ASSIGNMENT 2

Modeling-Software Design

Tobias Ohlsson

Sana Hameed
Sh223nw@student.lnu.se

Time Log

Hangman Game

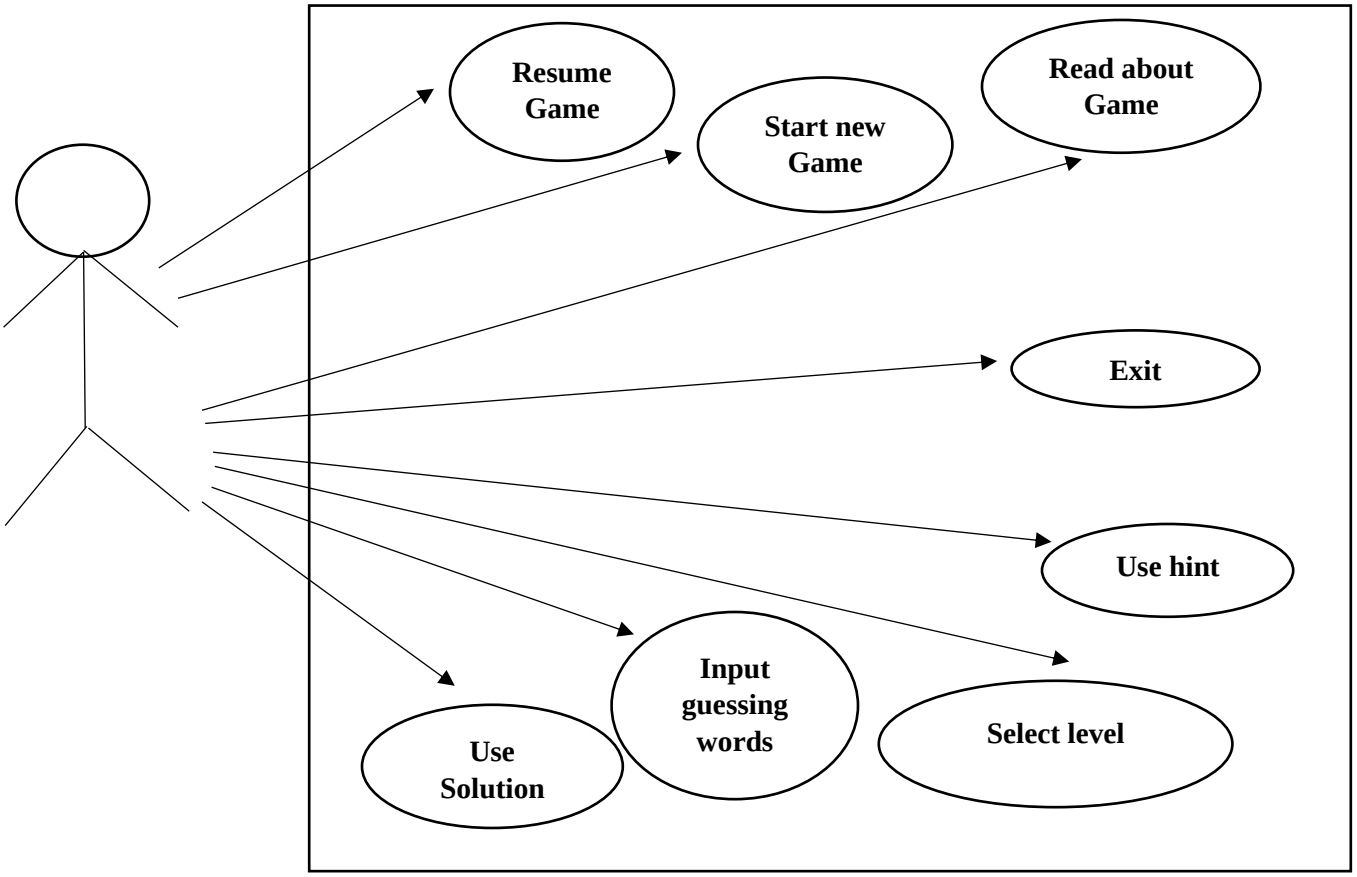
Hangman game Project Scheduling

Start Week	Jan 31, 2019

Week	1	2	3	4	5	6	Notes
Starting	Jan 31	Feb 7	Feb 14	Feb 21	Feb 28	Mar 7	
Phase One	Project planning						
	Project planning						
	Project planning						
	Project planning						
Phase Two		Working on					
		Diagrams					
		UML					
		UML					
			State Machine				
			State Machine				
Phase Three				Class Diagram			
				Class Diagram			
					Coding		
					Coding		
						Testing	
						Testing	

