# Software Development Project

**SINISA GVOJIC**

Linnaeus University, Sweden

7/2/2019

# Table of Contents

# 1 Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| 7/2/2019 | 0.25 | First project iteration and first version | Sinisa Gvojic |
| 18/4/2019 | 0.25.1 | Retake of the first iteration | Sinisa Gvojic |

## 2 General Information

| Project Summary | |
|---|---|
| Project Name | Project ID |
| Hangman | Sg222xg |
| Project Manager | Main Client |
| Sinisa Gvojic | People interested in learning either the Swedish or English language |
| Key Stakeholders | |
| Project Leader, Project Manager, Head Developer, Head Tester, End user | |
| Executive Summary | |
| The project is made for educational purposes for people who wish to learn proper spelling in either English or Swedish. The project will include several features other than the base game. It will be created through four iterations, with the final one being delivered on the 22/3/2019. | |

# 3 Vision

To design and develop a system based on the popular Hangman game. The system, from now on referred to as “the game”, will be text-based rather than creating a GUI with a graphical representation of the man being hanged. This is done due to time, knowledge and resource constraints under which the game is developed. The game will be developed using the Eclipse IDE, as the developer is most familiar with this environment. The user, henceforth referred to as “the player” will be presented with lines corresponding to the number of letters a word contains. The task of the player is to guess the letters within a certain limit of attempts in order to win and then will be given a score. ~~The score system will function as such: the player starts off with 1000 points and with each correct guess will be awarded with another 500, however if the player does not guess correctly 300 points will be subtracted instead.~~ The scores will be stored in a text file along with the name of the player. The game will have two languages available, Swedish and English. The player can select a language and then the word to be guessed will be in the selected language~~. The game will implement a multi-player option where the players will take turn to guess the randomly selected word~~. Additional features will include optional user registration, a survival with a time limit as well as a feature to add more words to the word pool.

## 3.1 Reflection

The vision for this project was created to display an overarching depiction of what the development of the project will look like, as well as what features and requirements will be necessary for the project to run as intended. It is written with the general, not-detailed idea of what features will be included in the game, as well as the developer’s expectations of what the final project will look like.

# 4 Project Plan

There will be 4 iterations of the project. The first iteration, with the due date 8/2/2019 will include the basic functions of the program. The basic functions include generating random words in the Swedish and English language, giving the user a certain amount of guesses and displaying a message whether or not the user won the round of the game or not. The second iteration, due to the 22/2/2019 will expand on the first by including additional features such as drawing the man to be hanged from different keyboard characters and counting the score and storing it along with the name the user input before starting the round. The third iteration will expand on the user and score side of the system by implementing user registration with an email and a single-player high score system, as well as test both manually and automatically the implemented features by the 8/3/2019. The final project is to be delivered by the 22/3/2019 at 23:55.

## 4.1 Introduction

The Hangman game is a Java program that allows the user to play a game of hangman in a text-based fashion.

## 4.2 Justification

The game is being developed in order to help people learn the correct spelling of words in either Swedish or English, since non-native speakers of the languages are the target audience

## 4.3 Stakeholders

Project manager – the person in charge making a schedule of the planned courses of action in regards to developing the project as well splitting resources to help develop the project faster and in higher quality

Head developer – the person in charge of making sure that the project requirements are fulfilled in a timely matter as well as communicating with the project manager and negotiating the proper course of action in case of an unpredictable circumstance (see Risks). Additionally, the head developer is in charge of making the code clear and well structured for easier maintainability

Head Tester – the person in charge of seeing through that the project has been thoroughly tested, both manually and via automated tests. The head tester communicates with the head developer to sort out any bugs or flaws in the source code.

## 4.4 Resources

The development and testing of this game will be done in the Eclipse Java IDE, using Java 1.8.201, installed on an Acer laptop with an i7-7700 processor, 16GB of RAM and 256GB of SSD memory. Additional resources for development will be gathered utilizing developer knowledge and the available material.

Note: The user does not need a GPU to play this game

## 4.5 Hard- and Software Requirements

As stated in the resources, the game will be developed in a Java IDE, and it being a text-based game played in a terminal, it does not require significant computing power regarding graphics. The user has to have java 1.8 installed on his/her device in order for the game not to run into any problems while running

## 4.6 Overall Project Schedule

The first iteration and the project skeleton is due to be delivered by the 8/2/2019, and the expansion with added additional core features(main menu and the option to quit the game without just closing the terminal) is to be developed, tested and delivered by the 22/2/2019 along with the use case documentation and the use case, class and state machine diagrams. The next iteration, due to the 8/3/2019 will implement more features such as the option to register users log in and track high scores. The project iteration will be delivered, alongside testing documentation showing both manual test cases and automated tests and their respective result.

## 4.7 Scope, Constraints and Assumptions

Under the scope of this project falls the development and deployment of the game. The game will not be released for sale/download on any online/digital game store nor will it be sold as physical copies. The features of the game will include selecting the language for the word to be guessed (English or Swedish), a high score count, user registration via email, multiplayer, survival mode as well as the option to add new words.

