REQUIREMENTS AND ANALYSIS DOCUMENT FOR 'TIS BUT A SCRATCH

TDA367

Ivar Josefsson, Alma Ottedag, Anna Nylander, André Samuelsson

Contents

Version: 3.0

This version overrides all previous versions.

Revised by: Alma Ottedag
Date: 14-05-25

1. Introduction

1.1 Purpose of application

The project aim is to create a top down 2D adventure game in which two players can cooperate in completing puzzles and fighting enemies located in separated areas. It will also aims to be a very easy modified application in which users will be able to construct their own maps, interactive objects and non-playable characters to use in the game.

1.2 General characteristics of application

The application will be a desktop, standalone and multi-player application with a graphical user interface for the Windows/Linux environment.

The game will be a real-time multiplayer game with network in which the player will be able to move freely over the room. The player will also be able to fight enemies and interact with different objects such as doors, as to take them to other rooms. The enemies and players stats will be shown using health bars and animations will play when the characters move.

In addition to the system itself the user will also be able to modify and construct his or her own worlds, weapons, non-playable characters and players using plugins and xml files which the application supports.

1.3 Scope of application

The application will not be able to save started games or collect any terminated games. The game will only use keyboard and therefore not support mouse or any other controllers. You will only be able to play multiplayer on separate computers, you cannot join a multiplayer game on the same computer. Neither is it within the scope to implement sound effects. As mentioned before, the user will be able to add several self-designed objects using xml and plugins, however any object not mentioned above is not within scope. Nor is it within scope for the application to play any animation when a character attacks.

1.4 Objectives and success criteria of the project
1. It should be possible to play multiplayer over network.
2. Hostile non-playable characters should be able to be attacked and defeated.

3. A self-designed map (using Tiled) should be able to play and a correctly modified xml-file for any of the characters or weapons should be runnable.
4. The player should be able to travel between rooms through doors.
5. Hostile non-playable characters must be able to be defeated, in other words, can die and be removed from the game.

1.5 Definitions, acronyms and abbreviations

Player = a user of the program and his or her corresponding representation in-game.

2. Requirements

2.1 Functional requirements

The application should support playing the game at the same time as it is hosted, as well as running a dedicated host.

The user should be able to;

1. Host a game.
2. Join a game as a client.
3. Play the game, including being able to:
    a. Move
    b. Attack
    c. Take damage
    d. Change room independent of other players
4. View credits

2.2 Non-functional requirements

2.2.1 Usability
The application will have a high focus on usability. The primary language in the game will be English and it is assumed that the user is somewhat familiar with games and other graphical applications. The game will primarily utilize graphics to convey information and text will be a secondary way to convey information.

The game will implement text instructions in the menu but the game will primarily be user friendly through intuitive controls and interfaces.

The application should support creation of your own maps and associated logic. A knowledgeable user should be able to create their own maps as well as their own monsters, this include AI, sprite and stats.

2.2.2 Reliability
NA

2.2.3 Performance
The application should not suffer from noticeable stuttering or delay. Any input from the user of the application should have an immediate response graphically, when playing locally the delay from user input may be a maximum of 0.05 seconds. The application should have a minimal footprint when it comes to networking in order to minimize delay and load on the infrastructure.

2.2.4 Supportability
The application should be built for desktop but steps should be taken to aid in porting the application to other environments.

There should be automated test verifying all use cases. Code related GUI should be tested manually.

2.2.5 Implementation
The application will use the Java environment, in order to run the application the user will need to have JRE (Java Runtime Environment) installed and configured. The application will have an executable file that it can be run through, no installation required.

2.2.6 Packaging and installation
The application will be delivered in a zip archive containing:
1. An executable file for running the program
2. All needed resources (maps, sprites etc.)

