hsordle

wordle in haskell

Описание

Автор: Александър Гуров, КН2, Група 6, 3MI0800270

Условието може да намерите тук

Github repo на този проект (тук правилно се изобразяват емоджита за разлика от pdf-а)

Популярната игра Wordle, реализирана в Haskell, за проект по ФП във ФМИ. Основната идея за решението е обхождане на списъци и ползването на IO функции за вход и изход в конзолата и произволни числа.

Структура на проекта

Използван е стандартния `cabal init`, като в директорията `App` има следните файлове:

- `Main-linux.hs` и `Main-windows.hs` - тъй като използвам емоджита, windows се нуждае от специален код за терминала, за да може да ги изобразява. Използвам библиотеката Win32, но тя се компилира само на windows, за това има два отделни main файла, които се избират от cabal спрямо операционната система, както е описано в `wordle.cabal`.
- `Play.hs` - в него се съдържа `play` функцията, която се изпълнява в началото на играта, в нея се набавя информация от потребителя чрез конзолата за дължината на думата, режима и нивото на играта. Също така чете и файла с думи `words.txt`, който е взет от следното github repo. Спрямо тази информация се извиква съответната функция от съответните модули. Тук е и константата `maxTurns`, в случая равна на 6.
- `Utils.hs` съдържа функции и дата типове, които се изпозват няколко файла. `Mode` - enum за ниво, `Game` - enum за режим, `Color` - enum за цветовете (сиво(макар че емоджито е бяло), жълто, зелено), с няколко функции за преобразуване от и към `Color`. `readLenFromConsole, chooseGame, chooseMode` са IO функции, които взимат данни от играча чрез входа на конзолата. `createResult` създава списък от `Color`, който отговаря по правилата на Wordle спрямо двете подадени думи - опит и днешната дума. `getRandomWord` използва System.Random за да даде произволна дума от списък с думи.
- `Wordle.hs` - тук се съдържа кода за режим Wordle. `playTurn, playTurnEasy, playTurnHard` са рекурсивни IO функции за съответните нива на играта, приемат думата, която е произволно избрана от списъка с думи във функцията `play`, дължината на думата, сегашния ход, максималния брой ходове и други помощни аргументи спрямо нивото - в `playTurnEasy` имаме списъци с букви, които са срещнати в съответния цвят и списък със думите от речника, за да могат да се използват в функциите `grayHints, yellowHints, greenHints, dictHints`, които се използват за предупреждаване на играча. В `playTurnHard` има булева променлива `haveLied` и `Data.Map (Int,Char) Color` от библиотеката Containers, който съдържа цветовете на срещани букви на специфични позиции до момента на "лъженето", за да може да няма противоречия с предишните резултати, като се подава на функцията `lyingResult`, която създава "лъжлив резултат". `lie` функцията връща цвят, който е различен от правилния за

буквата. `playTurn` е функция за нормалното ниво на Wordle и няма помощни аргументи. Функциите проверяват за валиден вход, изпълняват допълнителните си действия (hints, lying) и извикват себе си или приключват с `putStrLn`.

- `Helper.hs` - този модул съдържа рекурсивната IO функция `playHelper`, която приема думата, която алгоритъмът трябва да познае, предишния си опит, списъка с възможни думи, списък с буквите, които знае, че не са думата,`Data.Map (Int,Char) Color` за наредените двойки (позиция, буква), които са "жълти" и "зелени", дължината на думата и колко хода са останали. Проверява се входа дали отговаря на правилата на Wordle, дали съществуват думи, отговарящи на входа, след което в `getFilteredWords` се филтрира списъка с думи спрямо входа, като опитът на алгоритъма се избира като `getMostEliminatingWord` взима произволна дума от списък с "най-елиминиращи думи", създаден от `elimFilter`, който намира думите, чието множество от букви има най-много непразни сечения с множествата от букви на другите думи от филтрирания списък от думи. Тази функция има сложност `O(n^2)` и на първия ход може да се забави. Опитът се сравнява с думата на деня и или се извиква отново функцията с обновените помощни аргументи, или играта приключва с `putStrLn` заради край на играта. Също така се съдържа и функцията `isPresentAt`, която приема аргументи буква, индекс и дума и връща `Maybe String`, защото се ползва при думи с неизвестна дължина. Използвам функцията `?^` от библиотеката Control.Lens.

Употреба

В терминал се изпълнява `cabal run`

```
g8row-debian :: Documents/haskell/hsordle ‹master*› % cabal run
```

Първо играта чака вход - число, което да бъде дължината на думата на деня. Има валидация на входа - числото трябва да е между 1 и 31.

```
>> choose length of word:
5
```

След това трябва да се избере режим и ако е нужно ниво.

```
>> choose a mode
 - 1. Wordle
 - 2. Helper
>> enter a number (1-2):
1
>> choose a mode
 - 1. easy
 - 2. normal
 - 3. hard
>> enter a number (1-3):
1
```

За **режим Wordle**, ходът протича така - изисква се вход, дума с правилната дължина, след което се връща списък от емоджита (и при лесно ниво подсказки), след което се повтаря или играта приключва.

```
>> you are on turn 6, enter your guess:
vibes


□▩□□□
v i b e s
>> you are on turn 5, enter your guess:
night


□▩□□□
n i g h t
>> you are on turn 4, enter your guess:
finge

▥ this word is not in the dictionary
□ these letters are known to be useless: e,n,g

□▩□□□
f i n g e
>> you are on turn 3, enter your guess:
midek

▥ this word is not in the dictionary
□ these letters are known to be useless: e

□▩□□□
m i d e k
>> you are on turn 2, enter your guess:
pizda

▥ this word is not in the dictionary
□ these letters are known to be useless: d

□▩□□▩
p i z d a
>> you are on turn 1, enter your guess:
lumpa

▥ this word is not in the dictionary
▧ you already guessed the letter at index 1
▨ these letters are known to appear: i
□ these letters are known to be useless: m,p

▩□□□▩
l u m p a
licca was the word, u lose :/
```

За **режим Helper** се изписва правилната дума и опита на алгоритъма. Изисква се вход от потребителя, който е съставен от букви отговарящи на цветовете - w: gray/white, y: yellow, g: green (за съжаление gray и green започват с една и съща буква). Това ще бъде поредицата от цветове, която алгоритъма ще интерпретира.

```
answer: necia
guess:  chaja
>> you have 6 turns left, input a list of colors according to the guess
(w,y,g)
ywwwg
answer: necia
guess:  vicia
>> you have 5 turns left, input a list of colors according to the guess
(w,y,g)
wwggg
answer: necia
guess:  oncia
>> you have 4 turns left, input a list of colors according to the guess
(w,y,g)
wyggg
>> we won woooo !!!
```