The main constraint for the development of the game is time, as there are very clear and strict deadlines in order. That being said, with the lack of time comes a lack of top quality. The game will be text based with no graphics added through another program, such as Unity™. Seeing as the game is to be developed by a single student, there are no budget constraints for the game development. This is due to a lack of employees and other personnel related expenses.

The user/player of the game is assumed to have a functional computer that is able to run java programs with a mouse and keyboard connected to the device.

## 4.8 Reflection

Writing the project plan took a large portion of the time, strictly due to checking and re checking the deadlines listed on the MyMoodle page and making sure everything was detailed. This, however is done for the sake of precision which was lacking the first time I wrote this document.

# 5 Iterations

## 5.1 Iteration 1

Iteration 1 features the basic functions of the game. The basic functions include giving the user a set number of attempts to guess all the letters. The words will be randomly selected from the appropriate file in either Swedish or English. The player will then guess a letter one by one until he/she either guesses all the letters or the number of guesses reaches a certain threshold. This bare skeleton of the program is to be delivered by the 8th of February this year.

Tasks:

- Read chapters 2, 3, 22, 23 in the coursebook – getting familiar with the process of developing software. Estimated time for each chapter: 150 minutes
- Write vision – coming up with the software and how it is supposed to roughly function. Estimated time: 60 minutes
- Write project plan – a detailed plan for the project, excluding the writing of the source code. Estimated time: 120 minutes
- Write the first iteration plan – a fine grained plan about the tasks needed to be completed before submitting the first iteration of the project. Estimated time: 90 minutes
- Write a list of risks – potential factors that can negatively impact the development of the project. Estimated time: 45 minutes
- Write a time log. Estimated time: 30 minutes
- Write the method for the English word set in hangman. Estimated time: 30 minutes
- Write the method for the Swedish word set in hangman – same method as English, just the file name is different. Estimated time: 15 minutes
- Write the basic game source code. Estimated time: 50 minutes

# 6 Risk Analysis

## 6.1 List of Risks

| Risk | Probability | Effects |
|---|---|---|
| | | |
| The developer is ill during the key times in the development cycle | Low moderate | Tolerable |
| A critical flaw in the code is discovered with no possible way of handling it without affecting the overall code structure | Extremely low | Catastrophic |
| The device on which the project is developed is irreparably damaged in some way | Extremely low | Serious |
| The developer gets in a serious traffic accident | Low | Catastrophic |
| Environment used to develop the project does not generate code properly | Low | Insignificant |
| The developer needs to urgently return to his home country | Extremely low | Catastrophic |
| The workload is underestimated | Moderate | Serious |

## 6.2 Strategies

The developer is ill during the key times in the development cycle – appropriate medication will be provided to mitigate the symptoms of the illness, ensuring the timely delivery of the project

A critical flaw in the code is discovered with no possible way of handling it without affecting the overall code structure – structuring the code in such a way that no hard-coded component causes a fatal flaw

The device on which the project is developed is irreparably damaged in some way – regular antivirus scans are to be done and the device is to be kept in a safe container to avoid hardware damage

The developer gets in a serious traffic accident – reflective bands are provided to the developer to improve visibility. The developer is to avoid travelling by foot on busy roads during rush hours

Environment used to develop the project does not generate code properly – regular updates are to be installed to minimize the chances of code not being generated

The developer needs to urgently return to his home country – in case of this occurring the code will be uploaded to github for the possible continuation of project development

The workload is underestimated – a detailed analysis of the tasks is to be done and a work plan is to be assembled

## 6.3 Reflection

The list of risks was assembled by thinking about every-day-life that could actually have an impact on the project development. After the those, I moved on to thinking about the technical risks. As a one person project, risks such as specialists in different fields not being available and unable being to recruit qualified staff are not applicable, and many risks that usually involve multiple people are non existant.

## Time Log

First iteration/game skeleton:

| Action | Estimated time | Actual Time |
|---|---|---|
| Reading chapters 2, 3, 22, 23 in the coursebook | 150 minutes per chapter | On average 90 minutes per chapter |
| Writing the vision | 60 minutes | 73 minutes |
| Writing the project plan | 120 minutes | 158 minutes |
| Writing the first iteration plan | 90 minutes | 82 minutes |
| Writing the list of risks | 45 minutes | 52 minutes |
| Writing the time log | 30 minutes | 25 minutes |
| Write the method for the English word set in hangman | 30 minutes | 26 minutes |
| Write the method for the Swedish word set in hangman | 15 minutes | 4 minutes |
| Write the basic game source code | 45 minutes | 48 minutes |

Reflections:

- Reading the book took less time than expected since I played it safe and assumed it would take a significantly longer period to read the material
- Writing the project went above the estimated time due to the lack of experience in planning for a project
- Writing the method for the Swedish word set took less time than expected because it's mostly a copy-paste of the English word set method