

## Dokumentacja projektu Scrabble

### 1. Cel i opis projektu

Celem projektu jest stworzenie gry Scrabble. Polega ona na układaniu z dostępnych liter słów tak, aby uzyskać jak najwięcej punktów. Rozgrywka przeprowadzona może być pomiędzy graczami (od 2 do 4) lub pomiędzy graczem a botem. Kto zakończy rozgrywkę z największą ilością punktów, wygrywa.

Aktualny wynik gracza Kamil wynosi: 23

Runda numer 4: Teraz kolej gracza Janek

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0															
1															
2															
3															
4															
5						Z	B	A	W						
6							O								
7							N	U	R						
8									Z						
9									E			B			
10									P	O	L	O			
11												Ż			
12												K			
13												I			
14															

Stojak gracza Janek: U, E, N, F, Y, E, I

## 2. Podział kodu

Klasa Płytki – reprezentuje jedną płytkę dostępną w grze

Klasa PulaLiter – jest to odpowiednik worka z płytkami, z niego pobieramy płytki

Klasa Stojak – odpowiada za przechowywanie maksymalnie 7 płytek, każdy gracz posiada swój stojak

Klasa Gracz – reprezentuje każdego z graczy, jego wynik oraz stojak

Klasa Bot – dziedziczy po klasie Gracz, jest to po prostu bot, który w swoim ruchu dodaje na planszę najlepiej punktowane słowo, jakie może utworzyć

Klasa Plansza – przechowuje litery położone na planszy oraz odpowiada za graficzne przedstawienie wypełnionej planszy

Klasa Slovo – zarządza słowem, które gracz próbuje dodać na planszę. Sprawdza jego poprawność

Funkcja tura – pozwala przeprowadzić jedną pełną turę gry

Funkcja rozpocznij\_gre – inicjuje rozgrywkę

Funkcja zakoncz\_gre – wypisuje wynik zakończonej rozgrywki

## 3. Instrukcja użytkownika

W folderze z grą oprócz pliku scrabble.py musi znaleźć się także plik słowa.txt – jest to słownik z dostępnymi w grze słowami (do pobrania tutaj <https://sjp.pl/sloownik/growy/>). Aby zagrać, należy uruchomić przez interpreter plik scrabble.py, a następnie wybrać tryb gry. Podczas rozgrywki należy stosować się do wyświetlanych komunikatów. W swoim ruchu można dokonać wymiany płytek, położyć nowe słowo lub pominąć turę. Gra kończy się, gdy jeden z graczy nie będzie już miał płytek, a pula liter również będzie pusta. Alternatywnie koniec gry nastąpi również w momencie, gdy każdy gracz dwa razy z rzędu pominie swoją turę (czyli przy 3 graczach będzie to 6 tur pominiętych z rzędu). „#” odpowiada za pustą płytkę – może ona być dowolnie wybraną literą

## 4. Część refleksyjna

Pracę rozpocząłem od napisania wszystkich niezbędnych klas. Później zająłem się samym przeprowadzeniem rozgrywki. Zrealizowałem wszystkie funkcjonalności, które gra powinna posiadać. Problem sprawiła mi optymalizacja bota. Gdy na planszy pojawia się więcej liter, dodawanie słowa trwa długo.