Use Case Diagram

Hangman Game



Fully Dressed Use Case

Hangman game

1) Use case: New Game
Primary Actors: Any one playing the game

Goal: to start a new game

Precondition:

1. System support the game configuration
2. The file has been run and game screen appeared.

Scenario:

1. Go to the new game button and click on it.
2. New game is loaded on system.

2) Use case: Resume Game

Primary Actors: Any one playing the game

Goal: to resume game from previous play

Precondition:

1. Game was played before.

2. Game support to have a check point to start from.

Scenario:

1. Go to the Resume game button and click on it.
2. Game is loaded on system from the previous stop point.

3) Use case: Select Level

Primary Actors: Any one playing the game

Goal: to start game from required level

Precondition:

1. Required level has been unblocked.

2. Game supports loading level.

Scenario:

1. Go to the new game and click to the select level.

2. Select level and load the select level.

4) Use case: Exit Game

Primary Actors: Any one playing the game

Goal: to exit from the game.

Precondition:

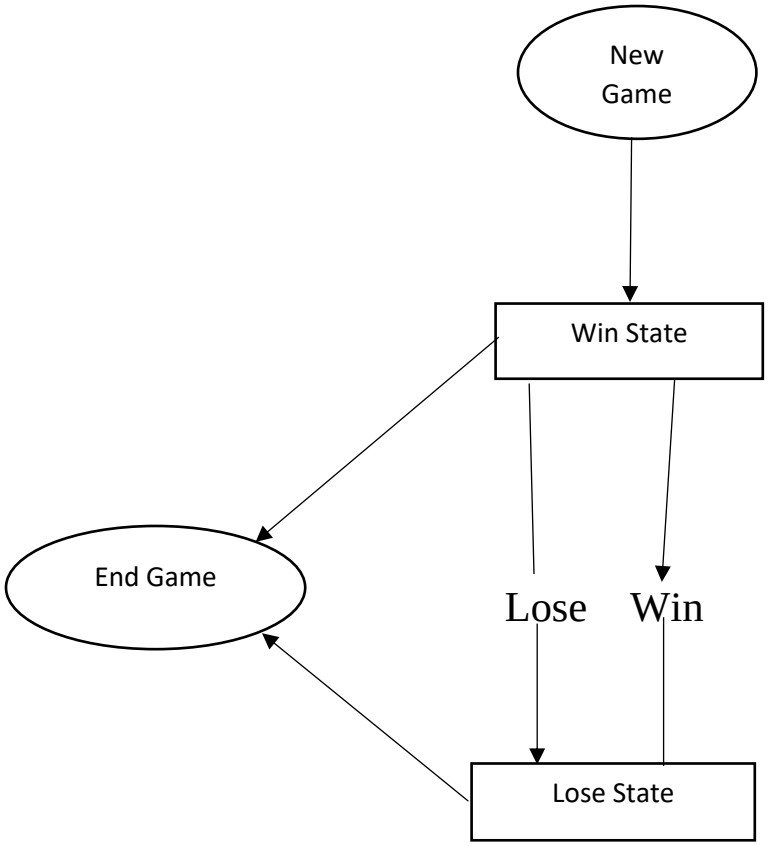
A game level is being played.

Scenario:

Click on exit game.

