Contents

SOFTWARE DEVELOPMENT HANGMAN PROJECT

Bhavya Shukla
Linnaeus University

8 February 2019

Contents

1 Revision History

Date	Version	Description	Author
08/02	1		Bhavya Shukla

2 | General Information

Project Summary	
Project Name	Project ID
Project Manager	Main Client
Key Stakeholders	
Executive Summary	

3 │ Vision

Hangman is a word game where there can be a minimum of 2 players and there is no maximum. One of the players chooses a word which is hidden from other players. The player then proceeds to make the same number of dashes or blank spaces as there are letters in the chosen word. The task of the other players is to guess the word which was chosen, 1 letter per guess. There can only be 8-10 incorrect guesses as the aim is to guess the word before a man hung by a rope (hangman) is made. The hangman is made one line at a time, 1 line for each incorrect guess.

The aim of the program is to recreate this game with the computer as the player who chooses the word at the beginning of the game. There will be a list of predefined nouns in the system and the computer will choose one at random. The player (user) has 8-10 guesses to guess the correct word before the hangman is completed. In addition the game will also have features like high score list, user registration and a time limit to make it more challenging.

Refection

Working on the vision of the project has helped me understand the rules of the hangman game. It has also increased my clarity on what type of program I will be making, for example adding other rules such as time limit and features such as high score and user registration. This will help me understand how to implement the code and what type of interface the user should receive. It has also helped me understand what parts of the code I still need to work on as they have not been completed.

4 | Project Plan

Write a project plan for the project. This project plan should show the way to the complete and finished application, something that you should be able to follow. Write as much as possible in the project plan, use the material available on mymoodle (deadlines etc.), and update the document throughout the course when you know more in the later assignments. Again, as an addition, write down your reflections on creating a project plan. This reflection should be about 100 words.

4.1 Introduction

The aim of the program is to recreate Hangman with the computer as the player who chooses the word at the beginning of the game. There will be a list of predefined nouns in the system and the computer will choose one at random. The player (user) has 8-10 guesses to guess the correct word before the hangman is completed. In addition the game will also have features like high score list, user registration and a time limit.

4.2 Justification

The application will help users play when they are unable to find other people to play with. The game can also be competitive, which is a quality that some enjoy.

4.3 Stakeholders

List and define the different stakeholders for the project.

4.4 Resources

Software Engineering 10th edition by Ian Sommerville
Stack overflow

4.5 Hard- and Software Requirements

4 │ Project Plan

Java, UML

4.6 Overall Project Schedule

21st February 2019 – Assignment 2
8th March 2019 – Assignment 3

4.7 Scope, Constraints and Assumptions

Detail what is part of the project and what is outside – specify the scope of the project.

Reflection

The project plan has helped me plan my time for each assignment accordingly and develop a structure of how I want to do my project while adhering to the guidelines. It also helps me realize what parts are still incomplete such as the UML which will be taking place next week. I believe this will help me avoid missing out on any necessary steps that are needed to complete the hangman game. The project plan will also help me identify mistakes earlier so I wil still have time to work on them.

5 | Iterations

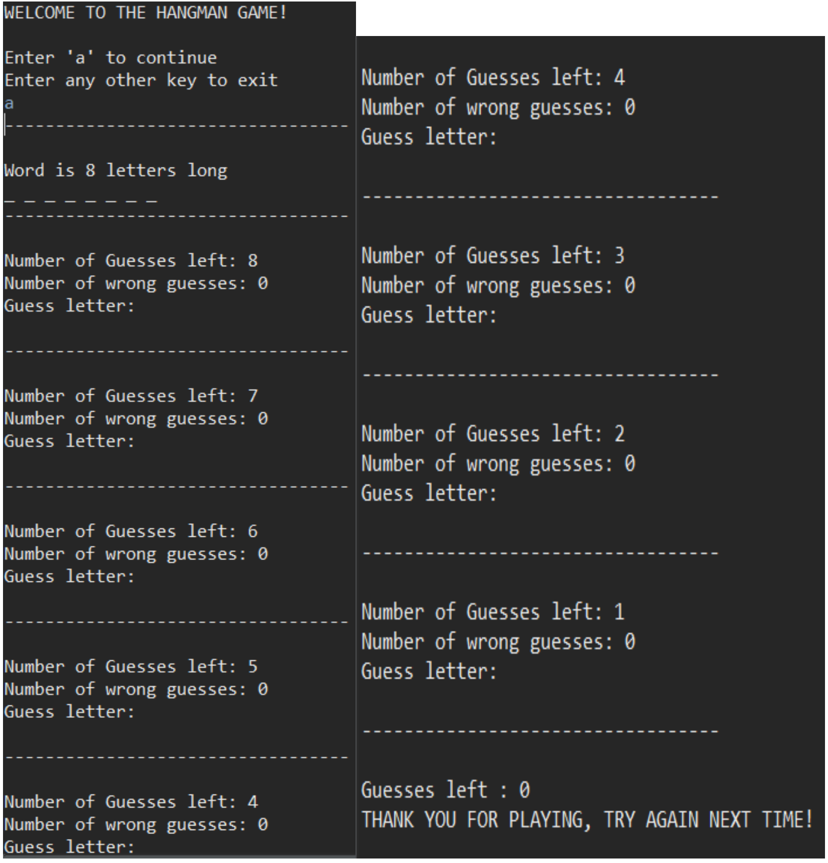
Plan for four iterations, including this. This is a fine-grained plan on what is to be done in each iteration and with what resources. To begin with, this is a plan of what we *expect* to do, update this part with *additions* (never remove anything) when plans do not match up with reality. Also make time estimates for the different parts.

