City simulation

**User Requirement Specification**

Dimo Popov

Peter Hadzhiev

Zhicheng Yu

Nikolay Kolev

Xiangkuan Peng

[**Document Revision History**](#_gjdgxs) **3**

[**Document Introduction**](#_30j0zll) **4**

[Purpose of This Document](#_1fob9te) 4

[Overview](#_3znysh7) 4

[**Requirements**](#_tyjcwt) **5**

[Functional Requirements](#_3dy6vkm) 5

[Non Functional requirement](#_2s8eyo1) 6

[**Use Cases**](#_17dp8vu) **7**

[Use case 1.1](#_3rdcrjn) 7

[Use case 1.2](#_1ci93xb) 7

[Use case 1.3](#_3whwml4) 8

[Use case 1.4](#_2bn6wsx) 8

[Use case 1.5](#_qsh70q) 8

[Use case 1.6](#_3as4poj) 9

[Use case 1.7](#_1pxezwc) 9

[**GUI**](#_2jxsxqh) **11**

[**Other**](#_2xcytpi) **12**

# Document Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Status | Description | Date | Name |
| V1.0 | C | Create initial URS | 18/2/2018 | Zhicheng Yu |
| V2.0 | A | Add use case | 8/3/2018 | All |
| V3.0 | M | Modify use case | 15/3/2018 | All |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

\*Change status: C means Create, A means Add, M means Modify, D means Delete.

\* The version number of the document is compiled from 0.1, and the document version number starts from 1.0. When a small change to the document, the version number to 0.1 advanced; large change version number to 1.0 advanced.

# Document Introduction

## Purpose of This Document

This document is a User Requirement Document for use in a project. It describes the requirements specification for the city simulation application which is intended to assist the relevant management or technical staff, the client to understand the requirements needed for the application.

## Overview

This document describes general software constraints of the application. The main focuses of this document is on the specific requirements for the application and followed by some use cases that will be implemented in the application.

## 

# **Requirements**

## Functional Requirements

Must Require:

1.1 Ability to place an object on the map.

1.2 Specifying the size of the map.

1.3 Ability to move the camera.

Could Require

1.1)

1.2) Ability to selecting and spectating the status of a building object or a vehicle object in the map.

1.3) It could have some explanatory note.

1.5) User saving and loading the map.

1.6) Ability to detecting collision when placing a building object.

## Non Functional requirement

Must Require

1.1 Automatically generating a grid on which can be placed objects.

1.2 The angel of the view of the camera should be changed according to the mouse.

Should Require

1.1) The application should be clear for user.

1.2) The source code should be kept clear for future use.

**Use Cases**

## **Use case 1.1**

**Name**: Adding an object  
**Actor**: A user (employee in the electric company).  
**Pre-condition**: /

**MSS**:

1. User selects an object in the toolbox.
2. User clicks on a fixed location for the object on the grid.
3. System sets the object in selected location on the grid.

**Extension:**

**2a**. If user tries to click on a location with an existing object, system will not allow the placement and will not create the object.

## **Use case 1.2**

**Name**: Select a created object  
**Actor**: user  
**Pre-condition**: /

**MSS**:

1. User clicks somewhere on the map.
2. System checks if there is an object and there is.
3. System selects that object.
4. System shows the info of the selected object.

**Extension:**

**2a.** If there is no object where user is clicking, end of use case.

## **Use case 1.3**

**Name**: Delete selected object from map  
**Actor**: user  
**Pre-condition**: Use case 1.2 Select a created object

**MSS**:

1. User clicks somewhere on the map.
2. System checks if there is an object and there is.
3. System selects that object.
4. System shows configuration panel of the selected object.
5. User clicks delete button on panel
6. System deletes the object from the map
7. System closes configuration panel

**Extension:**

**2a.** If there is no object where user is clicking, end of use case.

**Use case 1.4**

**Name:** Creating a grid

**Actor:** user

**Pre-condition:** application running

**MSS:**

1.User types the parameter for axle X

2.user types the parameter for axle Z

3.User pressing on create button.

3.System checks if the the two dimension are both integers and both negative/positive.

4.System creates grid x by z(example 2x2)

3.a If one of the dimension is positive and the other one negative exception is thrown.

3.b if the input of the user is not a integer is rounded up.

**Use case 1.5**

**Name:** Moving the camera

**Actor:** User

**Pre-condition:** use case 1.4

MSS:

1.User moves the pointer of the mouse to the edge window.

2.System moves the point of the camera toward the direction of the pointer

1.a if the pointer of the mouse haven’t gone beyond the edge of the window, camera doesn’t move

# GUI

The Graphical User Interface will be presented in 2D model. The application will contains several sections: Toolbox, Grid Area and Menu. They will always be provided in the interface. Toolbox will provide some items that can be interacted and used to fill in the Grid Area. The Grid Area will be the stage where the object being placed and displayed.

The toolbox includes the images as well as the names on the button as components able to be added into the Grid Area, in order to reduce any confusion.The menu button is located below of the grid as it is separates itself from the toolbox.

However for the end of the iteration 1 the GUI of the application will contain fields to input a value of the dimension of the map.

# **Other**

Development Tool

This application will be developed using C#

Language Choices

This application will be presented in English

Future choices

Possible changes for the use case