SOFTWARE DEVELOPMENT PROJECT TEMPLATE

Loic Galland

_

1. Revision History

Date	Version	Description	Author

2. General Information

3. Vision

Create a vision document for the system. This should be a document covering about half an A4 page describing the system. The purpose of the document is to make sure that everyone involved in the project has the same vision of what is to be created. Use the “Assignment Overview” and previous subtasks as your source for what to write. In addition, write down your reflections on creating a vision document. This reflection should be about 100 words.

This project consists of creating the “Hangman” game. The basic idea for this game is that the player is going to guess a word by suggesting letter by letter. It will be implemented in a text-based fashion with Java code language. If a letter given is inside the predefined word, then the letter will be showed at the right position of the word. If the player guesses a wrong letter, the game is building a part of a man getting hanged. The player can have 8 letters wrong before the man is hanged and therefore losing the game. (ground, pole, head, body, left arm, right arm, left leg, right leg).

The game will also let the user decide whether to write a letter or try a word. If they get wrong letter or wrong word it will be considered the same and they will therefore go to another part of the hanged man. The players will also be able to create an account by entering a username and a password. They will be able to log in and see the amount of times they won and other statistics that could be important for them.

By creating account, a leader board can be implemented to show the best players with their names, amount of wrong to win the game and in how much time they managed to finish the game.

Reflection:

I found that writing the vision was helpful to see what I wanted in the application. It made me think about all the possibilities and all the problem that could come with the program. It also made me think more in details to the different functionalities I wanted to implement for the application. It also helps to see how much time, will be needed to create the all program. And allow us to create a more realist view of what our program could look like.

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.

The player needs to be able to:

- Draw hangman after every wrong try
    - At the 8th wrong letter = Game Over
- Enter a Letter and see if and where it is on the word
- Choose whether write a letter or whole word.
- When guessing whole word = included in count of wrong words
- Single and Multi-player modes
- The app should have a nice graphical representation of the game that updates when the user is given wrong letter or words

Reflection:

The Project Plan help me to focus more into what could be done and make a more realistic plan. It is an important step into the Application cycle and will make it easier for the developers to code this application. It also able us to follow a structure so that the programmer can follow a procedure.

4.1. Introduction

For this project, the “Hangman” game will be created. It will be implemented in a text-based fashion with Java code language. The basic idea for this game is that the player is going to guess a word by suggesting letter by letter. If a letter given is inside the predefined word, then the letter will be showed at the right position of the word. If the player guesses a wrong letter, the game is building a part of a man getting hang. The layer can have 8 letters wrong before the man is hanged and therefore losing the game. (ground, pole, head, body, left arm, right arm, left leg, right leg).

4.2. Justification

Why should the application be made?

Because it is an interesting application and it let us use our coding skills. This game is also part of an assignment from the course .

4.3. Stakeholders

List and define the different stakeholders for the project.

There is no stakeholders.

4.4. Resources

What resources are available and used to create the application?

The resources available to create the application are Java coding language, Eclipse, Word, the Internet and my brain.

4.5. Hard- and Software Requirements

Specify what is used to develop and later run the software developed.

My computer is used to develop the software with Java on Eclipse. And it will be run on Eclipse too.

4.6. Overall Project Schedule

What are the important dates for deliverables?

Week 12 is the date where the final version of the “Hangman” game needs to be delivered.

4.7. Scope, Constraints and Assumptions

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

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

The first iteration is this project plan along with some degree of implementation. Complete the documentation first so that the implementation goals are met in code. You need to implement an idea and some skeleton code for your project to work with. This is assignment one.

5.2. Iteration 2

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 modelled.

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

List the identified risks and specify, as far as possible, the probability of them happening as well as the impact they would have on the project.

6.2. Strategies

Prepare for the risks by having strategies for avoiding the risks as well as minimising the impact of them if they do occur.

7. Time log

Each assignment must be accompanied with a time log. This time log should contain the date, time and task to be performed. The reason for doing this is for you to get some experience in estimating your own time – creating a time log is one of the best ways of doing this. Take into account the time for learning and understanding of the problem when you plan the time. Make your planning with 15 minutes as the minimum unit. In the time log you start by planning the amount of time you believe a task will take and after it is done you mark the actual time. If every entry that has a difference in planned and actual time spend, analyse the time difference.

Assignment 1 – Time Log

Task to Do	Time Estimated (min)	Time Taken (min)
Planning	60	90
Implementation	60	70

Assignment 2 – Time Log

Task to Do	Time Estimated (min)	Time Taken (min)
Class Diagram	15	20
Fully Dressed Use Case	30	40
Use Case Diagram	15	15
State Machine	30	60
Implementation	60	90

8. Handing in

All assignments have several files to hand in. The overall advice is to keep it simple. Make it easy for the receiver to understand what the files are by using descriptive file names. Use as few separate documents as possible. Always provide a context, that is do not send diagrams in “graphics format”, but always in a document where you provide the purpose and meaning of the diagrams. Remember that the “receiver” is 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.