Report Predicates:

(1) main/0:

- Displays 'Welcome to Pro-Wordle!'.
- Calls the (2)build_KB/0 predicate to start building the knowledge base.
- Displays the available categories by calling the (3)categories/1 predicate.
- Calls the (4)play/0 predicate.

(2) build_KB/0:

- Displays 'Please enter a word and its category on separate lines:'.
- Reads the entries line by line and asserts them as word and category as a word/2 structure in the knowledge base.
- Calls itself recursively.
- Keeps reading lines until the player enters ‘done’(base case).

(3) categories/1:

□ Collecting all the available categories in a list by checking if it (5)is_category/1, the (setof) ensuring no duplicates i.e. no category to be displayed twice.

(4) play:

- Calls the (6)choose_category/1 predicate, which makes the player choose a category.
- Calls the (7)choose_length/2 predicate, which makes the player choose a word length for a word in this category.
- Evaluates the guesses = the word length+1.
- Displays 'Game started’ and also the number of guesses the player has.
- Calls the (8)pick_word/3 predicate that is the game picks the first word from the knowledge base from category (Category) and with length (Length).

- Calls and passes the word with its length and category and number of guesses (Trials) to the (9)trytoguess/4 that asks the player to guess the word till winning (guessing correctly) or losing (number of guesses run out).

(5) is_category/1:

- Checking that there is a category that matches the given arity.

(6) choose_category/1:

- Displays 'Choose a category:'.
- Reads the category entered by the player.
- Calls the (3)categories/1 predicate and finds a substitution for (List) of the available categories.
- Checks if the chosen category is one of the available categories.
- Matches the entered category with the arity (L) i.e. (Category) that will be used in (4)play/0 and will be passed to(8) trytoguess/4.
- Displays 'This category does not exist.' If the chosen category is not one of the available categories and calls itself recursively till it finds a match.

(7) choose_length/2:

- Displays 'Choose a length:'.
- Reads the length entered by the player.
- Checks the availability of the length in the given category by using the (8)pick_word/3 predicate.
- Matches the entered length with the arity (Length) that will be used in (4)play/0 and will be passed to (8)trytoguess/4.
- Displays 'There are no words of this length.' If the chosen length is not available in the given category.

(8) pick_word/3:

- Holds if there is a word in the knowledge base in the given category with the given length.

(9) trytoguess/4:

- Displays 'Enter a word composed of ' followed by the word length.
- Reads the guessed word entered by the player.
- Here there are four cases:
    1. Displays 'You Won!' if comparing the entered word with the given word gives true, i.e. the player guessed it correctly.
    2. Displays 'You lost!' if the comparison gives false and it was the last trial for the player, i.e. number of trials runs out.
    3. If the player didn’t guess correctly (the Trial word isn’t of the same Length) and still has a number of Trails,
        - Displays 'Word is not composed of ' followed by the word length and asks the player to ‘Try again.'
        - Displays 'Remaining Guesses are ' followed by the number of Trials remaining. □ Calls itself recursively.
    4. If the guessed word is of the same length but isn’t the exact same word and the player still has a number of guesses (Trials2)
        - Using the atom_chars/2 predicate which holds if the (WW) is the list of characters in the word (W).
        - Using it again to hold when the (Trial1) is the list of characters in the word (Trial).
        - Passing the lists (WW) and (Trail1) to (11)correct_letters/3 predicate that holds if (L1) is the list of correct letters i.e. the common letters in both words.
        - Ensuring no duplicates in (L1) using the (12)remove/2 predicate that holds if (L11) is the result of removing duplicates from (L1).
        - Passing the lists (WW) and (Trail1) again to (13)correct_positions/3 predicate that holds if (L22) is the list of correct letters in the correct positions.
        - Displays 'Correct letters are: ' followed by the list of correct letters (L11).
        - Displays 'Correct letters in correct positions are: ' followed by the list of correct positions (L22).

- Evaluate the number of remaining (Trials2) = Trials -1.
- Displays the remaining guesses as 'Remaining Guesses are ' followed by (Trials2). □ Calls itself recursively.

(10) correct_letters/3:

□ Passing the arities to the intersect/3 predicate:

□ Recursive case: holds if the head of (L1) i.e. (X) is a member of (L2) i.e. (M) and that (Z) is the result of intersecting the tail (Y) with (M).

□ Recursive case: holds if the head of (L1) i.e. (X) is not a member of (L2) i.e. (M) and that (Z) is the result of intersecting the tail (Y) with (M).

□ Base case: intersecting an empty list with anything is an empty list.

(11) remove/2:

□ Base case: the result of removing duplicates from an empty list is an empty list.

□ Recursive case: holds if ([H|T1]) is the result of removing duplicates from ([H|T]) that if the head is not a member of the tail i.e. should occur in the result, and (T1) is the result of removing duplicates from (T).

□ Recursive case: holds if (TL) is the result of removing duplicates from ([H|T]) that if the head is a member of the tail i.e. should not occur in the result, and (TL) is the result of removing duplicates from (T).

