Requirements Documentation

Group 8: Team Eight
Stanley Liu (MacID: liuz23)
Toni Miharja (MacID: miharjat)
Zhi Zhang (MacID: zhangz1)

October 5 2017

Contents

1 Revision History

Date	Developer	Change	Revision
October 4, 2017	Stanley Liu	Added introduction.	Revision 0
October 5, 2017	Stanley Liu, Toni Miharja, Zhi Zhang	Added functional requirement, non-functional requirement, Off-the-shelf solution.	Revision 0
December 6, 2017	Zhi Zhang	Improved.	Revision 1

Table 1: Revision History: Requirements Documentation

2 Introduction

2.1 Purpose

The purpose of this requirement document is to give a detailed description for the tower defense game "Dragon Age". This document will include the stakeholders and the goals for the project, and mainly specifying the functional and nonfunctional requirements. At the end, it contains a off-the-shelf solution related to our project.

2.2 Scope

"Dragon Age" is a tower defense game running on personal computer. Basically, the user will send out three types of dragons to locate alongside a designed route to defend the homeland. As the enemy wave increases, the dragons can be upgraded to do more damages.

2.3 Stakeholders

The stakeholders of the project are:

- Software developer identifies our project group, which include project leader, project developer, and project tester. These people divides the project into different tasks to complete and make sure every task is done according to the project schedule.
- Project advisors include our professor and teaching assistants of the course, who give proper guidance and precious advices for the software development process.
- Ensuring a proper projectile collision detection between the projectile to the enemy.
- Consumers would be the users who play the game after the game launches, and we are targeting teenagers and adults specifically.

2.4 Goals

The project group has the following goals:

- The main goal of the project is to focus on the details of the development process, making sure every step is completed with teamwork and follows the software principles.
- The project should have well-defined assignments for each group member. Every member should participate in team meetings and contribute evenly.
- We aim to meet all the requirement for the original game concept and add features for optimization. Although the theme is different, the requirement such as the rules of the game should be achieved.

- The software development should be incremental and iterative. GitHub is used as the depository and each member will contribute by merging from their individual branch to master branch.
- The final game needs to be produced to an executable file from python and pygame code.

3 Functional Requirements

Functional requirements are those requirements that provide the fundamental reason for the product's existence

3.1 System Features

- The game shall start with providing 3 types of dragon (fire, ~~water~~ ice, and poison) and sufficient resource(money) for users to put at least one dragon on the map to defend the first wave of enemy.
- ~~The dragons shall have type attributes and each attacking form should have a specific effect on enemy.~~ Fire dragon shall do the most damage to the enemy and shall have the fastest attack speed compared to the other two dragons that are at the same level as the fire dragon.
- Ice dragon and poison dragon slows enemy down once hits the enemy.
- The users should earn resources after killing each enemy. The resource amount earned ~~is based on the wave~~ starts at
- When users earn enough resources, they have either use them to upgrade dragons or buy new dragons.
- The game shall have enemy waves that increase in movement speed and defense after each wave.
- If all the lives of ~~our~~ the home base are taken by enemies, the game shall display the "game over" message. ~~game shall enable the start and pause button for the ease of users to set up dragons in between waves.~~
- The game shall enable the "new game" button for the user to start a new game.
- The game shall enable the "quit" button for the user to end the game.

3.2 Use Cases

Name	Actor	Precondition	Postcondition	Flow of events
Build/buy dragon	User	The start up of the game	User is able to start defending the base using dragons.	i User checks if they have sufficient resources to buy a new dragon. ii User selects the dragon to build (either fire, water or poison). iii User locates the block where they want to place the dragon.
Destroy an existing dragon	User	There is one or more existing dragons placed.	The selected dragon is demolished.	i User selects the dragon that he wishes to demolish. ii User click on the demolish button.
Upgrade an existing dragon	User	There is one or more existing dragons placed.	The selected dragon is upgraded.	i User checks if they have sufficient resources to upgrade a dragon ii User selects the dragon to upgrade. iii User click on the upgrade button.

Table 2: Use Cases

4 Non-Functional Requirements

4.1 Look and Feel Requirements

4.1.1 Appearance Requirements

The layout of the game will induce the feel of Middle Earth. ~~The background may vary between stages and all will revolve around the theme.~~ The dragons will be different in colors, shape and size according to their types and level. The program will be sized to fit properly with the size of the window.

4.1.2 Style Requirement

The game should bring an adventurous and mystical feel to the user. The gameplay will provide the user with a more intense feeling of concentration rather than laid back.

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

The game will be easily picked up by ~~players of all ages.~~ age 12 and up.

4.2.2 ~~Personalisation and internationalisation requirements~~

~~Not applicable~~

4.2.3 Accessibility Requirements

The game will be compatible with ~~most operating systems~~ Windows, Mac, Linux.

4.3 Performance Requirements

4.3.1 Speed and latency requirements

The game will be highly responsive, responding to the user input ~~immediately~~ within 1 second.

4.3.2 Safety-Critical Requirements

~~The game will not compromise the user device.~~ The game will not store or transmit user's information or data.

4.3.3 Reliability and Availability Requirements

The game will be available for use anywhere ~~and~~ at anytime. It does not need internet connection and will be able to run smoothly as long as the user's device system is not heavily loaded.

4.3.4 ~~Scalability or Extensibility Requirements~~

~~The code will allow for scalability and extensibility.~~

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

The game will be available for use anywhere and anytime. It does not need internet connection and will be able to run smoothly as long as the user's device system is not heavily loaded.

4.4.2 ~~Requirements for Interfacing with Adjacent Systems~~

~~Not applicable.~~

4.4.3 Release Requirements

The game will be updated yearly.~~ according to the feedback, the changing taste and the preferences of the users.~~

4.5 Maintainability and Support Requirements

4.5.1 ~~Maintenance Requirements~~

~~The team will make sure that the game will have minimum maintenance needs.~~

4.5.2 Supportability Requirements

As the game is compatible with most operating systems, ~~most~~ people with a running PC or laptop should be capable of running the game on their machines.

4.5.3 Adaptability Requirements

The game shall be universally accessible by all users who are able to read English and to use mouse click function.

4.6 Security Requirements

4.6.1 ~~Access Requirements~~

~~The game will be accessible to those who are willing to play and have access to the necessary infrastructure.~~

4.6.2 Integrity Requirements

The game will not alter its own source code and will not accept invalid user input.

4.6.3 Privacy Requirements

The game will not be able to access data outside of the game and hence will not infringe user's privacy.

4.6.4 ~~Audit Requirements~~

~~Not applicable.~~

4.6.5 ~~Immunity Requirements~~

~~Not applicable.~~

4.7 Cultural Requirements

The game will be available only in English. The game will not be offensive in any ways to any religious or ethnic groups.

4.8 Legal Requirements

4.8.1 Compliance Requirements

This game will not compromise any laws.

5 Off-the-shelf Solutions

When it comes to the existing solution, there is a already made production on GitHub named “Pokemon Tower Defense” which is the game we are going to redevelop in this course. The requirements we are using come from this existing game, however, we write our own game code and add new features.