

# Dokumentacja projektu Scrabble

Barbara Zięba, Dominik Gulczyński

27 czerwca 2018

## 1 Wstęp

Jako projekt wieńczący kurs Programowania obiektowego napisaliśmy program umożliwiający przeprowadzenie rozgrywki w Scrabble na komputerze. Gra została napisana w języku Java. Grafika gry powstała na bazie standardowych bibliotek okienkowych AWT i Swing.

## 2 Instrukcja dla użytkownika

### 2.1 Początek gry

Po uruchomieniu programu użytkownik proszony jest o podanie imion graczy. Gracze będą prowadzili rozgrywkę w podanej kolejności zaczynając od podanego jako pierwszy. Można podać imiona od jednego do czterech graczy, i to właśnie tyłu graczy rozpocznie rozgrywkę.

### 2.2 Ruch gracza

Gracz musi w swoim ruchu wykonać jedną z następujących akcji:

**Położenie słowa.** Aby położyć słowo gracz powinien kliknąć kratkę planszy, w której będzie się znajdować pierwsza litera układanego słowa. Uwaga: Może to być litera już znajdująca się na planszy. W nowym okienku należy wpisać jakie to będzie słowo (również litery już znajdujące się na planszy) i zaznaczyć czy ma zostać położone pionowo (*vertical*) czy poziomo (*horizontal*). Wszystkie tworzone przy ruchu słowa muszą być poprawne, a dokładane litery łączyć się z już ułożonymi. Jeśli ruch będzie poprawny odpowiednie litery gracza pojawią się na planszy, a on otrzyma nowe.

**Wymiana liter** Gracz może w swoim ruchu wymienić od jednej do 7 (wszystkich) liter, które posiada. Za taki ruch nie otrzymuje punktów. Aby dokonać wymiany należy kliknąć przycisk *Exchange tiles* znajdujący się obok literek na ręku, a następnie wpisać bez spacji litery do wymiany.

**Pass** Po kliknięciu przycisku *Pass* gracz opuszcza swoją kolejkę. Może wykonać taki ruch, nawet jeśli ma literki do ułożenia.

Po ruchu litery gracza zostają schowane i pojawia się komunikat o ruchu następnego gracza. Należy przekazać mu komputer i kliknąć *OK*.

### 2.3 Koniec gry

Gra kończy się, gdy wszyscy gracze z rzędu spasuują, lub jeden z graczy wykorzysta wszystkie swoje płytki, a woreczek z płytkami będzie pusty. Zwycięzcą zostaje gracz o najwyższej liczbie punktów. Gdy gra kończy się, pojawia się okienko *EndGameDialog*.

### 2.4 Cofanie ruchu

Gracze sami powinni zdecydować, jak chcą korzystać z tej opcji. Sugerujemy, żeby w razie pomyłki we wpisywaniu użytkownik mógł zgłosić (niezwłocznie), że chce cofnąć ostatni ruch. Wtedy następny gracz powinien skorzystać z przycisku *Revert last move* i dać kolejną szansę koledze.

### 2.5 Szczegółowe zasady gry

Program kopiuje zasady gry Scrabble. Aby poznać reguły liczenia punktów i inne szczegóły gry, można zajrzeć na stronę <http://www.scrabble.info.pl/pobierz/zasady.pdf>. Nasz program korzysta ze słownika wyrazów dopuszczalnych w grach dostępnego na stronie Słownika Języka Polskiego <https://sjp.pl/slownik/growy/>.

## 3 Analiza obiektowa

### 3.1 Zaimplementowane klasy

#### 3.1.1 Logika gry

**Board** Jedna z głównych klas programu. Przechowuje informacje o stanie planszy: położonych literach, specjalnych polach i słowniku, który decyduje, które wyrazy są poprawne. Posiada dwie metody. Pierwsza, `placeWord` przyjmuje słowo, które chcemy położyć na planszy i jego pozycję. Jeśli jest to poprawny ruch zwraca parę listę płytek użytych przez gracza i liczbę punktów zdobytych za ten ruch. Kolejna, `value` służy do liczenia punktów zdobytych za ułożenie danego słowa.

