/junkdog/artemis-odb/wiki. [Online; accessed 23-February-2018].

Wikipedia contributors. Bomberman (1983 video game) — wikipedia, the free encyclopedia, 2018. URL https://en.wikipedia.org/w/index.php?title=Bomberman_(1983_video_game). [Online; accessed 25-February-2018].

Mario Zechner. Libgdx -goals and features. URL https://libgdx.badlogicgames.com/features.html. [Online; accessed 25-February-2018].