

# Zaawansowana sztuczna inteligencja do gry Scrabble

Jakub Turek  
J.Turek@stud.elka.pw.edu.pl

Promotor: dr inż. Jakub Koperwas

Wydział Elektroniki i Technik Informatycznych

25 kwietnia 2014

# Agenda

1. Wprowadzenie.
  - 1.1 Ogólny opis Scrabble.
  - 1.2 Spis zagadnień omówionych na pierwszym seminarium.
2. Omówienie podstaw teoretycznych.
  - 2.1 Optymalna strategia.
  - 2.2 Podział na fazy gry.
3. Opis wykorzystywanych algorytmów w podziale na fazy gry.
4. Wyniki testów.
5. Podsumowanie.
  - 5.1 Literatura uzupełniająca.

# Scrabble - definicja

- ▶ Gra planszowa dla 2-4 osób.
- ▶ Na początku gry każdy gracz otrzymuje po 7 klocków. Klocki należą do jednej z dwóch grup:
  - ▶ Reprezentują pojedynczą literę alfabetu i przypisaną do niej wartość punktową.
  - ▶ Reprezentują dowolną literę i nie mają wartości punktowej (blanki).
- ▶ Gra toczy się w turach. W każdej turze zadaniem gracza jest ułożenie na planszy wyrazu w układzie krzyżówkowym:
  - ▶ Dopuszczalne są dowolne wyrazy lub ich odmiany ujęte w słownikach języka i ortograficznych.
  - ▶ Wyjątki stanowią wyrazy rozpoczynające się wielką literą, skróty, przedrostki, przyrostki oraz słowa wymagające użycia łącznika lub apostrofu.
- ▶ Wartość punktowa jest zależna od sumy wartości klocków oraz ich położenia na planszy (premie literowe oraz słowne).

## Plansza do gry



**Rysunek:** Plansza wykonana z włókna węglowego, podświetlana diodami LED.

# Scrabble jako gra turniejowa w Polsce

- ▶ Polska Federacja Scrabble to oficjalna federacja zrzeszająca kluby Scrabble w Polsce. Została założona w 1997 roku.
- ▶ Ranking PFS zrzesza 335 graczy, którzy rozegrali minimum 30 partii turniejowych w przeciągu ostatnich dwóch lat.
- ▶ Zasady:
  - ▶ Ograniczony czas na wykonanie wszystkich ruchów - po 20 minut na gracza.
  - ▶ Dozwolone są ruchy uznawane za poprawne przez Oficjalny Słownik Polskiego Scrabblisty.
- ▶ Rekordy:
  - ▶ Najwyższy wynik w partii pełnej - Michał Alabrudziński, 721 punktów.
  - ▶ Najlepsze otwarcie - Radosław Sowiński, 112 punktów za słowo **źrebień**.

# Przypomnienie (1/2)

Podczas poprzedniego wystąpienia zostały omówione następujące zagadnienia:

1. Porównanie słowników do gier dla języka polskiego:

**OSPS** „Oficjalny Słownik Polskiego Scrabblisty”.

**SA** Słownik alternatywny.

2. Analiza statystyczna słownika alternatywnego.

3. Omówienie efektywnych struktur słownikowych zorientowanych na przeglądanie poprawnych sufiksów wyrazów:

**Trie** Drzewo poszukiwań.

**DAG** Directed Acyclic Graph.

**GADDAG** prefiksowo-sufiksowa odmiana DAG.

# Przypomnienie (2/2)

Omówione zagadnienia - ciąg dalszy:

4. Przedstawienie algorytmu Appela-Jacobsona wyznaczającego wszystkie legalne kombinacje ruchów dla ustalonego stanu gry.
5. Porównanie najlepszych algorytmów sztucznej inteligencji obecnej generacji:
  - ▶ Algorytm Maven.
  - ▶ Aplikacja Quackle.
6. Przedstawienie wybranych elementów algorytmu używanego w aplikacji Quackle.

