

BZ 214 Visual Programming Project Report

Group Number: <u>Group 39</u>	
No: 1030521025	Name Surname: Ahmet Kaan GÜMÜŞ
He prepared the loading screen, the language options option in the game interface(JavaFX), the creation of keys such as "shuffle", "enter", "delete", "clear", "reload" and "main menu", entering letters from the keyboard or the game and ending the game.	
No: 1030521231	Name Surname: Ufuk ALTUNBULAK
He reads the dictionary, adds he to the wordsmap hash, extracts the words that are pangrams from the wordsmap hash, transfers them to pangram setofwords and adds "pangrams dictionary" if he meets the criteria, then the controller gives pangrams in the dialog, takes pangrams, and checks it. He controls whether the word is derived from the pangram.	
No: 1030521018	Name Surname: Okan AFŞİN
He prepared the code that checks all the words in the dictionary and prints the pangram words in a not too long time, detected the errors in the program (tester), added explanations to the codes via JavaDoc and prepared the report sections.	

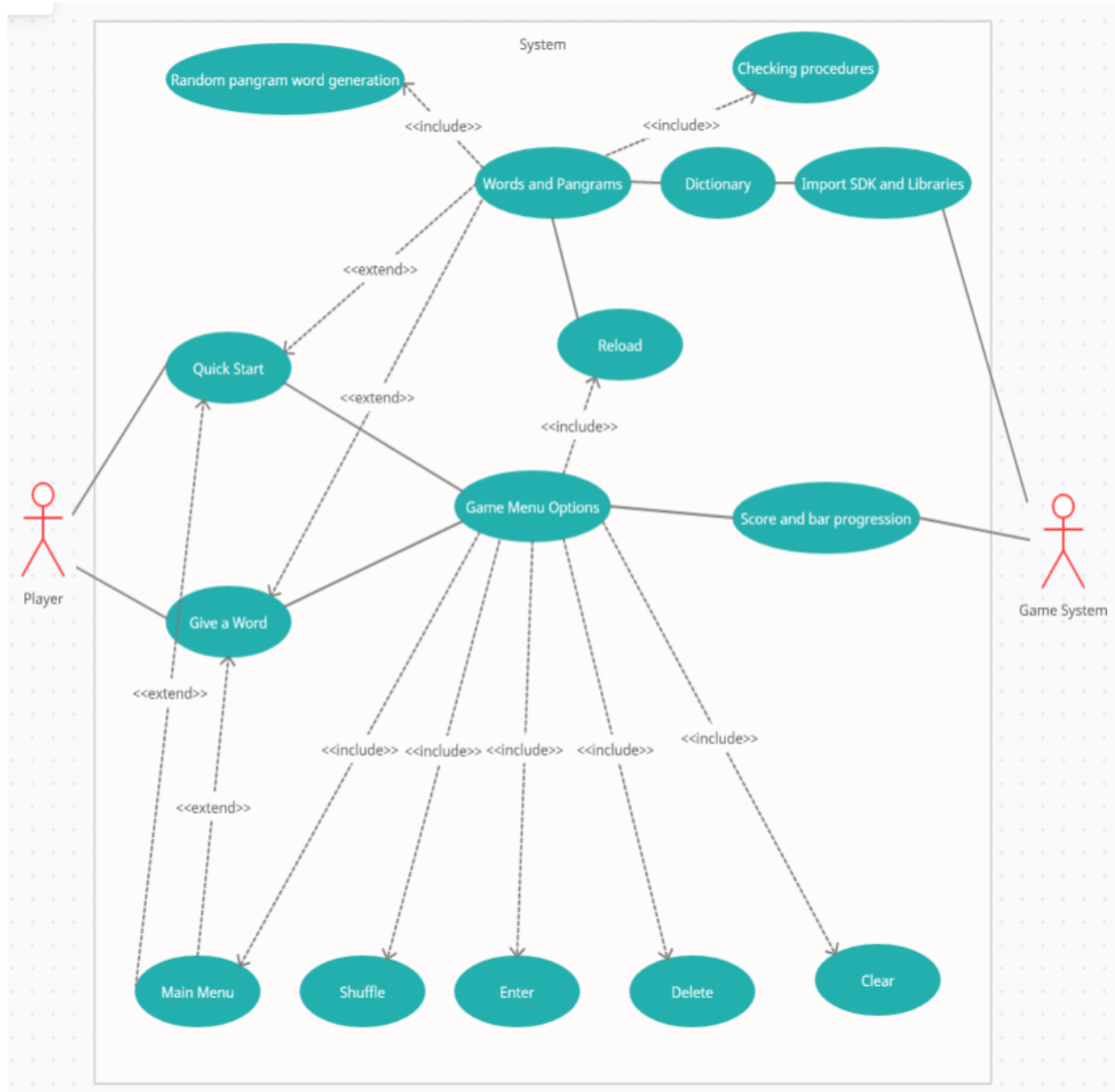
Abstract

The words read from the dictionary are determined through the codes written, and these codes are transferred to the game interface. prompts you to select and start the game. If you start with a quick start, the game starts with 7 letters determined by the system itself, you guess words to the given letters. If the word you create has at least 4 different letters and if there is in the dictionary, according to the number of letters or if you use the 7 letters given and form a word, you will get points "(pangram)" accordingly. The scoring detail for words is as follows: 1 point if it has 4 letters, 2 points if it is 5 letters, but an additional +7 points are added when the word is pangram. Expressions that motivate you according to the score you get from the words and the score bar are located in the upper right part of the interface. In the "give a word" part, which is the second gameplay part, a word with 7 letters and 7 letters different from each other is entered, and if that word is suitable for the game, the game starts by applying the same score conditions described in the quick start. 7 letters of the word are written in 7 hexagons and the game is played. During both games, the game interface has 6 buttons with various functions that make it easier for us to play games, these buttons are "Delete","Shuffle","Enter","Reload","Clear" and "Main Menu", and the detailed explanation of these buttons is in the software design. will be done.

Software Design

