



ДОКУМЕНТАЦИЯ НА ПРОЕКТ ПО ФП

WORDLE

Изготвен от Дара Коева ФН: 45756

Идея : За да бъде реализирана играта е необходимо да се ползва речник, откъде да се генерира тайната дума. За целта съм направила текстов файл (dictionary.txt), който съдържа думи на английски език с дължина в интервала [4;10] и за всяка дължина има по 20 думи. За началното на играта ще е направен потребителски тип GameMode (подобно на enum) с цел да се пази, какъв режим на игра е избрал играчът. В началото също се иска потребителя да въведе и желата от него дължина, с която после се вика съответния режим на игра.

- GameMode == Game : е необходима функция, която спрямо въведеното предложение за дума от играча връща съответните цветове. Както и такава, в която да се започне играта като се генерира произволна дума от речника с въведената от играча дължина и след което се извиква основната функция за съответната трудност.
 - STANDARD GAME – Тук единствено ни трябва една функция, която постоянно кара потребителя да въвежда своето предложение относно тайната дума, проверяме дали я е познал, ако Да – край на играта, иначе връщаме списъка от цветове отговарящ на съответния му вход и отново въвежда дума, докато познае или се откаже.
 - EASY GAME – тук вече е необходимо да се направи потребителски тип GameState , в който да се пази текущото състояние на играта. Жълтите и сивите букви ще се пазят в стрингове, а за зелените ще е необходим асоциативен списък, в който ключа ще е позицията, а стойността ще е самата буква. След като вече имаме генерираната дума и стартовата функция, откъдето ще извикаме основната функция за easy game. Основната функция постоянно ще кара потребителя да въвежда своето предложение относно тайната дума, проверяме дали я е познал, ако Да – край на играта, ако НЕ – му отговаряме дали тази дума е част от речника и информация относно използваните букви в нея :
 - 1) Дали в сегашното му предположение има сива буква, за която вече знае.
 - 2) Дали в сегашното му предположение липсва жълта буква, за която вече знае.

3) Дали всички зелени букви са на правилната позиция.

След което обновяваме GameState като добавяме новите сиви, жълти и зелени букви, функцията започва отново и така докато играчът познае думата или не пожелае да се откаже.

- EXPERT GAME – в стартовата функция освен генерирането на произволна дума, тук е необходимо и да се избере произволно при кой опит на играча да излъже относно цветовете. Това става чрез избиране на произволно число в интервала [1;3], по – малък интервал, за да има време играта да излъже дори и при по – кратките думи. В един списък се пазят всички използвани букви от играча до съответният момент. Тук ни е нужна и функция, която да връща грешни цветове, когато играта е стигнала съответният опит на играча, избран за лъжлив. Лъжливият масив от цветове се генерира по следния начин:

- 1) Гледам, кои букви принадлежат на тайната дума от сегашното предположение и за тези, които хем досега не са били въвеждани и хем са част от тайната дума ще излъжа играча, че са сиви.
- 2) Гледам, кои букви принадлежат на тайната дума от сегашното предположение и за тези, които хем досега не са били въвеждани и хем НЕ са част от тайната дума ще излъжа играча, че са жълти или зелени.

Основната функция на Expert game ще кара потребителя да въвежда своето предложение относно тайната дума, проверяме дали я е познал, ако Да – край на играта, ако НЕ – и не е настъпил моментът за лъжене връщаме правилните цветове и обновяваме въведените до този момент от играча букви. Ако текущият опит е равен на този за лъжа връщаме грешните цветове, но на следващите пъти след лъжата връщаме правилните цветове. И така докато играчът не познае тайната дума или не се откаже.

- GameMode == Helper – тук отново ще ползваме потребителския тип GameState от easy game, за да пазим текущото състояние на играта относно вече дадените цветове от играча за съответните букви. В стартовата функция ще ни е необходимо да вземем всички думи от речника с избраната от потребителя дължина. Първото а и всяко друго предложение за дума от програмата се реализира по следния начин:

- На база на текущият речник се избира дума, която има най – много поне една обща буква с останалите думи в текущият речник, така тя ще елиминира възможно най – много думи. За целта е необходим асоциативен списък, в който ключа ще е позицията на думата в текущият речник, а стойността – броят думи, с които има поне една обща буква (тези които е възможно да премахне).

В основната функция на всеки ход ще предлагаме дума, която е максималната, след което проверяваме дали сме познали думата, ако ДА - край на играта, ако НЕ проверяваме дали въведените цветове са коректни, тоест дали има такава дума в речника и дали не си противоречи т.е. сега да казва, че някоя буква е сива, е предният път е била жълта – това става като просто след обновяването на GameState и на базата на това променим и текущият речник. Ако той се окаже празен, то значи играчът е въвел грешни цветове и искаме отново да въведе цветовете, докато не е коректен входа му. Текущият речник се получава като от предишния махнем всички думи, които:

- 1) Няма зелени букви на съответната позиция
- 2) Не съдържат всички жълти букви
- 3) Съдържат някоя сива буква

Ако входът е правилен, то продължаваме докато програмата не познае думата или играчът не се откаже.