**Game** Najważniejsza klasa informująca o przebiegu gry. Dziedziczy z klasy **Board**, a oprócz tego przechowuje listę graczy, listę ruchów (historię) i woreczek dostępnych liter.

**Rack** Po angielsku oznacza stojak na literki. Zawiera listę płytek i udostępnia metody, które pozwalają na aktualizację tej listy.

**Player** Klasa dziedzicząca po **Rack** i odpowiadająca za informacje na temat gracza. Zawiera jego imię i bieżącą liczbę punktów.

**Bag** Reprezentuje woreczek z płytkami, które zostały do wylosowania. Jej pole to lista płytek, a metody udostępniają operacje do wykonania na tej liście, np. wyjęcie losowej płytki.

**Move** Klasa ruchu. Przydatna do pamiętania historii gry i wyświetlania jej przebiegu.

**Tile** Reprezentuje płytkę na planszy. Tutaj mamy również statyczne metody `getValueOf` (mówi ile punktów jest warta dana litera) i `paintTile` (służy do rysowania płytek).

**Bonus** Odpowiada za specjalne pola planszy. Zawiera informacje na temat bonusów powiększających liczbę punktów zdobywanych za ułożenie słowa.

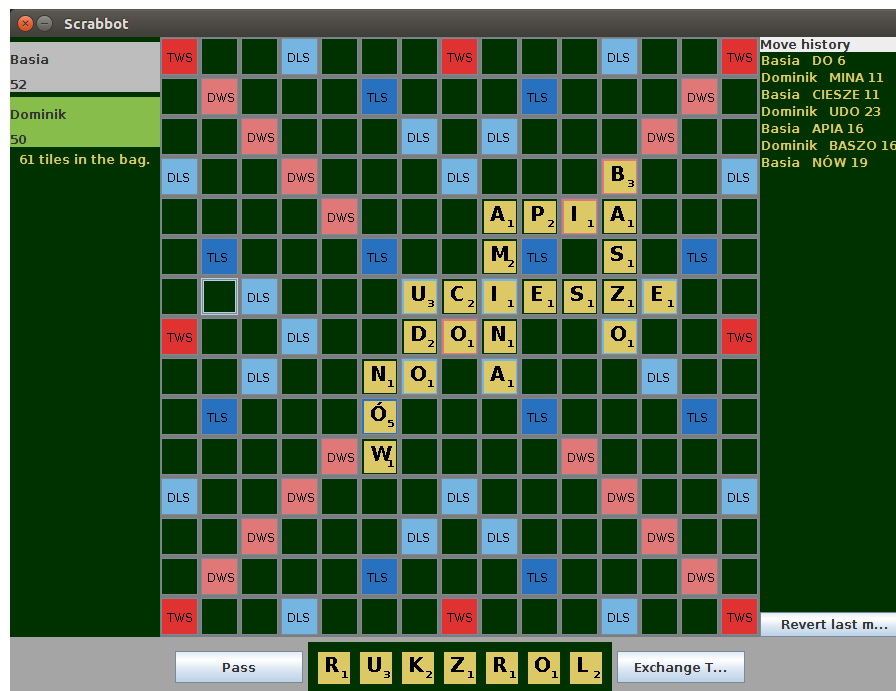
**Tree** Drzewo wyrazów dopuszczonych do użytku w grze. Zostaje utworzone za pomocą pliku txt upublicznionego przez stronę [www.sjp.pl](http://www.sjp.pl). Zawiera wewnętrzną klasę `Node`.

