CZECH TECHNICAL UNIVERSITY IN PRAGUE
FACULTY OF INFORMATION TECHNOLOGY

ASSIGNMENT OF BACHELOR'S THESIS

Title:	RPG game with augmented reality features - server part
Student:	Jakub Čech
Supervisor:	Ing. Miroslav Balík, Ph.D.
Study Programme:	Informatics
Study Branch:	Software Engineering
Department:	Department of Software Engineering
Validity:	Until the end of winter semester 2018/19

Instructions

The aim of the thesis is to specify, design, and implement a functional prototype of the server part of a RPG game with features of augmented reality (AR).
1. Create a story line and rules for the game. Consider intensive usage of geolocation and AR features.
2. Formalize the following requirements for the implementation of the server part:
 - the data layer consists of a database engine and a caching,
 - users can use their Google accounts to login and play,
 - the client part provides an API,
 - the communication between client and server parts of the application must be secure.
3. Design the server part of the game.
4. Design a suitable front-end part for server administration.
5. Discuss and choose a suitable implementation platform and related technologies (databases etc.).
6. Implement the functional prototype, document it, and perform suitable testing.
7. Tightly cooperate with Tomáš Zahálka who works on the client part.

References

Will be provided by the supervisor.

Ing. Michal Valenta, Ph.D.
Head of Department

prof. Ing. Pavel Tvrdík, CSc.
Dean

Prague March 6, 2017

Czech Technical University in Prague
Faculty of Information Technology
Department of Software Engineering

Bachelor's thesis

Location-based Role Playing Game

Jakub Čech

Supervisor: Ing. Miroslav Balík, Ph.D.

2nd June 2017

Acknowledgements

I would like to thank myself for doing this. I am an awesome and humble person. With great power comes great responsibility and no one else is as good or worthy as I am to be thanked. Ave Kuba!

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as school work under the provisions of Article 60(1) of the Act.

In Prague on 2nd June 2017 ……

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Čech, Jakub. *Location-based Role Playing Game*. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2017.

Abstrakt

Hvězdy jsou krásné, protože je na nich květina, kterou není vidět. Poušť je krásná právě tím, že někde skrývá studnu. Ať je to dům, hvězdy nebo poušť, to, co je dělá krásnými, je neviditelné!

Klíčová slova #deep, #thoughtoftheday, #follow4follow

Abstract

Place the 2 cups of crushed ice into a cocktail shaker. Pour the rum, lime juice, and simple syrup over the ice, cover, and shake well. Remove the ice from your serving glass and strain the drink into it. Serve immediately.

Keywords Daiquiri, Coctail, Rum, Cuba

Contents

List of Figures

Introduction

The world of mobile devices is quickly evolving. Smartphones and tablets are becoming more and more powerful and not only in terms of computational power and available memory. Mobile devices nowadays are packed with various sensors. It is possible to integrate data from GPS (Global Positioning System), accelerometer, gyroscope, magnetometer, and camera to quickly determine device's position and orientation. This opens us doors to augmented reality. In my bachelor thesis, I, in cooperation with Tomáš Zahálka, will create a mobile game for Android which will utilize elements of augmented reality. The game is set in a fantasy world and its genre is RPG (Role Playing Game).

Chapter 1

Project overview

1.1 Project timeline

1. **Analysis** – finished by the end of February
2. **Design** – finished by the end of March
3. **Implementation** – finished in the middle of May
4. **Testing** – begins with implementation and will be finished by the end of May
5. **Releases**
   a) **Pre-alpha** - 16. 4. 2017
   b) **Alpha** - 30. 4. 2017
   c) **Beta** - 14. 5. 2017
   d) **Final** - 1. 6. 2017

1.2 Specification and features

The content of this section is temporary.

1. Bare minimum
   - Zobrazení pohybu hráče po mapě
   - Generování monster
   - Generování budov
   - Interakce s herními objekty
   - Ukladaní postupu, stavu postavy

- Inventář
- Statistiky

2. Should be there
    - Mise, úkoly
    - Multiplayer
    - Skill systém
    - Dialogy
3. Nice to have
    - Navigace k objectives
    - Trading
    - Tutorial
    - Částečná offline podpora

Chapter 2

Analysis

2.1 Similar solutions

2.1.1 Parallel Kingdom - Age of Ascension

This game was on market for 8 years (2008-2016). Parallel Kingdom is a closest solution to ours.

