

Проект по Функционално програмиране

На

Габриел Стойков

КН2, 6 група

ОМ10800304

# Тема: Wordle

Съдържание:

- 0. Въведение
- 1. Помощни функции
- 2. Работа на режимите
  - 2.1 Нормален
  - 2.2 Лесен
  - 2.3 Експертен
  - 2.4 Помощник
- 3. Main функция

## 0. Въведение

За взимане на произволна дума са използвани функции от стандартната библиотека – например функцията `RandomRIO` от `System.Random` (лично при мен трябваше да я изтегля чрез `stack` или `cabal`). На места може би съм копирач ред-два, докато се учих как се работи с `do`-конструкции, монади и `IO` операции, но логиката и алгоритмите са измислени от мен. Речникът, който играта ползва, е текстов документ с име `“words.txt”`.

# 1.Помощни функции

Тук има кратко описание на функциите, чрез които са реализирани режимите на игра:

## **getRandomWordWithLength :: Int -> IO String**

Функцията приема дадена дължина на думата и връща произволна дума от речника ("words.txt") със зададената дължина. Използва се за начало на циклите на игрите.

## **getRandomWord :: [String] -> IO String**

Подава се като параметър списък от думи и връща произволна дума от списъка. Използва се в режима "Експерт", за да може програмата да излъже с цветовете.

## **checkGuess :: String -> String -> String**

Приема две думи като аргументи и връща списък от цветовете на думите, ако думите са различни, или връща String, който съответства на съобщение, че тайната дума е позната.

## **zipWithColor :: String -> String -> String -> String**

Приема дума, която съответства на опита на играча и два пъти отговора, и връща списък от цветовете на думата опит.

Например

```
ghci> zipWithColor "clown" "color" "clown"  
"[green] [yellow] [yellow] [yellow] [grey] "
```

Тайната дума е clown, а опитът е color.

### **zipWithColorSingle :: String -> String -> String -> String**

Подобна на миналата, но тук се връща само една буква от думата на цвета, използва се в режим помощник, за да може играчът да даде лесна обратна връзка на програмата.

```
ghci> zipWithColorSingle "clown" "color" "clown"  
"gyyyb"
```

g – green, y – yellow, b – grey(black)

### **getYellowLetter :: String -> String -> String -> String**

Връща низ, съдържащ жълтите букви от опита спрямо тайната дума. Тайната дума се приема два пъти, защото единият се ползва за итерация, а другият, за да помним цялата информация от думата.

```
ghci> getYellowLetter "color" "clown" "clown"  
"olo"
```

### **getGreyLetters :: String -> String -> String -> String**

Връща низ, съдържат сивите букви от опита спрямо тайната дума. Тук се приема тайната дума само веднъж, но подаваме и списък от букви, за които вече знаем, че са сиви. Това е доста удобно за лесния режим, защото можем да помним сивите букви от минала итерация.

```
ghci> getGreyLetters "color" [] "clown"  
"r"
```

**wordContainsAlreadyKnownGreyLetter :: String -> String -> Bool**

Подават се дума и низ от сиви букви и се връща дали думата съдържа поне една от тях.

**wordContainsAllGreen :: String -> String -> Bool**

Подават се дума и низ от зелени букви и се връща дали думата съдържа всички от тях. Низът от зелени букви изглежда така:

с\_l\_\_, където ако на дадена позиция имаме буква, то се знае, че тя е правилната буква от тайната дума, а ако е '\_', то не знаем буквата на тази позиция.

**getGreenLetters :: String -> String -> String**

Подават се две думи – опит и тайна дума и се връща низ, подобен на този от **wordContainsAllGreen**.

```
ghci> getGreenLetters "glow" "blow"  
"_low"
```

**getWeight :: String -> String -> Int**

Приема две думи – опит и тайна дума и връща оценка колко близко опитът е до тайната дума - + 1 за всяка жълта и +3 за всяка зелена.

**getWeightWords :: String -> [String] -> Int**

Приема дума и списък от думи и връща оценка колко общо има думата със списъка, като за всяка думи от списъка “map” - ва оценка и накрая събира всички тези оценки.

**optimalWords :: [String] -> String**

От списък от думи връща тази, която има най-висока оценка, изчислена с **getWeight** и **getWeightWords**.

**removeBadWords :: String -> String -> [String] -> [String]**

Приема тайна дума и опит и списък от думи – намира сивите и зелени букви на опита и според тях филтрира списъка от думи, като маха всички сиви букви и за дадена зелена буква оставя само думите, които има тази буква на същата позиция.

## **2.Работа на режимите**

### **2.1 Нормален режим**

Това е най-простият измежду режимите. Програмата намисля дума с предварително зададена дължина от речника и чака опит от играча. Ако опитът е дума с дължина различна от тайната, се извежда съобщение за това и опит на играча не се взима. На всеки ход програма изписва “цветовете” на буквите в опита.

```
Attempts left: 6
Enter your guess:
horse
[grey] [grey] [green] [yellow] [yellow]

Attempts left: 5
Enter your guess:
glass
[grey] [grey] [grey] [yellow] [green]

Attempts left: 4
Enter your guess:
brown
[grey] [yellow] [grey] [grey] [grey]

Attempts left: 3
Enter your guess:
hoogle
wrong input

Attempts left: 3
Enter your guess:
glass
[grey] [grey] [grey] [yellow] [green]

Attempts left: 2
Enter your guess:
house
[grey] [grey] [grey] [yellow] [yellow]

Attempts left: 1
Enter your guess:
great
[grey] [yellow] [yellow] [grey] [yellow]

The word was: terms
```

