1DV600 - The Hangman Project

Melina Cirverius
Linneaus University

2019-03-08

Contents

1 Revision History

Date	Version	Description	Author
2019-02-19	v1.1	Updated 5.2 – Iteration 2, and 7.2 – Time log for iteration 2	Melina Cirverius
2019-03-08	v1.2	Updated 5.3 – Iteration 3, and 7.3 – Time log for iteration 3	Melina Cirverius

2 General Information

Project summary	
Project Name	Project ID
The Hangman Project	mc222ap–1DV600
Project Manager	Main Client
Melina Cirverius	The Gamer
Key Stakeholder	
Project Manager, Development Team, End user.	
Executive Summary	
This project is focused on creating a hangman game. The main part is the planning process including this project plan. The goal of the project is the creation of the application. The work with the application will include modelling and testing.	

3.1 Vision

The goal for this project is to create a version of the hangman game. This will be a text based game played using the command line. At the start of the game the player will be greeted by a menu where they can begin a new game. The game will choose a random word from a list each time the player starts a new round. As an extra feature the game will have the words divided into three different difficulties and have the option for the player to choose the difficulty level at the start of the game.

The game will show the number of letters to be guessed in the randomly picked word, displayed as underscores, and the number of guesses available. The player can only guess one letter at the time and if the letter guessed is correct, the word will update to show where in the word this letter is situated. If the letter is present at more than one place in the word the game will show all occurrences of that letter. The player then continues by guessing again with a new letter.

If the player's guess is wrong, part of the hangman will be drawn and then they get to guess again. The number of guesses available for the player is determined by the number of parts present in the hangman.

The game continues either until the player is out of guesses and the whole hangman is drawn, or the player has guessed the whole word correctly. If the player manages to guess the whole word correctly they will get the option to either continue with a new random word or to end the game. If the player chooses to end the game they will be presented with a result screen where they can see how many words they have guessed correctly in a row, and the least number of guesses they needed to finish a word.

If the player fails to guess the word before the whole hangman has been drawn the game is over and they can start a new game. At this point a result screen will appear and show the player how many, if any, words that they guessed correctly. The result screen will also include a highscore list.

This project will be created in Java and the game will be played in the command line. The application will be built in such a way that the player can play as many rounds as they want of the game or until they hit Game over.

3.2 Reflection - Vision

Writing the vision document took some time as it was it was difficult to define the outline of the project. In the end I found that it was easier to just described how I wanted the finished hangman game to work with the different aspects and extra features I want to be implemented in the game if time permits. Having this vision written down in a clear way will help the project to move forward in terms of planning the work to be done. It will also be a huge help when it comes to actually building the game and implement the different aspects of it.

4 Project plan

4.1 Introduction

This project consists of the creation of a hangman game. This project plan details the different parts of the project.

4.2 Justification

The purpose of this project is learning how to properly execute a project according to a plan.

4.3 Stakeholders

The project manager wants to follow the set out plan and have a properly working product to precent to the client on time.

The development team wants to create an exciting hangman game with well structured code and zero bugs.

The end user wants a fun and creative game to play.

4.4 Resources

The main resource for this project is the time allocated for the course. The resource in terms of personnel is me since I am the only developer working on this project and I am also both the project manager and designer of the application. The resources in terms of research are selected chapters in the book *Software Engineering* by Ian Somerville as well as recorded lectures on related subjects.

4.5 Hard and software requirements

The software developed in this project will be made using a six year old laptop that will hopefully last throughout the project. The application will be developed in IntelliJ and written in Java. The same IDE will be used to play the game. For version control I will use Git and Github.

4.6 Overall project schedule

This project has set deadlines and important dates shown below. The project is divided into four separate iterations each following the completion of the previous iteration.

The first iteration including this project plan and some skeleton code for the hangman game, has a deadline on 8 February.

The second iteration will focus on modelling, UML and the completion of the main part of the game and has its deadline on 21 February.

The third iteration will include testing of the application and has a deadline of 8 March.

The fourth iteration is the main deadline of the whole project. For this iteration the whole hangman game should be completed and handed in together with the completed documentation. This last and final deadline is in week 12.

4.7 Scope, constraints and assumptions

The scope of the project is to create a working hangman game played in the command line. Out of scope is the possibility to select difficulty level at the start of a game as well as a high score list and the possibility for the game to keep track of the player's numbers of guesses and wins. Since this is only a command line based game there will be no advanced GUI for the game. Due to restricted time for the project there will be limitations to how much extra functionality can be included in the final application.

Assumptions for this project is that the end user will understand how to play a game in the command line as well as a basic understanding of hangman game logic.

4.8 Reflection - Project Plan

Starting out with the project plan I found myself over complicating it quite a bit. A lot of time was spent reading and re-reading material on planning as I had the impression that this part should be very comprehensive. When the time came to write it, following the directions in the template, it turned out to be better to have clear and concise explanations of the different parts of the project plan. As the vision details what the project is about, and the iteration plan details the different tasks to be made, the project plan could be a clear overview outlining the main parts of the project.

