Software Development Project

SINISA GVOJIC

Linnaeus University, Sweden

7/2/2019

Table of Contents

1 Revision History

Date	Version	Description	Author
7/2/2019	0.25	First project iteration and first version	Sinisa Gvojic
17/2/2019	0.50	Second iteration with added features and patches	Sinisa Gvojic

2 General Information

Project Summary	
Project Name	Project ID
Hangman	Sg222xg
Project Manager	Main Client
Sinisa Gvojic	Tobias Andersson Gidlund
Key Stakeholders	
Sinisa Gvojic – project leader, manager, developer, tester, resource manager Tobias Andersson Gidlund– main client and the project consultant	
Executive Summary	
My name is Sinisa Gvojic. I am going to be developing a somewhat simplistic version of the popular Hangman game which this document details. The target of this product is the person who asked for it's development and delivery. The main advantage of me developing this game is that I offer a prototype version of the product for the customer to test out in order to see if they would pursue the further development of the product or if they would deem it not worthwhile.	

3 Project Plan

There will be 4 iterations of the project. The first iteration, with the due date 8/2/2019 will include the basic functions of the program. The basic functions include generating random words in the Swedish and English language, giving the user a certain amount of guesses and displaying a message whether or not the user won the round of the game or not. The second iteration, due to the 22/2/2019 will expand on the first by including additional features such as drawing the man to be hanged from different keyboard characters and counting the score and storing it along with the name the user input before starting the round. The third iteration will expand on the user and score side of the system by implementing user registration with an email and a single-player high score system. The fourth iteration will implement multiplayer mode on one device without needing to register and a survival-style timed game mode.

3.1 Introduction

The Hangman game is a Java program that allows the user to play a game of hangman in a text-based fashion.

3.2 Justification

The application, if put on the market, is something so simplistic that it may be able to break through the market saturated with overly-complex storytelling and gameplay. Small indie games have been rising in popularity over the past several years, which provides assurance that the game may be worthwhile.

3.3 Stakeholders

Sinisa Gvojic – the project manager, project leader, chief developer, project tester and resource manager.

Tobias Andersson Gidlund – main customer and client

3.4 Resources

The development and testing of this game will be done in the Eclipse Java IDE, installed on an Acer laptop with an i7-7700 processor, 16GB of RAM and 256GB of SSD memory. Resources for development will be gathered utilizing programmer knowledge and the Internet.

3.5 Hard- and Software Requirements

As stated in the resources, the game will be developed in a Java IDE, and it being a text-based game, it does not require significant computing power in regards to hardware. Software-wise, it is necessary that the user's device has Java installed.

3.6 Overall Project Schedule

The first iteration and the project skeleton is due to be delivered by the 8/2/2019, and the expansion with added additional features is to be developed and tested by the 2/2/2019 and delivered by the 22/2/2019

3.7 Scope, Constraints and Assumptions

Under the scope of this project falls the development and deployment of the game. The game will not be released for sale/download on any online/digital game store nor will it be sold as physical copies. The features of the game will include selecting the language for the word to be guessed (English or Swedish), a high score count, user registration via email, multiplayer, survival mode as well as the option to add new words.

The main constraint for the development of the game is time, as there are very clear and strict deadlines in order. That being said, with the lack of time comes a lack of top quality. The game will be text based with no graphics added through another program, such as Unity™. Seeing as the game is to be developed by a single student, there are no budget constraints for the game development. This is due to a lack of employees and other personnel related expenses.

The user/player of the game is assumed to have a functional computer that is able to run java programs with a mouse and keyboard connected to the device.

4 Iterations

4.1 Iteration 1

Iteration 1 features the basic functions of the game. The basic functions include giving the user a set number of attempts to guess all the letters. The words will be randomly selected from the appropriate file in either Swedish or English. The player will then guess a letter one by one until he/she either guesses all the letters or the number of guesses reaches a certain threshold. This bare skeleton of the program is to be delivered by the 8th of February this year. The times of added features are detailed in the time log

4.2 Iteration 2

The second iteration of the project will expand upon the game skeleton with new features. The game will implement a menu that will enable repeatability as well as an option to quit the game in between sessions.

Revision: The game has been developed with these intended features with the diagrams and use cases provided alongside the documentation and the code. The program has been split into two java files, one containing the methods necessary for the game to run and the other containing the main method that starts the program.

5 Risk Analysis

5.1 List of Risks

Developer falling heavily ill. While very not highly probable, is possible to happen due to the season of the year. Depending on the illness symptoms, the developer could potentially work on the game from his home, thus mitigating the effects his temporary situation has on the delivery of the product.

Developer getting into a traffic accident. The developer mostly wears dark clothing without any reflective bands attached, even during the night. Extremely unlikely to happen, but the delivery of the product would be heavily delayed depending on the state of the developer.

Developer's computer being damaged or being under the attack of serious malware. Unlikely to happen, with catastrophic conseqnces regarding the delivery of the project.

Environment used to develop the game does not generate code properly. Unlikely to occur, however the results of it occurring the would not be noticeable.

Market risk. Another student delivering a project so outstanding that the standards are raised for the rest of the students, thus impacting the possibility o receiving a higher mark. Decently possible to happen, however the game would still be delivered on time.

Workload is underestimated. The developer underestimates the amount of time needed to complete a certain task. Moderate probability with tolerable effects on the project delivery and development.

5.2 Strategies

In order to avoid the developer-related risks stated above, wearing warm clothes and reflective bands is to be made mandatory for the developer, as well as the consumption of tea and vitamin dense foods. The IDE used to develop the game is receiving regular maintenance, however another environment is to be used as backup. Market risks are out of the control of the developer, so the best thing to be done is to design the best possible program, given the available resources.

Time Log

First iteration/game skeleton:

~~Implementing the getEnglish() and getSwedish() methods and connecting them to their respected file: Estimated time: 1.30 hours, actual time spent: 45minutes. The remaining 45 minutes were utilized to make further plans for the project.~~

~~Writing the checkVictory() method and the main method that, for now, only utilizes the three mentioned methods: Estimated time: 3hours, actual time spent: 2.25 hours. The remaining 35 minutes were used to do read the course book.~~

Action	Estimated time	Actual Time
Programming	3.5 hours	4 hours
Writing the documentation	2.5 hours	6 hours

Second iteration:

Action	Estimated time	Actual time
Programming	4 hours	5 hours 25 minutes
Expanding the documentation	4 hours 30 minutes	3 hours 40 minutes
Writing the use cases	3 hours	4 hours 15 minutes
Drawing the diagrams	2 hours	3 hours 20 minutes
Studying for the online exam	10 hours	13 hours 30 minutes