CONCEPTS OF PROGRAMMING LANGUAGES COURSE (CSEN 403)

SPRING SEMESTER 2022

REPORT OF 1ST PROJECT

TEAM NUMBER 124

This Report Contains the description of the implemented predicates, in addition screenshots of different runs of the game, the two runs show the KB building phase. One shows a winning scenario in the game play phase, and the other shows a losing scenario in the game play phase

Kareem Ahmed Eladl, T-16, 52-5934

Mariam Mohamed Fawzy, T-16, 52-6892

Omar Mahmoud Awad, T-14, 52-9213

Youssef Osama Kamal, T-5, 52-1907

PREDICATES USED IN THE IMPLEMENTATION

- is_category(C) succeeds if C is an available category in the KB.
- categories(L) succeeds if L is a list containing all the available categories without duplicates. Note that: we've implemented it in our project using the setof
- available_length(L) succeeds if there are words in the KB with length L.
- pick_word(W,L,C) succeeds if W is a word in the KB with length L and category C.
- correct_letters(L1,L2,CL) succeeds if CL is a list containing the letters in both L1 and L2. Note that: we've implemented it in our project using the insertion method
- 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.
- build_kb: reponsible of the KB building phase.
- items_in_category_list(C,L,Length) succeeds if L is a list containing all the available words in category C with length Length.
- main: the main predicate that will be queried to initiate the KB building phase then the game play phase. Note that we implemented this predicate by simply calling the bulid_kb predicate and the play predicate.
- play: running this predicate will firstly result in displaying a list L containing the available categories for the user to choose one from, these available categories are based on the bulid_kb that the user himself/herself has made. Then this predicate will then call playHelperCategory(L)
- playHelperCategory(L) it takes as a parameter list L containing the available categories. It is responsible for making sure that the user choose a category only from the options displayed by checking that CategoryOfWord (the category that the user chose) is a member of the L. If not the program will display a warning massage for the user to notify him/her that his/her choice does not exist, then the program will let him/her choose again, we did this by making playHelperCategory(L) a recursive predicate, However, if the user choose a category form the available ones, the program will go to the next step which is calling playHelperLength(CategoryOfWord) predicate.
- playHelperLength(CategoryOfWord) displays a message asking the user to choose the length of the word, if the user enters LengthOfWord (length of a word) that does not match the length of any word in the category he/she chose, the program will keep asking him/her to choose the length again , we did that by making playHelperLength(CategoryOfWord) a recursive predicate , the check itself is done by the use of pick_word(_,LengthOfWord,CategoryOfWord) predicate, However, if the user enters LengthOfWord correctly which means the category chosen has a word of this length, the program will display a message notifying the user that the game has started and the number of guesses is equal to LengthOfWord +1 . The continuegame(LengthOfWord,CategoryOfWord) predicate is then called.
- continuegame(LengthOfWord,CategoryOfWord) which firstly make a list containing all the words that have a length = LengthOfWord and present in CategoryOfWord(the category that the user has chosen), then the program chooses a word from this list randomly using random_member pre-defined predicate, we will call this word WinningWord , then guessStage(WinningWord , NumberOfGuessesLeft ,LengthOfWord) predicate is called.

Kareem Ahmed Eladl, T-16, 52-5934
Omar Mahmoud Awad, T-14, 52-9213
Mariam Mohamed Fawzy, T-16, 52-6892
Youssef Osama Kamal, T-5, 52-1907

In our implementation please note that there is 2 guessStage predicates.

- guessStage(WinningWord , NumberOfGuessesLeft ,LengthOfWord) the program will execute this predicate if NumberOfGuessesLeft > 1 , the program will ask the user to guess a word composed of LengthOfWord letters , and then the program will read GuessedWord, kindly note that is predicate is a recursive one.

  1) If WinningWord matches GuessedWord, "You Won!" message will be displayed and the program will end.

  2) If length GuessedWord does not match LengthOfWord , the program will display a message notifying the user that the length of the word he entered does not match the length the he/she chose before the game starts, the program will re-ask him/her to re-enter the GuessedWord remainding the user of the number of trials left which is NumberOfGuessesLeft. And note here that NumberOfGuessesLeft will not decrease in case of entering word of different length.