(12) correct_positions/3:

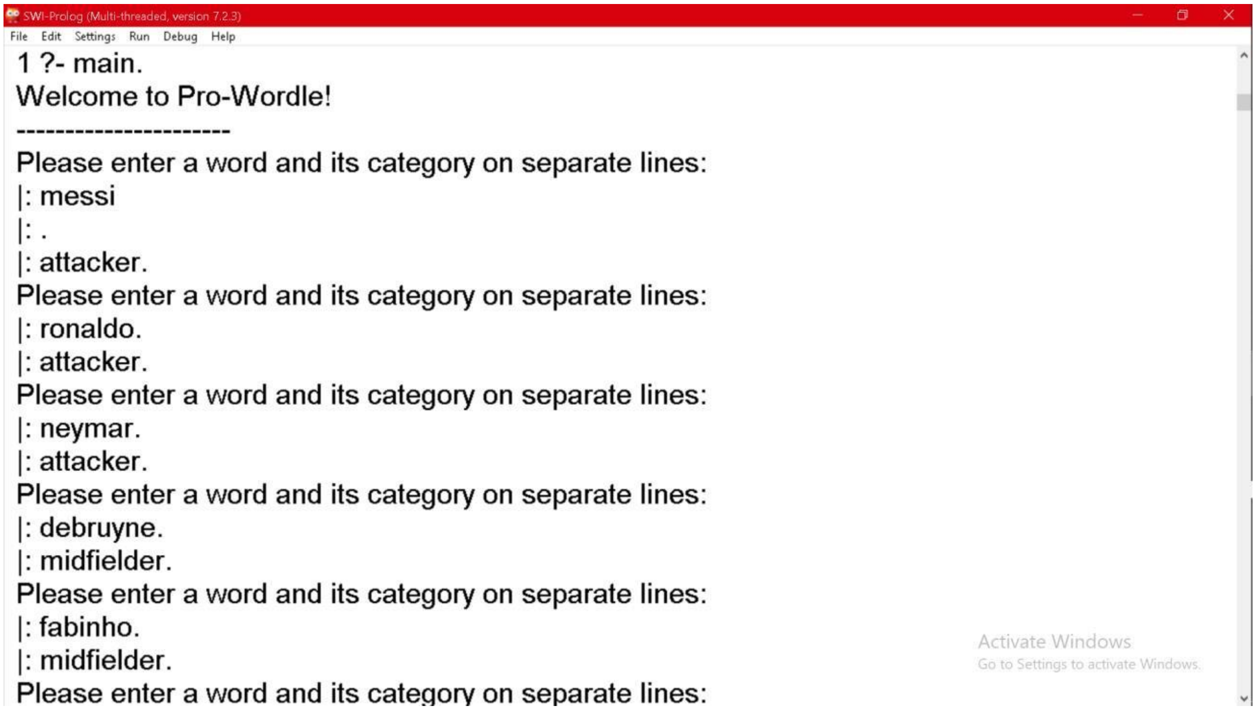
□ Passing the arities to the intersect1/3 predicate:

□ Recursive case: holds if ([X|Z]) is the result that's if (X) occurs in the head in both lists i.e. should be in the head of the result, and (Z) is the result pf intersecting1 (Y) and (T).

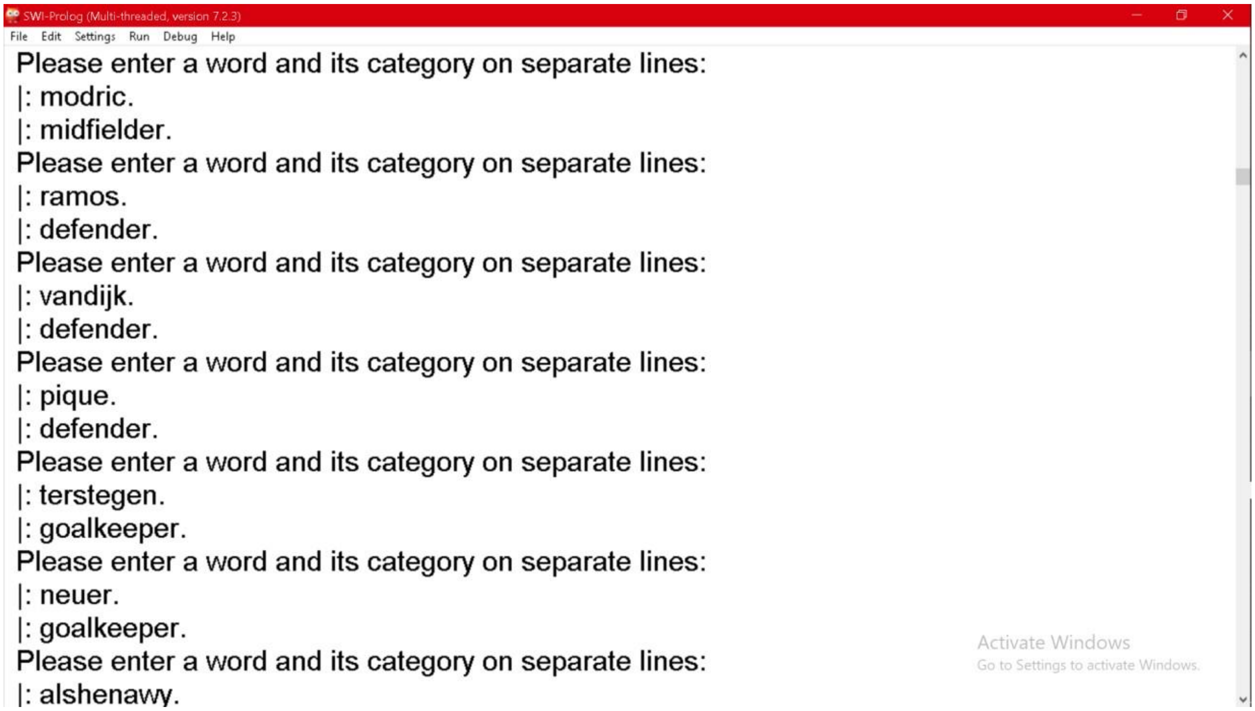
- Recursive case: holds if (Z) is the result that’s if comparing values of both lists’ heads gives false and (Z) is the result of itersecting1 (Y) and (T1).
- Base case: intersecting1 (finding same letters in same positions) of two empty lists is an empty list.

