SOFTWARE DEVELOPMENT PROJECT TEMPLATE

Xiaohe Zhu

Date 2020-02-02

Contents

1 | Revision History

Date	Version	Description	Author
2020-2-2	1	Initial version (Iteration 1)	Xiaohe Zhu
2020-2-20	2	Update iteration 2 UML diagrams and document: use case diagram, state machine diagram and class diagram Basic implementation	Xiaohe Zhu
2020-3-5	3	Update iteration 3 Unit test	Xiaohe Zhu
2020-3-20	4	The final version of the project	Xiaohe Zhu

2 | General Information

Project Summary	
Project Name	Project ID
Hangman Game	Hangman
Project Manager	Main Client
Xiaohe Zhu	University
Key Stakeholders	
Developer Project manager The end-user Tester Promoter	
Executive Summary	
This project aims to develop software that can implement the Hangman Game, by using Java language. The basic idea of this game is that the player is going to guess a word, and for every wrong guess the game is building a part of a man getting hanged. The game also contains additional functions like a high score list and a login system. This is a fun game that can also help people learn English.	

3 | Vision

This project aims to create a Hangman game written in the programming language Java. This game is a fun tool that can provide an exciting learning environment for English learners like children and other non-English speakers.

The basic idea of this game is that the player is going to guess a word by suggesting letter after letter. The player is presented with the number of letters in the word but for every wrong guess, the game is building a part of a man getting hanged. The number of wrongs that the player can have is eight. When the player guesses the whole words, the game shows the winning menu. When the player guesses 8 times wrong, the game shows the fail menu. And after that, the game promotes players to play again.

Additionally, to gain a smoother and provide a better experience for players, more functions will be added to the game like a high score list, user registration, persistence, multiplayer, time limit, point systems, the ability to add and remove words.

Learning English by playing the game is an interesting and efficient way since the learner can be more concentrated than traditional learning methods. The function of adding words can provide a customized test and makes it much fun, and the function of high score list can let the users see their improvements.

Reflection on writing vision:
To write the vision part, first I have to learn what the vision should cover. I read the textbook and the general review. I address the aim of this program and the reason we develop this game. The basic requirements can be recognized by the rules of the Hangman game, and I learned it on Wikipedia and played the game several rounds. As the reason for this report is used both the manager and development team, I should write it in nontechnical words. And this part should also cover the benefits and risk for the manager and it because the defined length of this vision part, I just mention the following parts which will address these things.

4 | Project Plan

4.1 Introduction

This project aims to develop software that can implement the Hangman Game, by using Java language. The basic idea of this game is that the player is going to guess a word, and for every wrong guess the game is building a part of a man getting hanged. The number of wrongs that the player can have is eight. When the player guesses the whole words, the game shows the winning menu. When the player guesses 8 times wrong, the game shows the fail menu. And after that, the game promotes players to play again.

The game also contains additional functions like a high score list multiplayer, and a login system. This is a fun game that can also help people learn English.

4.2 Justification

This game is a fun tool that can provide an exciting learning environment for English learners like children and other non-English speakers. Learning English by playing the game is an interesting and efficient way since the learner can be more concentrated than traditional learning methods.

Teachers can also use the game to test the students' English level. The function of adding words can provide a customized test and makes it much fun, and the function of high score list can let the users see their improvements.

4.3 Stakeholders

The end-user:
The teachers use the game as a teaching tool and want it that is easy to use.
The students use the game as a learning tool and want it that is interesting to use.

The project manager wants the game finished on time and correctly.

The developer wants the requirement clear and easy to implement.

4.4 Resources

The total available time is approximately 9 weeks. The literature is Software Engineering by Ian Sommerville. This project also refers to some online courses on Moodle.

4.5 Hard- and Software Requirements

This development language for this project is limited to Java language. Software: IntelliJ IDEA, the JDK 1.8, JRE 1.8.0, node version 10.9.0, npm version 6.2.0.

4.6 Overall Project Schedule

2020-2-3 first iteration and deliver the first version which implements the basic game rules.

2020-2-17 second iteration and deliver the second version which introduces the UML model.

