**Документация за проект Wordle**

Изготвил: Йордан Раданов 2MI0600025

1. **Въведение**

Проектът Wordle представлява имплементация на класическата игра "Wordle" на програмния език Haskell. Играта е налична в три нива на трудност - Easy, Normal и Expert, като всяко ниво е реализирано чрез инстанции на класа Game: **EasyGame**, **NormalGame** и **ExpertGame**. За wordlist е използван следния файл, взет от https://github.com/mahsu/IndexingExercise/blob/master/5000-more-common.txt

1. **Модулите и техните функции**

**2.1 Модул Generator**

**generateWord :: IO String**

Генерира тайна дума от речника, като потребителят въвежда желаната дължина. Използва се речникът "5000-more-common.txt".

**wordlist :: IO [String]**

Извлича списък от думи от речника "5000-more-common.txt".

**getRandomElement :: [a] -> IO a**

Връща произволен елемент от даден списък.

**getWord :: Int -> [String] -> IO String**

Избира дума от речника с определена дължина.

**guessTheWord :: String -> String -> String**

Определя цветовете за всяка буква в даден опит, като използва "g" за зелено, "y" за жълто и "r" за червено.

**2.2 Модул Colourize**

**colourize :: String -> Char -> String**

Оцветява символ в съответния цвят по зададен ANSI код.

**red :: Char -> String**

Оцветява символ в червено.

**green :: Char -> String**

Оцветява символ в зелено.

**yellow :: Char -> String**

Оцветява символ в жълто.

**printColoured :: Char -> IO ()**

Извежда оцветен символ в конзолата.

**2.3 Модул Contradictions**

**printContradictions :: String -> String -> String -> [String] -> IO()**

Извежда информация за противоречията в опита, включително грешки при въвеждане на дума.

**isElem :: String -> [String] -> String**

Проверява дали дадена дума се съдържа в речника.

**addToRed :: String -> String -> String -> String**

Добавя към списъка с "червени" букви буквите от последния опит, които са на "червено" място в думата.

**checkRed :: String -> String -> Bool**

Проверява дали въведените от потребителя букви са в списъка с "червени" букви.

**printCheckRed :: String -> String -> String**

Извежда съобщение, ако въведените букви противоречат с "червените" букви.

**checkYellow :: [String] -> String -> Bool**

Проверява дали в историята на опити има букви, които трябва да са на "жълто" място, но не са въведени от текущия опит.

**printCheckYellow :: [String] -> String -> String**

Извежда съобщение, ако липсват "жълтите" букви във въведения опит.

**checkGreen :: [String] -> String -> Bool**

Проверява дали в историята на опити има букви, които трябва да са на "зелено" място, но не са въведени от текущия опит.

**printCheckGreen :: [String] -> String -> String**

Извежда съобщение, ако липсват "зелените" букви във въведения опит.

**2.4 Модул EasyMode**

**data EasyGame = EasyGame { targetWord :: String }**

Представя структура от данни за играта на лесно ниво.

**instance Game EasyGame where**

Инстанция на класа Game за лесното ниво.

**easyGame :: EasyGame -> IO ()**

Функция за изпълнение на играта на лесно ниво.

**runEasyGame :: IO ()**

Стартира играта на лесно ниво.

**2.5 Модул NormalMode**

**data NormalGame = NormalGame { targetWord :: String }**

Представя структура от данни за играта на нормално ниво.

**instance Game NormalGame where**

Инстанция на класа Game за нормалното ниво.

**normalGame :: NormalGame -> IO ()**

Функция за изпълнение на играта на нормално ниво.

**runNormalGame :: IO ()**

Стартира играта на нормално ниво.

**2.6 Модул ExpertMode**

**data ExpertGame = ExpertGame { targetWord :: String }**

Представя структура от данни за играта на експертно ниво.

**instance Game ExpertGame where**

Инстанция на класа Game за експертното ниво.

**expertGame :: ExpertGame -> String -> [Bool] -> [String] -> IO()**

Функция за изпълнение на играта на експертно ниво.

**runExpertGame :: IO()**

Стартира играта на експертно ниво. Програмата има право да излъже потребителят един път, като това се случва с вероятност 1/8. Когато излъже програмата, отговорът не противоречи с подадените отговори преди това. Понеже е лесно да се разбере какъв е коректният изход при съответната дума, играчът не може да разбере изхода за дума, изписана по-рано в хода на играта.

**2.7 Модул Main**

**main :: IO()**

Главната функция, която стартира програмата. Потребителят избира ниво на трудност (Easy, Normal, Expert) и започва играта.

1. **Заключение**

Проектът Wordle предоставя интересна имплементация на класическата игра с различни нива на трудност. Модулите се използват за генериране на тайна дума, оцветяване на изхода и управление на противоречията. Потребителят има възможност да избира ниво на трудност и да се наслади на различни предизвикателства в играта Wordle.