Runs

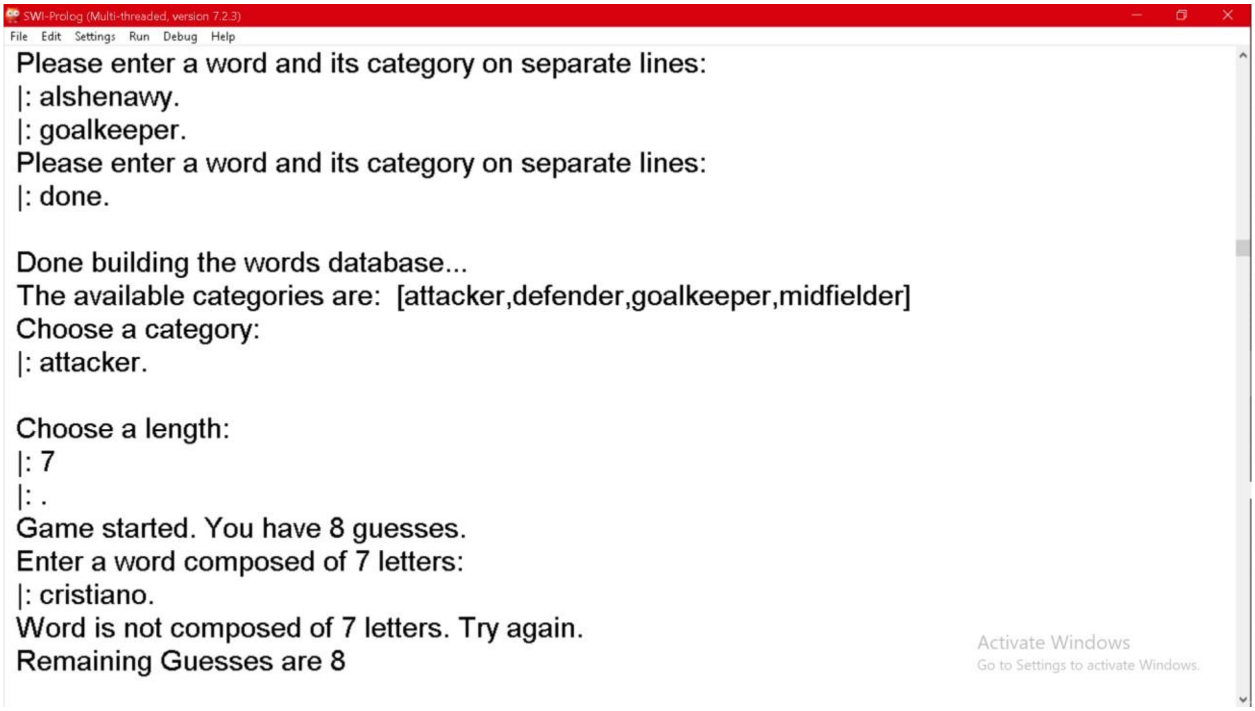
Winning scenario:


SWI-Prolog (Multi-threaded, version 7.2.3)
File
Edit
Settings
Run
Debug
Help
Activate Windows
Go to Settings to activate Windows.

