Requirement Documentation

Team 3
Shuying Chen, 400080161
Ziyang Huang, 400051063
Guiye Wu, 400089784

Oct 6,2018

Contents

1 Revision History

Table 1: Revision History: Requirement Documentation

Developer	**Date**	**Change**	**Revision**
Shuying Chen	Oct 5, 2018	Initial Draft	0
Ziyang Huang	Oct 5, 2018	Initial Draft	0
Guiye Wu	Oct 5, 2018	Initial Draft	0

2 Project Drivers

2.1 The Purpose of the Project

2.1.1 The User Business or Background of the Project Effort

This project is to redevelop a classic game minesweeper. The earliest minesweepers trace back to the 1960s, and this puzzle game style becomes popular during the 1980s. In modern time, minesweeper is the built-in game in window 7 system or the earlier version, for any other system it has to be downloaded in order to play this game. The game is very popular in earlier years, however, when the new systems are developed, the game is removed from the built-in games list and many people even don't know about the game nowadays. Our motivation is to renew this project and make it be well known.

2.1.2 Goals of The Project

The goal of this project is trying to redevelop the classic game – minesweeper with a new style of the user interface with some funny features to increase the enjoyment. This game is suitable for the users that would like practice logical thinking skills and being entertained at the same time. With different levels are included in this game, this will increase the playability for users from beginner to expert. The game will be re-implemented in python and can be easily accessed locally once it has been pre-downloaded on the computer. The advantage of this project is to provide our clients with a free-and-easy access minesweeper game whenever and wherever possible.

2.2 The Stakeholders

2.2.1 The Client

The client for our product will be the group of the general public that is interested in playing the new version of minesweeper game. They will also be the reviewer of the product.

2.2.2 The Customer

The customer of the project will a group of the general public which is interested in consuming the game media. The entity, which members can be from all ages, who have the access and interact with our product will be the customer.

2.2.3 Other Stakeholders

People, who have an interest in the game, either tangible or intangible, will be the other stakeholders for this software. This group of people may have different knowledge background for the game, so several levels of difficulties are provided, so this software will be suitable for the public and the basic tutorial is provided if needed.

2.3 Mandated Constraints

2.3.1 Solution Constraints

The product should re-size and re-allocate itself according to the different size of the dialogue box. The users can re-size the dialogue box as they wish, and the game should always be in the middle of the dialogue box and with the size of three-quarters of the dialogue box.
Also, the project must be the re-implementation of the classic minesweeper game, which is an open-source project that was written in Java. We will need to implement a project wrote in python that has similar functionality as the classic minesweeper game.

2.3.2 Implementation Environment of the Current System

The game will require the user to download ahead, and it will be a stand-alone game. The user can play the game locally on their computer.

2.3.3 Partner of Collaborative Applications

The product references to the classic Microsoft Minesweeper game, and the product is based on open source minesweeper codes which are written in Java. Furthermore, the product will be written in Python and pygame will be the collaborative application.

2.3.4 Off-the-Shelf Software

For the product to be executed the following off-the-shelf software is required:

a)Python(available from https://www.python.org)

b)A client where the interface can run

2.3.5 Anticipated Work-space Environment

The anticipated workplace environment for the product is anywhere. It can be played at any time, anywhere as you want, as long as you have the game downloaded before. It is an off-line application.

2.3.6 Schedule Constraints

The is no such a schedule constraint for our team. The basic deadline is about the beginning of December when we need to present our project.

2.3.7 Budget Constraints

The budget for the whole project is $0.

2.3.8 Enterprise Constraints

The game can be downloaded to local and it is free and easy to play, only with some logic.

2.4 Naming Conventions and Terminology

- Functional Requirement: Describe what services the software-to-be should provide
- Non-function Requirement: Constrain how such services should be provided
- Client: The group of people will provide their expectations on the product.
- Consumer: The group of people who are trying to be satisfied. They will be the entity that will consume the final product.
- Stakeholder: A person, group or organization that has interest or concern in the success of the project.
- User: The person who will eventually use the final product.
- Minesweeper: The name of the classic game, which we will re-implement on

2.5 Relevant Facts and Assumptions

2.5.1 Relevant Facts

The relevant fact is that the original implementation of this game contains about 1200 lines of code, and it needs to run on a web browser.

2.5.2 Business Rules

We need to work together at the same time like in the office and try to make the same effort for this project.

