

GUESS GAME

Summary of the desired functionalities

This elm program contains the typical ELM architecture:

Model: Data representation of the program's state, including the list of descriptions, victory state, loading state, error state, guess state, the word to be guessed, a list of words, display states for the guess and show guess.

Init: Initializes the Model by setting the loading state to True and sending a command to get words from the API.

Update: Handles updates to the Model based on the type of message received. For example, if a Convert message is received with a guess, the function checks if the guess matches the word to be guessed. If it does, the victory state is updated to True.

Subscriptions: None.

View: Renders the Model by showing either the loading state, error state, victory state, or the guess interface.

We will try to explain each field and function.

descriptions is a list of word descriptions.

victory is a boolean indicating whether the user guessed the word correctly or not.

loading is a boolean indicating whether the program is currently loading the data.

error is a string indicating if there is any error

guess_ is the guess of the user.

toBeGuessed is the word that the user has to guess.

wordsList is an array of strings which contains a list of words.

showGuess is a boolean indicating whether to display the answer to the user.

displayGuess is a boolean indicating whether to display the word to be guessed.

The program responds by sending messages of type Msg. In the update function, it is taken the current Model and it is returned an updated Model and a command. These are the types of messages we can find in the Update function:

Empty_model is a Model type variable which puts the default values to each field stored in Model.

Convert: updates the guess_ field in the model. If the guess is true, it sets the victory variable to True.

Print: updates the displayGuess field in the model.

HaveWords: updates the wordsList field in the model with the words from the words.txt file.

GenWord: updates the tobeGuessed field in the model with a new word to be guessed.

Json: updates the descriptions field in the model with the descriptions of the word to be guessed.

getting_words is a function that retrieves a list of words from a file using an HTTP GET request.

getting_descriptions is a function that takes a specific word as input and returns a description for it by making an HTTP GET request to an API.

takeString is a function that takes an input that might be a string and outputs it as a string.

takeWord is a function that takes a single word input that might be in a specific format and outputs it in that format.

descriptions_html takes a list of descriptions and transforms them into HTML elements that display each description.

random_function generates a random word from an input array of words and retrieves its description from the API.

errorToString converts an error that occurs during an HTTP request into a string message to better understand the error.