Requirements and Analysis Document for the Kelde,the wrath of Ra's al Ghul project (RAD)

Version: 2.0

Date 2015-06-29

Author Philip Tibom, Anders Bolin, Hossein Hussain, Daniel Olsson

This version overrides all previous versions.

1 Introduction

This section gives a brief overview of the project.

1.1 Purpose of application

The projects aim is to create a computer cross platform game that later can be moved to android with ease. The main differences will be the controls (Keyboard vs Touchscreen). The game will be heavily inspired by the 2D game from Nintendo, Zelda, A link to the past. The title of the game is inspired by the DC comics universe (Batman, Green Arrow etc) where the evil Ra's al Ghul (the demons head) from the middle east terrorizes the cities.

One of our main priorities will be to make the application very modular. Where a game designer rather than a programmer, later can add new monsters, maps and environmental objects that can be interacted with. By using drag and drop tools. Since we won't have the time to make a full scale storyline.

1.2 General characteristics of application

The application will be a desktop, standalone (non-networked), single-player application with a graphical user interface for the Windows/Mac OSX/Linux platforms.

The application will be real-time. Where the player will be able to move a character around in a 2D environment. To face enemies and interact with movable objects. The player will be able to fight and take damage. Collide with various objects and walls. And shoot projectiles.

There will be no time constraints and no ways to earn points. As this being an adventure/role playing game, the goal is simply to reach the end.

The game should have a simple GUI that displays the current health and inventory.

1.3 Scope of application

The application will include computer players in the form of monsters and enemies. They will be controlled by an AI whose goal is to attack the player. The game can only be played alone since there won't be a multi-player feature. The game should save interrupted play. Although when returning to the game some of the environment should be reset (to make it harder).

1.4 Objectives and success criteria of the project

1. The game should load the world, objects and characters.
2. The player should be able to move around with high performance collision detection.
3. The player should be able to attack and be attacked.
4. There should be monsters with simple Artificial Intelligence.
5. There should be an intro for the game.
6. There should be a menu screen.
7. It should be possible to easily add new creatures, objects and create maps.

1.5 Definitions, acronyms and abbreviations

- GUI, graphical user interface
- AI, Artificial Intelligence (computer controlled elements)
- Java, platform independent programming language.
- Host, a computer where the game will run.
- JRE, the Java Run time Environment. Additional software needed to run a Java application.
- Android, mobile operating system user in cellphones, tablets, watches and more.
- Monster, computer controlled enemy.
- Map, the environment that the player can move around in.

2 Requirements

In this section we specify all requirements

2.1 Functional requirements

1. Select new game from Menu.
2. Walk around the map.
3. Push/move loose objects such as barrels.
4. Deal damage to monsters.
5. Receive damage from monsters.
6. Trigger bomb explosions.
7. Pick up a wide variety of objects.
8. Open treasure chests that drops loot.
9. Teleport by walking into a door. (Moving into a new room)
10. Exit the game.

2.2 Non-functional requirements

Possible NA (not applicable).

2.2.1 Usability

Usability is high priority. Normal users should be able to play the game within a very short period. But the game is aimed towards people that have played similar games before.

Tests with at least four different non-computer-professional should be performed to verify the usability. Preferable someone that never played games before. And also someone that has played a lot of games before.

The game will be in English and so will the storyline. But it should not be necessary to understand English to play the game as it is only graphical apart from the story. A player should be able to play this game without a manual. The necessary information is provided within the game.

2.2.2 Reliability

NA

2.2.3 Performance

The game should be very fast and responsive. Any actions initiated by the player should not exceed 2 seconds response time in worst case. (The initial load time of the game is allowed to take longer since we will use a lot of graphics).

2.2.4 Supportability

The game will support Windows, Linux and Mac OSX or other JRE supported systems that receives input from keyboard and mouse.

The GUI will be designed for a computer system (Local application and Web application).

Controls and GUI will be easily replaced to fit mobile applications. The game itself should be able to be ported to Android with ease.

Use cases will be tested manually. All the underlying models will be tested automatically.

2.2.5 Implementation

To achieve platform independence the application will use the Java environment. All hosts must have the JRE installed and configured. The application needs to be installed on all hosts where it will run (possibly downloaded).

2.2.6 Packaging and installation

The application will be delivered as a zip-archive containing:

1. A file for the application code (a standard Java jar-file).
2. All needed resources, graphics, animations, scripts, and dialog text files.
3. Start programs (scripts) to start the game on the different platforms.
4. A README-file documenting installation and start of application.

2.2.7 Legal

There are no legal issues regarding our game. Graphics are open source licensed by the MIT license and Creative Commons. Appropriate artists has been credited and licensed in the project files provided on our github.

