



Concepts of Programming Languages (CSEN403) - Spring 2022
Project, Milestone 1
Team 32

REPORT

Milestone 1

The following report outlines a brief summary of the predicate descriptions provided in our code for milestone 1 of the project for [CSEN403] for the Spring Semester of 2022, as well as two screenshots for the different outcomes of the game (winning scenario & losing scenario).

Screenshots of different outcomes:

3 ?- main.

Please enter a word and its category on separate lines

|: ahmed.

|: names.

Please enter a word and its category on separate lines

|: mohamed.

|: names.

Please enter a word and its category on separate lines

|: ziad.

|: names.

Please enter a word and its category on separate lines

|: zain.

|: names.

Please enter a word and its category on separate lines

|: engineering.

|: faculty.

Please enter a word and its category on separate lines

|: pharmacy.

|: faculty.

Please enter a word and its category on separate lines

|: done.

Done building the words database...

The available categories are: [faculty,names]



Type here to search



Done building the words database...
The available categories are: [faculty,names]
Choose a category:
|: anything.
This category does not exist.
Choose a category:
|: names.
Choose a length:
|: 10.
There are no words of this length.
Choose a length:
|: 4.
Game started. You have 5 guesses.

Enter a word composed of 4 letters:
|: zain.
Correct letters are: [z,a,i]
Correct letters in correct positions are: [z]
Remaining Guesses are 4

Enter a word composed of 4 letters:
|: |



Type here to search



Remaining Guesses are 4

Enter a word composed of 4 letters:

|: zai.

Word is not composed of 4 letters. Try again.

Remaining guesses are 4

Enter a word composed of 4 letters:

|: noor.

This word does not exist

Remaining Guesses are 4

Enter a word composed of 4 letters:

|: zain.

Correct letters are: [z,a,i]

Correct letters in correct positions are: [z]

Remaining Guesses are 3

Enter a word composed of 4 letters:

|: |



Type here to search



Remaining guesses are 3

Enter a word composed of 4 letters:

|: zain.

Correct letters are: [z,a,i]

Correct letters in correct positions are: [z]

Remaining Guesses are 2

Enter a word composed of 4 letters:

|: zain.

Correct letters are: [z,a,i]

Correct letters in correct positions are: [z]

Remaining Guesses are 1

Enter a word composed of 4 letters:

|: zain.

You lost!

true .

9 ?- |



Type here to search



2 ?- main.

Please enter a word and its category on separate lines

|: iphone.

|: mobiles.

Please enter a word and its category on separate lines

|: samsung.

|: mobiles.

Please enter a word and its category on separate lines

|: oppo.

|: mobiles.

Please enter a word and its category on separate lines

|: salah.

|: players.

Please enter a word and its category on separate lines

|: messi.

|: players.

Please enter a word and its category on separate lines

|: jota.

|: players.

Please enter a word and its category on separate lines

|: sony.

|: mobiles.

Please enter a word and its category on separate lines

Activate Windows
Go to Settings to activate Windows.

Type here to search

Please enter a word and its category on separate lines

|: done.

Done building the words database...

The available categories are: [mobiles,players]

Choose a category:

|: players.

Choose a length:

|: 4.

Game started. You have 5 guesses.

Enter a word composed of 4 letters:

|: salah.

Word is not composed of 4 letters. Try again.

Remaining guesses are 5

Enter a word composed of 4 letters:

|: oppo.

Correct letters are: [o]

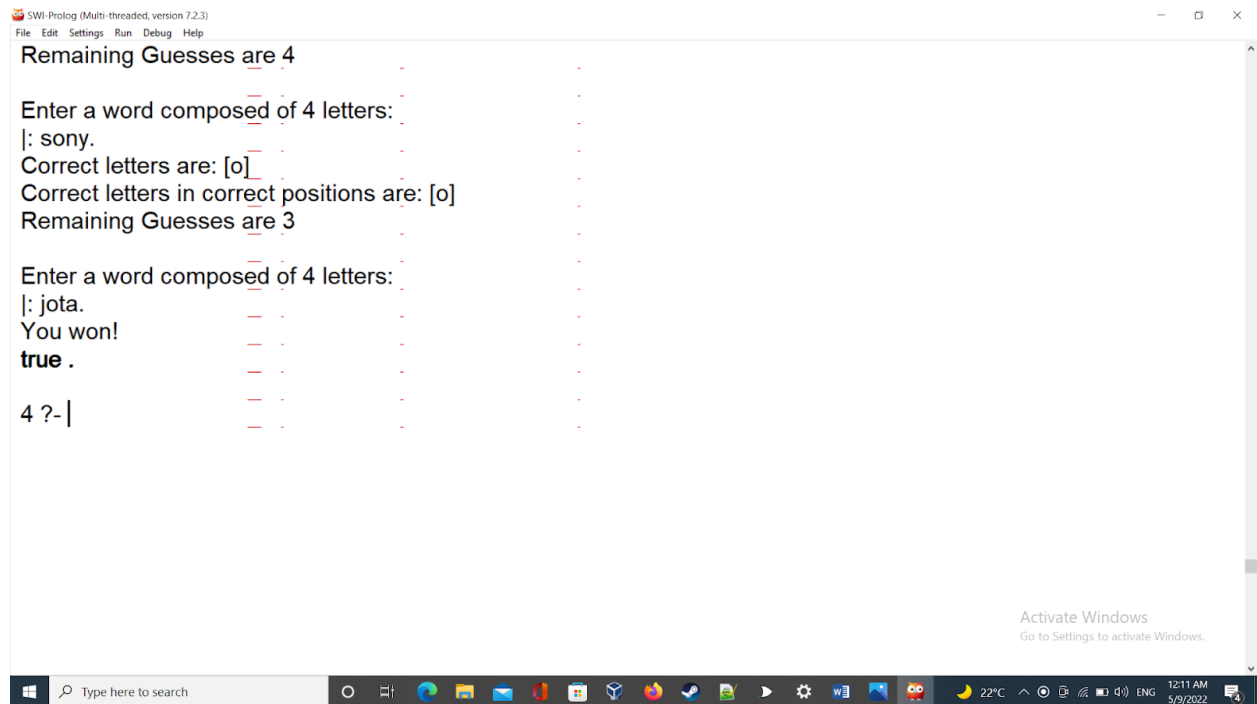
Correct letters in correct positions are: []

