Bilkent University

Department of Computer Engineering

CS319 – Object Oriented Software Project

Project short-name: An Object Oriented Approach to Zork-Like (Text Based) Games

Analysis Report

Project Group 1

Fatih Karaoğlanoğlu

Erdoğan Ege Tokdemir

Furkan Emrehan Kılıç

İpek Lale

Analysis Report

Oct 15, 2014

This report is submitted to the Department of Computer Engineering of Bilkent University inpartial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

[1 Introduction 1](#_Toc254263637)

[2 Current System 1](#_Toc254263638)

[3 Proposed System 1](#_Toc254263639)

[1.1 Overview 1](#_Toc254263640)

[1.2 Functional Requirements 1](#_Toc254263641)

[1.3 Non-functional Requirements 1](#_Toc254263642)

[1.4 Pseudo Requirements 2](#_Toc254263643)

[1.5 System Models 2](#_Toc254263644)

[1.1.1 Scenarios 2](#_Toc254263645)

[1.1.2 Use-Case Model 2](#_Toc254263646)

[1.1.3 Object and Class Model 2](#_Toc254263647)

[1.1.4 Dynamic Models 2](#_Toc254263648)

[1.1.5 User Interface 2](#_Toc254263649)

[4 Glossary 2](#_Toc254263650)

[5 References 2](#_Toc254263651)

Analysis Report

Project short-name: An Object Oriented Approach to Zork-Like (Text Based) Games

# Introduction

Text-based games are computer games that are displayed not using computer generated images, but with plain text. The user is, thus, free to use his/her imagination alongside a storyline expressed through text. This project was inspired by and aims to recreate the classical text-based game “Zork” (1977) using an object-oriented approach.

# 2 Current System

Actually there are many text-based games on the market. One of the most known ones is called ‘Zork’. It is an interactive fiction computer game which was written in 1977 using the MDL Programming Language[1]. The game begins in a house. Then player give some directions by writing according keywords on the terminal of the game and game goes on according to these directions.

# Proposed System

What we will done in this project is highly similar to Zork. We will change places and we will add some features. For example in Zork there is a comment ‘read’. If user enters it like ‘raed’ Zork does not understand it and only say; ‘I don’t know this word’. (figure 2) Our game will understand such typos and it will detect the command. In addition Zork is located on 2-dimensional plane. However our game will be on 3-dimensional plane which means there will be orientation options which allow to move up and down.

## Overview

The system to be developed is a text-based game. As the name of the game’s genre implies, the most important components of the game are user and game interactions because visual graphics does not exist. Written commands are the basic way to interact with the game. Unlike Zork, which is our inspiration, the game will provide flexibility for types and case sensivity. In addition instead of 2 dimensional the game will be played on three dimensional space support provided in text format.

The game plot will be based on survival. The game challenges a player to survive by displaying obstacles on his way such as hunger, heat, dark and wound. Player will increase or decrease his points based on his reactions to these obstacles. In addition, player has a health rate and when it drops to zero, the game ends. When the game ends, highscore table appears including players name, score and the finishing time.

Points and health will be calculated based on randomly generated values for some commands such as eat and drink. So there will be a chance factor in the game in addition to skill factor.

# Functional Requirements

* In the game there must be instructions screen which includes description of the game and keywords to proceed in the game.
* The game is controlled via mouse and keyboard.
* The game must have a score screen which shows the player’s score.
* The game should tolerate typos and must not be case-sensitive.
* The game should include a help section.
* The game should recognize the following commands: look, search, go, take, attack, kill, use, examine.
* The user should be able to login to the game.
* The user should be able to save his/her progress in the game.
* The user should be able to load a previously saved progress.
* The user should be able to logout.
* The user should be able to start from the beginning.?
* The user should be able to save only one instance of his playing.
* The user should be able to create a new account with a username and password.
* The game should display a high score table.
* The high score table should have dates, names and scores.
* The system should show the date in this format: dd / mm / yyyy.
* Multiple players should be able to play the game, although not simultaneously.

## Non-functional Requirements

* The response time for commands should be less than one second.
* The game should be able to run on Windows XP/Vista/7/8 and Unix.
* The content of the game should be easy to understand.
* There should be separate classes for each construction (locations, items, actions etc.) in the game to ease testability and increase flexibility of the system (working with separate classes make it easier to add features to the system).
* The level of expertise of the user shall be basic.
* The system should store a maximum of 10MB of data.
* The user shall not install the program.?
* The user should have JRE installed in his system in order to run the program.
* The user’s system should have at least 1 GB of installed memory and 1024x766 screen resolution.

**3.4 Pseudo Requirements**

1. The project should be completed within three months.
2. Java should be used as a programming language.
3. The system must be a desktop application.
4. The system must be distributable.
5. The system must use MySQL database. (eğer user management olcaksa OR txt)

System Models

### Scenarios

Overall usage

Lucy, a student at Bilkent University, has a lot of assignments and exams. She often gets bored and feels the need to play a video game to relax, so she runs our game, creates an account and starts playing. After a while, she gets bored, saves the game and quits. Later, she comes back and runs the game again. She loads her save file, and continues to play where she had left off. Eventually, she dies in the game. Then highscore table appears and her score, name and time of end game appears along with other players. She then quits the game and continues with her studies.

Controls

During the game she uses help command to learn how she plays the game. Then using the commands appeared in the help section she moves the character and try to survive from obstacles.

### Use-Case Model

A code segment is below:

for (i=1; i<=5; i++)

System.out.println(“report to write”);

If you need to inline code, use “this”style.

### Object and Class Model

Table 1is an example table.

Table 1 An example table

|  |  |
| --- | --- |
| Key | Value |
| key | Value |

### Dynamic Models

### User Interface

# Glossary

Glossary for any domain-specific terms you use in your report.

# References

1. http://en.wikipedia.org/wiki/Zork.