Hangman Revisited

Dragos Lucian

Linnaeus University

17 April 2019

Contents

1. Revision History

Date	Version	Description	Author
08.02.2019	1.0	Planning + Skeleton Code	Dragos Lucian
17.04.2019	1.1	Fixing the Project Plan	Dragos Lucian

2. General Information

Project Summary	
Project Name	Project ID
Hangman Revisited	1DV600_LD222JQ
Project Manager	Main Client
Dragos Lucian	Trivia enthusiasts
Key Stakeholders	
• Developer • Resource Manager • End-Users	
Executive Summary	
Hangman is a game concept in which users have a limited amount of chances to guess a randomly assigned word/phrase. “Hangman Revisited” gives a new and innovative perspective over the basic concept of the classic game by adding specific categories of words, progress tracking and individual score which make for prolonged play time.	

3. Vision

The system should allow users to play the game of “Hangman” i.e. guess the letters in a randomly assigned word/phrase from a specific category. The end goal is to guess as many words from the categories as possible.

The game will contain multiple categories of words and a timed mode which makes the game more challenging but at the same time more rewarding via the scoring system, all these features allow for diverse gameplay.

The individual progress will be seen for every category and a leaderboard will give the game a higher purpose than just guessing one word.

Every category should contain words that have something in common. The base categories will be Animals, Colors, European Countries, Movies and Songs. More categories can be added as updates for the game.

In the first mode (Classic), 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 is to guess all the words from each set.

The second mode (Timed) increases the difficulty of the game but also gives more rewards. Guessing a “timed” word gives 3 points (compared to just one in the classic mode) but a greater risk is involved since if you don’t guess a word the first time, it won’t go back into the list.

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

The game will be developed in an iterative manner, divided into four major iterations. Along with all the steps new features will be added and tested. A basic version of the game in which letters from a word are guessed should be implemented right after planning in order to set a starting point. The total available time to develop the project is 2 months, with an iteration being released every two weeks.

4.1 Introduction

"Hangman Revisited" is a game in which the player must guess all the letters from a randomly assigned word or phrase from a selected category. The round is won if the word is completely "guessed" and lost if the player makes too many wrong guesses. Every correct word gives the player points depending on the mode selected which gives the game two end-goals: guess all words from the categories and have the highest score while doing so.

4.2 Justification

"Hangman Revisited" follows a "campaign" game structure, allowing players to track individual progress and score in order to compete against other users with the aid of the scoring system or just aim to complete all the categories via the progress trackers. The game is designed to scale with the addition of new categories as "updates" that will keep it new and actual over a longer period. These features make the game stand out and directly target Trivia enthusiasts who love longer or competitive gameplay.

4.3 Stakeholders

The **developer** works directly on the source code and follows the specifications given. The developer should write scalable code that can be easily improved and modified over the course of iterations as well as after the release.

The **resource manager** should focus on distributing the resources evenly throughout the development period. The resource manager will influence the project by assigning the number of resources available for a specific feature, setting the pace and importance.

Lastly, the **end-users** (Trivia Enthusiasts) have a large impact in the project by setting the initial specifications and taking part in testing the product, giving valuable feedback that will bring the game closer to the client's preferences.

4.4 Resources

- Time: 8 weeks (57 days, from 22 January to 21 March 2019)
- Literature: Software Engineering 10th ed. by Ian Sommerville
- Hardware: Asus G551JW (Intel® i7, 8GB RAM, GeForce® GTX 960M)
- Software: Windows 10 (64bit) running Eclipse Java 2018-09, JDK 11.0.2, JUnit 5 & JavaFX 11

4.5 Hardware and Software Requirements

Requirements to develop the game:	
Operating System	Windows® XP/Vista/7/8/8.1/10 (32/64 bit)
Processor	Intel® Core™ i3 or higher
Memory	4096MB RAM
Graphics Card	Minimum 2GB
Software	Eclipse IDE, JDK 11.0.2, JUnit 5, JavaFX 11
Minimum requirements to run the game:	
Operating System	Windows® XP/Vista/7/8/8.1/10 (32/64 bit)
Processor	Intel® Core™ i3 or higher
Memory	2048MB RAM
Graphics Card	Minimum 1GB
JRE	11

4.6 Overall Project Schedule

- Project Plan & Skeleton Code: Friday, February 8, 2019, 23:55
- Basic Game & UML Models: Friday, February 22, 2019, 23:55
- Testing of the game: Friday, March 8, 2019, 23:55
- Final Iteration with complete features: Friday, March 21, 2019, 23:55

4.7 Scope, Constraints and Assumptions

This project will contain a game in which the user has to guess randomly assigned words/phrases. The scope of the game is represented by having 5 different categories to choose from, a user database to store credentials and a simple graphical user interface. Having more categories, storing the user credentials in a more advanced and secure manner, adding a category creator as well as having a professional-looking design is out of scope because of the time limitations as well as having a low functional impact.