5 Iterations

5.1 Iteration 1

The first iteration of this project include this project plan as well as some skeleton code for the application to show the general outline and design.

To even be able to begin with this project plan the first thing needed is research. This is of course needed to learn how to plan a project from the start and how to write the project plan, as well as to actually begin working on the project.

The next step to this iteration is to start working on the project plan documenting in as much detail as possible what this project includes and how to proceed. The detailed schedule should also be produced taking into account how much time both research and actual writing will take. When the project plan and planning this first iteration is done the next step is to write the skeleton code producing a rough draft of how the application will the organised. The first step to produce the application is to make up a good idea of how this application will look in the end and what features I want to be implemented.

Week 4: 23-27 January
Watch the first two lectures required to begin the project and read chapters two and three.
Set up a GitHub repository and invite collaborators.
Also start making the time schedule for this project and outline the project plan.

Estimated time:
Lectures 4h
Reading 6h
Writing 4h

Week 5: 28 January – 3 February
Watch the third lecture included in the research for iteration one and read chapters 22 and 23. Participate in a Q and A session with other teams to get a better feeling for the process and project planning. Continue working on the project plan, planning the project in more detail.

Estimated time:
Lectures 4h
Reading 6h
Writing 10h

Week 6: 4-8 February
Work on finishing all documentation for the first iteration, mainly the project plan, time schedule and time log. Outline and write the skeleton code for the application.
Hand in project plan and skeleton code.

Estimated time:
Writing: 14h
Code writing: 6h

Deadline for iteration 1: 8 February

5.2 Iteration 2

In the second iteration is time to model the game using UML and then to add features and start making a working game. Diagrams are also to be included in this project documentation and should be implemented in the way modelled.

For this iteration the first step is to create a use case model as well as writing the use cases. The use cases model will show the different use cases available and how they relate to the the gamer. The use cases should be fully dressed cases detailing how the game works.

Next up is to create the state machine for the play game use case. The state machine should clearly show how the game is intended to work. Even extra features that might not be implemented in the game during this iteration should be shown in the state machine.

A basic implementation of the hangman game will be created during this iteration, with extra features to be added in a later state. Lastly, a class diagram is to be created documenting the class structure of the implementation.

Read chapters 4, 5, 6, 7, 15 and 20 and watch lectures related to these subjects.

Week 7: 11-17 February
Watch the first lecture of theme two, read chapters 4, 5 and 20. Watch pre-recorded lecture related to these chapters. Create use case model, write uses cases and start implementing the Hangman game.

Estimated time:

Lectures	6h
Reading	6h
Creating Use Case Model	1h
Writing Use Cases	2h
Coding	3h

Week 8: 18-21 February
Read chapters 6, 7 and 15. Watch pre-recorded lectures related to these chapters. Create state machine for the play game use case. Watch second lecture of theme two. Continue to implement the game until a working version of the game is finished. Create Class Diagram.

Estimated time:

Lectures	8h
Reading	6h
Create State Machine	2h
Coding	18h
Create Class Diagram	2h

Deadline for iteration 2: 21 February

5.3 Iteration 3

The third iteration is mainly focused around testing. In this iteration it is time to plan and execute tests of the application as well as documenting this process.
Read chapter 9 and watch lectures related to this subject.

Objective

The objective is to test some of the code for the hangman game that was implemented in the last iteration.

What to test and how

The plan is to test UC1 and UC3 by running dynamic manual test cases. These two use cases were chosen as UC1 is a short but important part of the application as it is the start of the whole hangman game. UC3 on the other hand is the largest use case and also the most important one as it is the one showing how the game is played.

Automated unit tests will also be written for the methods createUnderscores() and evaluateGuess() in the class Hangman. These methods will be tested with at least two tests each. The reason behind choosing these two methods is since they are a crucial part of how the game should be played according to the project plan. The first method makes sure that the word is displayed in the right way for the player to start guessing and the second is a core part of the game as it evaluates the player's guess be comparing letters in the word with the one guessed.

One more unit test will be created for an unfinished method, drawHangman(), that should succeed once that method has been correctly implemented, but will fail at this stage.

Week 9: 25 February–3 March
Start by planning this iteration and creating a test plan for the tests to be written. Read chapter 8 in the book on testing. Watch the live lecture/workshop and participate. Watch pre-recorded lectures. Start creating the manual test cases.

Estimated time:	
Lectures	6h
Reading	2h
Planning	1h 30m
Writing test cases	4h

Week 10: 4–8 March
Run manual tests and record the results. Watch the live lecture/workshop and participate in discussion. Create automated unit tests, run these tests and record the results. Take screenshots and document. Write reflection.

Estimated time:	
Lectures	2h
Writing tests	10h
Testing	

Deadline for iteration 3: 8 March

5.4 Iteration 4

The fourth and final iteration is the completion of the game. In this iteration the project will be viewed as a whole, going through all the steps for any new features implemented into the game as well as make sure that all previous steps are completed and the game is working according to plan. This is also the time to look over the documentation and the planning made at the beginning of the project to see where it has been deviating from the plans and document the reasons for this is well as solutions.

