Requirements and Analysis Document

Version 0.1
Date: 22/3-16

Group 7

Andrea Buchholz
Rebecca Finne
Maija Happonen
Lisa Larsson

This version overrides all previous versions.

1 Introduction

This section gives a brief overview of the project.

1.1 Purpose of application

The aim of the project is to create a computer based generic version of a 2D, top-down action game. The game will have similar characteristics as Pokemon Silver [1], but instead of the aim of the game being to level, the project will focus on a more simple gathering of articles through missions. While the game will be created generically, a demo will be created - illustrating Chalmers campus and the collectable articles will be the mascot of each programme at chalmers.

For a wider set of game rules and explanations of definitions, see references.

1.2 General characteristics of application

The application will be a standalone, single-player, desktop application for Windows/Mac/Linux platforms.

Since the application is single player the game will be solely controlled by the main player.

1.3 Scope of application

The application is a single player game. The game will include interactions with a set amount of characters with three different types of information to give. The information will be given in a set order depending on chosen type of information. If the character is a mascot, and the player is unsuccessful in answering the question - the mascot will move in a set pattern.

The game can be paused and un-paused - but if the game is exited during a game, no progress is saved.

1.4 Objectives and success criteria of the project

1. The objectives in this game is to go to all the implemented facilities and collect the mascots for each programme.
2. The player will be able to go to all the programmes facilities on Chalmers' campus Johanneberg.

1.5 Definitions, acronyms and abbreviations

- GUI - A graphical user interface
- Java - The programming language used in this project [2].
- Emil/Emilia - The name for Chalmers-students of different genders.
- Mascot - A figure that belongs to a certain programme
- World - The map in which the player interacts (there are many in each game, for different facilities).
- IntelliJ - The IDE (Integrated Development Environment) used for developing the application.
- TileMap - The development environment for the interface.
- Progress bar - The bar in the game that shows the progress of the game.

2 Requirements

In this section we specify all requirements

2.1 Functional requirements

Create a list of high level functions here (from the use cases).

1. Start a new game
2. Choose character (Emil/Emilia)
3. Move around in the world. While moving he or she can:
    a) Interact with human
    b) Enter new worlds (i.e when going through a door)
    c) Catch mascots
4. Progress bar which shows the progress
5. Exit the application

2.2 Non-functional requirements

Possible NA (not applicable).

2.2.1 Usability

The language used will be english - as for the game to be available to as many people as possible. The information will be displayed in a fashion which is easily grasped quickly. There will be information displayed in the beginning of the game as to show the user what the goal is and how to accomplish the tasks. The same goes for when meeting the first mascot. The usability will be tested on friends and family throughout the development process as to communicate the right amount of information.

2.2.2 Reliability

NA

2.2.3 Performance

Any actions initiated by a player should not exceed a 2 sec. response time in worst case.

2.2.4 Supportability

The application will be able to work on different computers. The application will be able to run on a screen with at least 1366x768 resolution.

The application will not be a networked based game.

The application will be tested both through on-going GUI testing throughout the development, but also through JUnit-testing of specific methods.

Since the specific game is a demo, the main application should be able to work with any given map (world) - and this will also be tested.

2.2.5 Implementation

To be able to run the application a JRE have to be installed - which in our case will be IntelliJ.

2.2.6 Packaging and installation

The application will be able to download from git. A README-file will be written to assist in this.

2.2.7 Legal

This is not a commercial product. We will use trademarked figures and the game is overall a lot like Pokémon Silver, this may give some legal issues if this was a commercial product.

2.3 Application models

2.3.1 Use case model

The use cases that are further defined are:

- Move
- Interact with human
- Enter new world
- Catch mascot

- Progress bar

A closer look at each of these can be found in Appendix XX

2.3.2 Use cases priority

1. Move
2. Enter new world
3. Fight/answer question
4. Catch mascot
5. Progress bar

2.3.3 Domain model

UML, possible some text.
See Appendix XX

2.3.4 User interface

The application will be able to integrate most type of worlds with the same set of environments and ground rules, as the maps will be created in TileMap [3].
For images of the UI - see Appendix XX

2.4 References

1] Wikipedia, retrieved 2016-04-11:
https://en.wikipedia.org/wiki/Pok%C3%A9mon_Gold_and_Silver

2] Wikipedia, retrieved 2016-04-11:
https://en.wikipedia.org/wiki/Java_(programming_language)

3] Tiled, retrieved 2016-04-11http://www.mapeditor.org/

APPENDIX

GUI

(Picture of GUI here)

Domain model

(UML class diagram here)

Use case model

(UML use case diagram here)

Use cases

(Use cases texts here (five), see Course page for now)