"Parallel Kingdom is a mobile, location based, massively multiplayer game that uses GPS location and Google Maps to place users in a virtual world. Parallel Kingdom is the first location based RPG for the iOS and Android platforms. The game is set in a virtual world or "Parallel Kingdom" where users claim their territories based on their GPS location or by making friends who invite them to travel to new places. Parallel Kingdom is a freemium game and utilizes a virtual goods revenue model."

2.1.2 Ingress

Developed by Niantic, which was then part of Google, this game was released in 2013 for Android and in 2014 for iOS.[4] It is a location based, massively multiplayer game. A player have to choose one of the two factions, Enlightened or Resistance, and then as a part of his team capture regions of the game map. A faith of each faction relies on players' cooperation. Thanks to that players meet in real life and coordinate their actions.

Ingress was the first very successful augmented reality game with more than 10 000 000 installs.

2.1.3 Pokémon GO

After its success with Ingress, Niantic started working on a new game Pokemon GO. Once released, the game became incredible hit. Even though the game faced many problems during its launch, mainly caused by the unexpected

success and more active users than Pokémon GO was able to handle, in the first 80 days Pokémon GO reached about 550 millions downloads and earned about $470 million.

The game is very similar to Ingress and uses the same crowd-sourced geographical data.

2.2 Use Cases

2.2.1 Use case diagram

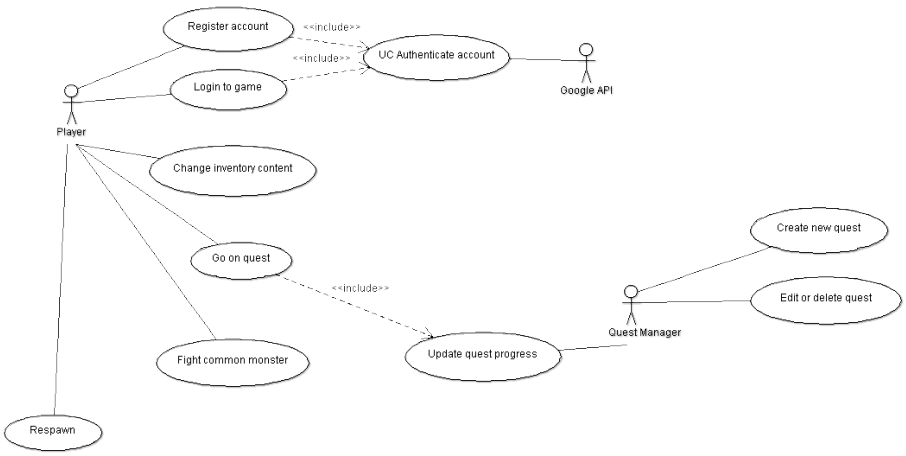

Figure 2.1: Use case diagram

2.2.2 Use case descriptions

The following section refer to the use-cases introduced in Figure 2.1.

1. **Register account**

2. **Login to the game**

3. **Change the inventory content**

4. **Go on a quest**

5. **Fight a common monster**

6. **Respawn**

7. **Update a quest progress**

8. **Create a new quest**

9. **Edit or delete a quest**

2.3 Requirements

2.3.1 Functional Requirements

2.3.1.1 Rules

1. **The server must provide API to clients**
   The key requirement for the server is to allow receiving HTTP(S) requests. When processed, the server responds in JSON format.

2. **The player's character has attributes**
   The character has a set of attributes, including health, experience, level, and owned gold. The maximum health increases with level. The experience is rewarded after certain actions, e.g. after killing a monster. The gold is primary in-game currency.

3. **A player can own items**
   A player has an inventory which can contain various types of items. The item can be for example sword, potion, armor etc.

4. **A game object has a type and inherits all its properties**
   The type of game object specifies allowed actions, its attributes, default name and description.

5. **A game object can be a monster**
   The monster can be killed but can also inflict damage to the player. It has its own inventory and there's a reward for killing the monster in a form of gold and experience.

6. **A game object can be a shop**
   The shop can contain several items with specified price.

7. **A game object can be an item**
   The item can be one of the many objects useful to a player. Examples of the items are health potion, sword, armor, necklace and similar.

