SOFTWARE DEVELOPMENT PROJECT – Hangman game

José Duarte de Sousa Barroca

2019-02-08

Contents

1 – Revision history

Date	Version	Description	Author

2 – General Information

<table>
<tr><th colspan="2">Project Summary</th></tr>
<tr><td>Project Name</td><td>Project ID</td></tr>
<tr><td>Hangman</td><td>JBarrocaHangman2019</td></tr>
<tr><td>Project Manager</td><td>Main Client</td></tr>
<tr><td>José de Sousa Barroca</td><td>Linnéuniversitetet</td></tr>
<tr><td colspan="2">Key Stakeholders</td></tr>
<tr><td colspan="2">Project manager – José Barroca
Project planning – José Barroca
Project development, validation, evolution – José Barroca
Client – Teachers and teaching assistants of the course 1DV600 – Software technology of Linnaeus University, Sweden</td></tr>
<tr><td colspan="2">Executive Summary</td></tr>
<tr><td colspan="2">This project consists of implementing a game of Hangman in the console using the programming language Java. In this single-player game, the player will be able to compete in single games or 5-round games and will be challenged by a score system that rewards guessing harder words in fewer tries and will be motivated by persistent high score tables.</td></tr>
</table>

3 – Vision

This project's goal is to create a console-based Hangman game, written in the programming language Java. When entering the application, the user can select (using the keyboard, as for all further interactions) either to start a game, view the high scores table or quit the application. If the player decides to start a game, he/she may choose to play a single word or a 5-word game. When a new game starts, a random word is chosen from a text file containing words in the English language, and is presented to the player with each of its letters replaced by underscore signs. At the beginning of each game, the hangman drawing will be empty, and the player will have a total of 10 chances to guess the letters included in the word. With each correct guess, the guessed letter is revealed in its position in the word, whilst the others remain replaced by underscore signs. With each incorrect guess, a new section of the hangman drawing is revealed and the number of remaining guesses is decremented. There are 10 different sections of the drawing (corresponding to the 10 tries): ground, vertical pole, horizontal pole, rope, head, body, right arm, left arm, right leg, left leg. At any time during the game, the player may guess the entire word by writing it in the console. If the word is correctly guessed, the game is won. After 6 wrong guesses, the drawing is complete and the player loses, revealing the word, the score and the option to either quit the application or start a new game. The score is calculated based on the player's guesses (right guesses increase score, wrong guesses decrease it), an extra bonus for guessing the correct word before finding all the letters and on a word-specific difficulty index based on the relative frequency of each of the word's letters in the English language. The score of a 5-word game will consist of the sum of the scores for each of the 5 words. When the player obtains a high-score (defined as one of the 5 highest scores in the record), the player is requested a username, which will be saved together with the high score. A different high scores table will be kept for single games and 5-word games.

Reflection on writing a vision statement: Writing a vision statement should encompass the goal of the project and summarize what everyone involved in the project should know and aim for. Since I have never had any contact with project management/planning and vision statements before, I felt the need to search for examples of vision statements. I noticed the content, style and idea behind vision statements in different contexts can vary quite a bit. For this project, we were supposed to write a description of the whole system, but in many other examples vision statements are short, captivating messages which strive to engage possible clients/users/investors in the product, and not so much descriptions of the product itself. If the goal of the vision statement is to establish a common ground between every team member, then I agree that a more objective and descriptive text is more useful. But if the goal is to captivate and capture outside interest, than I understand the need for more concise, appealing and innovative messages.

4 – Project Plan

4.1 Introduction

This application is an implementation of the "hangman" game in the console, written using the Java programming language. In this game, the player has a certain number of attempts to guess a random word by guessing its letters, one at a time. For each failed guess, a portion of a drawing of a hanging man is produced and, if the drawing is completed after a certain amount of attempts without the word being guessed, the player loses the game.

4.2 Justification

This application is a part of a project in the course 1DV600 (Software Technology) of the Software Development and Operations program from Linnaeus University, aimed at providing a first practical experience with structured project planning.

4.3 Stakeholders

I will be responsible for project planning and management (including activity planning, risk management and scheduling) as well as for the project's development (including specification, development, validation and evolution). Being an educational project, I consider the faculty's teachers who will be grading this project as 'clients', since the request for this project came from their behalf and they were the ones who defined the project's initial constraints and goals (just as the client usually defines at the start of a software project). This project lacks a specific end user, but, given that the end product is a simple game of Hangman, the end user will be everyone interested in playing it, which includes both myself and the 'clients' (teachers). Since there aren't any further managerial or financial ramifications to the project, there are no further stakeholders.

4.4 Resources

Coding will be done using the IDE Microsoft Visual Studio Code and Java as a programming language (requiring, therefore, an installation of JDK as well as several Visual Studio Code Extensions to support development in Java). Git version control will be done using Visual Studio Code's integrated terminal to update the project's repository in GitHub. An internet browser (Google Chrome, to be more specific) will be used for other miscellaneous tasks, such as communication with the project's "clients" and handling of project deliverables via the course's myMoodle platform. The project plan will be edited using Microsoft Office Word.

Regarding hardware, other than my own personal computer (either a laptop running Linux or an iMac running Mac OS X), no other hardware devices will be used.

4.5 Hard- and Software Requirements

To use this product (a text-based game which runs in the console), there are no software requirements other than an operating system with a JRE allowing for the execution of java programs. No specific hardware is required either other than a personal computer with the previously mentioned software installed.

