Hangman Game

Software Developement Project

Christopher Karlsson
LNU
2019-02-08

1. Revision History

Date	Version	Description	Author
2019-02-08	1.0	First draft of this document including general information, vision, project plan, iterations and risk analysis	Christopher Karlsson

2. General Information

Project Summary	
Project Name	**Project ID**
Hangman game	1
Project Manager	**Main Client**
Christopher Karlsson	LNU
Key Stakeholders	
Christopher Karlsson LNU	
Executive Summary	
Creating a Hangman game in which the user is presented with the task of guessing a sentence of words by electing what letters it might be made up of. The player has a given amount of attempts before the game ends and may at any time guess the full sentence to win the game.	

3. Vision

The aim of this project is to create a “hangman game”. The game will feature the task of guessing a predefined sentence before the the hangman character can be built and hanged, ending the game. The game progresses by the player guessing what letters are part of the sentence, the length of each words and the number of words is used by the player to guess what the sentence could be.

As players makes correct guesses letters are revealed while the hangman is built piece by piece from incorrect guesses. Once the sentence is correctly guessed or the hangman is complete the game ends in either the win or lose scenario.

Beyond the base mechanics of the game we also aim to develop multiplayer functionality where several players can make guesses together as well as communicate about the game they are playing. It should also be possible to compete for the highest score through difficulty progression.

Reflection on writing a vision document

A vision statement should in short explain what a project aims to accomplish, it should not contain specific goals but instead it is a concept of what the completed project is. Sommerville writes that “It should be written so that readers can immediately see how the system will be used.”[1]

I think a vague but guiding description of the system is useful especially to return to during development and when discussing features with others. If there’s a guideline for what a system should do but not an exact path of how to get there this allows new ideas along the way and makes it easier to drop others that you realised aren’t good because you are not constrained by a plan that was made before anyone knew what the final product would really look like.

[1] Software Engineering 10th Global Edition 2016 p565

4. Project Plan

4.1 Introduction

A hangman game, the game is played by guessing a sentence from the length of its words with a set number of tries.

4.2 Justification

The project is an assignment for the Software Technology course and its construction is necessary for grading in the course.

4.3 Stakeholders

- Christopher Karlsson
  Developer for this project. Aims to complete the course, learning about software development planning and working in teams to complete a project.

- LNU
  The client for this project is the University and the course teachers there that will receive and grade this project. Using it to judge what the student has learned.

4.4 Resource

To develop the hangman game we have a team consisting of one developer doing all the work. The developer is equipped with the computer system and software necessary to complete the project.

4.5 Hard-and Software Requirements

The software will be developed with Java in the Eclipse open extensible IDE. It should run on any java enabled device that can display text.

4.6 Overall Project Schedule

The project has three separate deadlines:

- 2019-02-08 This document and a barebone version of the project should be complete.
- 2019-02-21 Panning, breaking down the project into tasks and determining how long each task should take. Consolidating requirements. Modeling behaviour and structure.
- 2019-08-03 Testing the application, creating test scenarios to ensure the quality of the delivered end product.

5. Iterations

5.1 Iteration 1

The document as far as it can be completed should be done. The document will later be added to as the project developed but there should be a clear vision and project plan as well as some schedule to ensure that the project is completed on time. By now there should be a barebones demo to display to stakeholders which shows the basic functionality of gameplay including as well as the win and lose system states. This iteration is estimated to take 8 hours.

5.2 Iteration 2

UML modelling for the game where all of its features are decided is made, this is is expected to take 5 hours, we have a good vision for how most features should be implemented but more features may be discovered and multiplayer functionality is expected to take some time.

Implementing the features as they have been specified in the UML model will take longer, especially as some of the multiplayer functionality will take research to complete. This is estimated at 24 hours.

5.3 Iteration 3

The game should be finished at this point and only features that can be considered risk-free should be attempted. The focus is on testing the product and making sure we deliver quality software with all the planned features and without any unhandled exceptions. Constructing tests is estimated to take 8 hours.

5.4 Iteration 4

Delivery of the final product. A complete walkthrough of the game and all of its features should be possible. This should be done by 2019-08-03.

6. Risk Analysis

6.1 List of risks

- Outside events. The developer has a fulltime job and a family that takes up his time beyond this project and unforeseen events may at any time limit time that can be spent working on the project.

  Probability High **Impact** High

- Resource failure. A system crash can affect development.

  Probability Low **Impact** High

- Research failure. The implementation of some planned features are currently unknown, it may be outside the scope of the developers abilities.

  Probability Medium **Impact** Medium

- Motivation failure. The development team is made up of humans, humans are always a risk. If they can not be motivated to complete a task the likelihood that they do is affected.

  Probability Low **Impact** Medium

6.2 Strategies

- Outside events. It is difficult to handle these kinds of disruptions as they may occur at any time, the best course of action is to attempt to complete the tasks ahead of schedule always making sure that there is more time to complete a task than has been estimated incase the estimation is incorrect.

- Resource failure. A backup of the project should be kept offsite, this will be handled by uploading the project files to github. Backup systems are available to the team.

- Research failure. We have only planned features for which research is deemed necessary that is not project critical. If a feature that requires research cannot be completed within a reasonable timeframe it may be removed from the project.

- Motivation failure. As there is only a single developer motivation is not something that can be mitigated. The project is started with a high level of motivation, hopefully this carries through.

7. Timelog

2019-02-08	**Planned** **Actual**	5h 4h	Writing project document
2019-02-08	**Planned** **Actual**	15m	Setting up github repository
2019-02-08	**Planned** **Actual**	2h	Barebones demo