The game will run using a graphical user interface with a fixed resolution of 600x800px. The progress, user details and categories will be stored locally on the machine that runs the game.

To start the game, the GUI.java class located in the src/basicGame directory needs to be run. The interface is built using JavaFX 11 which was manually added to the project and run configuration. All the game resources (code, textures, categories and database) are stored in the source folder of the game. When a new account is created, it is added to the database (db.txt) and a copy of the categories assigned to the user is created inside the directory userLibraries.

Reflections

The project plan was the most difficult part of the documentation to put together since the size of the project makes it hard to come up with Stakeholders, Requirements and Scope. But on the other hand, going through all the planning of a project helped me a lot to visualize and get a better grasp on the project as a whole, not just individual assignments. The different sections in this part make the project feel “professional” and gave a glimpse of what I expect to be working with during my future career.

5. Iterations

The four iterations of this product represent important steps in developing an application. The final project needs to be the result of all the development steps that are represented by the different iterations (planning, modelling, testing and implementing)

TASK	DESCRIPTION	ESTIMATED TIME
Brainstorm ideas for game features	Come up with ideas for developing the game as well as unique features	2 days
ITERATION 1 (DUE DATE 8th of February)		
Writing the vision	Describe why the game should be developed	1:00 H
Writing the Project plan	Describe the process of development in detail	2:00 H
Analyzing risks	Write down all the risks that might be encountered while developing the app	1:00 H
Writing the Skeleton Code	Write the basic structure of the system (doesn't have to be playable)	2h

5.1 Iteration 1 (Planning)

This iteration started once the concept of the game was set. The concept should be represented by the vision and the Project Plan should showcase the steps to be taken in order to develop the game. Risks should also be taken into consideration and strategies to prevent or easily recover from any of the risks need to be developed.

After the vision and Plan are complete, the basic structure of the game should be implemented in the so-called "Skeleton Code", this will help plan the following iterations and divide the work-load better over the course of the following iteration.

6 Risk Analysis

As this project only has one developer, most of the risks have to do with his/her 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 schedule 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 prevent most of the risks encountered.

6.1 List of risks

Risk	Description	Probability	Impact
Computer breaking down	Loss of progress after the last release	Unlikely	Severe
Bad interpretation of requirements	Having more work than expected	Likely	Severe
Procrastinating	Rushed and unfinished project	Likely	Significant
Getting sick	Pause in development	Unlikely	Significant
Gold Plating	Not focusing on important aspects	Likely	Significant
Missing deadlines	Staying behind the required pace	Possible	Moderate
Not attending lectures	Missing key aspect of the project	Unlikely	Moderate

6.2 Strategies

Risk	Strategy
Computer Breaking down	Back-up project often and take care of computer
Bad interpretation of requirements	Read the assignments carefully and have slack
Procrastinating	Ignore distractions and focus on important things
Getting sick	Eat fruits, workout and in case of sickness use medicine
Gold Plating	Write tasks in order of importance
Missing deadlines	Check deadlines often and plan with slack
Not attending lectures	Watch live streams, read and set reminders

Reflection

Listing risks for an individual project like this one felt odd in the beginning but after I read into different risk categories and started writing them down, I realized that they could really impact the development process. I think writing the risks made me plan with more slack and made me feel a bit more confident towards the whole project.

10. Time log

Task	Date	Time Estimated	Actual Time
Writing the vision	05.02.2019	1:00 H	1:00 H
Project Plan	05.02.2019	2:00 H	3:30 H
Analyzing risks	06.02.2019	1:00 H	0:30 H
Skeleton Code	06.02.2019	1:00 H	1:30 H
Fix Project Plan	14.04.2019	2:00 H	2:30 H

There was a big difference between the estimated time for the Project Plan and the time it took to finish it. This is because putting together the plan didn't seem as difficult in the beginning but going through all aspects that need to be described turned out to be very time-consuming.

Fixing the project plan was required as multiple sections were lacking information or required adjustments. This was done by following the feedback provided in order to fulfill the requirements.