The Hangman Revisited

Dragos Lucian

Linnaeus University

8 February 2019

Contents

1. Revision History

Date	Version	Description	Author
08.02.2019	1.0	Plan + Skeleton Code	Dragos Lucian
22.02.2019	1.1	VanillaMode + Models	Dragos Lucian
08.03.2019	2.0	GUI + Testing	Dragos Lucian

2. General Information

Project Summary	
Project Name	Project ID
The Hangman Revisited	1DV600_LD222JQ
Project Manager	Main Client
Dragos Lucian	Linnaeus University
Key Stakeholders	
• Tobias Andersson Gidlund • Daniel Toll • Tobias Olsson	
Executive Summary	
Hangman is a game concept in which users have a limited amount of chances to guess a randomly assigned word/phrase. “The Hangman Revisited” gives a new and innovative perspective over the basic concept of the classic game.	

3. Vision

The system should allow users to play the game of “hangman” i.e. guess the letters in a randomly assigned word/phrase.

The game will be divided in two game modes, each game mode containing multiple difficulty levels which allows for a diverse gameplay.

The first mode (Campaign) should contain multiple sets of words and a progression tracker. In the first difficulty level (Peaceful), if a word is guessed, it’s taken out of the list, if not, it returns in the list, and will come back in the future. This mode allows for tracking the progress, the goal being to guess all the words from each set. The 2nd difficulty level (hardcore) gives the player a single chance to guess each word. This way of playing is motivated by the leaderboard that is created based on the amount of words from a set guessed by each player.

Every set of word should contain words that have something in common. The basic sets will be Animals, Movies and Countries. More sets could be implemented as updates for the game.

The second mode (Free Play) will have two levels of difficulty. The first level (Classic) takes a completely random word and the player can guess 6 wrong letters before losing the game. The 2nd level (Timed-Man) adds a timer element to the game which removes a “life” every 30 seconds, increasing the difficulty.

Reflection

The vision is probably the most important part of this project because one can figure out whether the project is worth all the resources or not just by analyzing it. It also puts every team member “on the same page”, action crucial creating and maintaining a united team with a common goal. The vision is used throughout every step of developing and is the root of the project. Although some parts of the completed system might eventually differ from the initial vision, the vision acts as a map during the entire development.

4. Project Plan

4.1 Introduction

"Hangman" is a game in which the player must guess all the letters from a randomly assigned word or phrase. The game is won if the word is completely "guessed" and lost if the player makes too many wrong guesses.

4.2 Justification

This game should be developed to apply the knowledge accumulated in the Software Technology course into a concrete project. Working on all aspects required to develop such an application will give an overview of the future career challenges some of the courses' students might face.

4.3 Stakeholders

- Tobias Andersson Gidlund - Process and Planning
- Daniel Toll - Testing
- Tobias Olsson - Modeling-Software Design

4.4 Resources

JDK 11, Java API, Eclipse IDE.

**TO BE UPDATED

4.5 Hard- and Software Requirements

Developed on:	
Operating System	Windows 10 Pro 64-bit
Processor	Intel® Core™ i7-4750HQ (8 CPUs)
Memory	8192MB RAM
Graphic Card	NVIDIA® GeForce® GTX 960M (8GB)

Requirements:	
Operating System	Windows® XP/Vista/7/8/8.1/10 (32/64 bit)
Processor	Intel® Core™ i3 or higher
Memory	2048MB R
Graphic Card	Minimum 1GB

**TO BE UPDATED (add JRE and HDD)

4.6 Overall Project Schedule

- Plan & First Iteration: Friday, February 8, 2019, 23:55
- Playable game (2nd iteration): Friday, February 22, 2019, 23:55
- Testing (3rd iteration): Friday, March 8, 2019, 23:55
- Final Product (4th iteration): Tuesday, March 19, 2019, 13:15

4.7 Scope, Constraints and Assumptions

**TO BE UPDATED

5. Iterations

The four iterations of this product represent the most important steps in developing an application.

- First Iteration should have the skeleton of the application, together with the plan and documentation
- The 2nd Iteration should contain more features that were previously designed using UML diagrams that must be included in the documentation
- The 3rd Iteration is focused on the testing aspect of a software
- The final Iteration should contain a complete game, the product of all previous steps and plans

5.1 Iteration 1

The draft for the project's plan has been written, but it requires some updates in the resource, requirements and scope section. A logo is also to be developed in the close future.

The Skeleton Code was developed, and it showcases a vanilla version of game that creates a game of Hangman for a predefined word and it allows a user to play via input/output of the console.

5.2 Iteration 2

The first implementation of the basic game mode (Vanilla) was done. Before implementing, the Use Case Model and the State Machine Diagram were made using UML. After the implementation, a Class Diagram was also made.

The focus on this Iteration was representing the program using different diagrams and models. All the diagrams are released with this iteration.

5.3 Iteration 3

The game now received a Graphical User Interface and the Category and User Login features.

Testing was done on all the classes of the system with the exception of the GUI, the 3rd assignment focused on test analysis and discovering bugs using unit tests.

Game has a rough aspect, but the functionality is close to being complete.

5.4 Iteration 4

6 Risk Analysis

As this is an individual project, most of the risks have to do with the developers Physical and Mental wellness. Planning and managing the time required to develop this project is crucial. The biggest risk that will be encountered has to do with the lack of motivation and external distractions. Having a well-organized life and work style will reduce the probability of any risk occurrence. Planning with slack and dividing the workload all throughout the time allocated is the ideal solution to all the risks encountered.

6.1 List of risks

Risk	Probability	Impact
Procrastinating	Likely	Rushed and unfinished project
Not submitting the assignments on time	Seldom	Losing the required pace of development
Not attending lectures	Unlikely	Missing key aspect of the project
Getting sick	Unlikely	Pause in development

6.2 Strategies

Risk	Strategy
Procrastinating	Ignore distractions and focus on important things
Not submitting the assignments on time	Check deadlines often and plan with slack
Not attending lectures	Watch livestreams, read and set reminders
Getting sick	Eat fruits, workout and in case of sickness use medicine

7. Time log

Task	Date	Time Estimated	Actual Time
Project Plan	05.02.2019	2:00 H	3:30 H
Skeleton Code	06.02.2019	1:00 H	1:30 H
Plan 2nd Assignment	19.02.2019	1:00 H	0:30 H
Use Case Model	20.02.2019	1:30 H	2:30 H
State Machine(basic)	20.02.2019	1:00 H	2:00 H
Implement Vanilla	20.02.2019	3:00 H	1:30 H
Class Diagram	20.02.2019	1:00 H	0:30 H
State Machine(extra)	21.02.2019	1:30 H	2:00 H
Plan 3rd Assignment	04.03.2019	0:30 H	1:00 H
Code Inspection	04.03.2019	1:00 H	1: 30 H
Manual TC	06.03.2019	1:30 H	1: 00 H
Running Manual Tests	06.03.2019	0:10 H	0:10 H
Unit Tests	08.03.2019	3:00 H	4:00 H
Test report	08.03.2019	1: 00 H	0:45 H