Time log

TASK	DATE	TIME ESTIMATION (h)	REAL TIME (h)	Accomplished
Getting knowledge and analysis	17-18.02. 2019	10	14	
Time Log	18.02.2019	0,5	0,5	18.02.2019
Updated Use Case Diagram	18.02.2019	2	1	18.02.2019
Fully Dressed Use Case for "Play Game" use case.	18.02.2019	2	4	19.02.2019
State Machine Diagram for "Play Game"	18.02.2019	2	3,5	21.02.2019
Implementation	19.02.2019	7	12	22.02.2019
Class diagram	19.02.2019	2	3	22.02.2019
Extended Use Case Model (for a higher grade).	19.02.2019	2	I made the basic version extended for my ideas	21.02.2019
Extended State Machine (for a higher grade).	19.02.2019	2	I made the basic version extended for my ideas	21.02.2019

//my comment on the time log

On this iteration I found the tasks generally more challenging. Especially implementation took me much more than I suppose. I think its due to lack of knowledge in JavaFx and of lots of doubts concerning class design. And of course one of my main Risks from planning came true – my son got sick and he is staying at home- this makes me extremely not concentrated but we deal with that risk as planned – switch with my husband. There is still some work to do with the project to cover the requirements. I will do it on the next iteration. Despite it all I am still satisfied- the program is running and it is the biggest program that I have built.

WORK TODO for iteration 3:

After Implementation for iteration 2 I still have some implementation work to do:

-the quit with confirmation!!!!!

-deal with win situation – change the button to start again maybe congratulation picture!!!

-win case when the all the letters are guessed – is not working properly!!!

-deal with Guess class- I don't use its fields – try to organize it better!!

-deal with PlayGameMain class – think weather it should and could be spited into classes
-learn how to organize JavaFX interface into classes

-while drawing class diagram I discovered I have some chaotic connections between classes- many of them share language for example

-now I think I should have consequently have put the guess into the guess object instead of using it just as String!! -TODO

-images for hangman version

-more passwords in every category

-polish text – decide weather make it bilingual or set depending on chosen language (OCH- lots of changes but maybe make another playGame main for polish

-work more on interface – start should be lower when playing a game

-check if buttons start and others get disabled in the proper moments

USE CASE FULLY DRESSED

UC1: Play Game

Scope

Hangman Game application

Level

User-goal

Primary Actor

Gamer

Stakeholders and Interests

- (**Gamer**): wants to play the game.

Preconditions

The gamer chose interface version (tree or hangman), language (English or Polish) and category (People, Food, Birds, All categories) and choose to start the game.

Postconditions

The game is finished. The final result is seen by Gamer.

The Gamer can choose to quit go to UC3 or Start game again go toUC1

Main Success Scenario for the Hangman game

1. The use case starts when the Gamer wants to start the game.
2. The game chooses a password from selected category.
3. The Gamer sees on the game window starting picture (where the hangman or tree drawing will be updated) and set of "_" in the amount of the passwords letters and a text box with text next to it "Enter the letter or guess the password ".
4. The Gamer enters the letter and confirms.
5. The game searches for that letter in a password:
    a) If the letter is in the password - the letter becomes visible in its position in the password- it is not hidden by the "_" anymore.

    b) Else if the letter is not in the password – the updated picture of the hangman is displayed and the letter is displayed at the bottom of the window as the "Wrong letter"

The Gamer repeats step 4 and steps 5,6 are repeated until the game is finished – step 7.

6. If the letters given by the Gamer fill every "_" in the password the password is guessed and the Gamer Wins- the win screen is displayed.
7. If the Gamer guesses the password and writes it down in the text box– the Gamer also wins (the Gamer can write the whole password at any moment of iteration 4, 5, 6) - the win screen is displayed.
8.Else if the Gamer writes the letters that are not in the password 13 times. After this time the Gamer losses the game and Try again screen is displayed.
9.The final screen is displayed with results, Win or Fail picture and possibility to Start the game- or Quit
10. UC2 ends

Extensions

(**at any time**): The Gamer quits the game

1.The US 3 starts

(**2**): Depending on the language choice in UC1:
1.The password is chosen from words in certain language:
a) The password is in Polish or
b) The password is in English

(**3**): Depending on the Version choice in UC1:
1.The primary picture is displayed:
a)If the Gamer selected Hangman:
The picture displayed is empty.
b)If the Gamer selected Apple Tree:
The picture displayed is the picture of apple tree with 13 apples.