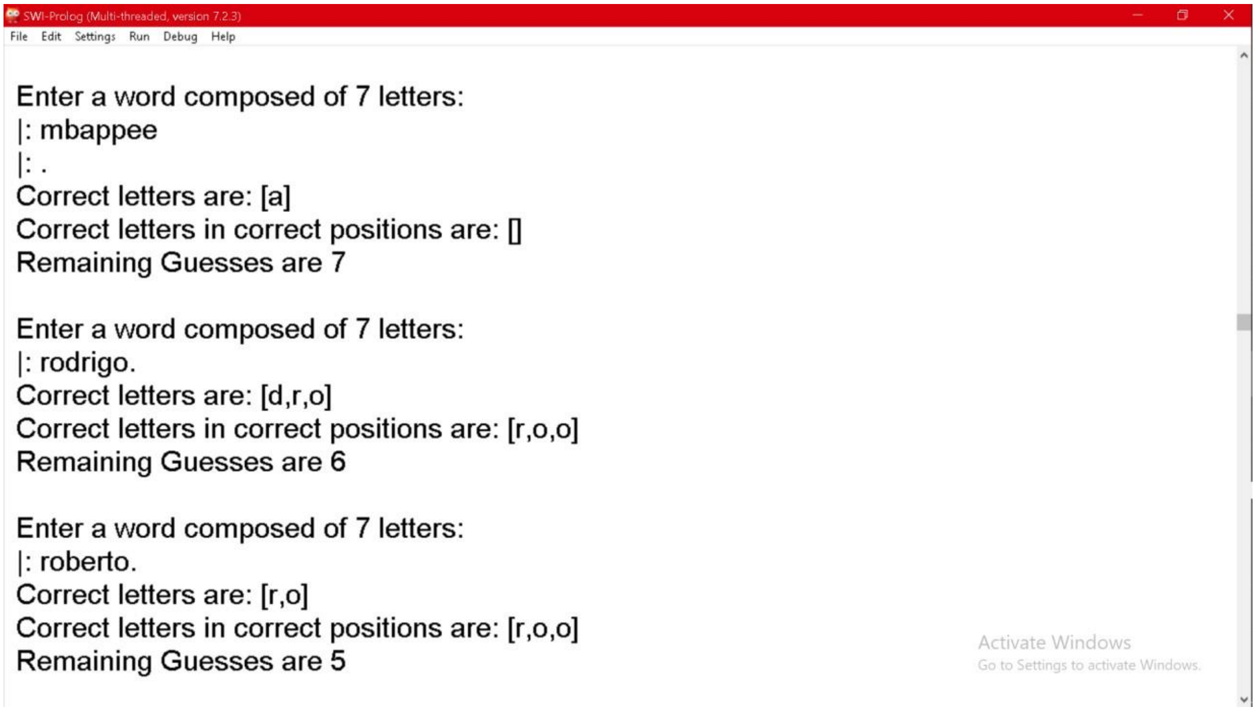
```
1 ?- main.
Welcome to Pro-Wordle!
----------------------
Please enter a word and its category on separate lines:
|: messi
|: .
|: attacker.
Please enter a word and its category on separate lines:
|: ronaldo.
|: attacker.
Please enter a word and its category on separate lines:
|: neymar.
|: attacker.
Please enter a word and its category on separate lines:
|: debruyne.
|: midfielder.
Please enter a word and its category on separate lines:
|: fabinho.
|: midfielder.
Please enter a word and its category on separate lines:
```


SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help

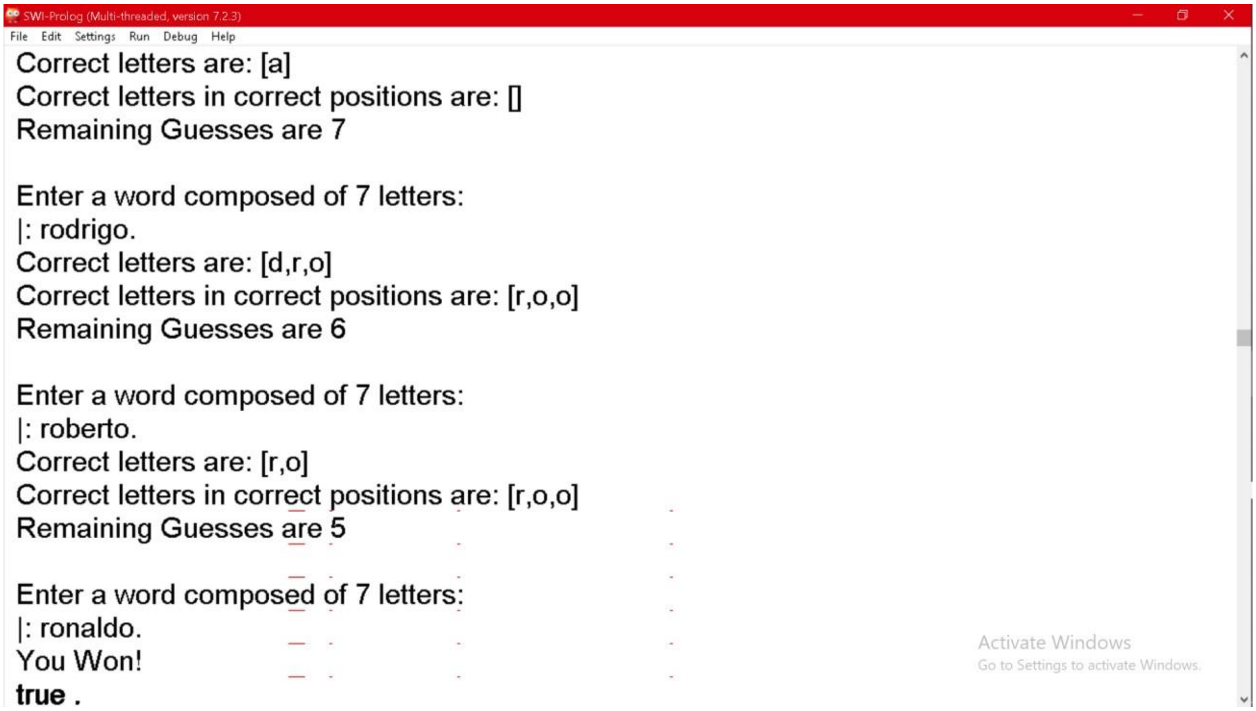
```
Please enter a word and its category on separate lines:
|: modric.
|: midfielder.
Please enter a word and its category on separate lines:
|: ramos.
|: defender.
Please enter a word and its category on separate lines:
|: vandijk.
|: defender.
Please enter a word and its category on separate lines:
|: pique.
|: defender.
Please enter a word and its category on separate lines:
|: terstegen.
|: goalkeeper.
Please enter a word and its category on separate lines:
|: neuer.
|: goalkeeper.
Please enter a word and its category on separate lines:
|: alshenawy.
```


SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help

```
Please enter a word and its category on separate lines:
|: alshenawy.
|: goalkeeper.
Please enter a word and its category on separate lines:
|: done.

Done building the words database...
The available categories are:  [attacker,defender,goalkeeper,midfielder]
Choose a category:
|: attacker.

Choose a length:
|: 7
|: .
Game started. You have 8 guesses.
Enter a word composed of 7 letters:
|: cristiano.
Word is not composed of 7 letters. Try again.
Remaining Guesses are 8
```

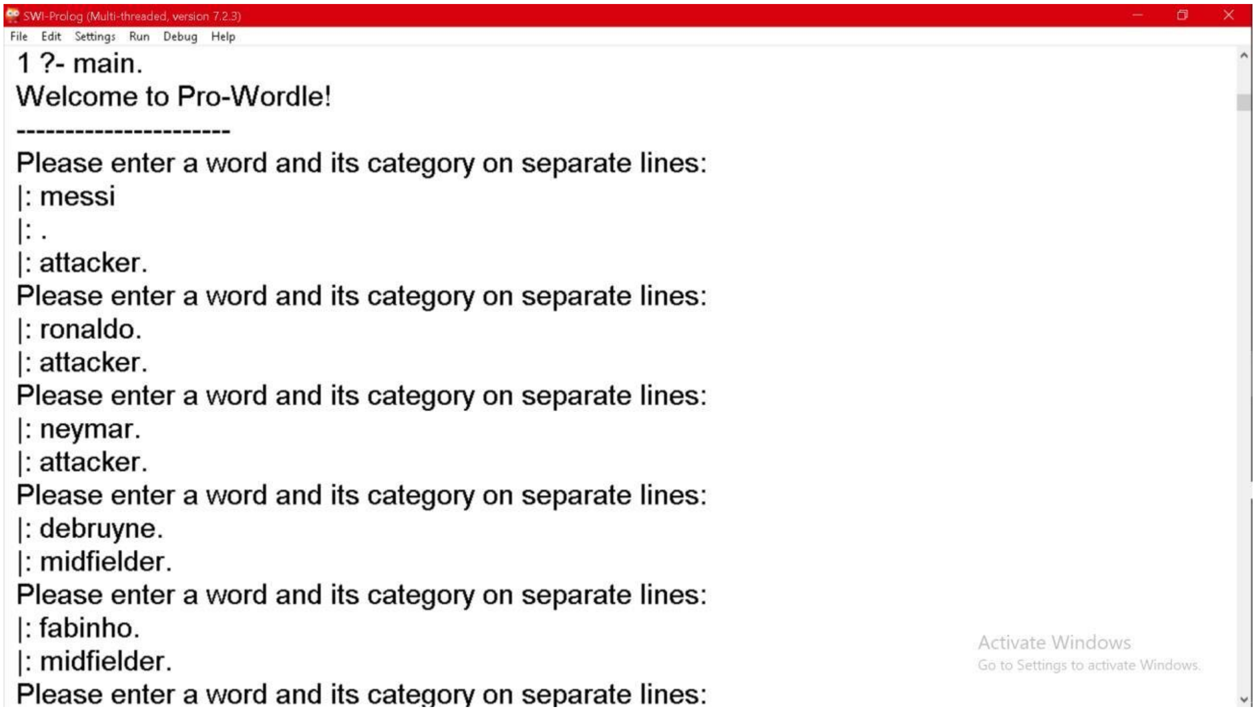

SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help
Activate Windows
Go to Settings to activate Windows.

```
Enter a word composed of 7 letters:
|: mbappee
|: .
Correct letters are: [a]
Correct letters in correct positions are: []
Remaining Guesses are 7

Enter a word composed of 7 letters:
|: rodrigo.
Correct letters are: [d,r,o]
Correct letters in correct positions are: [r,o,o]
Remaining Guesses are 6

Enter a word composed of 7 letters:
|: roberto.
Correct letters are: [r,o]
Correct letters in correct positions are: [r,o,o]
Remaining Guesses are 5
```

