**Project Report: Snowman**

Paula Acevedo (991699095), Simerpreet Kaur (991684906), Simranjeet Kaur (991675703)

Sheridan College

PROG24178 Object Oriented Programming 2

Parul Kantaria

Tuesday April 11th, 2023

**Final Project Overview and Description**

Our project is a java FX based application that will present the user a word or phrase to guess depending on the category (e.g., TV shows, movies, etc.) they choose. Player will take turns and try to guess the word or phrase. With every wrong attempt, the snowman will be built on the side until complete (seven wrong attempts will lead to complete snowman in our game). Once it is complete, the user loses the game. When guessed correctly, the number of attempts left will be added to the user’s total score.

During the duration of the game, the user will have the chance to use 1 hint. This will affect their total score and subtract 1 attempt from their current number. User can also guess a word completely for 2 extra points, but if wrong, it will subtract 2 from their attempts.

To obtain the word that will be guessed by the user, we will utilize file input to randomly retrieve it. For each category, there will be a file written with x number of words, each separated with a pipe symbol as a delimiter.

To implement dynamic arrays, we will used array lists throughout our program. One will be used to store the letters used by the user in the program for each gameplay (will reset when they try to guess another word). This is needed in order to implement the hint function, which checks all the letters the user used, and then moves to randomly select a letter from the from the word that needs to be guessed.

Another array list will be used to store the words in a select file (category). Once this is done, a random position will be generated using Math.random() to get a word from the array list, and that is the done that will be used for the user to guess. The array list will be emptied after the word is stored in a variable in the class Snowman.

Two extra array lists will be used to store the words guessed and words missed Strings.

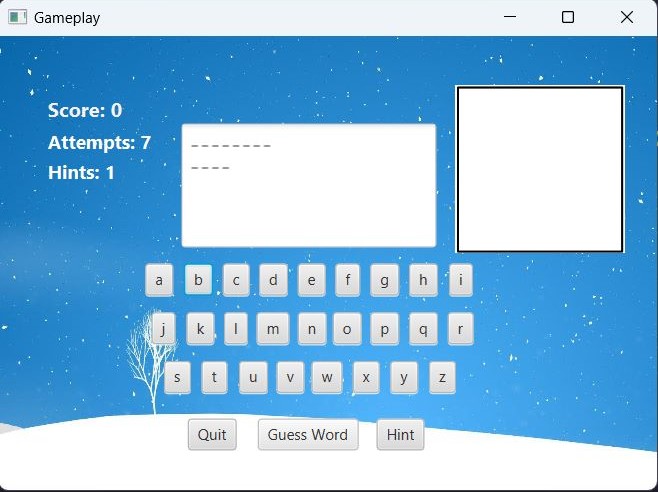
**Basic User Interface**

We envision 7 windows of our application:



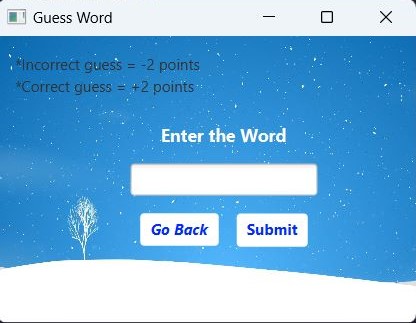
1. **Log in window**: allows user to enter the username they want to use to play the game. It also shows you the category of the words they want to guess from. These 2 fields *must* be filled out to proceed with the game (if user does not type a username, they will be assigned a default name).

Each button will read from a specific file and retrieve a word using a Math.random() generator. Values such as name and word are set in the object of the Snowman class created.



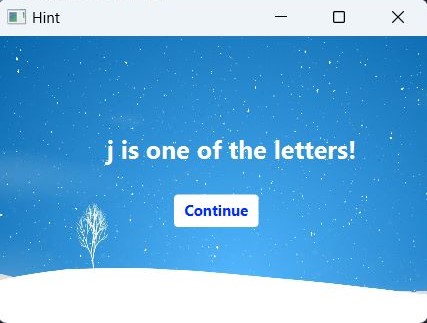
1. **Game window**: where the user will be guessing the letters which are present in the word. It will keep track of their total score, how many attempts they have left (starts at 7 and decreases by 1 for every wrong one they get) and build the snowman as the game progresses (1 wrong = 1 more piece of the snowman added). The program shows and allows user for 1 hint. In addition, they will be able to guess the word (2 attempts subtracted if guessed incorrectly) or quit the game.

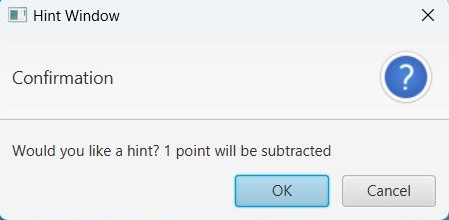
This window is the most complex out of all. It uses a variety of methods for each button to check that the letter/character selected is in the word. If it is, then a variable holding the progress of the word will be updated and the output in the text area will be changed accordingly. If it is not present, then an attempt will be subtracted, and the image will change. All characters are stored in a character array list. This ensures that when the user asks for a hint, the character given won’t be one of the ones chosen. In addition, all buttons will be disabled regardless of if they’re right or wrong.