Remaining Guesses are 4

Enter a word composed of 4 letters:

|: sony.

Activate Windows
Go to Settings to activate Windows.



Predicates Brief Description:

`is_category(C):`

Is a predicate that checks if the variable C is the same variable has been put by the user, it will give true if C is one of the categories in the KB.

`categories(L):`

Is a predicate that checks if the given list is containing the categories without repeating any of them using a (set of) predefined predicate

`available_length(L):`

Is a predicate that checks if the KB is having a number of words with the same number of input (L) using `atom_length(_,L)` and `word(W,C)`

`pick_word(W,L,C):`

Is the predicate which checks if the variable W is one of the words in the KB with the same length and category using word predicate and `atom_length` predicate also

`correct_letters(L1,L2,CL):`

Is a predicate that evaluates to true when CL is a list that consists of the common letters in both L1 and L2. This determines the correct letters guessed by the user, and not necessarily in a correct order.

`correct_positions(L1,L2,PL):`

Is a predicate that evaluates to true when PL is a list that consists of the letters that occur in the same locations in both L1 and L2. This determines the correct letters guessed by the user in a correct order.

Main: It initializes the game by calling 'build_kb' then 'play' predicates.

build_kb (puts categories and their corresponding words entered by users):

- Prompts the user to enter a word by using the write predicate to display 'Please enter a word and its category on separate lines' then takes the word entered by user using read(W) predicate and passes it the predicate processing1(W).
- processing1(W) first checks if the word is done but if the word is not done it prompts the user to enter a category then adds word(W,C) to the knowledge base using assert predicate.

play:

1. displays the list containing categories L where L is got by categories predicate.
2. then prompts the user to enter category X and length Y.
3. passes X to choose_a_category(X, Cuser) predicate which returns a valid Category entered by user .
 - choose_a_category(X, Cuser) predicate checks if X belongs to list containing all categories by using categories(L) and member(X,L) predicates.

4. passes `Y` to `choose_a_length(Y,Cuser,Luser)` predicate which returns a valid Category entered by user .

- `choose_a_length(Y,Cuser,Luser)` predicate gets all words corresponding to category `Cuser` and checks if any of those words have the length `Y` using `word(W,Cuser)` and `atom_length(W,Y)` predicates.

5. Gets a list containing words with the category and length entered by user using `allwordswithC(Cuser,Listallw)` and `spec_words(Luser,Listallw,Listspecw)` predicates .

- `allwordswithC(Cuser,Listallw)` predicate gets a list `Listallw` with all words with category `C`.
- `spec_words(Luser,Listallw,Listspecw)` predicate gets a list `Listspecw` with words from list `Listallw` having length `Luser`.

6. Gets a random word from the previous list using `random`.

7. Uses predicate `engine(G,Rand,Wrd)` which displays the number of remaining guesses, prompts user to enter words then checks if:

- The word is right,
- Or the number of guesses is zero,
- Or the word has invalid length,
- Or the word is not in the knowledge base by using `allwords(L)` predicate which gets a List of all words in the knowledge base regardless what is the category.
- Or the word has correct length and in the knowledge base but not same as `Rand` which is the random word chosen by the game, it decrements the number of guesses, displays the

correct letters and correct positions corresponding to Rand
using `correct_letters(L1,L2,Corrlet)` and
`correct_positions(L2,L1,Corrpos)` .Also it converts the Rand and
word entered by user to a list of letters using `atom_chars(W,L)`.