2.2.7 Legal
Sprites and tile sets used are allowed as long as credit is given. There might be legal issues pertaining to the name but that is not covered he

2.3 Application models

2.3.1 Use case model
RunGame, SetGameOptions, QuitGame, StartProgram, TakeDamage, UseEnvironmentalObject, UnlockDoor, Attack, GoThroughDoor, JoinGame, Interact, HostGame, MoveCharacter, KillEnemy, PickUpItem.UML and a list of UC names (text for all in appendix)

2.3.2 Use cases priority
1. Move

2. Attack - deal damage
3. Take damage
4. Swap to a different room
5. Interact with object

2.3.3 Domain model
See Appendix.

2.3.4 User interface
The application should utilize pseudo-3d to display the game world and game state. The game should have at least a resolution of 640x480 pixels with tiles of a minimum of 32x32px.The game should overlay some information during gameplay, health, items and weapon should be displayed in a easy to understand manner.
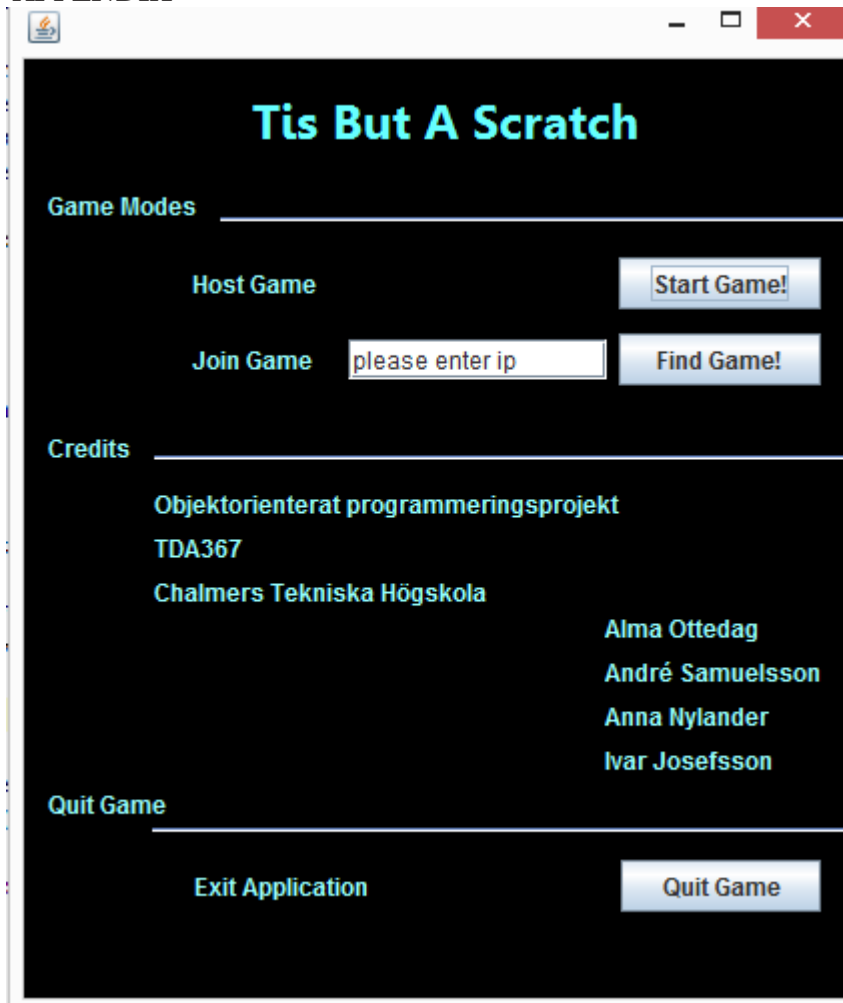
2.4 References