2.5.3 Assumptions

The project is developed for a single user, it can be run on a laptop or computer at any location because it is offline software. The project is not a large software, it does not have any requirements on CPU or memory for modern computers.

3 Functional Requirements

3.1 The Scope of The Work and the Product

3.1.1 The Context of The Work

Deliverable

- Required Documentation
- Final Software

Deadlines

- Requirements Document Revision 0: October 06
- Proof of Concept Demonstration: October 16
- Functioning Prototype: October 20
- Automated Testing prototype: October 21
- Test Plan Revision 0: October 27
- Design Documentation: November 10
- User documentation: November 17
- Test Report Revision 0: November 27
- Functioning Program: November 27
- Final Documentation: December 6

3.1.2 Work Partitioning

Table 2: Work Partitioning

Event numner	**Event Name**	**Input**	**Output**
1	Minesweeper Game Board	Code	User Interface
2	Minesweeper Graphic	Code and Graphics	User Interface
3	Minesweeper Rules	Code	User Interface
4	Minesweeper Audio Effects	Microphone	Audio Output Device
5	Minesweeper Final Edit	Code	User Interface

3.1.3 Individual Product Use Cases

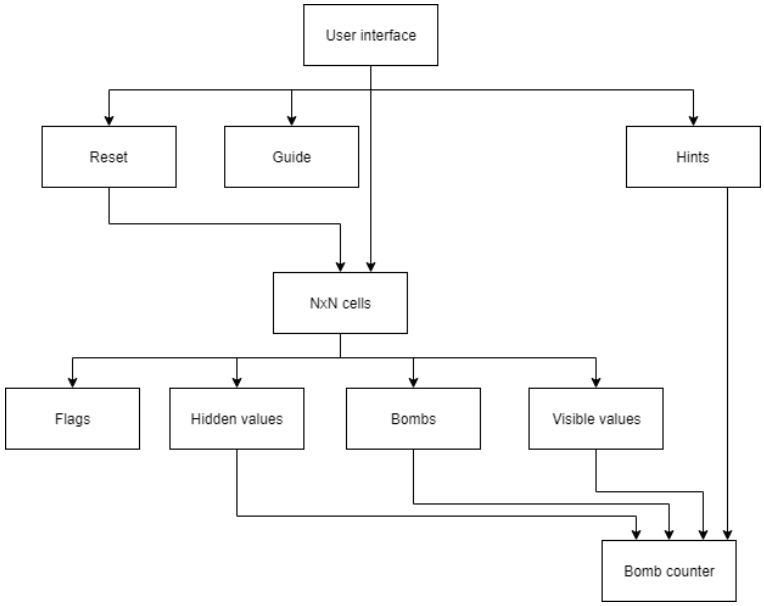

Figure 1: UseCases

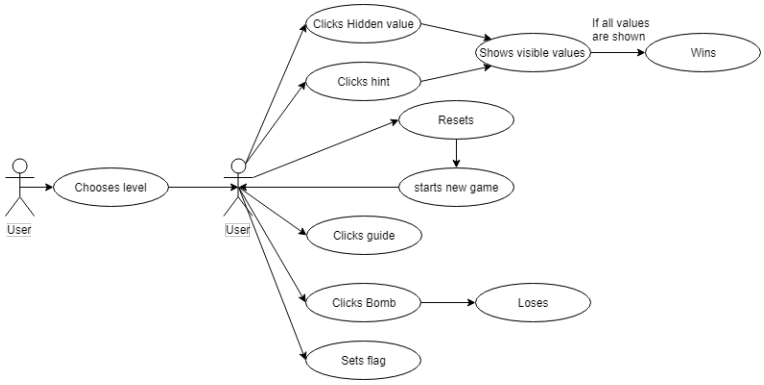

Figure 2: UseCasesDiagram

3.2 Functional Requirements

3.2.1 Input

1. The software is able to read user's position of clicking blocks and record the coordinate for other functional operations.

2. Left clicks on the cell will be considered as visualizing the hidden values
3. Right clicks on the cell will be considered as setting flags
4. Click the icon above the playing area will be considered as an input.

