Control flow

DEADLINE: 09/07/2019

FOLDER STRUCTURE

```
FL11_HW7/*
  └ homework/*
    └ src/
      └ js/
        └ task1.js*
        └ task2.js*
      └ .eslintrc.js*
      └ task1.html*
      └ task2.html*
```

* - required

TASK

Task #1. Check the email and change the password

Write the code which verify user rights.

Step 1. Check login

- Ask user for an email // use prompt()
- If the input is an empty line or Esc – show “Canceled.” // for showing - use alert()
- If the input length less than 6 symbols - show “I don't know any emails having name length less than 6 symbols”.
- If the visitor enters "user@gmail.com" or "admin@gmail.com" then prompt for a password.
- If it’s another string – then show “I don’t know you”.

Step 2. Check password:

- For an empty string or cancelled input, show “Canceled.”
- For email "user@gmail.com" correct password is “UserPass”, for "admin@gmail.com" correct password is “AdminPass”. In other case, show “Wrong password”.

Step 3. Change the password:

- 1) Suggest user/admin to change his password – “Do you want to change your password?”. //use confirm()
- In case the user clicks the 'Cancel' button, the message “You have failed the change.” //use alert()
- 2) If user clicked ‘Ok’ – ask to write the old password (use prompt()) and validate it as at Step 2.

- 3) If the visitor enters correct old password for current email then prompt for a new password.
- If the input length less than 5 – show "It's too short password. Sorry."
- 4) If the new password is valid ask to enter it again.//use prompt()
- If the inputted value doesn't match the new password from 3) – show "You wrote the wrong password."
- If user write the same new – show "You have successfully changed your password." //use alert()

Task #2. Guessing game

Your task is to write a simple simulator of casino roulette.

Requirements:
Step 1:

- Create a prompt window (use confirm()). Show the message inside the window 'Do you want to play a game?'.
- In case the user clicks the 'Cancel' button, the message 'You did not become a billionaire, but can.' should be shown (use alert).

Step 2:

- If user clicked 'Ok' – start a game: randomly (use Math.random()) choose an integer number in range [0; 8] (including 0 and 8) and ask user to enter a number of pocket on which the ball could land (use prompt()).
- User has 3 attempts to guess a number.
- If user guessed the number on which ball landed, on 1-st attempt prize is 100$ (maximum prize for current numbers range), 2-nd attempt – 50$, 3-rd attempt – 25$.
- If user did not guess a number show the message 'Thank you for your participation. Your prize is: … $' (Use alert) and ask if he wants to play again (use confirm).

Step 3:

- If user did guess - Show the message 'Congratulation, you won! Your prize is: … $. Do you want to continue?'.
- If user does not want to continue – show the message 'Thank you for your participation. Your prize is: … $' (Use alert) and ask if he wants to play again (use confirm).
- If user does want to continue, make number range bigger at 4 as the previous one (for example [0; 8] -> [0; 12]), and two times bigger maximum prize (for example on 1-st attempt prize will be 200$, 2-nd attempt – 100$, 3-rd attempt – 50$). Prize must be added to the previous one and number of attempts should be set to 3 (user should have 3 attempts to guess a number for each numbers range)
- Each time you ask user to enter a number you should show him a range of cells, how much attempts he has left, his total prize and possible prize on current attempt. See Figure 1:

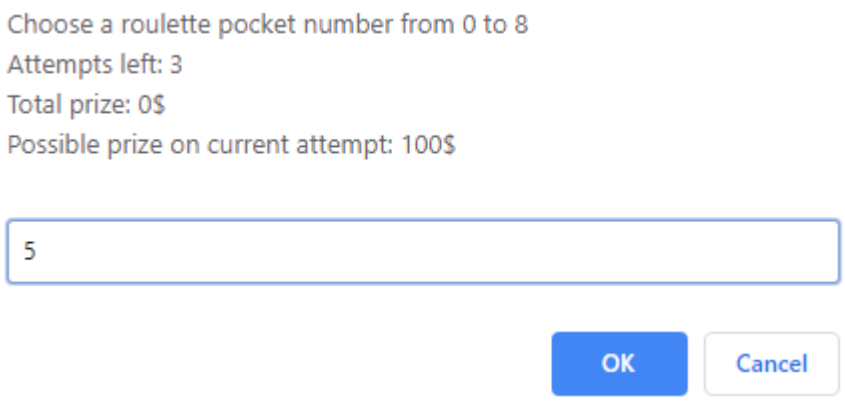

Figure 1 – The prompt window

— All these stuffs should be repeated until user lose or decide to quit

RESTRICTIONS

- Using JS functions is forbidden.
- Adding **task/** folder is forbidden. Do not push it to repository.

HOW TO

- To run linter for JavaScript (JSX) files use `npm run jslint`.

BEFORE SUBMIT

— Remove all unnecessary files that you might have included by mistake
— Verify that all functionality is implemented according to requirements
— Add comments if the code is difficult to understand
— Fix warnings/errors in the browser console
— Verify that the name of the folders and files meet the requirements
— Make sure there are no errors/warnings in the browser console
— Run the linter and fix all warnings and errors.

SUBMIT

- The folder should be uploaded to github repository 'FL-11' into master branch