HANGMAN

Xingrong Zong

AAA

02.04.2020

Logo

Optional logo

Contents

[HANGMAN 1](#_Toc32094225)

[1. Revision History 1](#_Toc32094226)

[2. General Information 2](#_Toc32094227)

[3. Vision 3](#_Toc32094228)

[4. Project Plan 4](#_Toc32094229)

[4.1 Introduction 4](#_Toc32094230)

[4.2 Justification 4](#_Toc32094231)

[4.3 Stakeholders 4](#_Toc32094232)

[4.4 Resources 5](#_Toc32094233)

[4.5 Hard- and Software Requirements 5](#_Toc32094234)

[4.6 Overall Project Schedule 5](#_Toc32094235)

[4.7 Scope, Constraints and Assumptions 5](#_Toc32094236)

[5. Iterations 6](#_Toc32094237)

[5.1 Iteration 1 6](#_Toc32094238)

[5.2 Iteration 2 6](#_Toc32094239)

[5.3 Iteration 3 6](#_Toc32094240)

[5.4 Iteration 4 7](#_Toc32094241)

[6. Risk Analysis 8](#_Toc32094242)

[6.1 List of risks 8](#_Toc32094243)

[6.2 Strategies 8](#_Toc32094244)

[7. Time log 9](#_Toc32094245)

[8. Handing in 10](#_Toc32094246)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 02.02.2020 | 1 | Project Plan | Xingrong Zong |
|  | 2 | Test Plan | Xingrong Zong |
|  | 3 | Project | Xingrong Zong |
|  | 4 | Project | Xingrong Zong |

# General Information

|  |  |
| --- | --- |
| Project Summary |  |
| Project Name | Project ID |
| Hangman | 20200202 |
| Project Manager | Main Client |
| Xingrong Zong | Player |
| Key Stakeholders |  |
| Project Manager | Player |
| Executive Summary |  |
| The project Hangman is to create a program selects a word and the player is going to guess the word by suggesting letter after letter. For every wrong guess, the game is building a part of a man getting hanged. The number of wrongs that the player can have is about wight or ten depending on how many parts are used to hang the man. | |

# Vision

The project Hangman is to create a program selects a word and the player is going to guess the word by suggesting letter after letter. For every wrong guess, the game is building a part of a man getting hanged. The number of wrongs that the player can have is about wight or ten depending on how many parts are used to hang the man.

The program is written in Java. For this text based version the player should be greeted with a menuand when beginning the game a word from a predefined list of nouns should randomly be picked and the number of letters displayed with equally many underscore signs. Build the "image" of the hanging man using the available characters on the keyboard or by using Unicode characters. In addition to the requirements described here, it is recommended that you find and complement it by add your own ideas like a high score list, user registration, persistence, multiplayer, time limit, point systems, the ability to add and remove words and much more. You decide and outline it in your project documentation.

Reflection: To create this vision document for the system, the team has to understand the concept and rules of the creating game.

# 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 project starts with project planning about tasks, time-consuming, labor

Reflection:

## Introduction

The project Hangman is to create a program selects a word and the player is going to guess the word by suggesting letter after letter. For every wrong guess, the game is building a part of a man getting hanged. The number of wrongs that the player can have is about wight or ten depending on how many parts are used to hang the man.

## Justification

Why should the application be made?

For playing

## Stakeholders

List and define the different stakeholders for the project.

Project Manager

Project Customer / User : player

## Resources

What resources are available and used to create the application?

Java program

## Hard- and Software Requirements

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

## Overall Project Schedule

What are the important dates for deliverables?

03.02.2020 – Project Plan

## Scope, Constraints and Assumptions

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

# 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.

## Iteration 1

Implement idea and some skeleton code for the project. Complete the documentation.

Estimated time: 120 mins

Actual time: 180 min

Generate words for game

Player input

## **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.

## 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.

## 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.

# 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.

## 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.

Underestimated development time - The time required to develop the software is underestimated

Hardware unavailability

Requirements change

Product competition

Recruitment problems - Alert customer to potential difficulties and the possibility of delays, investigate buying-in components

## Strategies

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

Avoidance strategies - The probability that the risk will arise is reduced.

Minimization strategies - The impact of the risk is reduced

Contingency plans - Prepare for the worst and have a strategy in place to deal with it

# Time log

Eachassignmentmustbeaccompaniedwithatimelog. Thistimelogshould 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Estimated t | Actual t | Task | t |
| 02.02.2020 | 120 min | 240 min | Project Planning | 120 min |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Handing in

All assignments have a number of files to hand in. The overall advice is to *keepitsimple*. Make it easy for the reciever 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 a number of diagrams in “graphics format”, but always in a document where you provide the purpose and meaning of the diagrams. Remember that the “reciever” is in reality 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.