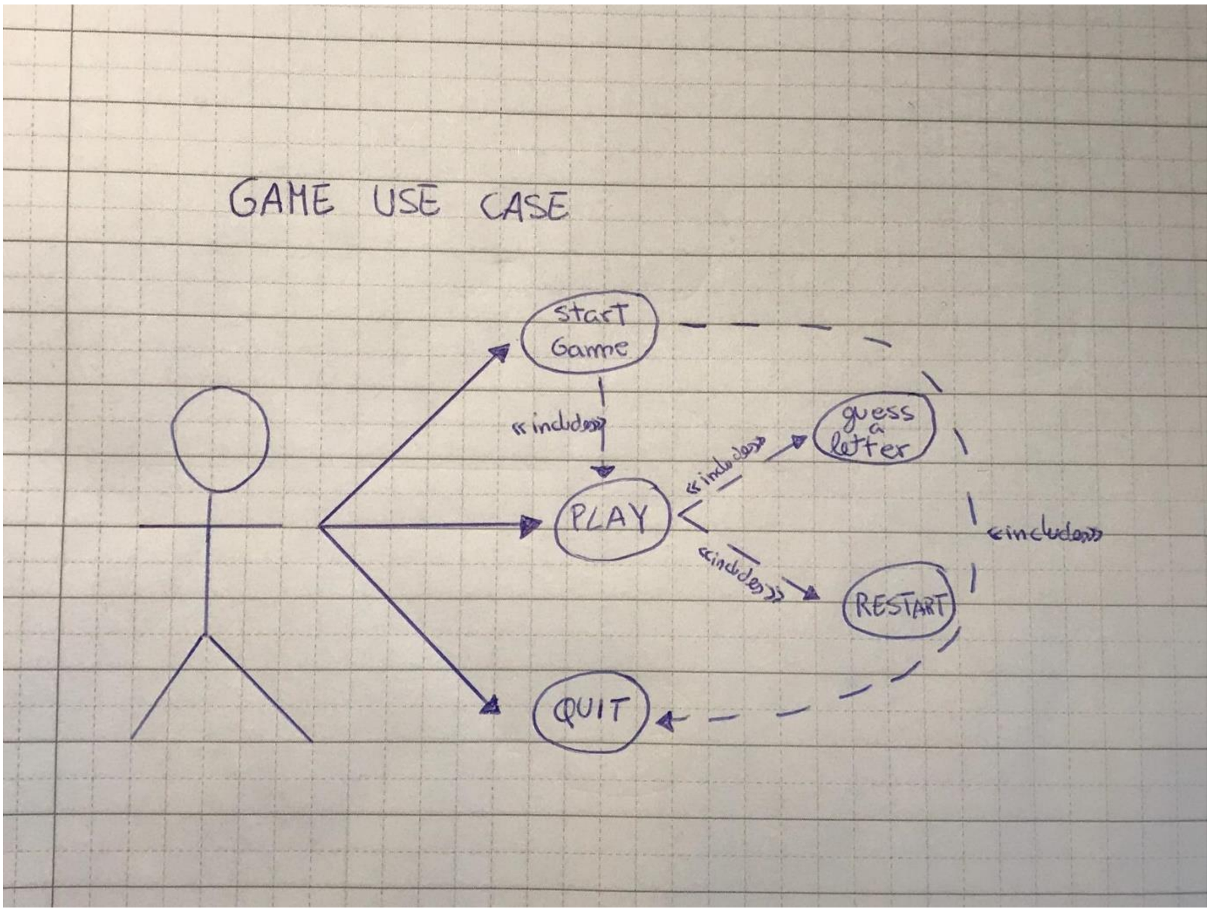