In this course the overall planning has in some ways already been decided, so use the template to provide more details on specific tasks that define *your* project. Remember that you can plan to add features to any of the phases as long as the main focus is also met.

The first assignment is to complete iteration one.

5.1 Iteration 1

Example of execution:



```
WELCOME TO THE HANGMAN GAME!

Enter 'a' to continue
Enter any other key to exit
a
---------------------------------

Word is 8 letters long
_ _ _ _ _ _ _ _
---------------------------------

Number of Guesses left: 8
Number of wrong guesses: 0
Guess letter:

---------------------------------

Number of Guesses left: 7
Number of wrong guesses: 0
Guess letter:

---------------------------------

Number of Guesses left: 6
Number of wrong guesses: 0
Guess letter:

---------------------------------

Number of Guesses left: 5
Number of wrong guesses: 0
Guess letter:

---------------------------------

Number of Guesses left: 4
Number of wrong guesses: 0
Guess letter:
```

```

Number of Guesses left: 4
Number of wrong guesses: 0
Guess letter:

---------------------------------

Number of Guesses left: 3
Number of wrong guesses: 0
Guess letter:

---------------------------------

Number of Guesses left: 2
Number of wrong guesses: 0
Guess letter:

---------------------------------

Number of Guesses left: 1
Number of wrong guesses: 0
Guess letter:

---------------------------------

Guesses left : 0
THANK YOU FOR PLAYING, TRY AGAIN NEXT TIME!
```

5.2 Iteration 2

5 | Iterations

In this iteration you need to add some features to the game *but* after you have first modelled them using UML. All diagrams need to be included in the project documentation and should be implemented in the way mod- elled.

5.3 Iteration 3

You may include additional features to the game in this iteration, but the main focus is on *testing*. Plan, perform and document your tests in this iteration.

5.4 Iteration 4

The outcome of this iteration is *the complete* game. Reiterate the steps in iteration 1 – 3 for a set of new features but also remember to see the project as a whole, not only its parts.

6 | Risk Analysis

All projects face risks that make it important to prepare for what might happen. Use the chapters in the book as well as the content of the lectures to identify the risks within this project. As always, write down your reflections on creating a risk analysis. This reflection should be about 100 words.

6.1 List of risks

1. Testing is not done thoroughly, therefore user may input values which do not compliment the program.
2. Productivity is low as more time is taken than originally assigned for each task.
3. As the project is made, new features are included which make the code more complex and harder to accommodate later on.

6.2 Strategies

It is better to do testing in groups or teams as people have different ways to approach a program, therefore their implementation can be different. It is best to stick to the original schedule with minimal delay or tasks will be lest incomplete. The original structure must be completed before adding in new features to the program.

Reflection

While identifying the risks I took in the account the problems I had with previous assignments so that I did not repeat the same errors again. One of these was adding new features which I found quite difficult as sometimes I had to delete code and make a new one again, therefore it made me understand that it is better to plan the structure beforehand. Another issue is time management, sometimes assignments take longer to finish as there are unexpected mistakes, so it is better to start earlier.

7 Time log

Activity	Estimated Time	Time Taken
Assignment 1	8 days	8 days
Assignment 2	12 days	
Assignment 3	26 days	

8 Handing in

All assignments have a number of files to hand in. The overall advice is to *keep it simple*. Make it easy for the reciever to understand what the files are by using *descriptive* file names. Use as *few* separate documents as possi- ble. Always provide a *context*, that is *do not* send a number of diagrams in “graphics format”, but always in a document where you provide the pur- pose and meaning of the diagrams. Remember that the “reciever” is in reality a customer and as such has very little knowledge of the diagrams and documents – always provide context that make anything you hand in understandable to a non-technical person.

To hand in an assignment, make a git release and hand in the link via
Moodle to that release.