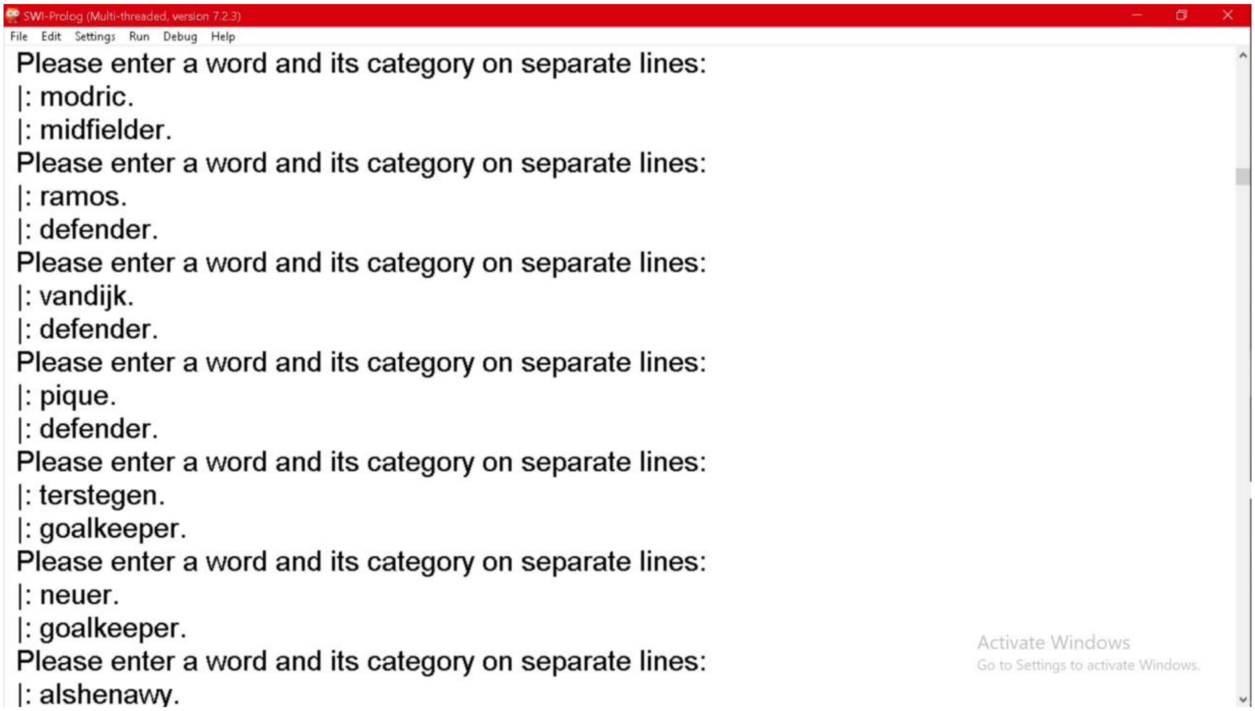

SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help
Activate Windows
Go to Settings to activate Windows.

```
Correct letters are: [a]
Correct letters in correct positions are: []
Remaining Guesses are 7

Enter a word composed of 7 letters:
|: rodrigo.
Correct letters are: [d,r,o]
Correct letters in correct positions are: [r,o,o]
Remaining Guesses are 6

Enter a word composed of 7 letters:
|: roberto.
Correct letters are: [r,o]
Correct letters in correct positions are: [r,o,o]
Remaining Guesses are 5

Enter a word composed of 7 letters:
|: ronaldo.
You Won!
true .
```

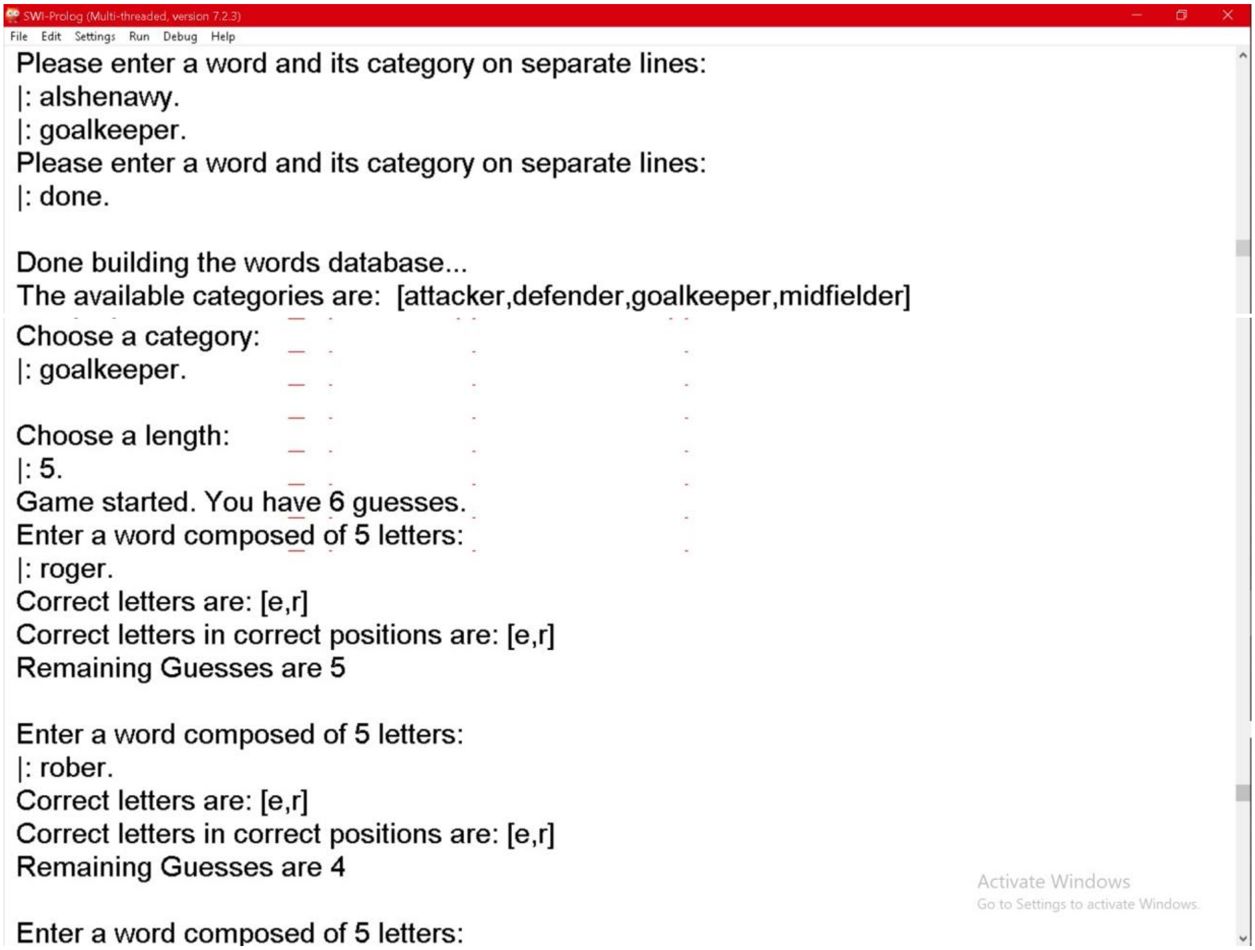
Losing scenario:


SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help

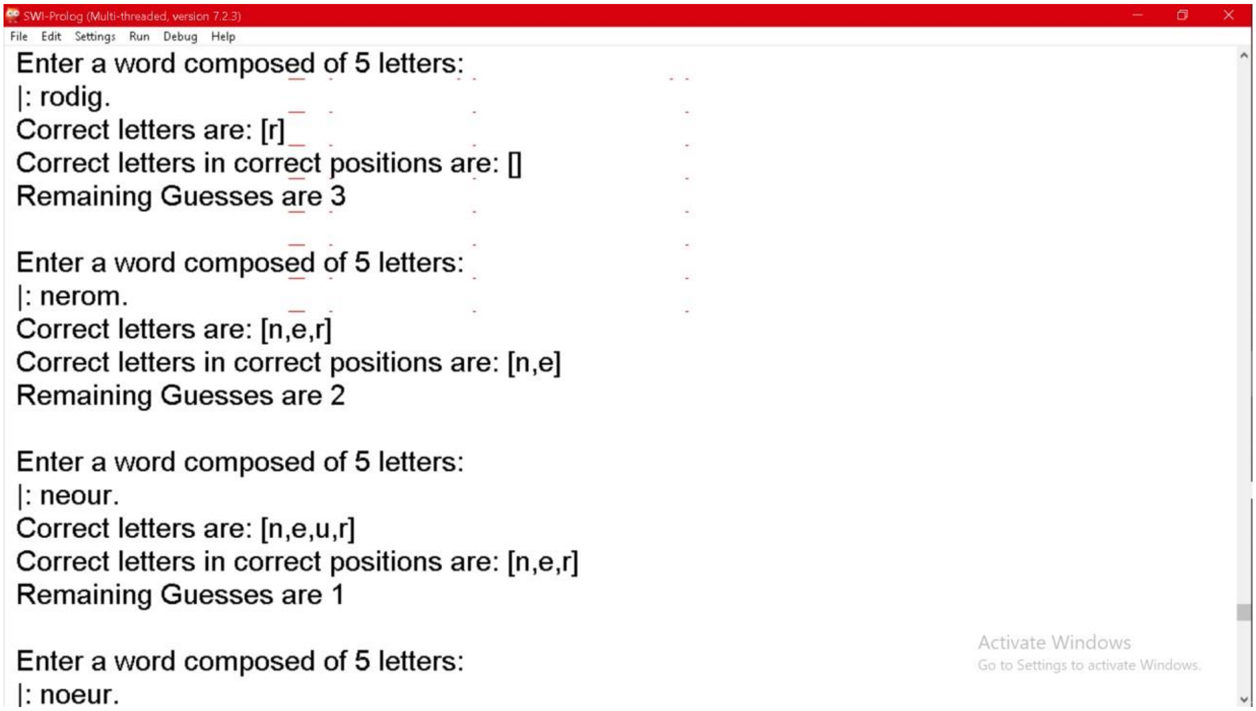
```
1 ?- main.
Welcome to Pro-Wordle!
----------------------
Please enter a word and its category on separate lines:
|: messi
|: .
|: attacker.
Please enter a word and its category on separate lines:
|: ronaldo.
|: attacker.
Please enter a word and its category on separate lines:
|: neymar.
|: attacker.
Please enter a word and its category on separate lines:
|: debruyne.
|: midfielder.
Please enter a word and its category on separate lines:
|: fabinho.
|: midfielder.
Please enter a word and its category on separate lines:
```


SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help

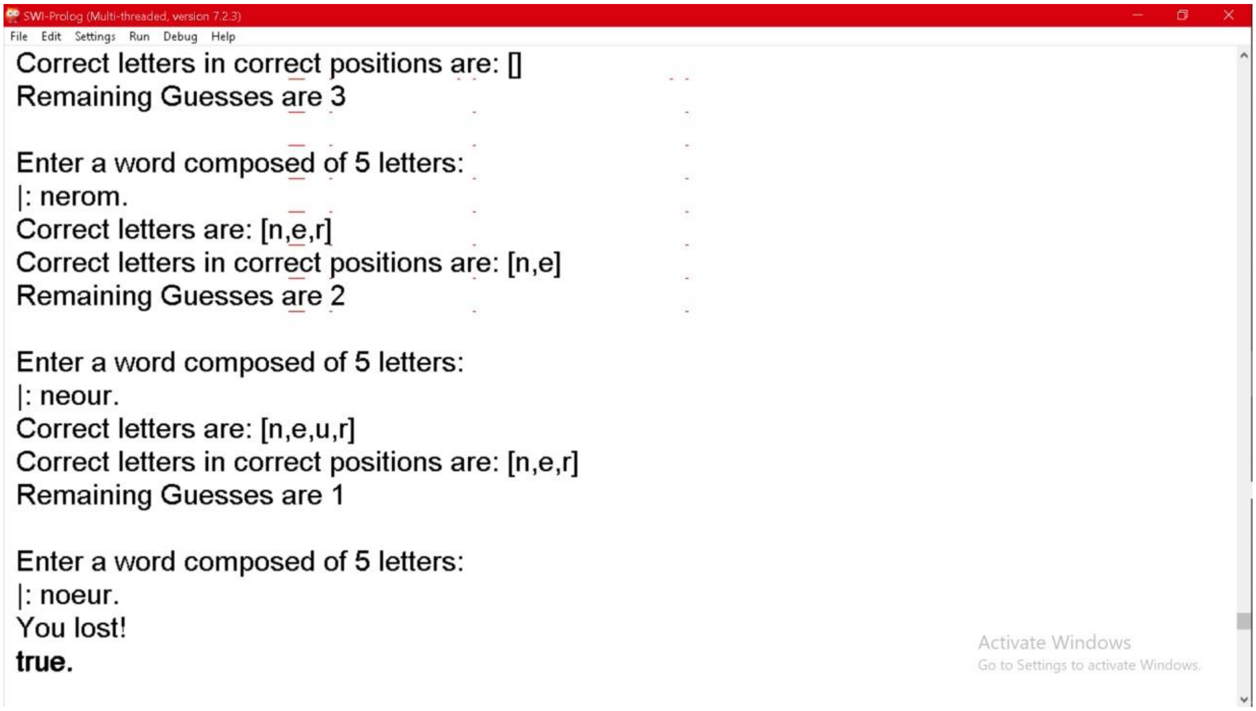
```
Please enter a word and its category on separate lines:
|: modric.
|: midfielder.
Please enter a word and its category on separate lines:
|: ramos.
|: defender.
Please enter a word and its category on separate lines:
|: vandijk.
|: defender.
Please enter a word and its category on separate lines:
|: pique.
|: defender.
Please enter a word and its category on separate lines:
|: terstegen.
|: goalkeeper.
Please enter a word and its category on separate lines:
|: neuer.
|: goalkeeper.
Please enter a word and its category on separate lines:
|: alshenawy.
```


SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help

```
Please enter a word and its category on separate lines:
|: alshenawy.
|: goalkeeper.
Please enter a word and its category on separate lines:
|: done.

Done building the words database...
The available categories are:  [attacker,defender,goalkeeper,midfielder]
Choose a category:
|: goalkeeper.

Choose a length:
|: 5.
Game started. You have 6 guesses.
Enter a word composed of 5 letters:
|: roger.
Correct letters are: [e,r]
Correct letters in correct positions are: [e,r]
Remaining Guesses are 5

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

Enter a word composed of 5 letters:
```


SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help
Activate Windows
Go to Settings to activate Windows.

```
Enter a word composed of 5 letters:
|: rodig.
Correct letters are: [r]
Correct letters in correct positions are: []
Remaining Guesses are 3

Enter a word composed of 5 letters:
|: nerom.
Correct letters are: [n,e,r]
Correct letters in correct positions are: [n,e]
Remaining Guesses are 2

Enter a word composed of 5 letters:
|: neour.
Correct letters are: [n,e,u,r]
Correct letters in correct positions are: [n,e,r]
Remaining Guesses are 1

Enter a word composed of 5 letters:
|: noeur.
```


SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help
Activate Windows
Go to Settings to activate Windows.

```
Correct letters in correct positions are: []
Remaining Guesses are 3

Enter a word composed of 5 letters:
|: nerom.
Correct letters are: [n,e,r]
Correct letters in correct positions are: [n,e]
Remaining Guesses are 2

Enter a word composed of 5 letters:
|: neour.
Correct letters are: [n,e,u,r]
Correct letters in correct positions are: [n,e,r]
Remaining Guesses are 1

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