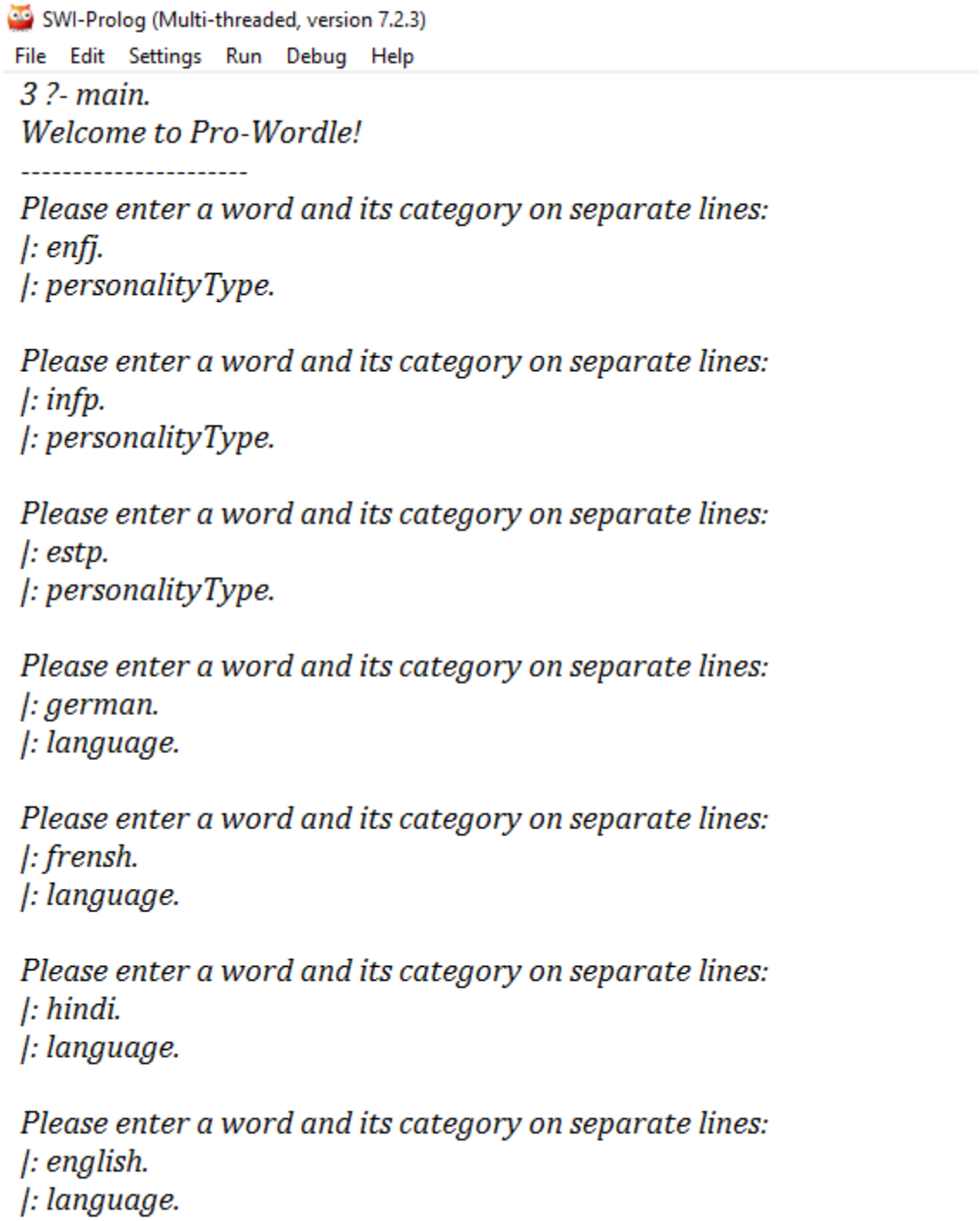
  3) If length of GuessedWord does match LengthOfWord but the GuessedWord itself does not match WinningWord , in this case the program will tell the user which letters of the GuessedWord is correct using correct_letters(GuessedWordList , WinningWordList , CorrectLettersList) predicate , it will also notify him/her which letters of the GuessedWord are in the correct position using correct_positions(GuessedWordList , WinningWordList , CorrectPositionList) , and lastly the program will re-ask the user to re-enter the GuessedWord remainding the user of the number of trials left which is NumberOfGuessesLeft . And note here that NumberOfGuessesLeft will decrease.