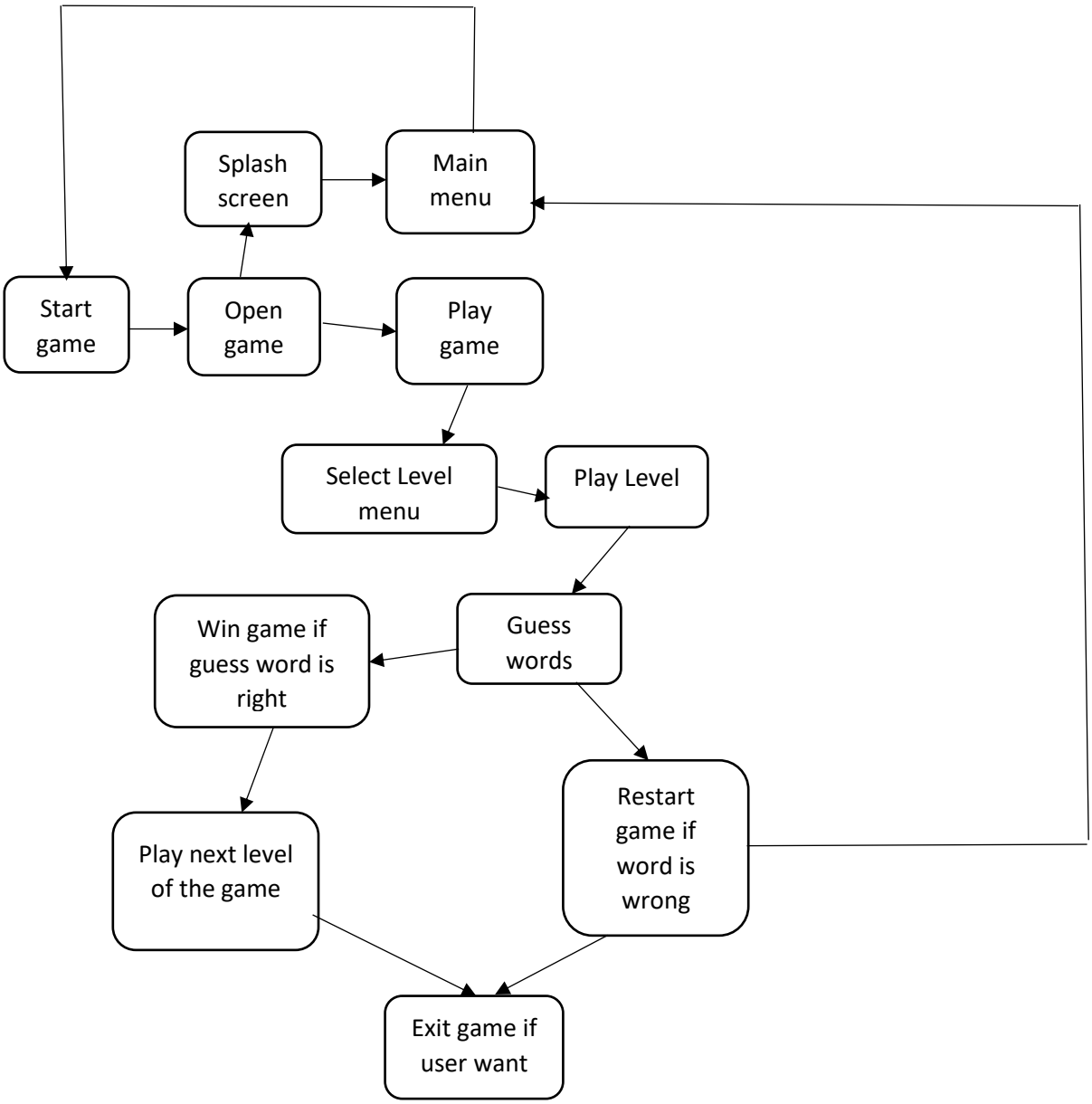
State Machine Diagram

Hangman Game



Extended State Machine Diagram

Hangman Game



Implementation:

This is a game made to simulate the classical game of Hangman; in which there will be a 'questioner' who will present the player with a word whose letters are blanked out, and the player will then have to guess what the word is correctly. A wrong guess would result in the player's lose the game, a stick figure, getting closer and closer to be 'hanged'. Hence, the Hangman name of the game.