**Main** Klasa, która uruchamia program.

### 3.1.2 Klasy rozszerzające Swing

#### Główne okno gry

**GameWindow** Główne okno gry, zarządza komunikacją gracz-komputer.



Rysunek 1: Główne okno gry

#### Panele

**BoardPanel** Zajmuje centralną część okna gry. Zawiera pola-przyciski typu `CellButton`, których przyciśnięcie wywołuje `MoveDialog`.

**HistoryPanel** Znajduje się po prawej stronie okna gry. Informuje o historii gry poprzez wypisywanie kolejnych ruchów.

**PlayerPanel** Znajduje się po lewej stronie okna gry. Informuje o wynikach poszczególnych graczy. Obecny gracz jest podświetlony na zielono.

**RackPanel** Znajduje się na dole okna gry. Wyświetla `TilePanel` obecnego gracza, oraz zawiera przyciski `Pass` i `Exchange`.

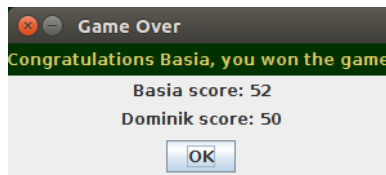
**TilePanel** Wyświetla stojaczek wybranego gracza.

#### Przyciski

**CellButton** Wyświetla pojedyncze pole na planszy gry.

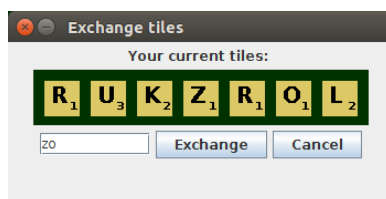
## Okienka typu *pop-up*

**EndGameDialog** Informuje, że gra się zakończyła. Wyświetla tabelę wyników graczy.



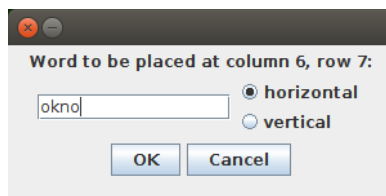
Rysunek 2: Okienko zakończenia gry

**ExchangeDialog** Wyświetla się, gdy gracz, naciśnie przycisk **Exchange**. Pozwala wymienić literki na stojaczku (kosztuje to jedną turę). Jeżeli literki które gracz chce wymienić nie zgadzają się z jego stojakiem, pojawia się komunikat o błędzie.



Rysunek 3: Okienko wymiany liter

**MoveDialog** Jest wyświetlany po kliknięciu wybranego pola. W okienku gracz wpisuje wybrane słowo. Jeśli jest poprawne — okienko znika, jeśli nie — pojawia się komunikat o błędzie.



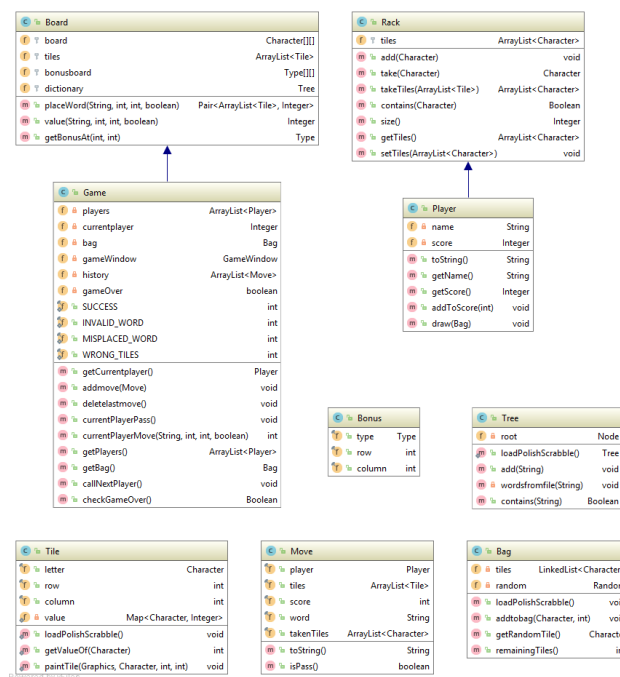
Rysunek 4: Okienko nowego ruchu

**NewGameDialog** Pierwsze okienko, które widzi gracz. Pozwala wprowadzić imiona do czterech graczy, którzy będą prowadzić rozgrywkę.