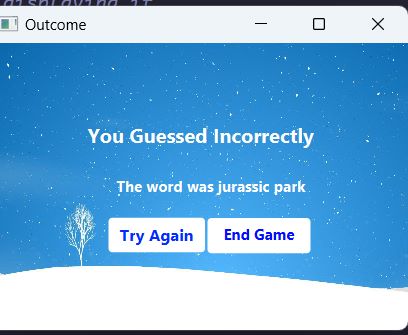
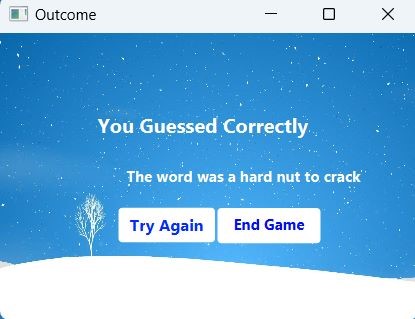
1. **Guess word window:** allows user to enter a word to guess. In here, the word typed by the user will be compared to the word that is currently being guessed. If correct, the user will end the gameplay and displayed the ***outcome window*** (explained further down). If not, the user score will be subtracted 2 attempts (output messages displayed accordingly). The user will also have the option to return and not guess without any penalties. In addition, a pop-window will be shown to the user to indicate if they got the word right or not. 2 extra points plus the remaining attempts will be added to the total score if guessed correctly.

User will only be able to access this window if they have more than 3 attempts left. If not, then an error window will appear indicating the user so and they button “guess word” will be disabled.

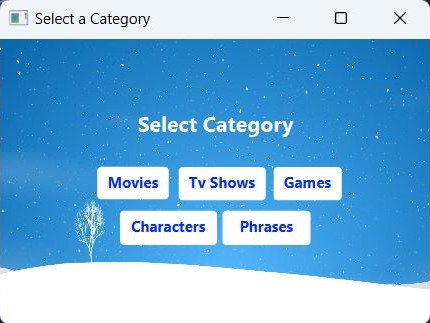




1. **Hint window:** designed to show user hint (one letter) of the word or phrase. A confirm alert will be shown before it moves to the other window. Once user presses OK, they cannot go back and 1 point will be subtracted from their score. Continue is clicked to continue the game. The hint value is subtracted by 1 and the hint button will be disabled.

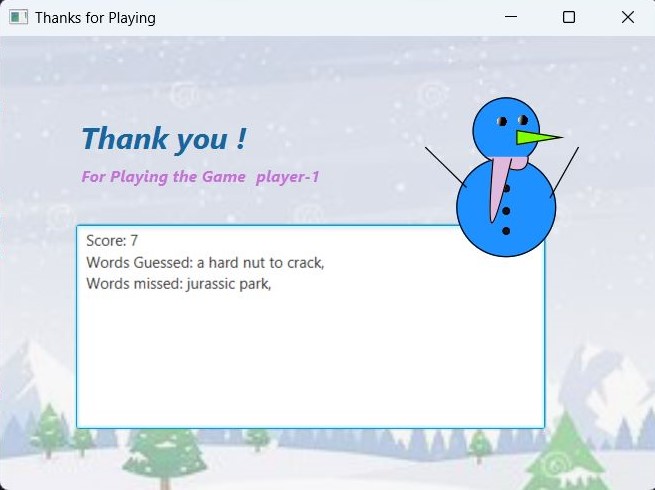


1. **Outcome window:** displays the outcome of the player and whether their gameplay ended up being successful or not. When correct, the user will get the number of attempts they have left as their score and the word will be added to the correct words array list. If incorrect, the word will be added to the incorrect words array list. It also gives the user the options to play again or quit the game.



1. **The category window:** allows user to choose from the categories of word again (new word from the file selected will be retrieved). It is only shown if the user selects “try again” on the ***outcome window***.