# Założenia algorytmu

Cel pracy oraz przyjęte założenia:

- ▶ Zwiększenie procentowej liczby wygranych najlepszych algorytmów obecnej generacji:
  - ▶ Skuteczność mierzona w starciu z przeciwnikami klasy mistrzowskiej.
- ▶ Algorytmem bazowym (oraz referencyjnym) jest wykorzystywany przez aplikację Quackle.
- ▶ Średni czas wykonania ruchu nie może być większy niż w algorytmach obecnej generacji.
- ▶ Złożoność pamięciowa algorytmu nie jest istotna.
- ▶ Słownik dopuszczalnych wyrazów jest znany z góry:
  - ▶ Dodanie obsługi nowego języka wymaga przeprowadzenia automatycznej analizy, która może być operacją czasochłonną.



# Strategia optymalna

## Twierdzenie

Istnieje optymalna strategia gry w Scrabble.

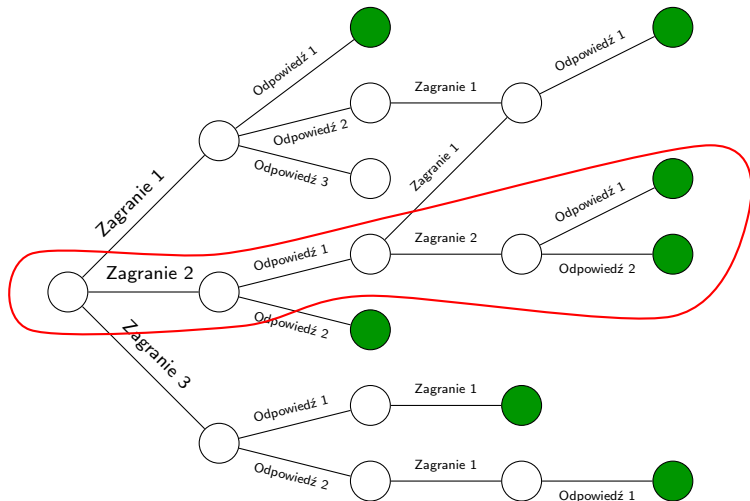
## Przestrzeń stanów

Stan rozgrywki po danej turze opisują parametry  $P$  - rozmieszczenie klocków na planszy oraz  $Z$  - zagranie. Przejście między stanami determinuje zmiana  $(\Delta P, \Delta Z)$ .

## Dowód

Dla dowolnej rozgrywki tworzymy graf możliwych stanów wychodząc od stanu końcowego. Do stanu końcowego można wejść tylko poprzez skończoną liczbę legalnych zagrań. Rozumując iteracyjnie dochodzimy do stanu początkowego, na każdym etapie analizując skończoną liczbę przejść między stanami. Wynika z tego, że ilość stanów jest skończona. Można więc w każdym kroku wybrać optymalną strategię, która maksymalizuje prawdopodobieństwo wygranej.

## Strategia optymalna - ilustracja dowodu



# Strategia optymalna - następstwa

- ▶ Wyznaczenie optymalnej strategii należy do klasy problemów *PSPACE-complete*.
- ▶ Analiza przestrzeni stanów jest możliwa wyłącznie dla bardzo ograniczonej przestrzeni stanów:
  - ▶ W praktyce analiza przestrzeni stanów jest możliwa, gdy w worku nie ma już klocków lub gdy pozostał jeden/dwa klocki.
- ▶ **Należy zmieniać strategię w zależności od fazy rozgrywki:**
  - ▶ Nie zawsze można użyć strategii optymalnej.
  - ▶ Wykorzystanie metod heurystycznych.

# Fazy gry

Rozgrywkę w Scrabble można podzielić na cztery zasadnicze fazy:

- OP** *opening-play*. Faza obejmuje pierwsze zagranie.
- MG** *mid-game*. Faza trwa od momentu rozpoczęcia rozgrywki do momentu rozpoczęcia fazy *pre-endgame*.
- PEG** *pre-endgame*. Faza rozpoczyna się, gdy w worku pozostaje  $\leq 7$  klocków i trwa do rozpoczęcia fazy *end-game*. W PEG każde kolejne zagranie może skutkować opróżnieniem worka. Dodatkowy podział:
  - PEG-1** W worku pozostał jeden klocek. Przez tę fazę przechodzi około 50% gier.
  - PEG-2** W worku pozostały dwa klocki.
  - PEG-X** W worku pozostało  $x \leq 7$  klocków.
- EG** *end-game*. W worku nie ma już żadnych klocków.