2.3 Application models

2.3.1 Use case model

UML and a list of UC names (text for all in appendix)

2.3.2 Use cases priority

- New game button (high)
- Load game button (low)
- Exit game button (high)
- Move (high)
- Attacked by Bat (high)
- Attacked by Eye (high)
- Attacked by Ghost (high)
- Attack Bat (high)
- Attack Eye (high)
- Attack Ghost (high)
- Open treasure (medium)
- Close treasure (medium)
- Pick up axe (medium)
- Pick up sword (medium)
- Pick up key (medium)
- Pick up red potion (medium)
- Pick up blue potion (medium)
- Shoot arrow (low)
- Trigger bomb (medium)
- Teleport (high)

2.3.3 Domain model

See APPENDIX.

2.3.4 User interface

The GUI will be drawn by the graphics card inside our game loop. We will use our own images and types of buttons. The GUI will be scalable to suit **any** screen resolution. Hence it will work as good for computer screens, projectors, 4K screens or mobile phones.

Text to motivate a picture.

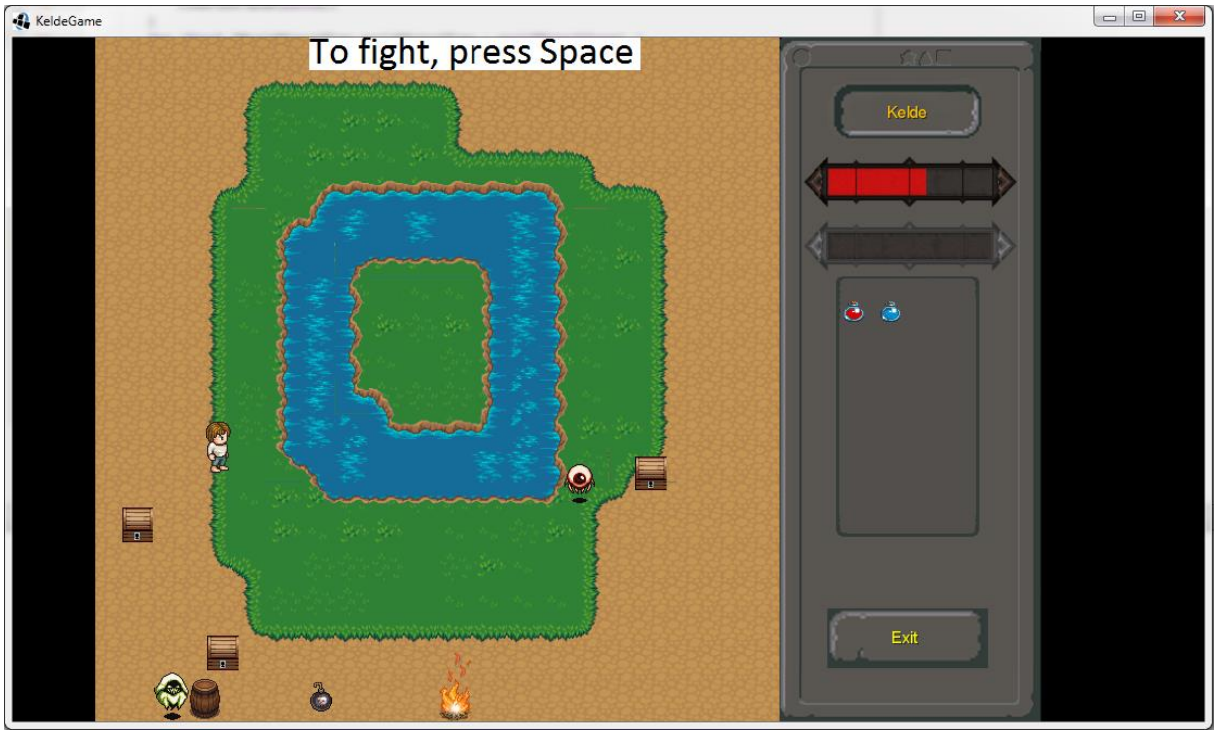
2.4 References

Zelda, A link to the past:
https://en.wikipedia.org/wiki/The_Legend_of_Zelda:_A_Link_to_the_Past

APPENDIX

GUI

Preliminary GUI.



Domain model

Our application has become very large and therefore our UML cannot fit in a single page. Please have a look at our github repository.

Direct link to UML:
https://raw.githubusercontent.com/ptibom/Kelde/master/Documents/UML/diagram-all-dependencies.png

Use case texts

For use cases please see our github repository folder dedicated to use-cases.

https://github.com/ptibom/Kelde/tree/master/Documents/UseCases