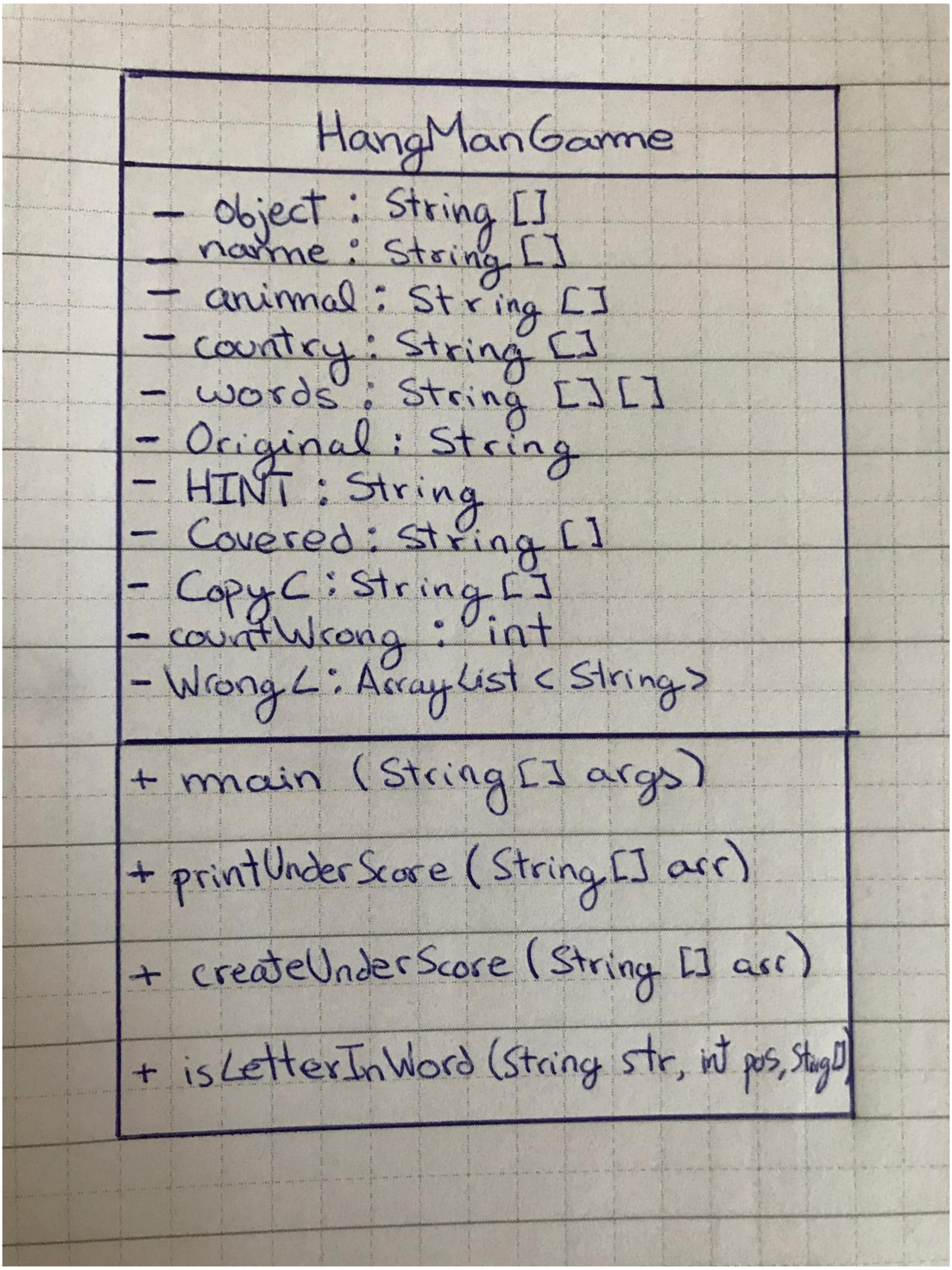
Use Case, State Machine, Class Diagram

