

Frankfurt University of Applied Sciences

Escape the Room

Project Report

Participants:

Purit Thong-on 1106398

Piyapat Russamitinakornkul 1106291

Instructor:

Professor Doctor Eicke Godehardt

This project report is for Software Engineering Design Exercise Course, Summer Semester 2015

2015

Tong

Project Report

1/1/2015

Table of Contents

**Contents 1**

**Chapter 1: Introduction 3**

* Objective 4
* Rules 4
* Game Theme 5
* Basic Game Details 5

**Chapter 2: Development Tools 7**

* Scheduling 8
* Designing Diagrams 8
* SCM Server 8
* Development Process 9
* Programming Language 9
* IDE 10
* Documents and Reports10

**Chapter 3: Project Estimation 11**

* Activity Schedule 12

**Chapter 4: Use-Case Diagram 13**



*Chapter 1:*

*Introduction*

Introduction

# Objective of "Escape the Room"

“Escape the Room” is a game where the player is situated in a small room with the door locked from the outside. The player has to find a key to get out of the room. A key may be hiding somewhere in the room.

# Rules:

* A Player can move the camera angle around the room, which are flexible enough for exploring the whole room.
* Each room contains only ONE exit door.
* Rooms are decorated with many different types of furniture. Some of the furniture can be interacted with and will contain items, including a key and safe containing it.
* The key to unlock the exit door and exit the room is sealed within a securely locked safe.
* The player will be provided an inventory to store all picked up items.
* The player needs to find a password to open the safe and get the key. The password can only be obtained when a player has solved various puzzles and interact with many different items hidden in the room.
* Puzzles and items for finding the password to unlock the safe are chained together. A player starts solving one puzzle using items to receive a clue to unlock next puzzle until the player obtains a complete password to unlock the safe.
* Items can interact with each other or other components in a room to trigger or solve a puzzle, or even obtaining a password for opening a safe.
* The game will end when a player can successfully exit the room.

# Game Theme:

* A category of the game is “Puzzle”, and the theme is mystery and horror. Nevertheless, it is not completely a ghost game. The atmosphere of the game is meant to be spooky and uncomfortable.
* The soundtrack in this game will provide a mysterious feeling. Different tracks will be played during gameplay. The playlist will repeat itself once every track has been played.

# Basic Game Details:

* Each item in the game has got a name attached to it. Player can see this when the mouse is hovered over the item.
* When any item has been clicked, a displaying screen pops-up displaying information about the item, including name, picture and description which may be a clue to solve a puzzle.
* The player can rotate 4 directions: up, down, left, and right.



(This page is intentionally left blank)

*Chapter 2:*

*Development Tools*

Developing Tools

# Scheduling

## Microsoft Office Excel:

Since all team members have quite a lot of experience using this program, a scheduling table, activity timeline, and the activity diagram, will be done by using an excel spreadsheet.

For activity timeline, a simple calendar will be used to fill in the details of what task will be done each day, and by whom.

Another spreadsheet will divide cells into an equal size for drawing an activity diagram.

# Designing Diagrams

## Magic Draw 18.1:

This program is required for drawing software design diagrams. The software itself can also generate basic codes for the diagram. In addition, the interface of the program is not too sophisticated to understand, and the program is also simple and easy to use.

# SCM Server

## GitHub:

Since group members of this project are very familiar with using GitHub for program version control, and all of them have their own accounts in this service. Therefore, the project will be created on the repository of this service.

The framework is that a task is split to every team member. Each member pushes his/her work to the repository. Other members can pull the work from a repository.

# Development Process

## Spiral Model:

In general, the game is always harder as the player progresses. A requirement of earlier stages can change if the current designing stage will be too hard to be played, or impossible to beat. When developing a game, balance of hard and easy stages can be very crucial. Therefore, as the player solves puzzles in the game, each puzzle has to be harder, but containing feasible solutions.

When harder puzzles are designed later, the earlier one can be modified to decrease the difficulty of the current designing puzzle, and to make future harder puzzles feasible.

In conclusion, the spiral model features revising requirements after the prototype is created for multiple times, which is important for maintaining difficulty in a game to be harder either constantly or unstably as the game progresses.

However, there will be no risk analysis stage. Therefore, the upper right section of the spiral will be skipped.

# Programming Language

## Java:

The game is intended to be made available on computer of all platforms which can support Java. Java is a complete object-oriented language and is very easy to maintain. In addition, Java features a package called “swing”.

“swing” is a Java package which contains its own graphic user interface (GUI). This package contains its own margins and layout, which allows a GUI design to look all the same regardless operating systems. This will maintain a consistent game layout throughout every platform.

# IDE

## Eclipse Kepler:

Eclipse is one the most popular Integrated Development Environment for developing software with Java language. In addition, the IDE itself is easy to install and use.

# Documentations and Reports

## Microsoft Office Word (2007 or newer)

All of project participants have a lot of experience in using Microsoft Office. In addition, all of their computers have this software installed. Therefore, it is much easier to use a program which everyone has in common and can function with this report file.

## Adobe Acrobat (PDF)

This program is mostly contained in every computer nowadays. Other users who do not have a Word Document copy of this file cannot modify the contents of this report. Therefore, it is more secure to let other people read this project report using a PDF file.



*Chapter 3:*

*Project Estimation*

Project Estimation

# Activity Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Task No. | Description | Time Duration | Dependencies |
| *Phase 1:* | ***Concepts, introductions, and requirements*** | | |
| T1 | Project concept | 1 days |  |
| T2 | Requirements Gathering | 2 days | T1 |
| T3 | Writing Introduction and basic idea about the project | 1 day | T1 |
| *Phase 2:* | ***Responsibilities*** | | |
| T4 | Divide and list all tasks | 1 day | T1, T2, T3 |
| T5 | Assign each task to each member | 1 day | T1, T2, T3 |
| *Phase 3:* | ***Design*** | | |
| T6 | Arrange, analyze, and revise requirements | 2 days | T2, T3 |
| T7 | Design diagrams | 10 days | T6 |
| *Phase 4:* | ***Implementation*** | | |
| T8 | Coding | 10 days | T7 |
| T9 | Revise Diagrams | 5 days | T7 |
| T10 | Test the first version of the program | 1 day | T8 |
| T11 | Re-code and finalize the program | 4 days | T9 |
| *Phase 5:* | ***Test and Deployment*** | | |
| T12 | Fully test the program | 1 day | T11 |
| T13 | Finalize diagrams | 3 days | T12 |
| T14 | Documentation | 3 days | T7, T13 |



*Chapter 4:*

*Use-Case Diagram*

Use-Case Diagram