(**4**): The Gamer enters the password
1.The password is checked by the system:
a)If the password is correct :
The game goes to step 7 in US2
b)If the password is incorrect :
The notification incorrect password is displayed and the game can be continued.
One password guess counts as one letter guess.

(**4**): The Gamer enters other character than upper or lower case letter
1.The notification about error is displayed and the game can be continued.
This mistake is not counted as a guess.

(**5b**): Depending on the Version choice in UC1:
1.The picture is upgraded:

a)If the Gamer selected Hangman:

With every wrong guess the hangman's part is added and after 13 wrong guess he gets hanged.

b)If the Gamer selected Apple Tree:

With every wrong guess one apple falls down from the tree and after 13 wrong guesses there are no apples on the tree.

Special Requirements

- The letters of the password must be displayed in upper case
- The interface must be colorful
- All text other than the password must be bilingual English/Polish

Variations in Technology and Data

- The passwords for every category should be easy – possible to guess by children
- In the future more passwords will be needed
- The passwords and letters can be entered upper or lower case

Frequency of Occurrence: Unpredictable, but not more than few times a day

Miscellaneous

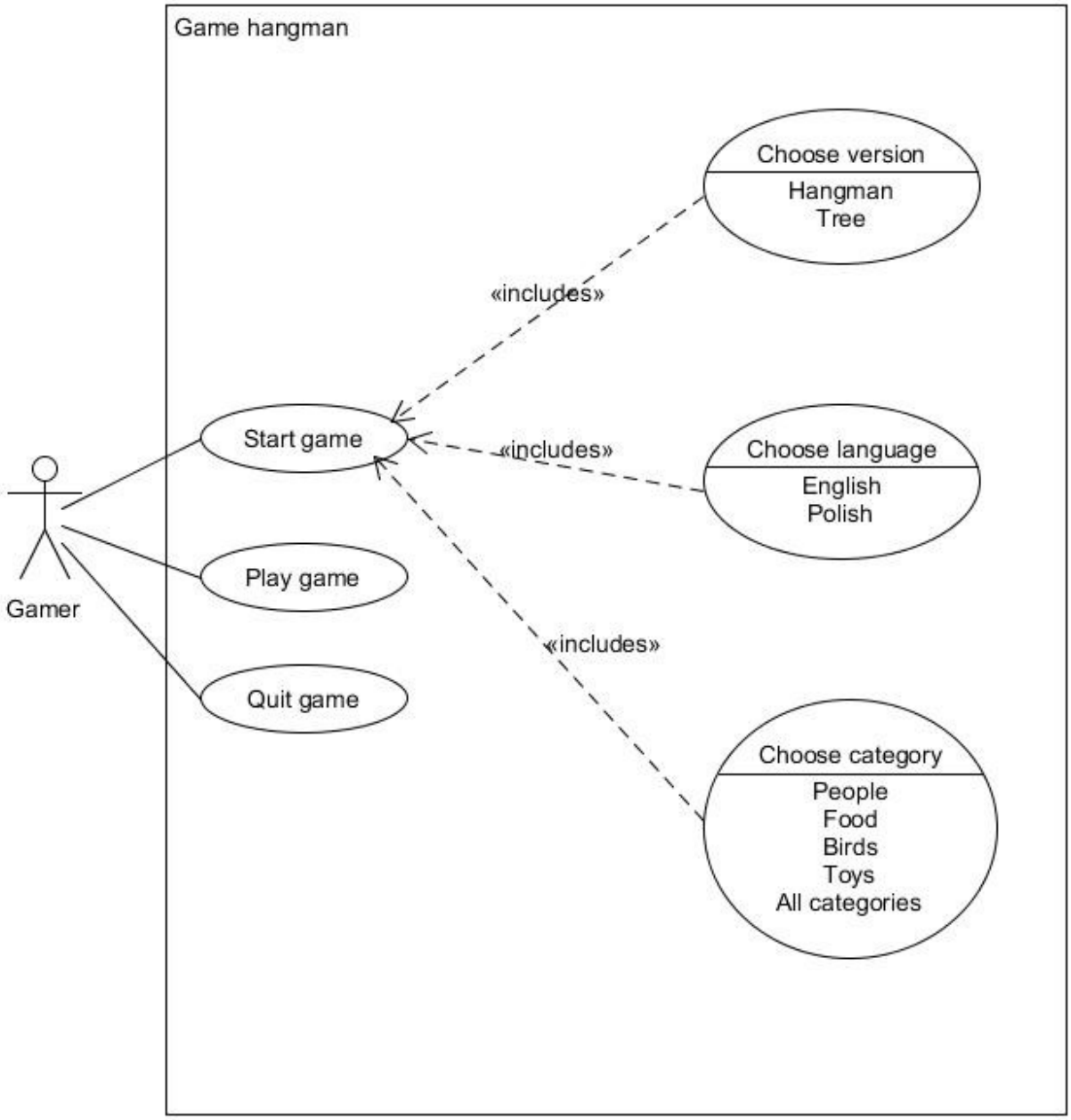
1.The possibility of adding levels of difficulty in the future.

