Programátorská dokumentace hry Wordle

**RANDOM**

Nejprve importujeme modul ***random***, který je nutný k náhodnému výběru slov pro hádání.

**SLOVNÍK**

Dále potřebujeme mít nějaký seznam slov, ze kterého bude hra slova vybírat.

* Proto jsem si stáhl z internetu seznam nejpoužívanějších anglických slov a pak z nich vybral pouze ta pětimístná.
* Vytvoříme tak soubor **possible\_words.txt**

**WORDLE**

1. Načtení slovníku

Vytvoříme funkci ***load\_words(file\_path)***, pomocí které program otevře soubor **possible\_words.txt** a z jednotlivých slov ve slovníku vytvoří seznam – **words**.

1. Zkoumání jednotlivých písmen a poskytnutí nápovědy

Funkce ***check\_place(guess, wordle)*** nám bude sloužit při zkoumání slova daného hráčem.

* Jejím výstupem bude seznam **feedback**, který bude obsahovat nápovědy
* Parametr **guess** reprezentuje dané slovo hráčem a parametr **wordle** zas hledané slovo náhodně vybrané hrou
* Funkce postupně porovnává písmena na stejných pozicích těchto dvou parametrů
  + Pokud jsou na stejných pozicích stejná písmena, do výsledného seznamu **feedback** se uloží pozice, písmeno a zpráva, že je písmeno na SPRÁVNÉM, tedy stejném, místě
  + Pokud jsou na stejné pozici jiná písmena, ale písmeno se v hledaném slově nachází (in **wordle**), do výsledného seznamu **feedback** se uloží pozice, písmeno a zpráva, že je písmeno na ŠPATNÉM místě
  + Pokud se písmeno v hledaném slově nenachází vůbec, do seznamu **feedback** se neuloží nic
* Např:
  + guess = PLANE, wordle = APPLE => check\_place(guess, wordle)

=> feedback = [“1. Letter – P: WRONG place”, „2. Letter – L: WRONG place”, “3. Letter – A: WRONG place”, “5. Letter – E: RIGHT place”]

* Funkce vrací seznam **feedback**

1. Kontrola správného formátu zadaných slov

Funkce **get\_valid\_guess(words)** kontroluje, zda je slovo zadané hráčem (**guess**) validní, tedy jestli je pětimístné a jestli se jedná o slovo, které existuje, jinými slovy, zda jej můžeme nalézt v našem seznamu **words**. Tento seznam je jediným parametrem této funkce.

* Kontrola je umožněna while cyklem, který končí v momentě, kdy je zadáno validní slovo.
* Nejprve vyzveme hráče, aby zadal slovo. Toto slovo uložíme do proměnné **guess**. Pak následují kontroly:
  + Pokud slovo není pětimístné, vrací program zprávu: *"That was not a five letter word!"* a cyklus pokračuje
  + Pokud slovo je pětimístné, ale nelze jej nalézt v našem seznamu **words**, vrací program zprávu: *"Word not found!"*
  + Pokud slovo je pětimístné a lze jej nalézt v seznamu **words**, je validní a tedy jej můžeme použít. Cyklus je zastaven.
* Funkce vrací proměnnou **guess**, tedy validní slovo

1. Samotná hra

Nejprve využijeme funkci ***load\_words(„possible\_words.txt“)*** a vytvoříme tak seznam slov - **words**.

Poté s pomocí příkazu ***random.choice(words)*** z tohoto seznamu do proměnné **wordle** uložíme náhodně vybrané slovo. Toto slovo chce hráč uhodnout.

Uživatel má šest pokusů na to, aby slovo uhádl, proto vytvoříme while cyklus, který se zastaví po šestém pokusu, a nebo v případě, že hráč slovo uhodne.

Následuje výzva hráče k hádání. Využijeme funkci ***get\_valid\_guess(words)*** a obdržíme tak od uživatele validní slovo, které uložíme do proměnné **guess**.

Pokud je **guess** stejné jako **wordle**, hráč vyhrál. Program mu tuto skutečnost sdělí pomocí zprávy„You won!“ a poté ukončí cyklus – break.

Jinak použijeme funkci ***check\_place(guess, wordle),*** která postupně srovná písmena obou parametrů **guess** a **wordle** na první až na páté (poslední) pozici. Hráč tak dostane nápovědy a hádá znovu.

V momentě, kdy dojdou všechny pokusy a slovo není stále uhodnuto - cyklus skončil přirozeně, uživatel prohrává. Program vytiskne zprávu „You lost!“ a sdělí správnou odpověď: „The right answer was: {**wordle**} “

Konec programu