Deadline for iteration 4: Week 12

6 Risk analysis

6.1 List of risks

Risk	Probability	Effects
Staff becoming ill and unable to work	Moderate	Catastrophic
Hardware used to complete project failing	Moderate	Tolerable
Time required to complete project being underestimated	High	Serious
Extra features not being included in final product	Moderate	Tolerable

6.2 Strategies

Risk	Strategy
Staff becoming ill	Give staff resources and time for healthy eating and exercise.
Hardware failing	Continuously backup all work made and have the ability to borrow my sister's computer to complete work.
Underestimated time	Creating a realistic schedule and constantly keep track and update time spent on specific tasks, as well as prioritising tasks so the most important parts of the project get done first.
Features not included	Creating realistic goals and not be overly ambitious about what features can be added in the time allocated.

7 Time log

7.1 Iteration 1

Date finished	Task	Scheduled time	Actual time
2019-01-23	Watch lecture 1 / Introduction	2h	2h
2019-01-23	Created GitHub repository	1h	0,5h
2019-01-24	Read chapter 2 – Software Processes	3h	3,5h
2019-01-25	Read chapter 3 – Agile Software Development	3h	3h
2019-01-29	Watch lecture 2	2h	1h 40min
2019-01-29	Plan the time schedule for iteration 1	3h	3h
2019-01-30	Read chapter 22 – Project Management	3h	3,5h
2019-01-30	Participate in Q and A	2h	2h
2019-01-31	Read chapter 23 – Project Planning	3h	4h
2019-01-31	Watch lecture 3	2h	1h 40min
2019-02-01	Outline the project plan	5h	7h
2019-02-01	Write the vision document	4h	4h
2019-02-04	Complete first draft of the project plan	8h	9h
2019-02-04	Start outlining skeleton code for the application	2h	1h
2019-02-04	Update and finish time schedule	2h	1h
2019-02-05	Write risk analysis	2h	2h
2019-02-06	Finish skeleton code	1h	1h
2019-02-08	Error check and finish up project plan	6h	8h
Total time	**Iteration 1**	**54h**	**58h 20min**

7.2 Iteration 2

Date finished	Task	Scheduled time	Actual time
2019-02-06	Watch live lecture on theme 2	2h	2h
2019-02-13	Plan iteration 2	1h	1h
2019-02-18	Expand use case model	1h	1h
2019-02-15	Write fully dressed use case for Play Game	1h	1h
2019-02-17	Write additional use case for Set up Game	1h	0,5h
2019-02-19	Create State Machine Diagram for Play Game	2h	2,5h
2019-02-15	Implement functionality to start game and display a word to guess in Hangman game.	3h	2h
2019-02-15	Read chapter 4	2h	2h
2019-02-16	Read chapter 5	2h	2h
2019-02-18	Read chapter 20	2h	2h
2019-02-18	Watch Lecture 4 – Systems and Software Modeling	2h	2h
2019-02-18	Read chapter 7	2h	2h
	Watch Lecture 5 – Modeling with UML	2h	-
2019-02-20	Read chapter 6	2h	2h
	Watch Lecture 6 – Software Architecture	2h	-
2019-02-20	Read chapter 15	2h	2h
	Watch Lecture 7 – Software Design	2h	-
	Watch Lecture 8 – From Software Design to Implementation	2h	-
2019-02-20	Watch live lecture 2 on theme 2	2h	2h
2019-02-21	Expand functionality in game by implementing updated word after guesses and end game scenarios.	8h	6h
2019-02-22	Finish implementing functionality in Hangman to complete a working game.	10h	8h
2019-02-22	Create Class Diagram	2h	2h
Total time	**Iteration 2**	**55h**	**42h**

7.3 Iteration 3

Date finished	Task	Scheduled time	Actual time
2019-02-25	Plan iteration/theme 3	30m	30m
2019-02-25	Create a test plan	1h	45m
2019-03-01	Read chapter 8	2h	3h
2019-02-27	Watch live lecture/workshop on theme 3	2h	2h
2019-03-01	Watch pre-recorded lecture 9	1h 45m	1h 45m
2019-03-05	Watch video on the “Greeter Example”	30m	30m
2019-03-04	Create manual test case for Use case 2	2h	1h
2019-03-04	Create manual test case for Use case 3	2h	1h
2019-03-04	Run manual tests and record the results	2h	30m
2019-03-06	Watch pre-recorded lecture 10	1h 45m	1h 45m
2019-03-06	Watch pre-recorded lecture 10.5	30m	30m
2019-03-06	Watch live lecture/workshop on theme 3	2h	1h 30m
2019-03-07	Create automated unit tests for method createUnderscores()	4h	3h
2019-03-07	Create automated unit tests for method evaluateGuess()	4h	2h
2019-03-07	Create automated unit test for unfinished method drawHangman()	2h	30m
2019-03-08	Run automated tests for method (1) and record the results	2h	30m
2019-03-08	Run automated tests for method (2) and record the results	2h	1h
2019-03-08	Run automated tests for method (3) and record the results	2h	30m
2019-03-08	Write reflection	1h	30m
Total time	**Iteration 3**	**35h**	**22h 45m**

7.4 Iteration 4