2020-2-24 third iteration and deliver the third version which implements additional functions.

2020-3-2 Fourth iteration and hand the final project.

4.7 Scope, Constraints and Assumptions

Scope: The Hangman game can be run as a console application. The core function is picking words and displaying the lines that represent each letter. The player shall be able to guess letter after letter. If guessed letter exists in the word, the letter shall be showed. If the letter doesn't exist in the word, the system draws one stroke of the hangman picture. The number of wrongs that the player can have is eight. When the player guesses the whole words, the game shows the winning menu. When the player guesses 8 times wrong, the game shows the fail menu. And after that, the game promotes players to play again.

An extra function is including user registration and creating an account before the game start. And the game also includes a multiplayer mode that allows two-player play at the same time. The game also allows the player to add words that can be guessed.

Outside the scope of this game is the limited words database. The database only includes a limited number of words.

The constraints for the project are time that is limited to 9 weeks. The developer team only contain one person. The developer has limited knowledge about developing and database. This game shall be run as a console application in the terminal environment without any installation and setup. And For that reason the interface is limited.

The assumptions are that the user has a delightful experience. The user can play the game on their personal computer in a terminal environment

without any installation and setup.

Reflection on writing project plan:

To write the project plan, I leaned the structure from the textbook and write is followed by the template. At this stage, I write as much as possible for the project plan. My plan mainly focusses on the requirements from the client's descriptions and Wikimedia and my plan basic be designed by following the requirements. And since I am not sure the next courses, the Hard- and Software Requirements may not clear and I may fix this part latter. As an improvement, the Scope, Constraints, and Assumptions may be added more details.

5 | Iterations.

5.1 Iteration 1

Start Date 2020-1-22

End Date 2020-2-3

The goal of this iteration is to write a project plan and a skeleton code. The sessions of this project plan are including vision, project plan, detailed iteration one, risk and analysis, time log for iteration one, and reflection.

Iteration one also defines project name, project id, introduction, justification, stakeholders, resources, hard-and Software Requirements, overall schedule, scope, Constraints and Assumptions.

The skeleton code for iteration one is focusing the basic classes and interface. The relation between each class will be addressed on later iterations.

5.2 Iteration 2

Using UML modeling and implementing the basic game.

Start Date 2020-2-3

End Date 2020-2-24

5.3 Iteration 3

Testing

Start Date 2020-2-23

End Date 2020-3-9

5.4 Iteration 4

Finishing the project.

Start Date 2020-3-9

End Date 2020-3-24

6 | Risk Analysis

This part is to anticipate the possible risks and set strategies for coping with these risks.

Risk	Probability	Impact	Strategy
Time underestimate	High	Serious	keep attention to the progress of development and change the schedule if necessary
Size underestimate	High	Serious	Carefully plan for each task and keep attention to the current progress.
Hardware crash	Low	Catastrophic	Always save the code on several places, in this case, push into GitLab frequently.
Staff leave since illness	Moderate	Serious	Always starts work early and try to finish before the deadline.
Developer lacks knowledge	High	Tolerate	Read Software Engineering by Ian Sommerville and view the lecture on Moodle
Requirement change	Low	Serious	prepare to change the requirements list and add it to the code.

Reflection on writing risk analysis:

This is a hard part for me since I never did this part before. I read the textbook and just followed the risk list written in the book. The risks analysis should anticipate the future risk during development, and I should not only consider the coding processes and also think about all the processes throughout the whole time. The outside environment should also be covered. At the beginning stage, I just list 4 possible ricks since the project is relatively small and easy. In the future stage, I may continually add new risks to this list.

7 | Time log

Time log for the first assignment.

Task	Date	Estimate Time	Actual Time	Difference
Reading book	1-27	3 hours	4 hours	1 hour
Project plan	1-31	2 hours	3 hours	1 hour
Time log	1-31	1 hour	45 mins	15 mins
Vision	1-31	30 mins	1 hour	30 mins
Iteration plan	2-1	1 hour	45 mins	15 mins
Risk analysis	2-2	30 mins	30 mins	0
Coding	2-3	1 hours	1 hours	0