- guessStage(WinningWord , 1 , LengthOfWord) the program will execute this predicate if NumberOfGuessesLeft = 1 , the program will ask the user to guess a word composed of LengthOfWord letters , and then the program will read GuessedWord, kindly note that is predicate is NOT a recursive one. We made it that way simply because it is the user's last trial to guess the word.

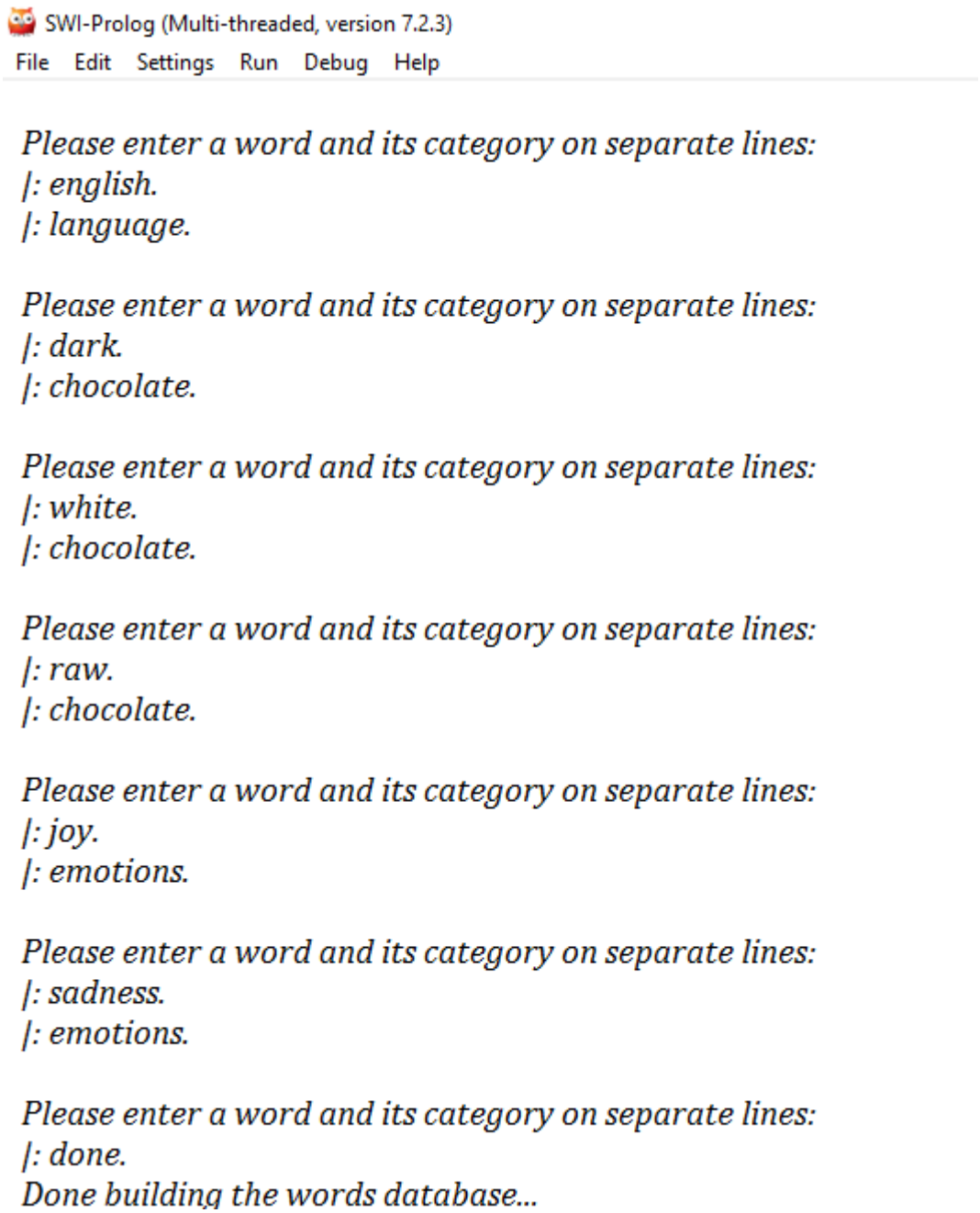
  1) If WinningWord does match GuessedWord, "You Won!" message will be displayed and the program will end.