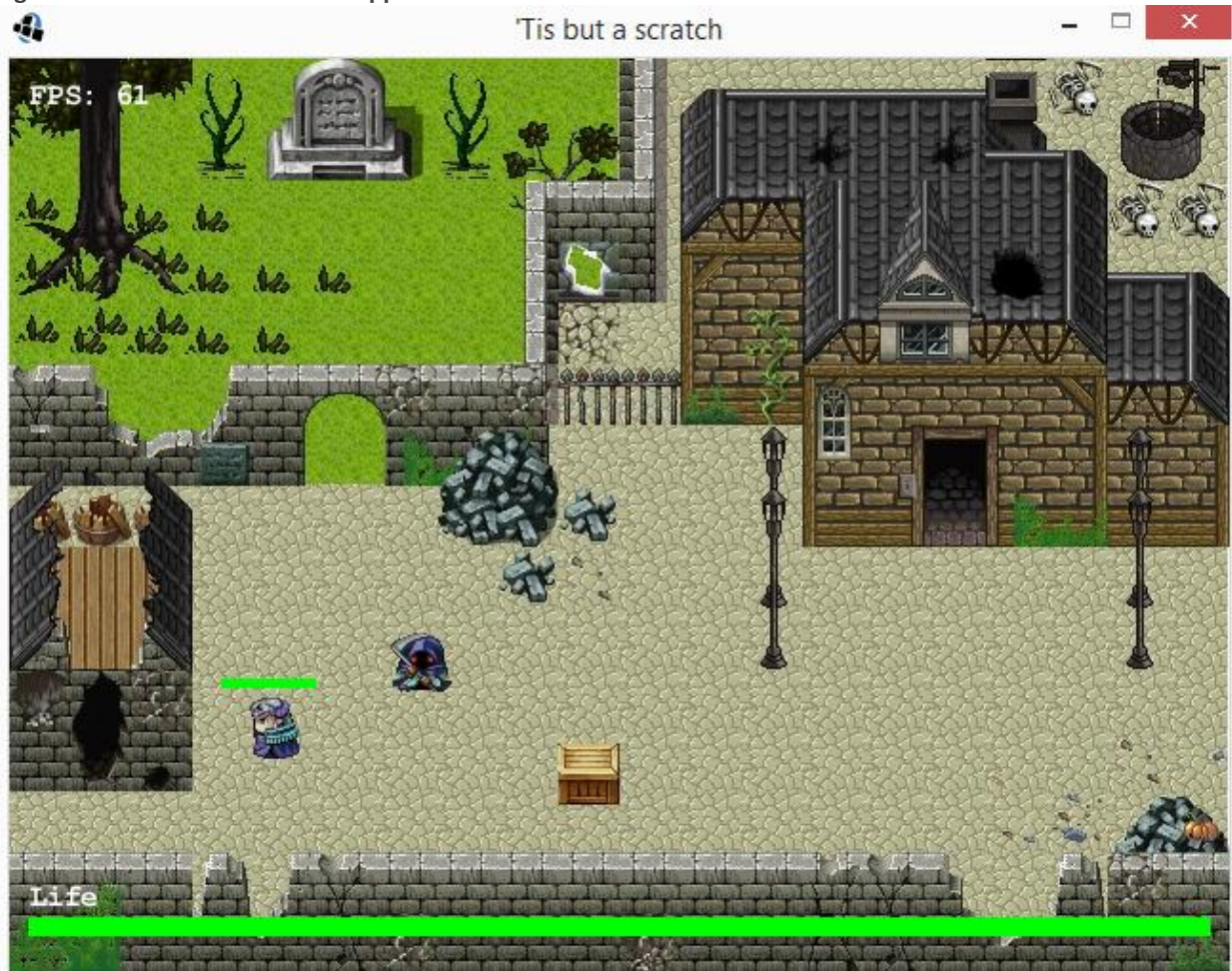APPENDIX

Figure 1 : Main menu GUI of the application

Figure 2 : In-game display GUI

Figure 3 : GUI when attacking

Figure 4 : GUI when attacking enemy and enemy taking damage

Figure 5 : Domain model for the application