2.Is 13 a comfortable-medium amount of wrong guesses to fail? To make the game not too frustrating for the children and not to easy for adults.

3.Should it be possible to get exactly the same password in the next game or they should"t repeat for some time at least (depending on the amount of possible choices)

4.The interface tree problem- when the last apple falls is this the end of the game?

UPDATED USE CASE DIAGRAM



UC 1 Start Game

Precondition: none.

Postcondition: the game menu is shown in one of the options:

-Tree English in chosen category

-Tree Polish in chosen category

-Hangman English in chosen category

-Hangman Polish in chosen category

Main scenario

1.Starts when the user wants to begin a session of the hangman game.

2.The system presents the main menu with a title, the option to choose game interface- Tree or Hangman,the option to choose the language (English or Polish) -inactive , option to choose the category (People, Food, Birds, Toys, All categories)- inactive, the option to play- inactive and quit the game-active .

3.The Gamer makes the choice of interface :

a) The Gamer selects Hangman.

or

b) The Gamer selects Tree.

4.The choice of language gets activated.

5.The Gamer makes the choice of language:

a) The Gamer selects Polish. The passwords in UC2 will be in Polish

or

b) The Gamer selects English. The passwords in UC2 will be in English

6.The choice of category gets activated.

7.The Gamer makes the choice of one category:

a) The Gamer selects People. The passwords in UC2 will be from this category.

or

b) The Gamer selects Food. The passwords in UC2 will be from this category.

or

c) The Gamer selects Birds. The passwords in UC2 will be from this category.

or

d) The Gamer selects Mixed. The passwords in UC2 will be mixed.

8. The choice of start gets activated.

9. The Gamer makes a choice to Start a game – the choice of interface,language and category is visible but becomes frozen- can't be changed.

10. The system starts the game (see Use Case 2).

UC 2 Play Game

In the document Use_Case_Fully_Dressed

Alternative scenarios

2.1, 3.1, 4.1, 5.1, 6.1 The Gamer makes the choice to quit the game.

1. The system quits the game (see Use Case 2)

6.1 Invalid menu choice

1. The system presents an error message.

2. Goto 2

UC 3 Quit Game

Precondition: The game is running.

Postcondition: The game is terminated.

Main scenario

1. Starts when the user wants to quit the game.

2. The system prompts for confirmation.

3. The user confirms.

4. The system terminates.

Alternative scenarios

3.1. The user does not confirm

1. The system returns to its previous state