  2) length GuessedWord does not match LengthOfWord , the program will display a message notifying the user that the length of the word he entered does not match the length the he/she chose before the game starts, the program will re-ask him/her to re-enter the GuessedWord remaining the user of the number of trials left which is NumberOfGuessesLeft which is 1 trial left. And note here that NumberOfGuessesLeft will not decrease in case of entering word of different length and become a zero.

  3) If WinningWord does NOT match GuessedWord, "You Lost!" message will be displayed and the program will end. Kindly note that this time neither a list containing correct letters CorrectLettersList using correct_letters(GuessedWordList , WinningWordList , CorrectLettersList) nor a list containing letters in correct positions CorrectPositionList using correct_positions(GuessedWordList , WinningWordList , CorrectPositionList) will be displays.



Kareem Ahmed Eladl, T-16, 52-5934
Omar Mahmoud Awad, T-14, 52-9213
Mariam Mohamed Fawzy, T-16, 52-6892
Youssef Osama Kamal, T-5, 52-1907


SWI-Prolog (Multi-threaded, version 7.2.3)
File Edit Settings Run Debug Help
Please enter a word and its category on separate lines:
|: english.
|: language.
Please enter a word and its category on separate lines:
|: dark.
|: chocolate.
Please enter a word and its category on separate lines:
|: white.
|: chocolate.
Please enter a word and its category on separate lines:
|: raw.
|: chocolate.
Please enter a word and its category on separate lines:
|: joy.
|: emotions.
Please enter a word and its category on separate lines:
|: sadness.
|: emotions.
Please enter a word and its category on separate lines:
|: done.
Done building the words database...

SWI-Prolog (Multi-threaded, version 7.2.3)

File Edit Settings Run Debug Help

```
The available categories are: [chocolate,emotions,language,personalityType]
Choose a category:
|: personalityType.
Choose a length:
|: 4.
Game started. You have 5 guesses.
Enter a word composed of 4 letters:
|: ntej.
Correct letters are: [n]
Correct letters in correct positions are: []
Remaining Guesses are 4
Enter a word composed of 4 letters:
|: ejnt.
Correct letters are: [n]
Correct letters in correct positions are: []
Remaining Guesses are 3
Enter a word composed of 4 letters:
|: ipnt.
Correct letters are: [i,n,p]
Correct letters in correct positions are: [i]
Remaining Guesses are 2
Enter a word composed of 4 letters:
|: ipnf.
Correct letters are: [i,n,f,p]
Correct letters in correct positions are: [i]
Remaining Guesses are 1
Enter a word composed of 4 letters:
|: ipnf.
You lost!
true .
```

Kareem Ahmed Eladl, T-16, 52-5934
Omar Mahmoud Awad, T-14, 52-9213
Mariam Mohamed Fawzy, T-16, 52-6892
Youssef Osama Kamal, T-5, 52-1907

SWI-Prolog (Multi-threaded, version 7.2.3)

File Edit Settings Run Debug Help

```
The available categories are: [chocolate,emotions,language,personalityType]
Choose a category:
|: science.

This category does not exist.
Choose a category:
|: personalityType.

Choose a length:
|: 5.
There are no words of this length.
Choose a length:
|: 4.
Game started. You have 5 guesses.
Enter a word composed of 4 letters:
|: pnit.
Correct letters are: [i,n,p]
Correct letters in correct positions are: [n]
Remaining Guesses are 4
Enter a word composed of 4 letters:
|: pnif.
Correct letters are: [i,n,f,p]
Correct letters in correct positions are: [n]
Remaining Guesses are 3
Enter a word composed of 4 letters:
|: infpp.
Word is not composed of 4 letters. Try again.
Remaining Guesses are 3
Enter a word composed of 4 letters:
|: infp.
You Won!
```

Kareem Ahmed Eladl, T-16, 52-5934
Omar Mahmoud Awad, T-14, 52-9213
Mariam Mohamed Fawzy, T-16, 52-6892
Youssef Osama Kamal, T-5, 52-1907