

Hangman Project template

Mattias Gustavsson mg22ws

Vision

The project is to create an Hangman game, it shall contain all the required features for a working Hangman game aswell as several others for a more flexible game. The basic features that you need is those that make you can play and win or lose by getting a random word you should guess on. In addition you should be able to add your own words to the list of words, if you want you should be able to choose a word for another to play. There shall be some different levels of difficulty like word easy (5-6 letters) ,word medium (7-8 letters) and word hard (9-10 letters). Then aswell the number of lives the player has easy, medium and hard.

You shall have different Users that saves the number of wins and losses. There will be a graphical interface aswell as an multiplayer where you guess on the same word.

Project plan

Introduction

This application is an Hangman game, in the game you can login as an existing or a new player. You can add words, play with standard difficulty or play with some specific difficulty. There will be an graphical interface for making playing mutch easter. The Software will be delivered in 3 increments where additional functionality will be added each increment.

Justification

The software is ordered from the Website [www.games.com\(made Up\)](http://www.games.com(made Up)) they have had an old Hangman software game available to be downloaded on there site. But there customers started to complain that the word library of words was to small and that they wanted different difficulties. So the purpose for this software was to replace the old software and aswell provide all the features that was requested.

Stakeholders

- Top-manager of games.com
- The owner of the company
Adam "a end user of the software"

Resources

The resources for this software is one Software engineer that has 2 months to complete the software with his computer.

The code is only made of Java.

Hard-and software Requirements

Hardware Requirements

- There are no hardware requirements.

Software requirements

1. Functional requirements

- Start a game
- Win a game
- Lose a game
- Get random word
- Create new word
- Login to existing player
- Create a new player
- Choose different difficulties
- Choose the number of lives

2. Non-functional requirement

- Easy to use and learn
- It should take less than 15 seconds to add a new word
- The software should have low response time

Scope and Constraints and Assumptions

The plan is to successfully complete all the requirements of the softwares.

So that all the functional requirements features shall be working as planned and the Non-functional requirement shall as well be working although they are harder to evaluate. But multiplayer is a little bit put to the side and will probably not be done in this project.

Iteration 1

1. Writing the document Estimated 4 hours
2. Estimated 30 min planning classes and methods
- 3.
4. Plan for basic setup of classes then make the game work to play, the things i did here was get a random word, made the game possible to start and end and although you can guess on letters in the word and it will show if you were right but you can't win or lose. I worked on the classes Interface, Player, Words, Game. Estimated 5 hours
5. Estimated 15 min planning
6. Make that you can win and lose in the game by counting those you were right and wrong on and improving the interface. Estimated 2 hours
7. Final fixes some change to the text based interface. Estimated 1 hour

Risk analysis

My strategy is to get a good ground to build the software on by carefully chosen classes to fit into the software. I will start with making it possible to run the game basically only getting a random word and then guessing what letters is in it and saves the letter if it is right.

Then adding that you can lose lives if you are wrong. Then that you dies when you have 0. And this will be how i will work, starting with the absolute necessary for the game to work and then working down to less the important features. So if something that is really important for the hole project comes up it probably do it early. The thing that might be the most likely to be a risk is the graphical interface since I have not worked so much with interfaces earlier but the rest is known area.

Time log

01-28 1. Writing the document. 5 Hours

01-29 2. 20 min planing classes and methods

01-29 3. Made it possible to play the game without be able to lose or win. 4 Hours

01-30 4. Implemented that you now lose lives when wrong and now win when all letters are right. 1 Hour

02-06 5. Final fixes mostly in the interface. 30 min

Reflections

This Assignment was pretty fun because i like to code in java and it wasn't especially hard to get the Hangman program to work. I have before done similar programs like this so it was no new experience although i enjoyed coding very much. The Project plan was less fun then the coding but have to say that it probably was more educational to have to re update the memory of how to make a Project plan.