SOFTWARE DEVELOPMENT PROJECT - HANGMAN

Mattias Ruljeff
Rulles Hack

Project name	Hangman
Revision	2
Project ID	1
Project manager	Mattias Ruljeff
Main Client	Linnaeus University
Key stakeholders	Linnaeus University
Programmer	Mattias Ruljeff
Executive Summary	

Content

1. Revision History

Date	Revision	Description	Author
2020-02-03	1	Added this document.	MR
2020-02-13	2	Added User Cases, iteration 2 and Check-state diagram.	MR

Full name of Author	
Initials	**Author name**
MR	Mattias Ruljeff

2. Vision

We are going to create a game called Hangman. In the game, a player should guess a given word, letter by letter. If the player guesses the wrong letter more then 8 times, the game will end showing a man being hanged. The game will be played in the console. The visual part of the game will be shown in Unicode letters in the console window.

Our vision is to create the hanged man, step by step, for each wrong guess. The drawing steps are:

1. The ground.
2. The first vertical pole.
3. The first horizontal pole.
4. The supporting pole for the vertical pole.
5. The rope.
6. The man’s torso.
7. The arms.
8. The legs.

When the player loose a scoreboard of the best 5 players will be shown, then the game restarts.

3. Project plan

Listen to all the lectures, read the chapters in Software Engineering, structure the Hangman-game.

The hangman game will be created in 2 separate JS-files. The first file will contain the game logic, the second file the Unicode-characters for the “animation” of the man being hanged.

The player should be able to insert a player name.

The player will be given eight chances to guess a random picked English word, letter by letter.

The animation-steps of the hangman are:

1. The ground.
2. The first vertical pole.
3. The first horizontal pole.
4. The supporting pole for the vertical pole.
5. The rope.
6. The man’s torso.
7. The arms.
8. The legs.

When the player loose, the game restarts.

The first file with the game logic will contain an array of words that the game can use to present to the player.

3.1 Introduction

A game of hangman, played by a player in the computers console window.

3.2 Justification

We will make the Hangman-game to succeed with the course’s goals.

3.3 Stakeholders

Linnaeus University.

3.4 Resources

Visual studio code.

3.5 Hard- and Software Requirements

Hardware: A computer.

Software: Visual studio code and Git bash.

3.6 Overall Project Schedule

Week 6 (Complete assignment 1)

Week 9 (Complete assignment 2)

Week 11 (Complete assignment 3)

Week 12 (Submit complete project)

3.7 Scope, Constraints and Assumptions

A game of Hangman

4. Iterations

Included parts:

- Read Software engineering.
- Create Hangman-game.
- Create documentation.
- Plan iterations.

4.1 Iteration 1

- Create documentation.
- Start to build the Hangman-game. Add all necessary files for the project.
- Read chapter 2, 3, 22 and 23.
- Watch video-lecture 2 and 3.

4.2 Iteration 2

- User case diagrams.
- User cases.
- Check-state diagram.
- Read chapter 4, 5, 6, 7, 15 and 20.
- Watch lecture 4, 5, 6, 7 and 8.

4.3 Iteration 3

as

4.4 Iteration 4

asd

5. Risk Analysis

This is a small project that is for a school course, so the risks are not that extensive. Some risks still need to be observed.

5.1 List of risks

Risk	Affect	Description	Probability	Effects
Hardware failure	Project	If the computer/hard-drive crashes	Low	Catastrophic
Sickness	Project and product	If I get sick and can't go to school	Moderate	Serious
Size underestimate	Project and product	If the added features are too complicated/hard to integrate	Moderate	Serious
Requirement change	Project and product	If the product specifications change	Moderate	Tolerable

5.2 Strategies

Strategies to avoid and minimize the impact of the risks:

a. Hardware/software – Save all the files to Gitlab. Backup-computer available.
b. Sickness – Work from home in case of sickness
c. Requirement change – Build modules that can easily be modified
d. Size underestimate – Skip extra functions

6. Time Log

Task	Planned time (hours)	Actual time (hours)	Starting date	Latest update	End date
Project plan	10	7 so far	2020-02-03 11:00	2020-02-13 22:45	-
Hangman-game	30	1 so far	2020-01-31 15.30		-

7. User Cases

UC 1 Start Game	
Preconditions:	None
Post conditions	The game menu is shown
Main Scenario	
1. Starts when the player starts a session of Hangman. 2. The systems presents the main menu, presenting a title, the options to play and to quit the game. 3. The gamer enters a name and starts the game	
Alternative scenarios	
2.1 The player doesn't enter a name and start the game. 2.2 The system asks the player to enter a name. 3.1 The player makes the choice to quit the game. 3.2 The system quits the game. (See UC 3)	

UC 2 Play the game	
Preconditions:	Player has chosen a name and started the game
Post conditions	None
Main Scenario	
1. The player enters a letter. 2. The systems check the letter and prompts the player to enter another letter. 3. Repeat step 1-2 until the player has guessed the right word. 4. The system presents the high score and prompts the player to play again.	
Alternative scenarios	
2.1 The system detects that the letter is wrong, creates parts of the hanged man and presents it to the player. 2.2 The player guessed wrong 8 times, loses the game. (See UC 3)	

UC 3 Quit the game	
Preconditions:	The game session is running
Post conditions	The game session is terminated
Main Scenario	
1. Starts when the player wants to quit the game or loses. 2. The system prompts for confirmation. 3. The player confirms. 4. The system terminates.	
Alternative scenarios	
3.1 The user does not confirm. 3.2 The System returns to its previous state.	

8. Handing in