In summary, our project is a puzzle game that includes 2 different language options and 2 different game methods within the words in the dictionary, giving points to the player according to the length of the word and whether all the letters are used or not, giving expressions according to the score, and buttons that provide convenience in the game. We used coretto-17 sdk, javafx-sdk-18.0.1, Maven and AnimateFX libraries in the development of our pro. In case of any problem, we consulted each other first, and if we could not find any results, we did research on the internet. Let's explain the 6 buttons in the game interface briefly mentioned in the abstract section. The "Delete" button deletes the written letters, the "Shuffle" button mixes 6 letters out of the 7 letters except the middle hexagon, the "Enter" button sends the word to the query, the "Reload" button is in the quick start game option. New random 7 letters are given to start a new game, "Clear" clears the part where the letter is written, and "Main Menu" allows to return to the main menu. In this project, the user, that is, the player, is asked to create a word

according to the given letters, as a result, the user can be told that he has formed a correct word and points can be given. There are 11 classes in the project. The project is started with the FXStarter class. Other classes are ControllerDialog, Dictionary, Finish, Game, Loading, Main, Message, Pangram, Word and WordPicker. Controller dialog class deals with words, transfer of words and score. The Dictionary class takes care of dictionary reading, dictionary manipulation, and pangrams. The Finish class is the class where the codes for the last part of the game are written. The Game class is the class that concerns the most detailed part of the game section. The Loading class is the class that concerns the loading area. The Main class is the class that concerns the main parts of the project. The Message class deals with message transmission. The pangram class is the class in which pangram words are created and controlled according to conditions. Word class is word check and score calculation class. Finally, the WordPicker class is a class where the word is entered and the word is selected. Use case diagram is given below.



Conclusion

There is no problem during the work of the project, it works smoothly. No deficiencies have been identified. Our project meets the requirements in the project request pdf published in ERUDM, for example, one of the requests was that there were two types of games, these two games are working in our project. Another request, scoring and message transmission according to the score, also works and works on the buttons during the game. Our personal opinion about this project has increased our development in java language well, because we have actively prepared applications, we have constantly tested and because it is a group work, we have seen and fixed the problems that sometimes a single person cannot see as a group.