РАЗПРЕДЕЛЕНИЕ ПО МОДУЛИ :

- Main.hs – тук е функцията за стартирането на програмата.
- Beginning.hs – тук са всички функции свързани с началото на Wordle, т.е. за избирането на режим на играта и дължината за тайната дума.
- Colors.hs – тук се намират всички функции свързани със списъците от цветове.
- WorkWithDictionary.hs – тук са всички функции свързани с извличането на думи/дума от речника.
- Game.hs – тук са всички функции свързани с режим Game, както и за Standard game.
- Easy.hs – тук са всички функции свързани с лесното ниво на играта Easy game.
- Expert.hs - тук са всички функции свързани с експертното ниво на играта Exper game.

- Helper.hs - тук са всички функции свързани с режим помощник на играта Helper mode.

ОСНОВНИ ФУНКЦИИ :

- Beginning.hs :
 - 1) chooseMode – кара играча да въведе каква игра иска да играе и връща какво е избрал.
 - 2) getInt – кара играча да въведе с каква дължина да е тайната дума, функцията връща избраната дължина.
- Colors.hs :
 - 1) colors – взима тайна дума и текущото предположение на играча и връща съответния масив от цветове.
 - 2) getColors – кара играчът да въведе цветовете отговарящи на съответната дума, която е предложила програмата. Функцията връща този списък от цветове.
 - 3) isCorrectEnteredColors – взима списък с цветове и проверява дали са правилно въведени.
- WorkWithDictionary.hs :
 - 1) getWordsOfLengthN – взима дължина и път до файла и връща списък от всички думи с желаната дължина от файла.
 - 2) getRandomWord – взима дължина и връща една произволна дума от файл с тази дължина.
- Game.hs :
 - 1) getSuggestion – взима дължина и кара потребителя да въведе дума с тази дължина и връща думата на играча.
 - 2) continueGame – взима тайната дума и кара играчът да я познае, връща или тайната дума или „изход“.
- Easy.hs :
 - 1) updateGameState – приема тайна дума, предположение и текущото състояние на играта, връща новото състояние на играта като добавя новите сиви, жълти зелени букви към старите.
 - 2) bonusInformation – приема тайна дума, предположение и текущото състояние на играта и връща на терминала информация относно какви букви е изпуснал играча да ползва и какви не е трябвало да ползва.
 - 3) continueGame - взима тайната дума и текущото състояние на играта и кара играчът да я познае, връща или тайната дума или „изход“.
- Expert.hs :

- 1) `getRandomTry` – връща произволно число в интервала [1;3]
- 2) `cheater` – взима тайна дума, предположение, въведените за първи път букви от играча и на базата на това, кои букви какъв цвят трябва да са, връща списъкът с грешни цветове.
- 3) `updateEnteredLetters` – приема предположение и досегашните въведени букви от играча и ги обновява и връща новия стринг
- 4) `continueGame` – взима тайна дума, опитът, при който трябва да излъже, текущият опит и вече въведените букви от играча. Връща или тайната дума или „изход“.

➤ `Helper.hs` :

- 1) `updateGameState` – взима текущото предложение на програмата, цветовете върнати от играча и текущото състояние на играта и на базата на цветовете връща новото състояние на играта.
- 2) `newDictionary` – взима текущото състояние на играта, текущия речник и на базата на състоянието на играта, връща новия речник.
- 3) `countRemovedWords` – приема текущия речник и връща асоциативен списък с ключ позицията в речника и ст-т с колко думи има поне една обща буква (броейки и себе си).
- 4) `getSuggestion` – приема текущия речник и асоциативен списък с ключ позицията в речника и ст-т с колко думи има поне една обща буква (броейки и себе си). Връща тази дума, която има максимална стойност.
- 5) `continueGame` – приема текущото предложение на програмата, текущия речник, асоциативния списък и текущото състояние на играта. Кара потребителя да въведе правилни цветове за текущото предположение за дума на програмата, след което на базата на цветовете или избира ново предположение, което е максимално или играта приключва. Връща списъка от цветове.

ИНСТРУКЦИИ ЗА РАБОТА :

- 1) **ВАЖНО** : ако нямате библиотеката **System.Random** преди стартирането на програмата си я свалете. Може да ползвате следните инструкции :
<https://discourse.haskell.org/t/how-to-install-modules/1363/5>
Чрез нея се реализирани функциите за избиране на произволна дума от речника, както и произволния опит, при който програмата да излъже.
- 2) За стартирането на програмата пуснете `Main.hs`