# Strategia a faza gry

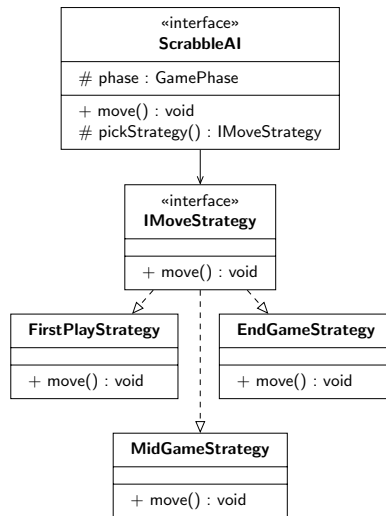


**Rysunek:** Zmiana strategii wraz z progresją rozgrywki.

# Wzorzec projektowy - strategia

## Strategia

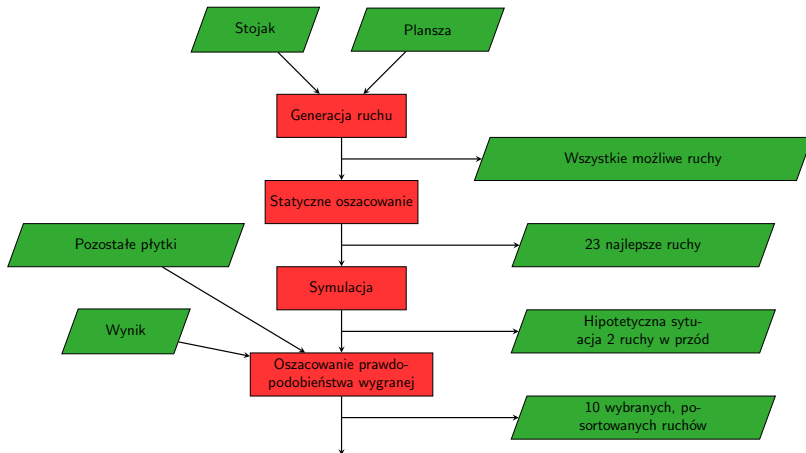
Wzorzec definiuje rodzinę algorytmów, pakuje je jako osobne klasy i powoduje, że są one w pełni wymienne. Zastosowanie strategii pozwala na to, aby zmiany w implementacji przetwarzania były całkowicie niezależne od strony klienta, który z nich korzysta.



# Otwarcie gry

- ▶ Celem jest wybranie najwyższego punkowanego zagrania.
- ▶ W algorytmie aplikacji Quackle jest przeszukiwany słownik i dla każdej prawidłowej kombinacji liter wyznaczana jest wartość punkтова.
- ▶ Wprowadzona optymalizacja wydajnościowa:
  - ▶ Ponieważ słownik jest znany z góry można przeprowadzać wstępne obliczenia.
  - ▶ Każdą możliwą kombinację liter indeksujemy 7-wyrazowym ciągiem znaków, które przedstawiają litery ułożone w porządku alfabetycznym.
  - ▶ Dla każdego indeksu obliczamy najlepsze otwarcie.
  - ▶ Najlepsze otwarcie wyznaczone w czasie jednostkowym (wyjątek: blanki).

## Algorytm dla fazy mid-game



**Rysunek:** Schemat algorytmu referencyjnego dla fazy MG.



## Faza mid-game: statyczne oszacowanie

Algorytm referencyjny wykorzystuje do statycznego oszacowania poniższą funkcję celu.

### Funkcja celu

$$F(x) = P(x) + LV(x),$$

gdzie  $P(x)$  to liczba punktów za zagranie  $x$ , a  $LV(x)$  to *leave value* klocków pozostałych po wykonaniu zagrania  $x$ .

