

Requirements and Analysis Document for Meteor Defense

Table of Contents

Version: 1.0

Date: 2014-03-19

Author: Jacob Gideflod, Emma Lindholm, Simon Nielsen, Andreas Pegelow

This version overrides all previous versions.

1 Introduction

This section gives a brief overview of the project.

1.1 Purpose of application

The purpose of our application is to entertain people with a simple game they can play whenever they want from their android smart phone devices. The main target is young people who are looking for a way to be entertained during microbreaks.

1.2 General characteristics of application

This application is a singleplayer arcade-style game for android where you progress through several levels. Since this is a mobile application it's meant to be used anywhere you are, whether it's in your bedroom in silence or on the bus.

1.3 Scope of application

The features of the game includes:

- Shooting down meteors falling from the sky
- Progressing through multiple levels of varying difficulty
- Many different kinds of meteors
- Buying new weapon from an armory with in-game currency
- Buying upgrades to your weapon
- Selling your weapon

Features not to be included:

- Multiplayer
- In-game purchases for real money

1.4 Objectives and success criteria of the project

For this application to be considered a success it needs to be released on Google Play by the end of the course with the features specified in 1.3 Scope of application.

1.5 Definitions, acronyms and abbreviations

- Android – An operating system for smartphone devices.
- LibGDX – A library for creating applications for desktop, android, iOS and the web with the same code.
- Arcade games – Early simple games running on big machines such as space invaders.
- Google Play – The marketplace for android devices where you can buy and download applications and more.
- Continent – Representing a “game world” containing many different cities.
- City – Representing a “level”.
- Armory – The ingame place where you handle your weapons.
- Weapon – A utility to help defend the city from incoming meteors.
- Meteor – An object falling from the sky that will hurt the city if it hits it.

2 Requirements

2.1 Functional requirements

The player should be able to:

1. Browse between Continents and Cities (levels)
 1. Choose a level
2. Play a level
 1. Choose weapon as you play
 2. Fire the chosen weapon
 3. Hit and destroy meteors
 4. Lose if the city is destroyed
3. Get a star rating 1-3, score and coins depending on how well the player did
4. Manage weapons in armory
 1. Buy weapons
 2. Upgrade weapons
 3. Sell weapons
5. Change settings for sounds and music
6. Exit the application

2.2 Non-functional requirements

2.2.1 Usability

Great usability is extremely important for this application since it's meant for mobile devices where people have very low tolerance for bad applications. We will try to achieve this by utilising clear entry points and distinct icons. Tests with a couple of people from the target audience will be conducted to ensure that this goal has been completed.

2.2.2 Reliability

N/A.

2.2.3 Performance

Since this application will be running on mobile devices with very different performance it has to be efficient regarding memory usage, power consumption, etc. Latency will not be tolerated. To reach this goal among others we will use a library called "libGDX" which will help with the graphics rendering and other maintenance issues.

2.2.4 Supportability

The application should be able to run on all android smartphones from version 2.2 and up with varying screen sizes.

2.2.5 Implementation

Because this application will be running on android smartphones it will be coded in Java with android implementation. To make this easier the library libGDX will be used which helps with the translation to android and the graphics.

2.2.6 Packaging and installation

The application will be downloaded and installed from the Google Play store.

2.2.7 Legal

N/A.

2.3 Application models

2.3.1 Use case model

The following list contains all the use cases so far:

- Play
- Shoot
- Hit Meteor
- Fire Unimmediate Effect Special Weapon
- Fire Immediate Effect Special Weapon
- Buy Weapon

The UML can be found in the appendix.

2.3.2 Use cases priority

The following list is ranked with priority where 1 has the highest priority.

1. Shoot
2. Hit Meteor
3. Play
4. Fire Unimmediate Effect Special Weapon
5. Fire Immediate Effect Special Weapon
6. Buy Weapon

2.3.3 Domain model

See Appendix.

2.3.4 User interface

Application will adapt to different screen sizes but fits the aspect ratio 16:9. The screen can't however be tilted to landscape mode since this does not suit the game play. The interface presumes that the device has a functional back button, this function is not provided in any other way. The user cannot customize the interface.

2.4 References

Arcade game: http://en.wikipedia.org/wiki/Arcade_game

LibGDX: <http://libgdx.badlogicgames.com/>

APPENDIX

Use cases overview

Use cases text

Use Case: Play

Summary: From Start-screen, choosing to play which leads you to "level-choosing"-view

Priority: high

Extends: N/A

Includes: N/A

Participators: User

Normal flow of events

Actor	System
1. Press “Play”-button	
2.	Show “Choose Continent”-view
3. Press current continent	
4.	Show “Choose City”-view
5. Press current city	
4.	Start game

Alternate flow

Actor	System
3.1 Swipe	
3.2	Show next continent in carousel

Actor	System
5.1 Swipe	
5.2	Show next city in carousel

Use Case: Shoot

Summary: The use-case play has been completed.. We are now using the default weapon.

Priority: high

Extends: N/A

Includes: N/A

Participants: User

Normal flow of events

Actor	System
1. Player taps the screen	
2.	Changes direction of the cannon barrel.
3.	Draws a projectile on the screen that follows a straight line.
4.	Cooldown bar shows up.

Use-case: Hits meteor

Alternative flow from Shoot

Actor	System
3.1	Projectile hits meteor. If it's a standard meteor, meteor is destroyed with appropriate animation. If it's a bigger one it gets smaller, approaching the standard meteor size.
3.2	Update meteor score

Use Case: FireImmediateEffectSpecialWeapon

Summary: The use-case play has been completed.. User decides to use a special weapon with an immediate effect during game

Priority: mid

Extends: N/A

Includes: N/A

Participators: User

Normal flow of events

Actor	System
1. Player taps the symbol of the special weapon	
2.	The appropriate action of the special weapon is preformed
3.	Cooldown bar shows up.

Use Case: FireUnimmediateEffectSpecialWeapon

Summary: The use-case play has been completed.. User decides to use a special weapon, which requires a gesture to be performed, during game

Priority: mid

Extends: N/A

Includes: N/A

Participators: User

Normal flow of events

Actor	System
-------	--------

1. Player taps the symbol of the special weapon	
2.	Weapon symbol is shown as active
3. Player performs the required gesture for the weapon	
4.	The appropriate action of the special weapon is preformed
5.	Cooldown bar shows up.

Use Case: BuyWeapon

Summary: Assumes that the application shows the Continent- or City-chooser view. User decides to buy a special weapon.

Priority: mid

Extends: N/A

Includes: N/A

Participators: User

Normal flow of events

Actor	System
1. Player taps the armory button	
2.	Armory-view is shown
3. Player presses a locked weapon symbol	
4.	The detailed view of the weapon is shown
5. Player presses Buy button	
6.	Buy button is replaced with Sell button
7.	Lock icon is removed from weapon
8.	Upgrade button is enabled

Alternate flow of BuyWeapon 5

Actor	System
5.1	User does not have enough points in its wallet to buy the chosen weapon, Buy button is disabled

UML