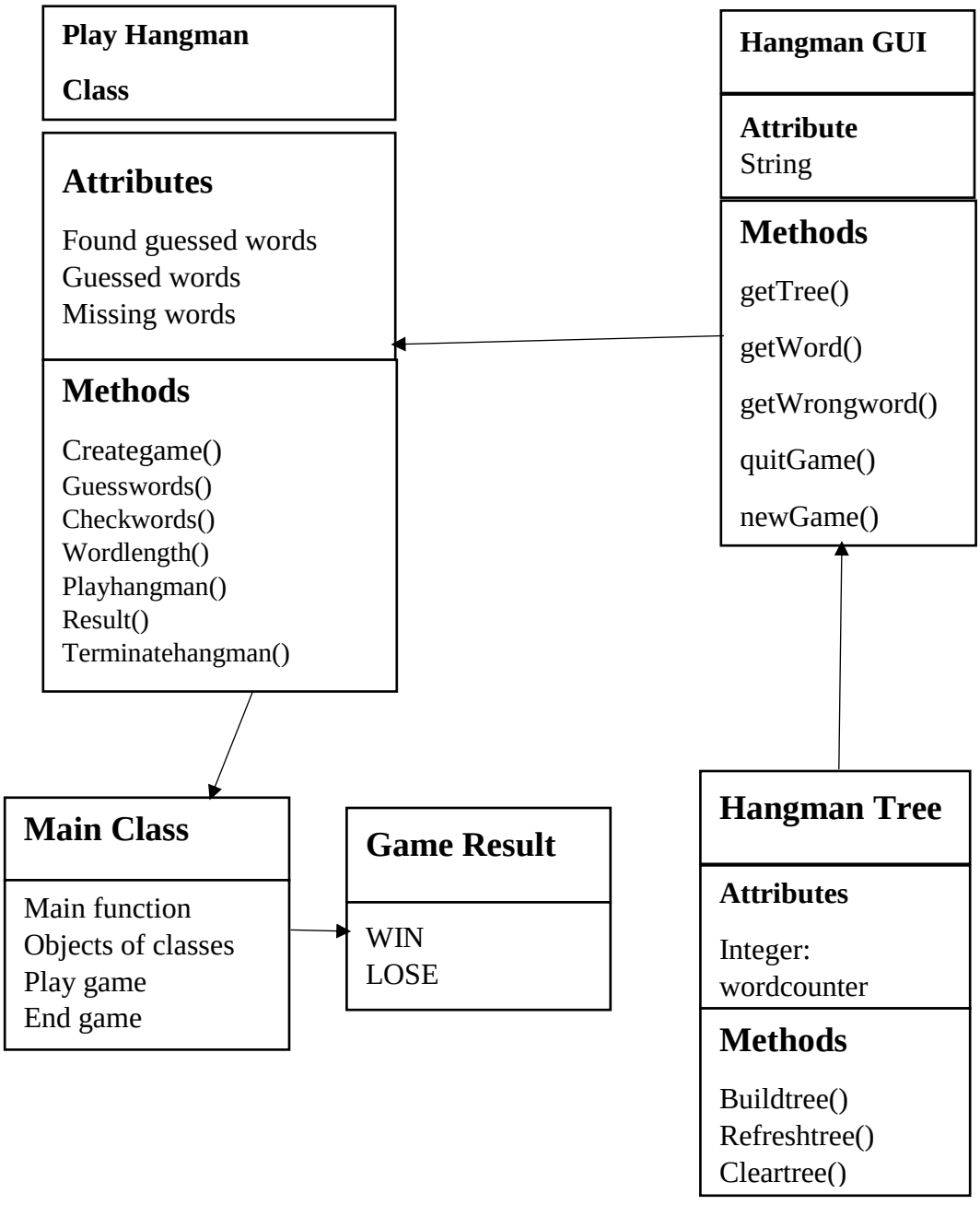
In this version, the game takes on the role of the 'questioner', with the player allowed to choose the difficulty he (or she) wishes. The game assumes that longer words are more difficult, thus the higher the difficulty, the longer the word given towards the player. The rest of the game just follows the pen-and-paper version.

Unlike the pen-and-paper version however, this Hangman game lacks the ability to come up with new words, and thus is somewhat limited in the sense that the words 'questioned' are all hard-coded into the app itself.

The original draw of this idea was to somehow implement the Hangman game with the ability to generate words on its own; we eventually found out that such an algorithm would probably be quite costly and difficult to implement. However, this being the first game we develop, we hope that our subsequent ones would rely less on hard-coded data.

Class Diagram

Hangman Game



Handed in:

https://github.com/Sanahameed/sh223nw_1dv600