Understanding and extending the flashcard application

MUHAMMAD AZFAR YAQUB

MYAQUB@UNIBZ.IT

Flashcard

- In this lab you will start with the flash card application available here.
- In its current state, the flash card application supports:
  - selecting a deck of cards
  - showing the cards in a deck, one by one
  - recording whether a guess was right or wrong
  - re-ordering the cards in a deck, putting the wrong cards first, then the unseen cards, then the right cards
  - making sure the order of cards persists
- It does so using the following components
  - A SelectDeck component to pick a deck among the available one, that has the overall state of the application
  - A Deck component that keeps track of the guesses, and can re-order cards according to guesses
  - A Card component that keeps track of whether the front or the back of the card is shown
  - A Front component showing the front of the card, allowing to flip it
  - A Back component showing the back of the card, allowing to record a guess as right or wrong

The goal of this lab is to extend the flash card application in a variety of ways.

Code understanding task

- The first task is to understand how the application works and how it is structured. To do so:
  - list all of the components that the application defines, along with their props, state, and the types of props and states
  - on another sheet of paper, display the components in a tree that matches the way they are organized in the UI
- Show how control flow affects the application. Annotate on the tree (possibly with different colors):
  - how callbacks are passed down from parents to child's components
  - how these callbacks change the state of the applications
  - how changes of state change the tree of the application

Starter task

- Show some statistics while reviewing the cards in a deck
- Show the number of cards reviewed, the number of cards left, and the number of correct and incorrect guesses
- These statistics should be defined in an independent component, that can be shown or hidden by the user
- Another possible starter task is to better style the components so that the card looks more like an actual card

Editing a deck

- Define a "deck editor" component, that allovws the following operations:
  - allow for the creation of new cards in a deck
  - allow for the deletion of existing cards
  - allow for the edition of an existing card
  - allow for the renaming of a deck
- The main application should allow to select a deck either for reviewing flashcards, or to edit the deck
  - Any change to the data structure should persist if you leave the screen
  - Example, if you switch from reviewing one card deck to reviewer another one
- Hint
  - give an id to components to allow the name of a deck to be edited
  - Note that at this stage, we don't know enough (yet) to make changes to the data persist if you leave or relaunch the application.

Going further

- allow for the deck to be saved automatically, no need to press "restart" for this
- order the cards in a smarter fashion, by taking into account the time taken to flip the card
- allow for the addition of a new deck of cards
- allow for the deletion of a deck of cards
- improve the style the application, for instance by adding spacing between elements, using TouchableOpacity instead of buttons, adding a white background to flashcards front and back
- simplify the code by using a single callback, with an argument, to record guesses, instead of using two callbacks

Partial solution

- A version of the flashcard application with the deck editor can be found here