8. **Each game object has its own inventory**
   The inventory contains other game objects. Example of this requirement is a monster with a potion and a sword in its inventory; both will be given to the player who kills the monster.

9. **The server stores a list of predefined locations**
   Real geographic locations for the game objects are stored on the server to ensure every player has the same location-object pair.

10. **A game object can independently exist at many locations**
    This requirement aims to help maintain the game objects efficiently by administrators. It allows creating small set of abstract game objects with predefined inventories and other attributes.

11. **If a player kills a monster at a location, the monster will be hidden for a period**
    To prevent the player from killing the same monster continuously without a need of moving somewhere else, the location should be hidden for a certain period after the kill.

12. **The server should provide API for administration**
    Such API will be used to manage locations, create and edit game objects or to assign a game object to some locations. It is necessary to protect the administration endpoints from unauthorized access.

13. **The server must persist player's profile between sessions**
    All the player's attributes, his inventory and equipment must be stored between sessions. Player will continue from the state in which he ended.

2.3.1.2 Features

1. **A player registers and logs in the game using Google account**
   For the player's convenience, a Google account is required to play. The server does not have to store or handle any password. Most of the authentication process is delegated to Google servers.

2. **A client can get nearby game objects based on his location**
   The major feature of this application is being location-aware. Server must provide a method to retrieve game locations near the requested latitude and longitude. The "near area" should be circular, defined by its radius; the size have to be carefully chosen so it's big enough to cover client's maps but also small to limit the response size and the spatial search overhead.

3. **A player can kill a monster**
   When the player wins the fight, he will be rewarded by experience and gold.

4. **A player can be killed be a monster**
   The player can lose health during the fight with a monster. If the health reaches zero, the player dies and loses an amount of gold based on his level.

5. **A player can collect items from the monster he killed**
   When the player wins the fight, he's offered to collect items from the monster's inventory. He can chose any subset of these items.

6. **A player can equip an item**
   Many items in the game can be equipped. These items have predefined equipment slot, for example a sword have to be held in hand, an armor worn on chest, shoes put on feet and so on.

7. **A player can buy object from a shop**
   Gold can be exchanged for various items in shops.

8. **A player can use an item from his inventory**
   Some of the items in the game are consumables. When used, an action defined by the item is executed. For example a health potion heals the player.

9. **A player can purchase in-app product**
   The application allow a user to exchange real-life currency for the in-game one. The server should verify such purchase and add the currency to his profile.

2.3.2 Non-functional

1. **The data layer consists of a database engine and a caching**
   Game Server never accesses database directly. Data are retrieved from the database upon request and then cached. Upper layers communicate exclusively with cache.

2. **The server provides an API for client**
   The API support at least following:

   a) Retrieve nearby game locations

   b) Create player

   c) Update player's data (inventory, experience, resources, quests progress etc.)

   d) Provide information about a quest

3. **The communication between client and server parts of the application must be secure**
   All data sent from and to a client has to be encrypted.

4. **Client can only connect to a Connection Server**
   Several Connection Servers exist to prevent a bottle-neck. Client selects the Connection Server by an algorithm. Client does not have an access to any other part of the server.

2.3.3 System and Interface

1. **System uses Java 8 SE as an execution environment**

2. **Operating system for the server is Debian 8**

3. **Database engine is MySQL**

4. **Cache engine is Redis**

2.4 Technology

2.4.1 Frameworks

Chapter 3

Design

3.1 Server-Client communication

How the data are exchanged between client and server. Will contain format, technologies and protocol specification.

3.2 Security

Basic methods for verification of legitimacy of the data. How the server API will be secured.

3.3 Architecture

Diagrams and overall architecture of the server belongs here.