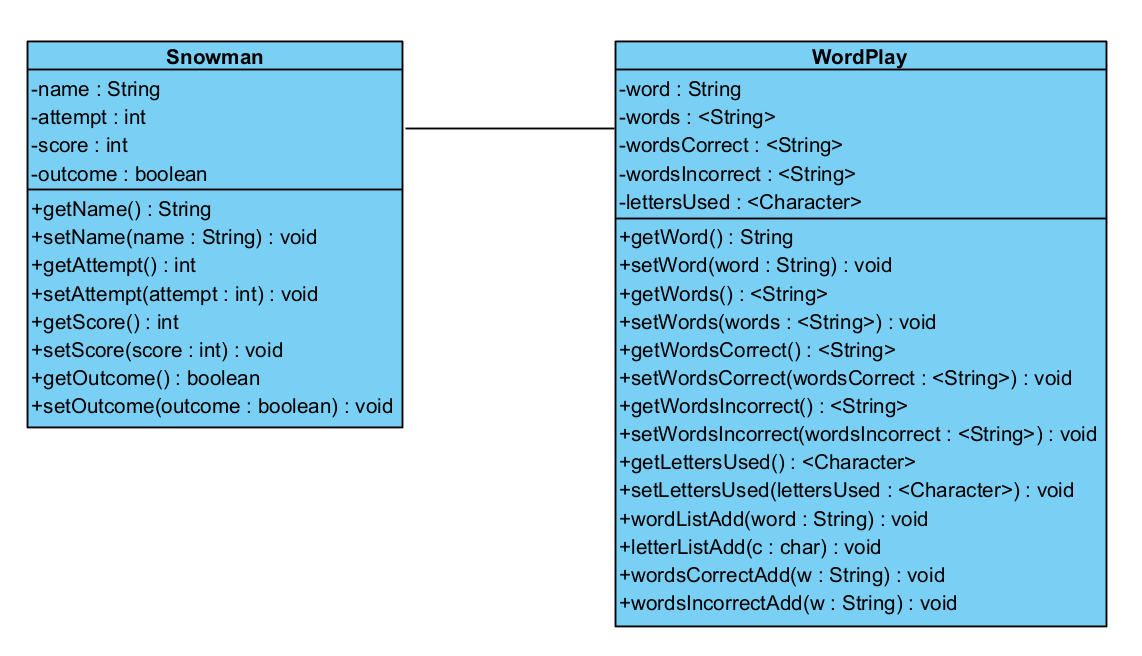
For this window, a new stage got gameplay is retrieved. In addition, it also uses the array list in the Snowman class called “words” to make sure to not add a word that was previously guessed by the user.



1. **Result window:** displays final score of the player, the words that they guessed, and the words that they missed. It will also display a message thanking them for playing, along with their username. This is the end of the game.

To do this, it retrieves the name and score value of the player object, as well as the array lists containing the words guessed and the words not guessed.

**Class Structure**

****

We have designed the class structure using UML diagrams and it includes 2 regular classes and 7 controllers.

1. **Snowman**: It stores the username of the player, the number of attempts the user has per each gameplay, the total score they have, and the outcome for each gameplay (along with their setters and getters).
2. **WordPlay**: It stores the word and array lists for total words used, words guessed correctly, words guessed incorrectly, and letters used with the setters and getters. There are also 4 other methods used to add values to the array lists within the class.
3. LogController
4. GamePlayController
5. GuessController
6. OutcomeController
7. CategoryController
8. FinalController

**Conclusion**

***Paula***

While working on the project, I have learned that I need to be more lenient when it comes to letting others work on their part. It has been hard for me to let go and allow others to do their tasks without monitoring every aspect of the project 24/7. It can be tedious sometimes, since inconsistencies arise, and it feels like doing most of it myself can save us time and effort. However, if I continue taking on all the work (I am not forced to do this. I do it voluntarily and have been asked to reconsider my teamworking methods by them and I agree), my team members will never learn how to properly work with Scene Builder and implement the techniques learned in class by themselves.

One thing I would change would be to assign the workload evenly. However, since the code is not complete, we can still try giving each other different tasks and continue to work together the way this project was designed to be.

That being said, the teamwork has been great from beginning to end. Everyone does what they are assigned and when difficulties arise, each of us seek others out for help. My group members are capable enough and I need to understand this is not a “one-man” job in order for this project to be successful.

***Simerpreet***

Our group project named as Snowman is completed by three of us with great excitement and we also learned a lot in terms of skills and teamwork. We all contributed equally and did a proper planning of how to create the project. We proposed multiple ideas and wrote them down after that we go with the one which seemed best. In starting we were so excited but in the middle of the project we got confused in functionality and code. Everyone was discussing their own ideas and solution and it was getting little mess. Then we used internet and cleared our mind with proper discussion. Fortunately, things get better and we were able to complete this project. It was great time doing the project and we really learnt some coding skills and team work importance. The end result was so pleasing that we are going to remember the time spend and stuff we did as a pleasant memory.

We think that by writing out proper functionality and expressing our ideas more effectively would make us complete the project more earlier and in an organised way.

***Simranjeet***

Snowman project was great opportunity to know how working in a project is, as of our group members paula, simerpreet and me all of us really go well with each other, we have good coordination and everyone is just ready to do the work assigned and configure the solution to any problem ahead.

Me as a person in this project was responsible for doing the interfaces part from the scene builder I did my part well and since there were some issues as per the standard way to do it, paula helped me configure them out, that was when I learned this from her. Simer was good with the file that stores all the words to be guessed throughout the game. And paula did the coding part, which was the largest part in it, I personally wanted to do some things in it as well . However, paula did it really well. The issues that we had while working on this project were relating to the acess of same code. Otherwise as a group we did great and after working on this project I quite felt like I should take part in the project more actively and then I am good.