Terminal Hangman Project Plan

Viktor Ödman
Linnéuniversitet
1 februari 2020

1 | GENERAL INFORMATION

PROJECT SUMMARY	
Project Name	Terminal Hangman
Project Manager	Viktor Ödman
Main Client	Linnéuniversitetet
Executive Summary	The product is going to be a single player text-based console game. The game is going to be based on the traditional pencil and paper game "Hangman" The product is going to be developed by me. The project is going to be spilt up to 4 iterations.

2 | REVISION HISTORY

DATE	VERSION	DESCRIPTION	AUTHOR
01/02/2020 - 18:00	Iteration 1	Added Vision and Vision reflection	Viktor Ödman
02/02/2020 - 19:05	Iteration 1	Made Changes in the Vision Document	Viktor Ödman
02/02/2020 - 19:11	Iteration 1	Updated Vision reflection	Viktor Ödman
02/02/2020 - 19:30	Iteration 1	Added The project plans introduction.	Viktor Ödman
02/02/2020 - 21:32	Iteration 1	Added Justification, Stakeholders and Resources	Viktor Ödman
03/02/2020 - 11:00	Iteration 1	Added Hard and Software Requirements. Also added Overall project schedule.	Viktor Ödman
03/02/2020 - 14:40	Iteration 1	Added Scope, Constraints and Assumptions	Viktor Ödman
03/02/2020 - 17:20	Iteration 1	Added Iterations and Risk Analysis	Viktor Ödman
03/02/2020 - 17:45	Iteration 1	Added Time Log and a reflection to the project plan.	Viktor Ödman

3 | VISION

My vision for this project is to be able to plan, design and develop a simple console application based on the traditional paper and pencil game "Hangman".

The game will differ from the traditional game in some ways. Traditionally the game is played by several players, but this game is going to be a single player game. The word is not going to be chosen by the player, but will be generated from a list of words.

Even though the application is limited to the console, and is not going to have a graphical user interface, the application will be easy to understand and easy to use. By using a text-based interface in the console, the user will have an easy time navigating though the start menu and the game.

The project will be developed by me, using techniques that i've acquired from this course. The game will be developed using techniques that I've acquired from previous courses.

My hopes with this project is to go from just developing an application with almost no strategy, (as I have done in previous applications in previous courses) to be able to develop an application by seeing it as a project.

3.1 | Vision reflection

When creating the vision I tried to present a simple overview of what is going to be developed without going into too much detail. In the vision I was describing what the final product should be, how it should work, how the user should use the product, what the user will see, who is going to develop the project, why I'm developing the project and what hope to get out of this project.
The difficult part of writing the "Vision" was to know what details that should be included and what should be left out.

4 | PROJECT PLAN

4.1 | Introduction

The application that I'm going to create is a text-based console game that is based on the old pencil and paper game "Hangman" https://en.wikipedia.org/wiki/Hangman_(game). Traditionally the game is played by at least two players, where one player comes up with a word and the other player tries to guess the word letter by letter.

In this game the objective is going to be too guess a word based on a number of underscores, each underscore representing a letter and the letters combined will represent a word. This application will be a single player game, where the words are randomly picked from a list.

There will be at least two wordlists, each following a specific theme. The first list will be a list with all the countries in Europe. The second list will be a list of car brands.

There will be a menu when starting the game where the user should be able chose what list the word should be generated from and then start the game. The game is going to randomly pick a word from the chosen list.

The players task will be to figure out what word that the underscores represent by guessing a letter. After each guess, if the word consist of the guessed letter, the underscore/underscores that represented that letter will be changed to that letter. If the guessed letter is not included in the word, a part of the building of a man getting hanged begins. Each part of the man represents the number of wrong guesses allowed to guess the word.

4.2 | Justification

The reason for creating this application is to get experience on how to develop an application from the perspective of a project. To learn how to plan, structure and design the development of an application. So hopefully i've learnt something new after this project is over.

4.3 | Stakeholders

Name	Role
Viktor Ödman	Project Manager / Developer

4.4 | Resources

The available resources for this project is:

- About 20 hours a week in case of time available for this project.

- Knowledge gained from this course.

- Software Engineering - TENTH EDITION by Ian Sommerville (Book)

4.5 | Hard and Software Requirements

For development the hard and software used was:
- My computer.
- A keyboard and mouse
- The latest version of Node.js.
- Visual studio code.
- A bash terminal.

Hard and Software requirements for running this application:
- The latest version of node
- Any pc that can run node applications.
- A keyboard
- A terminal/console

4.6 | Overall Project Schedule

The schedule for this project will be based on the different deadlines for this course. The deadlines for this course is split in to 4 iterations spanning over 9 weeks. The different iterations are:

Iterations	Weeks	Theme
Iteration 1	4-5	Process and Planning
Iteration 2	6-8	Modeling and Software Design
Iteration 3	9-10	Software Testing
Iteration 4	11-12	Final iteration

4.7 | Scope, Constraints and Assumptions

This application will be a single player game based on the traditional pencil and paper game hangman. The game will only be played in a console/terminal environment, and will not have any graphical user interface. Here are some Requirements, Constraints and Assumptions.

