German University in Cairo
Department of Computer Science
Dr. Nourhan Ehab

Concepts of Programming Languages, Spring Term 2022
Project 1: Pro-Wordle

Due: 10th May 2022

1. Project Description

In this Project you are going to implement a modified version of the game Wordle [1] in Prolog (Pro-Wordle). The Pro-Wordle game is made up of two main phases: (1) a knowledge base building phase and (2) a game play phase.

In the KB building phase, the player is prompted to enter words with their categories. Corresponding facts are added to the KB using the predicate `word/2` where `word(W,C)` is true if `W` has a category of `C`. This continues until the player enters `done`. Below is a sample of the interaction between the game and the player in the KB building phase.

```
Welcome to Pro-Wordle!
----------------------

Please enter a word and its category on separate lines:
|: horse.
|: animals.
Please enter a word and its category on separate lines:
|: panda.
|: animals.
 Please enter a word and its category on separate lines:
|: hello.
|: greetings.
Please enter a word and its category on separate lines:
|: banana.
|: fruits.
Please enter a word and its category on separate lines:
|: bison.
|: animals.
Please enter a word and its category on separate lines:
|: hoard.
|: collections.
Please enter a word and its category on separate lines:
|: done.

Done building the words database...
```

In the example above, the following Prolog KB will be constructed.

---

[1] Check out the game here: `https://www.nytimes.com/games/wordle/index.html`

```
word(horse,animals).
word(panda,animals).
word(hello,greetings).
word(banana,fruits).
word(bison,animals).
word(hoard,collections).
```

After the KB building phase, the game play phase begins. The game displays to the user the available categories to pick one from. The player is then prompted to pick a length and category for the word to be guessed. The game picks a word of the given length and category and the guessing game begins. The player is allowed a maximum number of guesses equal to the entered length plus one. In each guess, the user must enter a word of the chosen length. The game then displays to the user the correctly entered letters (not necessarily in correct positions), and the correctly entered letters in correct positions. Below is a sample of the interaction between the game and the player in the game play phase based on the example KB provided earlier.

```
The available categories are: [animals,greetings,fruits, collections]
Choose a category:
|: animals.
Choose a length:
|: 9.
There are no words of this length.
Choose a length:
|: 5.
Game started. You have 6 guesses.

Enter a word composed of 5 letters:
|: hello.
Correct letters are: [h,e,o]
Correct letters in correct positions are: [h]
Remaining Guesses are 4

Enter a word composed of 5 letters:
|: hell.
Word is not composed of 5 letters. Try again.
Remaining Guesses are 4

Enter a word composed of 5 letters:
|: hoard.
Correct letters are: [h,o,r]
Correct letters in correct positions are: [h,o]
Remaining Guesses are 3

Enter a word composed of 5 letters:
|: horse.
You Won!
```

Here is another sample where the player loses.

```
The available categories are: [animals,greetings,fruits,collections]
Choose a category:
|: things.
This category does not exist.
Choose a category:
|: animals.

Choose a length:
|: 5.
Game started. You have 6 guesses.

Enter a word composed of 5 letters:
|: bison.
Correct letters are: [s,o]
Correct letters in correct positions are: []
Remaining Guesses are 4

Enter a word composed of 5 letters:
|: bison.
Correct letters are: [s,o]
Correct letters in correct positions are: []
Remaining Guesses are 3

 Enter a word composed of 5 letters:
|: bison.
Correct letters are: [s,o]
Correct letters in correct positions are: []
Remaining Guesses are 2

 Enter a word composed of 5 letters:
|: bison.
Correct letters are: [s,o]
Correct letters in correct positions are: []
Remaining Guesses are 1

 Enter a word composed of 5 letters:
|: bison.
 You lost!
```

Your implementation must contain the below predicates. You are allowed to use any other helper predicates if needed. All the examples given below are according to the KB at the beginning of Page 2.

- `is_category(C)`: succeeds if `C` is an available category in the KB.
  Examples:

  ```
  ?- is_category(animals).
  true.

  ?- is_category(C).
  C= animals;
  C= animals;
  C= greetings;
  C= fruits;
  C= animals;
  C= collections.
  ```

- `categories(L)`: succeeds if `L` is a list containing all the available categories without duplicates.
  Examples:

  ```
  ?- categories(L).
  L=[animals,greetings,fruits,collections]
  ```

- `available_length(L)`: succeeds if there are words in the KB with length L.
  Examples:

  ```
  ?- available_length(5).
  true.
  ```

- `pick_word(W,L,C)`: succeeds if `W` is a word in the KB with length L and category C.
  Examples:

  ```
  ?- pick_word(W,5,animals).
     W=horse;
     W=panda;
     W=bison.
  ```

- `correct_letters(L1,L2,CL)`: succeeds if `CL` is a list containing the letters in both L1 and L2.
  Examples:

  ```
  ?- correct_letters([h,e,l,l,o],[h,o,r,s,e],CL).
  CL=[h,o,e];
  false.
  ```

- `correct_positions(L1,L2,PL)`: succeeds if `PL` is a list containing the letters that occur in both `L1` and `L2` in the same positions.
  Examples:

  ```
  ?- correct_positions([h,e,l,l,o],[h,o,r,s,e],CP).
  CP=[h];
  false.
  ```

- `build_kb`: reponsible of the KB building phase as described above.
- `play`: responsible of the game play phase as described above.
- `main`: the main predicate that will be queried to initiate the KB building phase then the game play phase.

2. **Teams.** You are allowed to work in teams of four members. You must stick to the teams submitted in the team submission form. IDs for the submitted teams are posted on the CMS.

3. **Report.** You should submit a short report with your code containing a brief description of the implemented predicates and screenshots of two different runs of the game. The two runs must show the KB building phase. One run should show a winning scenario in the game play phase, and the other should show a losing scenario in the game play phase. The two runs must be different from the two runs shown in this description.

4. **Deliverables.** You should submit a single `.pl` file named with your team ID, and a single `.pdf` file with your report also named with your team ID. The submission link will be posted on the CMS prior to the submission.