SOFTWARE DEVELOPMENT PROJECT: HANGMAN GAME

**Bengt Svedling Cruz**

*LNU SOFTWARE DEVELOPMENT*

*2/18/2019*



**1 | Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
|  |  |  |  |
|  |  |  |  |

Project Name – Version – Author – Date

**2 | General Information**

**3 | Vision**

The vision for this project is to develop a traditional hangman game of either one or multiple players. A hangman game consists of a user/player guessing letters within a hidden word, until the word is completed. The game should have different features as for example words focused on one theme or topic depending on the user’s choice and leader boards with the different players involved in the game. A traditional hangman game includes graphics in which every time the user misses a letter a figure of the hangman appears. Steadily adding each “human” part or limb. Usually taking up to 6 chances before the whole human figure is completed and the man is “hanged”. Meaning the game has come to and end and the user/player has lost the game. With extra features like showing which is the actual word being guessed, in the case of the user losing the game. Developing the game should give a clear view to the individuals involved in the production of the game on what the user should feel and experience while running the game.

Reflection:

Writing the vision section of this project document has made me reflect on cutting down on future ideas I had planned for the developing of the game and focus more in the basic features mentioned above. Focusing mainly in the developing of a better-looking and functioning program or game in other words. Main point of focus are for instance: interesting graphics that make the game lifeful and more enjoyable for the user, and the developing of a game with minimum flaws or so called lags.

**4 | Project Plan**

1. Coming up with a basic skeleton code for the game.
2. Trial and error.
3. Main user features specification.
4. Feature completion.
5. Game completion.

Reflection:

Reflection on creating the project plan for this project helped aim and identify the single details and steps on this project. Going from simple user steps in which the user has to guess a single letter at first and keep checking different letters in 6 chances before the man is hanged. In such order:

* User starts the program.
* Then chooses to either start the game or exit the game.
* User chooses theme for the word that is going to be guessed.
* User starts guessing letters for the word.
* If letter doesn’t fit the word, hangman figure starts building.
* From head to legs. Head, body, arms (2), legs (2).
* If the user manages to guess all the letters I the word a message will appear showing the user has won the game.
* If the user doesn’t manage to finish the game before six tries, the gam should show the hidden word being guessed.
  1. **Introduction**
* The development is a simple and traditional hangmn game. Implemented in a text based fashion with java code language.
  1. **Justification**
* First of all most, it is a fun and productive way of developing a game in java language. Second, it is also a simple way of testing our skills and delivering a so called “product” for a user or a consumer.

* 1. **Stakeholders**

-Me, my self and I. (No stakeholders.)

* 1. **Resources**
* Java language using Eclipse Integrated Development Environment (IDE). Research done in internet navigation mostly.
  1. **Hard- and Software Requirements**
* Java language runned by Eclipse.
  1. **Overall Project Schedule**
* Deadlines:

1st Assignment = week 6

2ndAssignment = week 8

3rd Assignment = week 10

4th Assignment = week 12

* 1. **Scope, Constraints and Assumptions**
* N/A.

**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.

* 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.

* 1. 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.

* 1. 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.

* 1. 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 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.

* 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.

* 1. 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.

Estimated Time Plan:

|  |  |  |
| --- | --- | --- |
| Time | Day | Task |
| 2 hours | 7/02/2019 | Assignment 1 Part 1 of Hangman Code |
| 1 hour | 7/02/2019 | Assignment 1; 4 | Vision and 5 | Project Plan |
| 2-3 hours | 16/02/2019 | Assignment 2 Implementation  Part 2 of Hangman Code |
| 2 hours | 19/02/2019 | Assignment 2; Understanding Use Cases, Diagrams and Modelling. |
| 1 hour (16-17) | 19/02/2019 | Assignment 2; Task 4 Class Diagram. |
| 1 hour (21-22) | 19/02/2019 | Assignment 2, Tasks 2,3 and 4 adjustments. |
|  |  |  |
|  |  |  |

Actual Time Plan:

|  |  |  |
| --- | --- | --- |
| Time | Day | Task |
| 1 hour | 7/02/2019 | Assignment 1 Part 1 of Hangman Code |
| 1.5 hours | 7/02/2019 | Assignment 1; 4 | Vision and 5 | Project Plan |
| 1.5 hours | 18/02/2019 | Assignment 2 Implementation  Part 2 of Hangman Code |
| 2.5 hours | 18/02/2019 | Assignment 2; Understanding Use Cases, Diagrams and Modelling. |
| 5-6 hours | 18/02/2019 - 19/02/2019 | Assignment 2; Task 1, 2 and 3. |
|  |  |  |
|  |  |  |
|  |  |  |

Time Difference Analysis:

|  |  |
| --- | --- |
| **Time Difference** | **Task** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**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.