Use case diagram

This use case diagram represents the user's interaction with the system and the relationship between the user and different use cases in which the user is involved.



Modelling Structure

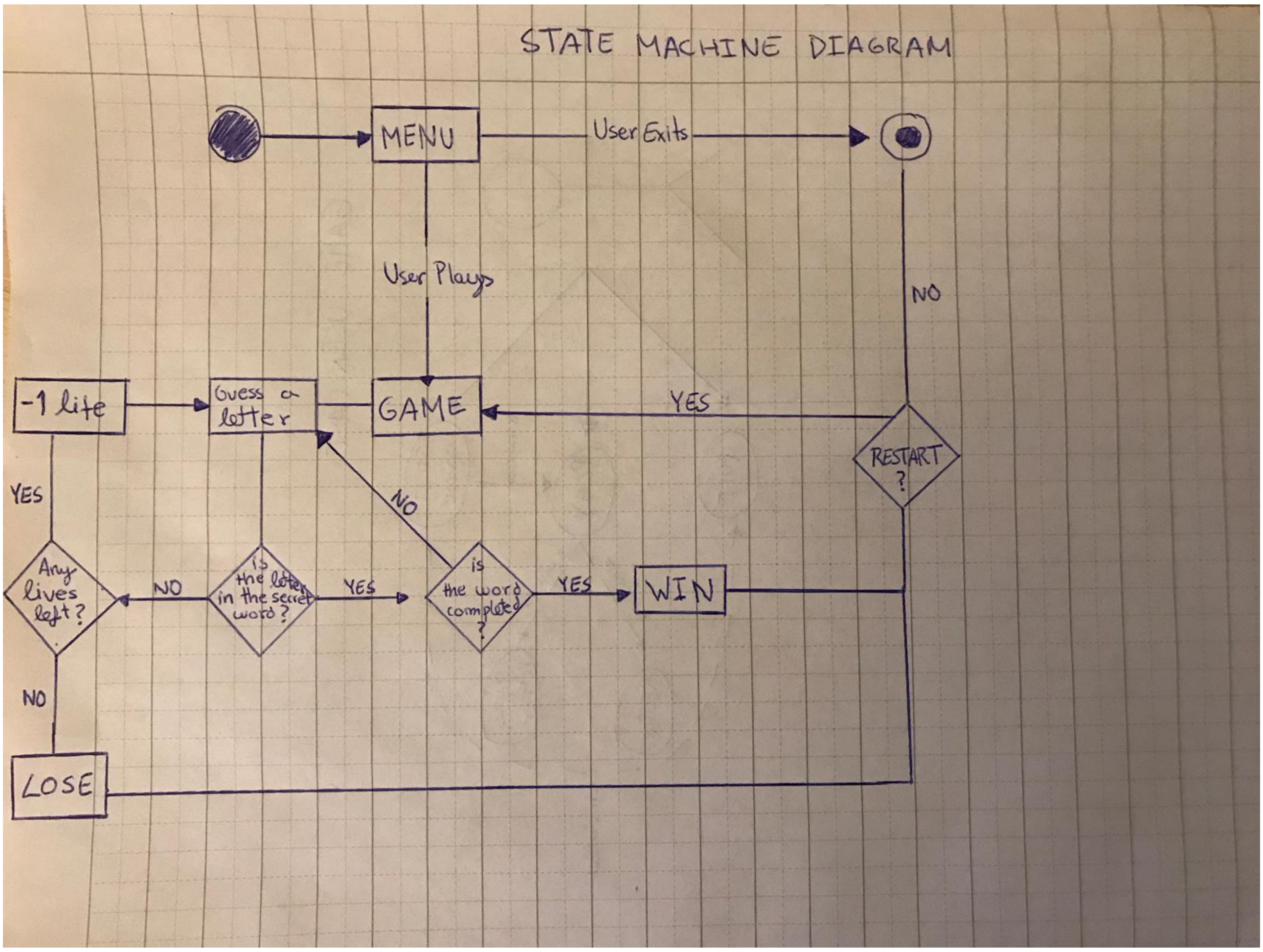


HangManGame

- object : String []
- name : String []
- animal : String []
- country : String []
- words : String [] []
- Original : String
- HINT : String
- Covered : String []
- CopyC : String []
- countWrong : int
- WrongL : ArrayList < String>

+ main (String [] args)
+ printUnderScore (String [] arr)
+ createUnderScore (String [] arr)
+ isLetterInWord (String str, int pos, String[]

State Machine Diagram



UC2 – Play Game Main Scenario

Precondition: The user knows the game rules.

Postcondition: The user received an outcome from the game.

1. The user selects the play button.
2. The system starts the game and generates the secret word.
3. The user inputs a letter as a guess.
4. The system reads the input letter.
   If the user didn't get 7 guesses wrong go to 3.
5. The user guesses the secret word and wins the game.
6. The system shows that the user won the game and gives the option of playing again or exit the game.
7. The user selects the play again button.
8. The system restarts the game and generates a new secret word.
   Go to 2.

Alternative Scenario

3.1 Invalid user input, the system counts as a wrong guess and resets the text field for a new input.

3.2 User input is a duplicate, the system counts as a correct guess and proceeds with the game.

5.1 The user gets 7 guesses wrong, the system shows the user that he has lost the game and displays the option to play again or exit the game.

7.1 The user selects the exit game button, the system closes the game.