German University in Cairo
الجامعة الألمانية بالقاهرة

CSEN 403: Concepts of Programming Languages

10^{th} of May, 2022

Pro-Wordle

An Implementation of Josh Wardle's Wordle game in Prolog

Authors

Team #99

Omar Hesham, 52-8724, T10

Nardy Mechael, 52-8695, T08

Omar Tamer, 52-11870, T15

Mohammed Khaled, 52-21246, T05

Introduction

Pro-Wordle is an implementation of the Josh Wardle's word guessing game Wordle. Initially, the player is prompted to enter a word and a category to which that word belongs. The game then prompts the player to choose a category and the length of the word. The game will select a random word belonging to the selected category which is of the selected length. The objective then is to try to guess the word which the game has selected. The player has a number of guesses equal to the chosen length + 1 to guess the word before they lose.

The game provides some feedback after each guess made by the player to point the player in the right direction in guessing the word, namely:

- The game will display the correct letters in the player's guess (the letters that appear in the selected word)
- The game will display the letters which are in the correct position. That is, they appear in the same position in the selected word as the position in which they appeared in the player's guess.

If the player did not guess any letters correctly, the game displays two empty lists.

Implementation

This section describes the technical implementation of the game in Prolog. It gives a brief description of each predicate implemented and its intended purpose.

Note: predicate names are displayed in cyan.

The format predicate\arity has been avoided here by choice, except for predicates with 0 arguments or when a predicate is mentioned in the description of another. We chose instead to write the predicate's head as it is. We think this is an easier format to understand, and the arity of the predicate is inferred from the head.

main\0: The main driver predicate to play the game. It calls both build_kb\0 to build the knowledge base of the game and play\0 to then play the game.

build_kb\0: Builds the knowledge base by calling the helper predicate words_and_categories\0.

words_and_categories\0: Prompts the player to enter a word and its category on separate lines and adds them to the game's knowledge base as a predicate word\2. This continues until the player enters 'done.'

word(Word, Category): A predicate to associate each word with its category.

play\0: The predicate to play the game after having built the knowledge base. It displays the available categories to the player and prompts the player to choose a category and a word length. The game then selects a random word with the selected length from the selected category using pick_random_word\3 and proceeds to handle the logic of evaluating the player's guesses using guess\3.

pick_word(W, L, C): Succeeds if W is a word such that it belongs to category C and has length L.

pick_random_word(W, L, C): Creates a set of all valid words by using pick_word\3 and then selects a random word from this set.

play\0: Handles the logic of progressing the gameplay. It first shows the player the available categories using categories\1, prompts him to choose a category using choose_a_category\2 and a length using choose_a_length\2. The game then chooses a random word using pick_random_word\3 and starts to ask and process the player's guesses using guess\3.

guess(Word, N, Length): Handles processing the player's guesses. It asks the player for a guess and first checks if the player guessed the correct word, in which case the player would win. If not, then it checks then the validity of that guess (that it has the correct length and is a valid word to guess belonging to the knowledge base of the game). If the guess is invalid, the program prompts the player with the appropriate error message and asks the player to enter another guess (note that entering an invalid guess in this way does not deduct from the available guesses the player has).

If the player enters a valid guess, but is not the correct word, it then proceeds to call string_chars\2, correct_letters\2, and correct_positions\2 in order to provide the player with feedback on the guess.

string_chars(String, Chars): A bidirectional predefined predicate in prolog that converts a string to a list or vice-versa.

correct_letters(L1, L2, CL): Succeeds if CL is a list of the common elements between L1 and L2 (without duplicates).

correct_positions(L1, L2, CL): Succeeds if CL is a list of all elements which appear in the same position in both L1 and L2.

choose_a_category(Category, List): Prompts the player to choose a category and succeeds if Category is a member of List.

choose_a_length(Length, Category): Prompts the player to choose a length and succeeds if at least one word of that length exists in the given Category.

is_category(C): Succeeds if C is a category.

categories(L): Succeeds if L is a list of all available categories.

available_length(L): Succeeds if there is at least one word with length L.

clear_kb\0: Clears the knowledge base of the game by retracting all word\2 predicates.

Demonstration

The following section demonstrates two runs of the game: one where the player wins and one where the player loses. In addition, it demonstrates some errors in input by the user and the game's response to them in these two runs.

Please note that these runs do not demonstrate all possible errors in input in order to keep them at a reasonable length for this report.

Run #1 (Player Wins):

```
?- main.
Welcome to Pro-Wordle!
----------------------

Please enter a word and its category on separate lines:
|: train.
|: transport.
Please enter a word and its category on separate lines:
|: car.
|: transport.
Please enter a word and its category on separate lines:
|: plane.
|: transport.
Please enter a word and its category on separate lines:
|: pyramids.
|: landmark.
Please enter a word and its category on separate lines:
|: eiffel.
|: landmark.
Please enter a word and its category on separate lines:
|: brandenburg.
|: landmark.
Please enter a word and its category on separate lines:
|: done.

Done building the words database...

The available categories are: [landmark,transport]
choose a category:
|: transport.
Choose a length:
|: 5.

Game started. You have 6 guesses.

Enter a word composed of 5 letters:
|: car.
Word is not composed of 5  letters. Try again.
Remaining Guesses are 6

Enter a word composed of 5 letters:
|: crane.
Word is not in the Knowledge Base! Try again.
Remaining Guesses are 6

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

Enter a word composed of 5 letters:
|: plane.
You Won!
Note: to clear kb write ?- clear_kb.
true .
```

Run #2 (Player Loses):

```
?- main.
Welcome to Pro-Wordle!
----------------------

Please enter a word and its category on separate lines:
|: abbey.
|: singer.
Please enter a word and its category on separate lines:
|: appel.
|: singer.
Please enter a word and its category on separate lines:
|: blake.
|: singer.
Please enter a word and its category on separate lines:
|: barry.
|: singer.
Please enter a word and its category on separate lines:
|: boehm.
|: singer.
Please enter a word and its category on separate lines:
|: bosch.
|: singer.
Please enter a word and its category on separate lines:
|: leger.
|: actor.
Please enter a word and its category on separate lines:
|: lewis.
|: actor.
Please enter a word and its category on separate lines:
|: monet.
|: actor.
Please enter a word and its category on separate lines:
|: nolan.
|: director.
Please enter a word and its category on separate lines:
|: poole.
|: director.
Please enter a word and its category on separate lines:
|: puvis.
|: director.
Please enter a word and its category on separate lines:
|: done.

Done building the words database...

The available categories are: [actor,director,landmark,singer,transport]
choose a category:
|: singer.
Choose a length:
|: five.
The length you entered is not a number.
Choose a length:
|: 5.

Game started. You have 6 guesses.

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

Enter a word composed of 5 letters:
|: barry.
Correct letters are: [b]
Correct letters in correct positions are [b]
Remaining Guesses are 4

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

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

Enter a word composed of 5 letters:
|: abbey.
Correct letters are: [b]
Correct letters in correct positions are []
Remaining Guesses are 1

Enter a word composed of 5 letters:
|: puvis.
You lost!
Note: to clear kb write ?- clear_kb.
true .
```