3.4 Deployment Diagram

3.5 Database Model

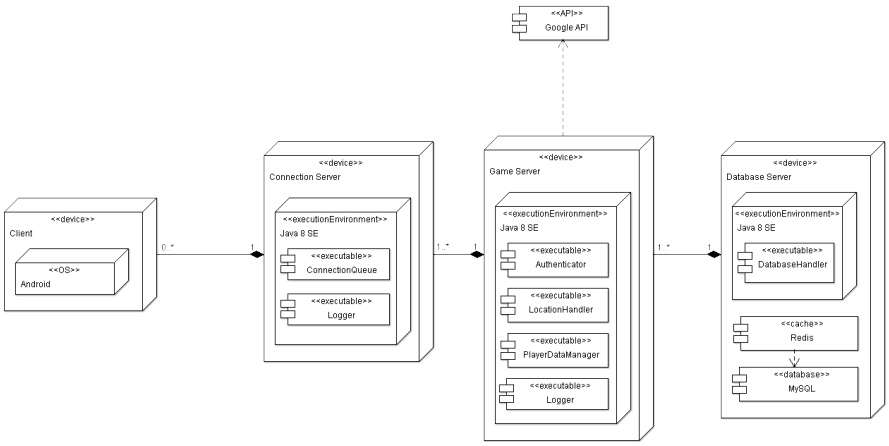

Figure 3.1: Deployment diagram for the game

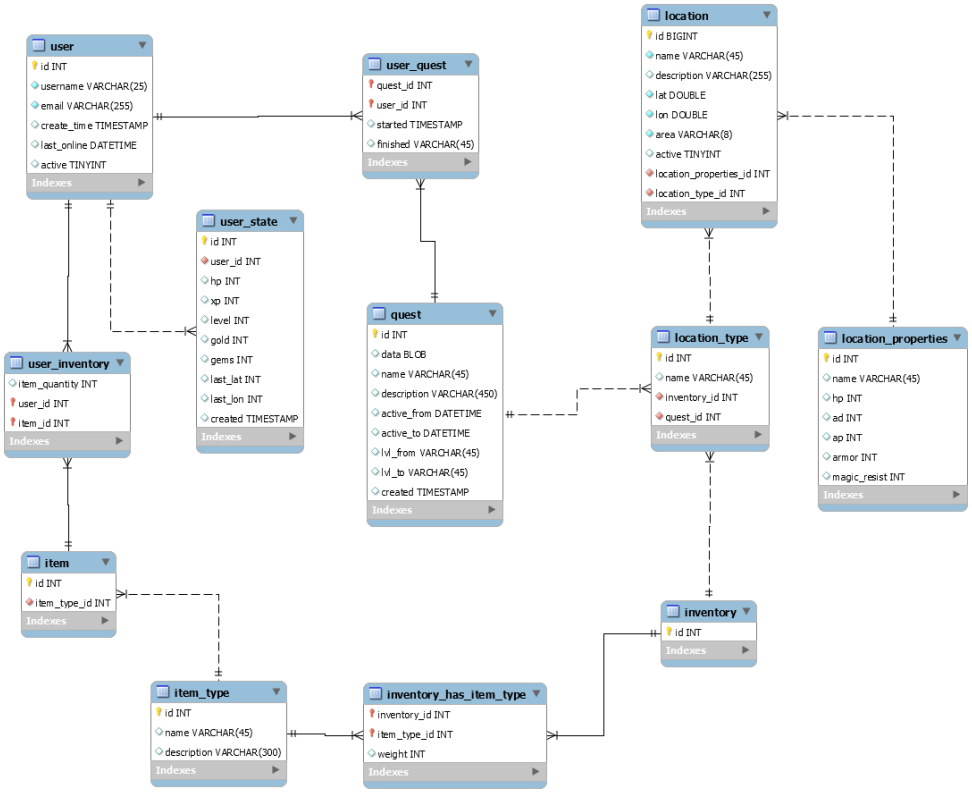

Figure 3.2: Diagram of the database structure implemented on the server

Chapter 4

Geo-data mining

Description of how the geographical data for the monsters and building will be obtained.

4.1 Methodology

4.2 Output

Chapter 5

Implementation

Implementation-specific aspects of the game. Will include a description interesting parts of the server.

Chapter 6

Deployment

Chapter 7

Testing

Results

Appendix A

Acronyms

GUI Graphical user interface

XML Extensible markup language

Appendix B

Contents of enclosed CD

- `readme.txt` ..... the file with CD contents description
- `exe` ..... the directory with executables
- `src` ..... the directory of source codes
  - `wbdcm` ..... implementation sources
  - `thesis` ..... the directory of LaTeX source codes of the thesis
- `text` ..... the thesis text directory
  - `thesis.pdf` ..... the thesis text in PDF format
  - `thesis.ps` ..... the thesis text in PS format