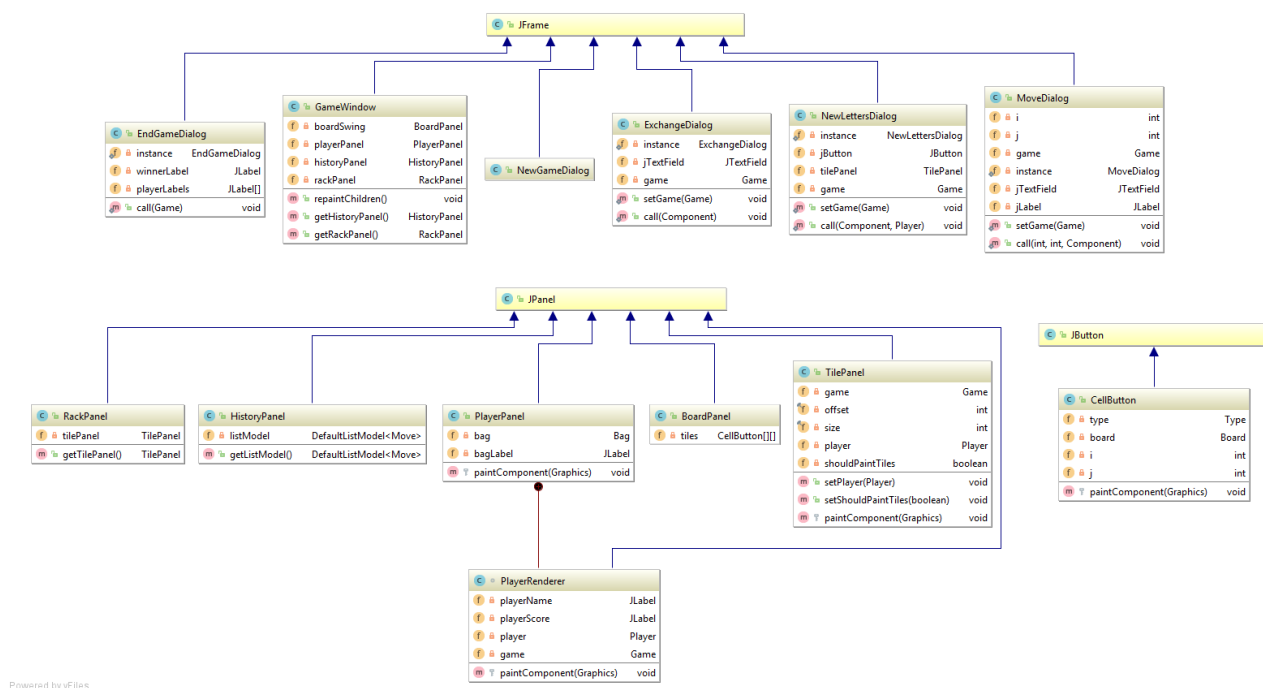
**NewLettersDialog** Pojawia się zawsze na koniec tury. Informuje gracza o stanie jego stojaczka.

## 3.2 Diagram UML klas

Diagramy wygenerowane za pomocą IntelliJ IDEA.



Rysunek 5: Logika gry



Rysunek 6: Wyświetlanie gry

## 4 Wskazówki dla programistów

Projekt jest otwarty na dalszą pracę nad nim. Jedną z opcji jego prostego rozbudowania jest umożliwienie gry w innych językach lub na innych zasadach. Można również pokusić się o dodanie możliwości zmierzenia się w grze z komputerem. **Uwaga!** Pliki gry kodowane są w formacie UTF-8.