## 2.2 Лесен режим

Тук играчът получава помощ от програмата и съобщения за “лош” опит. Лош опит в смисъл, ако играчът е играл дума, за която е изпълнено едно или повече от следните:

- В думата има вече известни сиви букви
- В думата не присъстват всички зелени букви на съответните си позиции
- Думата е била играна на минал ход
- Думата не е в речника

Attempts left: 6  
Enter your guess:  
bring  
[grey] [grey] [green] [grey] [grey]

Attempts left: 5  
Enter your guess:  
plant  
Word contains letters that are already known not to be in the answer

Attempts left: 5  
Enter your guess:  
doing  
Word contains letters that are already known not to be in the answer

Attempts left: 5  
Enter your guess:  
/info  
Tried words: ["bring"]  
Grey Letters: bgnr  
Green Letters: \_\_i\_\_

Attempts left: 5  
Enter your guess:  
cried  
Word contains letters that are already known not to be in the answer

Attempts left: 5  
Enter your guess:  
child  
[yellow] [green] [green] [grey] [grey]



```
Attempts left: 4
Enter your guess:
/info
Tried words: ["child","bring"]
Grey Letters: bdglnr
Green Letters: _hi__

Attempts left: 4
Enter your guess:
third
Word contains letters that are already known not to be in the answer

Attempts left: 4
Enter your guess:
ships
[grey] [green] [green] [grey] [grey]

Attempts left: 3
Enter your guess:
thick
[grey] [green] [green] [green] [grey]

Attempts left: 2
Enter your guess:
which
Congratulations! You guessed the word.

The word was: which
```

На всеки ход играчът може да въведе вместо дума командата “/info” и програмата ще изведе съобщение за думите, опитани до момента, сивите букви и зелените букви.

```
Attempts left: 4
Enter your guess:
/info
Tried words: ["child","bring"]
Grey Letters: bdglnr
Green Letters: _hi__
```

## 2.3 Експертен режим

Този режим е почти същият като нормалният, но тук допускаме, че програмата може веднъж да даде лъжлив отговор, като този шанс е  $1/3$  на всеки ход и може да се случи само веднъж по време на играта. Този лъжлив отговор не си противечи с миналите отговори, защото програмата избира да сравни опита на играча не с реалната тайна дума, а с друга дума, но такава, че тя не съдържа вече знаещи се сиви букви и съдържа всички знаещи се зелени букви.

```
[grey] [grey] [yellow] [grey] [grey]
Attempts left: 4
Enter your guess:
clear
[grey] [grey] [yellow] [grey] [grey]
Attempts left: 3
Enter your guess:
clear
[grey] [grey] [yellow] [grey] [grey]
Attempts left: 2
Enter your guess:
clear
[grey] [grey] [yellow] [grey] [grey]
Attempts left: 1
Enter your guess:
clear
[grey] [grey] [green] [grey] [grey]
Attempts left: 0
Enter your guess:
clear
[grey] [grey] [yellow] [grey] [grey]
```



## 2.4 Помощен режим

Тук играчът въвежда тайната дума и програмата трябва да я познае. Има два варианта – програмата сама да си избира дума и после да я провери с тайната или на всеки ход чака от играча низ, съответстващ на цветовете на опита, ако низът е грешен, се извежда съобщение за това и от играчът се очаква нов низ.

На всеки ход програмата се опитва да играе такава дума, че да се отстранят максимално много думи( в общия случай – т.е не знаем тайната дума) Това става със следния алгоритъм:

От всички възможни думи ( на първия ход това са всички думи в речника) се причислява някаква оценка на всяка дума – това става с ф-ята **getWeightWords :: String -> [String] -> Int**. Взима се думата с най-висока оценка( това ще е дума, която има максимално общи букви с останалите възможни думи. Тази дума се играе – ако тя е вярната – то програмата е познала думата – иначе ще се види колко сиви и зелени букви и ще се филтрират думите от възможните заедно със самата въведена дума( защото ако има само жълти букви, може да стане безкраен цикъл) и алгоритъмът ще се завърти отново.

- Без вход от играча:

```
secret word:
might
Automatic play? (y/n)
y
The word "tones" was played
[yellow] [grey] [grey] [grey] [grey]

The word "faith" was played
[grey] [grey] [yellow] [yellow] [yellow]

The word "right" was played
[grey] [green] [green] [green] [green]

The word "light" was played
[grey] [green] [green] [green] [green]

The word might is the correct word
```

- С ВХОД :

```
secret word:
clear
Automatic play? (y/n)
n

The word "tones" was played
Type Colors
bbbyb

The word "place" was played
Type Colors
bgyyy

The word "black" was played
Type Colors
bgyyb

The word "claim" was played
Type Colors
ggybb

The word clear is the correct word
```

## 4.Main функция

Програмата се стартира от нейната main ф-я. Избира се режим и дължина на думата.

```
ghci> main
Welcome to Wordle!
Choose game mode
1. Normal - No additional help
2. Assisted - additional help
3. Expert - The program can lie once
4. Helper - choose the secret word and let the program guess it
4
Word length:
5
secret word:
black
Automatic play? (y/n)
y
The word "tones" was played
[grey] [grey] [grey] [grey] [grey]

The word "child" was played
[yellow] [grey] [grey] [yellow] [grey]

The word black is the correct word
```