HANGMAN

IPEK UYANIK

LINNEAUS UNIVERSITY

08/02/2018

Linnæus University

Contents

1 | Revision History

Date	Version	Description	Author
07/02/2019	1.0	Iteration 1	Ipek Uyanik

2 | General Information

Project Summary	
Project Name	Project ID
Hangman	iu222au_1dv600
Project Manager	Main Client
Ipek Uyanik	User of all ages
Key Stakeholders	
Developer: Ipek Uyanik Tester: Ipek Uyanik	
Executive Summary	
The project is implementing hangman which is a game of guessing a Word or phrase one letter at a time.	

3 | Vision

This project’s purpose is implementing the game "Hangman" in a text based fashion in Java.

For this text based version the player will be greeted with a menu and when beginning the game a word from a predefined list of nouns will randomly be picked and the number of letters displayed with equally many underscore signs. By using the available characters on the keyboard or by using Unicode characters, the "image" of the hanging man will be builded.

In addition to the this requirements, features like high score list, user registration, multiplayer, time limit, point systems, the ability to add and remove words will be added to the game.

Don’t think that is a simple game. It will be a greatest version of Hangman. Users will have lots of fun while playing that game and having a hangman game that can be easily used by users.

REFLECTION

While i writing the vision for the hangman game, i could be more flexible or more creative. But i preferred being more concrete. Even those are can be a little bit much for me. Because this project will be my first game project and i am not sure that i am be able to do that much features for a a game. Nevertheless, even in thought, I did not refrain from mentioning the basic features that should be present in a game. I am strongly hoping that i will be able to implement that features to the game.

4 | Project Plan

4.1 Introduction

The project is implementing the “Hangman” in a text based fashion.Hangman is a game that a player trying to guess a word using a strategy based solely on letter frequency. As the player continues, a part of the stick figure on the noose is added. Once a full body is drawn, the game is over, and the player lost.

4.2 Justification

This game provides students to get to know sounds and letters better. It also expands vocabulary knowledge. And of course the greatest purpose is fun.

4.3 Stakeholders

Manager: Plans and manages the project

Developer: Builds and creates software and application

Tester: Tests the project

End-users: Use the product

4.4 Resources

Money: 0 SEK

Time: 2 months

Software: Java-Eclipse

Hardware: Macbook-air, 1,8 GHz Intel Core i5

Personal: Manager, Developer Team, Test Team, End-Users

4.5 Hard- and Software Requirements

Java is used for developing and Eclipse is for environment.

4.6 Overall Project Schedule

Phase I: 08/02/2019

Phase II: 22/02/2019

Phase III: 08/03/2019

Phase IV: 22/03/2019

4.7 Scope, Constraints and Assumptions

Scope The Project will introduce a game; including the following features:
A high score list
User registration
Multiplayer
Time limit
Point systems
Constraints
The system will be developed using platform tools and languages supported by the developer, including:
-Java
-Eclipse
System must be multi-platform and support Windows, Linux, and MacOS standard terminals.
Where possible, the system will be designed in such a way that its features can be exposed as services.
Where possible, the system will use existing services and data.

REFLECTION

Planning the project was a little bit complicated. At the first, I did not understand exactly what i was suppose to do. It is not stil clear as i want but i think it because of the content of the project. In my opinion this is a student project which includes implementation of the game and if we compare the other big projects we do not have the exact and extensive knowledge about the content Despite of this i tried to rely on the content of the template and complete the documentation according to the request as much as i could.

5 | Iterations

5.1 Iteration 1

Estimated time: 2 days

Task1-Complete the documentation	
Creating the vision	40 min.
Planning the project	75 min.
Iterations	40 min.
Risks and strategies	15 min.
Setting the time log	30 min.
Task2-Create the skeleton code	30 min.

Due to-08/02/2019

5.2 Iteration 2

Task1-Creating UMLs	3 hrs.
Task2- Implementing the code	5 hrs.
Task3- Adding features to the game	7 hrs.
Task4-Updating the documentation	1 hrs.

5.3 Iteration 3

Task1-Adding new features to the game	11 hrs.
Task2-Planning the test	1 hrs.
Task3-Performing the test	4 hrs.
Task4-Documenting the test	4 hrs.
Task5-Updating the documentation	1 hrs.

5.4 Iteration 4

Task1-Reiterationing the steps 1-3	6 hrs.
Task2-Completing the game	10 hrs.
Task3-Completing the all game and project	6 hrs.

6 | Risk Analysis

6.1 List of risks

List of risks
1-The time required to develop the software is underestimated.
2-The organization is restructured so that diffirent management are responsible for the Project.
3-Changes to requirements that require major design rework are proposed.
4-Key staff are ill and unavailable at critical times.

6.2 Strategies

1-Investigate buying-in components; investigate use of automated code generation.
2-Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
3-Derive traceability information to assess requirements change impact; maximize information hiding in the design.
4-Reorganize team so that there is more overlap of work and people therefore understand each other's jobs.

REFLECTION

While i analysing the risks i used the chapter's book so it was not difficult that much. All i had to done was to try to finding the appropriate risks for my project like time estimationing or illness etc. That step was pretty clear to understand. And again, in my opinion the number of risks are not that much. Due to the project's size or the number of employees,

7 | Time log

Date	Task	Start Time	End Time	Total Hrs.	Actual Hrs.
05.02.2019	Complete documentation	17.00	21.00	4	6
06.02.2019	Skeleton code	18.00	18.30	0.5	0.5
09.02.2019	Create UMLs	17.00	20.00	3	
10.02.2019	Add features to the game	16.00	22.00	5	
13.02.2019	Add features to the game	15.00	18.00	3	
17.02.2019	Add features to the game	16.00	20.00	4	
19.02.2019	Update the documentation	16.00	17.00	1	
23.02.2019	Add features to the game	18.00	21.00	3	
24.02.2019	Add features to the game	16.00	21.00	5	
27.02.2019	Add features to the game	18.00	21.00	3	
01.03.2019	Testing	15.00	16.00	1	
02.03.2019	Testing	17.00	21.00	4	
04.03.2019	Testing	18.00	22.00	4	
07.03.2019	Update the documentation	17.00	19.00	2	
09.03.2019	Reiteration	16.00	22.00	6	
11.03.2019	Reiteration- Complete the game	17.00	22.00	5	
20.03.2019	Complete the game	18.00	23.00	5	
21.03.2019	Complete	15.00	21.00	6	

8 | Handing in

All assignments have a number of files to hand in. The overall advice is to *keep it simple*. Make it easy for the reciever to understand what the files are by using *descriptive* file names. Use as *few* separate documents as possi-ble. 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 pur-pose 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.