4.7.1 | Requirements

Functional requirements

- When starting the application the user shall be greeted with a start menu.
- One menu item shall lead to a description on how to play the game.
- The words shall be randomly picked from at least two lists.
- Every wordlist shall follow a theme.
- The user shall be able to choose what list the word should be picked from.
- The letters of the word shall be displayed as underscores.
- The maximum number of wrong guesses before the player loses the game shall be 8.
- The parts of the man getting hanged shall be represented by the maximum of wrong guesses.
- When the guessed letter is correct, the underscores that represent that letter shall be changed to that letter.
- When the guessed letter is incorrect, a new part to man getting hanged shall be added.
- If the user is able to guess the full word, the game shall present that the user has won.
- If all the parts of the man getting hanged is added, the game shall present that the user has lost.
- The user should be able to play only using the keyboard.

Non-functional requirements

- The code should follow the code and JSDoc conventions of https://www.npmjs.com/package/ @lnu/eslint-config.
- There should a describing README file on how to install the application and how to start the application.
- While being in the game directory, the user should be able to install the game and it's dependencies by running "npm install" in the console/terminal.
- While being in the game directory, the user should be able to start the game by running "npm start" in the console/terminal.

4.7.2 | Constraints

- The game will be a text-based console application and will not have any graphical user interface.
- The application will be written in javascript using the node.js platform.
- The time for developing this application will be limited to the timeframe of this course.
- Limited knowledge and experience in project planning.

4.7.3 | Assumptions

- That I will be able to allocate at least 20 hours a week for this project.
- That node and the npm dependencies that I use will still work when the assignment is done.
- That the second course that I'm attending, will not interfere with this project.
- That the limited knowledge that I have in project planning will be enough to finish this project.

4.8 | Project Plan Reflection

In the project plan I tried to paint a complete picture of what the actual application should do when the project is done. I also tried to describe how the application should work, who is involved and why the application should be developed. I think that this part of the project documentation will really help me to define the application and make it easier to design and develop it. I not sure if was supposed to add functional and non-functional requirements, but I thought it was fitting when describing the scope of the application.

5 | ITERATIONS

In this part I will be going through the different iterations of this project. I will try to make an estimate of what I will be able accomplish in the different iterations. I will also make some time estimations regarding the different parts within the iterations.

5.1 | Iteration 1

Week 4-5 Process and Planning(Both Time Estimation and Actual time are estimated in this iteration)

Tasks	Time Estimation	Actual Time	Description
Reading Chapters 2-3 and 22-23	12 hours	12 hours	The chapters includes information about Software processes, Agile software development, Project management and project planning
Watching pre-recorded lectures	3.23 hours	3.23 hours	The course provided lectures
Create Project Documentation	15 hours	15 hours	By following the project template
Add skeleton code	1 hour	1 hour	Add basic skeleton template

5.2 | Iteration 2

Week 6-8 Modeling and Software Design

Tasks	Time Estimation	Actual Time	Description
Reading Chapters 4, 5, 20 and 7.1	15 hours	Unknown	The chapters includes information about Requirements engineering, System modeling, System of systems and Object oriented design using the UML
Watching pre-recorded lectures	6 hours	Unknown	The course provided lectures
Update the project plan	Unknown	Unknown	Change the project plan with the new information gathered in this iteration.
Modeling and designing the application	Unknown	Unknown	Create UML diagrams and implementing code based on those diagrams

5.3 | Iteration 3

Week 9-10 Software Testing

Tasks	Time Estimation	Actual Time	Description
Reading Chapter 8	4 hours	Unknown	The chapter includes information about Software Testing
Watching pre-recorded lectures	4 hours	Unknown	The course provided lectures
Update the project plan	Unknown	Unknown	Change the project plan with the new information gathered in this iteration.
Software Testing	Unknown	Unknown	Testing the application

5.4 | Iteration 4

Week 11-12 Final Iteration

Tasks	Time Estimation	Actual Time	Description
Iterating the different iterations again	Unknown	Unknown	Define the remaining tasks to be done in the project
Update the project plan	Unknown	Unknown	Change the project plan with the new information gathered in this iteration.
Add new features	Unknown	Unknown	Add new features

6 | RISK ANALYSIS

Here I will cover some of the risks that might occur during this project and some strategies to avoid or minimize the impact of the risks.

6.1 | List of risks

Risk	Affects	Description	Effects	Probability
Not meeting deadlines	Project / Product	Not being able to hand in the assignments before the deadline.	Serious	Moderate
Tasks taking up more time then I estimated	Project / Product	Self describing	Tolerable	High
Computer Malfunction	Project / Product	My computer is malfunctioning and I can't get access to my project files.	Tolerable	Low
Staff illness	Project / Product	Im to ill to able to work on the project during the course.	Serious	Low

6.2 | Strategies

Risk	Strategy
Not meeting deadlines	Avoidance strategy: Put it the time that is needed to meet deadlines. Contingency plan: Hand in the assignment on a later occasion.
Tasks taking up more time then I estimated	Minimization strategy: Allocating more time from other Tasks.
Computer Malfunction	Minimization strategy: Doing frequent commits to GitHub so that my files are available elsewhere.
Staff illness	Contingency plan: Hand in the assignment on a later occasion.

7 | TIME LOG

Here I will estimation about how much time a particular task Is going to take in terms of learning about the task and implementing it. I will also keep track on how much time the task actually took, and compare it to the estimation.(This will not be done in iteration 1)

Task	Estimated Time	Actual Time	Analysis