CS314 HW 4

Spring 2019

Hangman

Write an implementation of the game "hangman" in Haskell, where the user must guess a word. Use the provided `words.txt` file.

- Pick a random word from the `words.txt` file
- Display underscores for each unguessed letter
- Read a character from the user and replace the underscores with that letter where it matches the actual word (if any)
- Continue reading guesses from the user
- When the user gets all the letters correct, exit

To keep things simpler, you don't have to keep track of the number of guesses or letters already guessed, so the user will always win eventually. You can assume the input format will be correct.

For example, here the word randomly picked was "strongest". The first, third, etc. lines are the current state of the word being guessed. The second, fourth, etc. lines are the user inputting a letter to try:

```
_________
e
______e__
s
s_____es_
h
s_____es_
t
st____est
r
str___est
a
str___est
o
stro__est
n
stron_est
g
strongest
```

Some possibly useful functions:

- `getLine :: IO String`
- `lines :: String -> [String]`
- `randomRIO :: Random a => (a, a) -> IO a`
  (in System.Random)
- `readFile :: FilePath -> IO String`
  (`FilePath` is another name for `String`)
- `words :: String -> [String]`

Note that in recent versions of ghc, `import System.Random` will give an error, since random is no longer included by default. It should work on ilab, though, since that uses an older ghc.

If you want to develop using a more recent version, you can look into how to set up a project using Stack (https://docs.haskellstack.org/en/stable/README/) and then add “`- random >= 1.1`” under `dependencies` in `package.yaml`. Note that your submission will be run on ilab without stack, though.

Submission

Please submit a single file named `hw4.hs` on Sakai.

You should have a `main` function, so that your hangman game can be executed using `runhaskell hw4.hs`.