### Leave value

Obliczona na podstawie bazy danych gier wartość, która faworyzuje kombinacje liter o większym prawdopodobieństwie dopełnienia wysokopunktowych zagrań w nadchodzących ruchach. Może przyjmować dodatnie oraz ujemne wartości.

# Faza mid-game: symulacja

Algorytm referencyjny wykonuje następującą symulację:

1. Gracz wykonuje zagranie  $P_1$ .
2. Przeciwnik wybiera 7 losowych klocków, wyznacza wszystkie możliwości ruchu i wykonuje zagranie  $P_2$  dla ruchu z najlepszym statycznym oszacowaniem.
3. Gracz uzupełnia klocki, wyznacza wszystkie możliwości ruchu i wykonuje zagranie  $P_3$  dla ruchu z najlepszym statycznym oszacowaniem.
4. Gracz oblicza wartość  $PV_1$  zagrania  $P_1$  odejmując od liczby swoich punktów po zagranie  $P_3$  liczbę punktów przeciwnika po zagranie  $P_2$ .
5. Gracz dodaje do  $PV_1$  *leave value* po zagranie  $P_3$ .

# Faza mid-game: oszacowanie prawdopodobieństwa wygranej

Algorytm referencyjny szacuje prawdopodobieństwo wygranej zgodnie z poniższą zależnością.

## Estymata prawdopodobieństwa wygranej

$$W : PV, TR \rightarrow [0; 1],$$

gdzie  $PV$  to wartość zagrania obliczona na etapie symulacji, a  $TR$  to liczba klocków pozostałych do wykorzystania w partii.

Wartość funkcji jest wyznaczana na podstawie bazy danych gier.

# Faza mid-game: słabe strony algorytmu (1/2)

Symulacja pomija istotne aspekty rozgrywki:

- ▶ Zawartość stojaka jest losowana z pozostałej puli klocków z użyciem rozkładu jednostajnego. Można lepiej wyznaczyć rozkład prawdopodobieństwa:
  - ▶ W poprzednim ruchu przeciwnik ułożył słowo **RADO**. Łatwo wywnioskować, że nie posiadał on liter **U** oraz **I**, gdyż w przeciwnym razie prawdopodobnie ułożyłby lepiej punktowane słowa - **URODA** lub **RADIO**.
  - ▶ Trzy litery są pozostałością po poprzednim ruchu, więc przy ich losowaniu wykluczamy z worka **U** oraz **I**.
- ▶ Symulacja nie uwzględnia zjawiska *łowienia* korzystnych zagrań:
  - ▶ Niektóre kombinacje liter dają 90% prawdopodobieństwo, że po dobraniu do nich kolejnych liter będzie można ułożyć cenny 7-literowy wyraz.
  - ▶ Próba przewidzenia klocków przeciwnika przy podejrzeniu *łowienia*.

## Faza mid-game: słabe strony algorytmu (2/2)

Oszacowanie prawdopodobieństwa wygranej wykorzystuje nieprecyzyjne informacje wejściowe:

- ▶ Pozycja „w przyszłości” wyznaczona na etapie symulacji może być nieadekwatna do rzeczywistości:
  - ▶ W trakcie symulacji pominięte zostało *bingo* przeciwnika.
- ▶ Podział zagrań na ofensywne i defensywne na podstawie ilości otwartych premii oraz ilości otaczających klocków po wykonaniu zagrania.
- ▶ Rozważanie opłacalności zagrania ofensywnego / defensywnego na obecnym etapie rozgrywki.
  - ▶ Wykorzystanie sieci neuronowej.