3.2.2 Output

5. If the player clicks a non-block, the block will either show the blank space of corresponding numbers.
6. If the player clicks a bomb, the block will show the icon of the bomb and terminate the game.
7. If the player right-lick the mouse on to the blocks, the will be a "flag" shown on the blocks that the block is being tagged.
8. If the player click the icon above the playing area, the current game will clear out and a new game is restarted.

3.2.3 Other functional Requirements

9. The game has a convenient button for the user to click, which will show all the number around the clicked cell, if the neighboured bombs are flagged right.
10. The game has three difficult levels that are easy, normal and hard. The easy model is 10 times 10 cells, the normal model is 20 time 30 cells and the hard model is 40 times 60 cells.
11. The game can be reset during any condition of the game.
12. The game is over when user hits the bomb.
13. The game is passed when user finds out all the hidden values on the board.
14. The hidden values on the board will convert to visible values when the values are clicked.
15. All the cells and the reset button are clickable on the board.
16. The game provides 'flag' that allows user to put on the board to indicate a bomb.

4 Non Functional Requirements

4.1 Look and Feel Requirements

1. The game interface shall looks comfortable according to the funny features

4.2 Usability and Humanity Requirements

2. The software shall be easy to use for people older than 5 years.
3. The game provides guide for new players, and the guide contains all the game rules in details.

4.3 Performance Requirements

4.3.1 Speed Requirements

4. The response time of the system should be fast.
5. Setting up a new game should be fast.

4.3.2 Safety Critical Requirements

6. The software shall not make any injures on user.

4.3.3 Reliability and Availability Requirements

7. The software can be run any time and it will be usable for 24 hours per day, 365 days or 366 days per year.

4.3.4 Precision Requirements

8. Each cell contains either value or bomb.
9. The size of the value, bomb and flag should fit to the size of each cell.
10. The reset button, the guide button and each cell are large enough to be clicked.
11. The values shall be clear to recognize.
12. The location of value, bomb and flag should be aligned to the cell position.

4.3.5 Capacity Requirements

13. The software takes less than 10 mb memory to be downloaded.

4.4 Operational and Environmental Requirements

4.4.1 Excepted Physical Environment

14. The software is used when user clicks the game icon and chooses a difficult level.

4.4.2 Excepted Technological Environment

15. The software can only be used after downloading.

4.5 Maintainability and Support Requirements

4.5.1 Maintainability

16. The software can run without internet.

4.5.2 Portability

17. The software is expected to run on Windows, Linux and Mac environments.

4.6 Security Requirements

18. The software does not require any information from the user.

19. The software will not read or destroy or execute any data from the local machine.

4.7 Cultural Requirements

20. The software does not include any feature that offends user's culture.

4.8 Legal Requirements

21. The software is redevelopment from a open-source code.

4.9 Health and Safety Requirements

22. The software shall not make any injures on user.

5 Project Issues

5.1 Open Issues

N/A

5.2 Off-the-shelf Solutions

5.2.1 Ready-Made Products

- Interface

5.2.2 Reusable Components

Modularized code components

5.2.3 Products That Can Be Copied

The original code cannot be copied, but it can be a reference and a prototype that we can rely on.

5.3 New Problems

5.3.1 Effects on the Current Environment

The new product will be an off-line user interfaced game. It will be played locally without any interrupting of the laptop. It is really safe and stable.

5.3.2 Effects on the Installed Systems

There is an interface along with the system.

5.3.3 Potential User Problems

Some potential users who do not know the rules and logic may be a headache about the game and very stressful.

5.3.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

The server that will host our product will not be powerful enough to hold the desired amount of users with our project growth pattern.

5.3.5 Follow-Up Problems

The will be no follow-up problems that we take together, make the same effort to the program.

5.4 Tasks

5.4.1 Project Planning

Table 3: Project planning

Task	**Roles of Completes**	**Time**
Model Implementation	Software Engineers	OCT 15
Model Revision	Client	OCT 26
Python Design	Software Engineers	NOV 9
Interface Implementation	Software Engineers	NOV 15
Revision	Client	NOV 23
Publishing	Software Engineers	NOV 26

5.5 Migration to the New Product

None.

5.6 Risks

There is no risk for this project.

5.7 Costs

There is no cost for this project.

6 User Documentation and Training

6.1 Waiting Room

Additional functionality of the game functionality and visual as well as audio effects.

6.2 Ideas for Solutions

Proper hierarchy and documentation of python code.

7 Appendix

7.1 Symbolic Parameters

N/A