4.6 Overall project schedule

2019-02-08	Project plan, GitHub repository, skeleton code
2019-02-21	Requirements, UML behavior- and structure modelling
2019-03-08	Testing
2019-03-22	Complete project

4.7 Scope, constraints and assumptions

Scope: As mentioned in the project vision, the game will include a main menu, two different game modes (single game and 5-round game) and separate high score tables for each game mode, with data persistency between different uses using text files. This game is to be played locally by only one player and will not have any kind of multiplayer functionality. This game will be implemented as a console application and, as such, will not have a graphical user interface besides the text and symbols allowed in the console. The game is a purely text-based application and will not include any type of media files such as images, videos or audio files.

Constraints to the project: Since this is not a commercial project, its realization is not dependent on any budget and there are no financial constraints to consider. Time, however, will be an important constraint, since I have an infant son which requires my care during the day, so I depend mostly on late afternoons and evenings to develop this project (as well as remaining university work). A related constraint is my lack of knowledge – I'm currently learning everything about project management for the first time, so I need to devote a lot more time than probably necessary to planning this project, while also requiring some time for the development tasks due to my inexperience with programming Java.

Assumptions: This project assumes the end user is able to perform basic tasks with their personal computer (such as turning it on and using its operative system) and has sufficient knowledge with a terminal application (or an IDE) to be able to run the program and interact with it through key commands.

Reflection on writing a project plan: Writing a project plan is an essential step when developing a project. The project plan is created at the beginning of a project with the goal of structuring the work that has to be done in different activities, assessing the time and effort required for each, assigning them to team members and anticipating risks and strategies to deal with them. This project plan is reassessed and updated throughout the entire project. This is my first contact with elaborating a project plan, but I can already understand the enormous asset it is in a project, especially if it is a complex project involving many team members and many different tasks. I was able to reflect on my project in a much more structured and objective way thanks to the planning that this document requires, and now that this first version of the document is created I feel I have a solid foundation to organize all the work that lays ahead.

5 – Iterations

5.1 Iteration 1

2019-01-23 to 2019-02-08

Planned activities and estimated time:

- Writing project plan
  - Vision (1 hour)
  - Project plan subcategories (2 hours)
  - Iteration planning (1 hour)
  - Risk analysis (1 hour)
  - Time log (30 minutes)
- Creating project GitHub repository, sharing it with teachers (15 minutes)
- Writing and pushing skeleton code (15 minutes)
- Pushing project plan to the GitHub repository as a documentation file (<15 minutes)
- Doing a release in GitHub (<15 minutes)

5.2 Iteration 2

2019-02-09 to 2019-02-21- Develop a working game (includes requirements, modelling behavior and modelling structure)

5.3 Iteration 3

2019-02-22 to 2019-03-08 - Testing

5.4 Iteration 4

2019-03-09 to 2019-03-22 - Finished product

6 – Risk analysis

6.1 List of risks

Risk	Probability	Impact
Estimation risks: - Underestimate the time required to develop the software - Underestimate the degree of difficulty associated to specific features of the project's implementation	High Moderate	Serious Tolerable
People risks: - I become ill and unable to work on the project - I don't have enough knowledge to develop the project according to planned specifications	Low Low	Catastrophical Serious
Requirement risks: - Changes to requirements that require major design rework are proposed	Moderate	Serious
Technology risks: - I lose access to my project files - The hardware I use to develop the project stops functioning/becomes inaccessible	Low Low	Catastrophical Serious
Tools risks: - Microsoft Visual Studio Code IDE's Java Extensions don't work as expected	Low	Tolerable

6.2 Strategies

Risk	Strategy
Estimation risks	Anticipate which requirements are non-essential and can be sacrificed in case time becomes critical.
People risks	No avoidance or minimization possible against staff illness in this case (only me). If encountered, be prepared to discuss with the course's staff the possibility of an extended deadline for a deliverable or the need for a re-take of the project.
Requirement risks	Discuss the changed requirements in detail with the client (course's staff) in order to re-plan and resume development as soon as possible.
Technology risks	Using a light, open source IDE, version control and an online github repository minimizes the dependency on a specific computer (in case I loose access to my home computers, it's easy to install the needed tools in another unit to carry on with the project). Using cloud-based tools such as Microsoft Office Online and OneDrive allows me to also work in the project's plan independently of any specific hardware unit and without needing to physically backup my files.
Tools risks	If any problem occurs due to the use of Microsoft Visual Studio Code with Java Language extensions, I can easily change the IDE to Eclipse and use Git version control via a separate terminal application.

Reflection on risk analysis: When planning a project, it is very important to plan ahead for the many different types of risks that can happen. This allows the project manager to anticipate strategies for handling these different risks in order to minimize their impact on the project. Although some categories can be defined for the different types of risks, this is an aspect of project planning that is not entirely objective, and it's easy to understand how much this process depends on the project manager's experience when you have no experience at all and try to plan this step. However, trying to apply risk management principles to this particular project, which is very different than the examples one usually reads about (large software developers with multiple teams, etc.) definitely makes it more understandable.

7 – Time Log

Activity	**Anticipated time** (hours:minutes)	**Real time** (hours:minutes)
Project plan – Vision	01:00	01:15
Project plan – Section 4 (sub-categories)	02:00	04:30
Project plan – Iteration planning	01:00	00:45
Project plan – Risk analysis	01:00	01:30
Project plan – Time log	00:30	00:30
Creating and sharing GitHub repository	00:15	00:15
Writing and pushing skeleton code	00:15	00:15
Pushing project plan into GitHub repository	<00:15	<00:15
Doing a release in GitHub	<00:15	<00:15