# Faza end-game (1/2)

- ▶ Możliwe wykorzystanie drzewa przestrzeni stanów:
  - ▶ Dla rozgrywki 7 na 7 klocków ilość gałęzi wynosi 200, a maksymalna ilość zagłębień wynosi 14.
  - ▶ Nie można zastosować efektywnie algorytmu  $\alpha - \beta$ .
  - ▶ Wymagane jest przeszukiwanie drzewa z ograniczeniami.
- ▶ Wykorzystanie programowania dynamicznego:
  - ▶ Założenie, że sytuacja na planszy jest statyczna. Szacowana jest wyłącznie wartość stojaków.
  - ▶ Wartość każdego możliwego ruchu jest szacowana przy założeniu, że gra zakończy się dokładnie w  $N$  turach.
  - ▶ Oszacowanie wartości ruchu  $F_N(x) = P_N(x) + LV_{N-1}$ , gdzie  $P_N$  to liczba punktów uzyskanych za ruch, a  $LV_{N-1}$  to oszacowanie wartości pozostałych klocków przy założeniu, że do końca gry pozostało  $N - 1$  tur.

## Faza end-game (2/2)

Przykład oszacowania:

1. Gracz 1 ( $G_1$ ) zakończy grę w 8 ruchach. Wtedy gracz 2 ( $G_2$ ) rozegra 7 ruchów. Powstaje ścieżka, którą przetwarzamy algorytmem minimax.
2.  $G_2$  zakończy grę w 7 ruchach.  $G_1$  rozegra wtedy 7 ruchów. Jeżeli ta ścieżka będzie lepsza dla  $G_2$ ,  $G_2$  wybierze właśnie ją.
3.  $G_1$  będzie miał okazję do poprawy jeżeli zakończy grę w 7 ruchach...

Algorytm jest powtarzany do momentu, gdy żaden z graczy nie może się poprawić.

# Faza end-game: modyfikacje

- ▶ Nie zawsze istnieje jedna optymalna ścieżka zagrania dla przeciwnika:
  - ▶ Przykładowo, jeżeli przeciwnik może zagrać słowo **ALE** w dwóch pozycjach (za 25 i 28 punktów) korzyść z zablokowania drugiej pozycji względem zablokowania pierwszej wynosi tylko 3 punkty.
- ▶ Rozwiązaniem problemu jest wprowadzenie dwóch oszacowań:
  - ▶ optymistycznego,
  - ▶ pesymistycznego.
- ▶ Wykorzystanie algorytmu  $B^*$  do przeszukiwania przestrzeni (minimax nie wspiera przedziałów).



# Faza pre-endgame

Algorytm w fazie *pre-endgame* wymaga połączenia podejść wykorzystywanych w fazach *mid-game* i *end-game*:

- ▶ Obliczenia wykonywane w fazie *mid-game* nadal pozostają poprawne.
- ▶ Istnieje możliwość zakończenia gry w dwóch turach, więc dla uzyskania pełnego obrazu należy przeszukać przestrzeń stanów jak w fazie *end-game*.
- ▶ Hybrydowy algorytm dla fazy *pre-endgame* jest uruchamiany począwszy od 9 klocków pozostałych w worku.

# Wyniki testów przeciwko algorytmowi referencyjnemu

- ▶ 1523 rozegrane partie.
- ▶ Teoretyczny rozkład dla przeciwników na jednakowym poziomie powinien być bliski 50% – 50%.
- ▶ Wyraźne odchylenie na korzyść algorytmu po modyfikacjach.

	Algorytm zmodyfikowany	Algorytm referencyjny
Wygrane	987	536
% wygranych	<b>64,8%</b>	35,2%

# Wyniki testów przeciwko ludziom

- ▶ Nie udało się przeprowadzić miarodajnych testów przeciwko ludziom.
  - ▶ Przewaga, którą daje znajomość całego słownika jest tak duża, że algorytm osiągnął 100% wygranych w dotychczasowych rozgrywkach przeciwko ludziom.
  - ▶ Niemożność przeprowadzenia testów na przeciwnikach klasy mistrzowskiej.
- ▶ W przyszłości skuteczność algorytmu testowana będzie przeciwko graczom serwisu kurnik.pl.

# Literatura uzupełniająca



M. Lampis, V. Mitsou, K. Sołtys.

Scrabble is PSPACE-Complete.



Brian Sheppard.

World-championship-caliber Scrabble.

*Artificial Intelligence*, vol. 134, p. 241-275, January 2002.



J. Katz-Brown, J. O'Laughlin.

How Quackle Plays Scrabble.

# Dziękuję za uwagę!