

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.
2. Przegląd zagadnień omówionych na pierwszym seminarium.
3. Istnienie optymalnej strategii gry.
4. Ogólny opis algorytmu.
5. Omówienie wykorzystanych mechanizmów w podziale na fazy gry.

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.

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. Podział rozgrywki na cztery fazy - MG, PEG-1, PEG-2, EG.
6. Porównanie najlepszych algorytmów sztucznej inteligencji obecnej generacji:
 - ▶ Algorytm Maven.
 - ▶ Aplikacja Quackle.
7. 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.

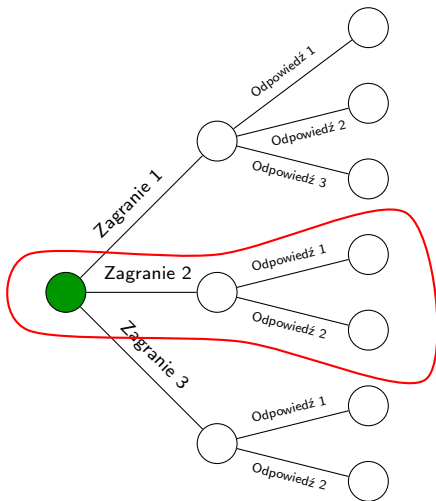
Przestrzeń stanów

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

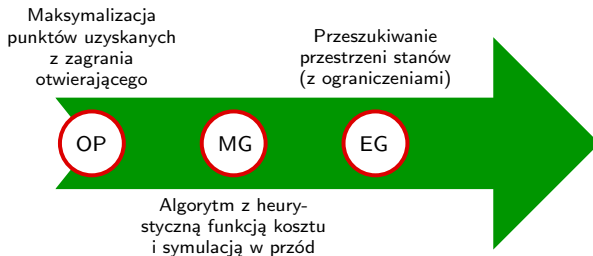
- ▶ Wyznaczenie optymalnej strategii należy do klasy problemów *PSPACE-complete*.
- ▶ Analiza przestrzeni stanów jest możliwa wyłącznie dla bardzo ograniczonej przestrzeni stanów:
 - ▶ W praktyce analiza przestrzeni stanów jest możliwa, gdy w worku nie ma już klocków lub gdy pozostał jeden/dwa klocki.
- ▶ **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, w którym w worku pozostaje jeden/dwa klocki lub nie ma już klocków (w zależności od przebiegu rozgrywki).
- PEG** *pre-endgame*. Faza trwa, gdy w worku pozostał pojedynczy klocek. Według szacunków przez tę fazę przechodzi około 50% gier. Można wyróżnić dwie podfazy:
 - PEG-1** W worku pozostał jeden klocek.
 - PEG-2** W worku pozostały dwa klocki.
- EG** *end-game*. W worku nie ma już żadnych klocków.

Strategia a faza gry

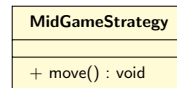
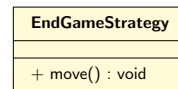
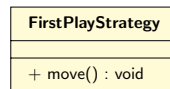
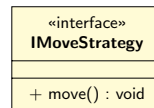
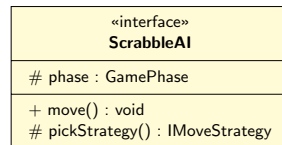


Rysunek : Zmiana strategii wraz z progresją rozgrywki.

Wzorzec projektowy - strategia

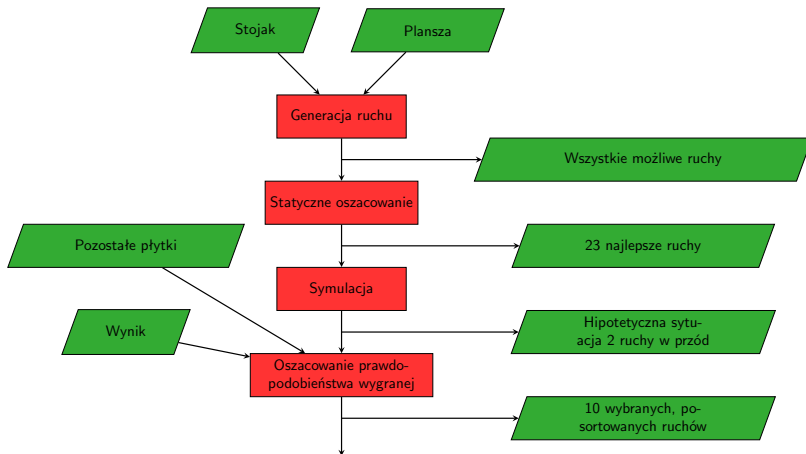
Strategia

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 punktowanego 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).



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 zagranii P_3 liczbę punktów przeciwnika po zagranii P_2 .
5. Gracz dodaje do PV_1 *leave value* po zagranii 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.

Prawdopodobieństwo występowania bigramów

N-gram

Sekwencja składająca się z n liter, znaków lub wyrazów.

- ▶ Unigram.
- ▶ Bigram.
- ▶ Trigram.
- ▶ 4-gram.
- ▶ ...
- ▶ N-gram.

Bigram	Wystąpienia
ni	1 077 436
ie	1 028 249
ow	645 018
an	507 205
wa	484 295
za	313 370
po	301 636
ch	296 749
ał	294 734
ia	284 247

Prawdopodobieństwo występowania n-gramów

Trigram	Wystąpienia
nie	635 196
owa	307 277
ani	195 186
wan	180 460
cie	148 513
nia	142 201
jąc	131 792
prz	130 283
wał	126 134
rze	116 370

4-gram	Wystąpienia
owan	127 626
ował	88 130
wani	78 095
niep	77 449
prze	73 230
ując	67 062
ania	61 398
ając	59 499
ście	56 462
łaby	55 380

5-gram	Wystąpienia
owani	54 991
niepo	40 329
ałaby	37 581
yście	33 161
owała	28 175
niewy	26 193
owane	25 555
wania	25 551
owany	25 542
ałyby	25 282

Prawdopodobieństwo występowania n-gramów (2)

6-gram	Wystąpienia
owania	17 609
wałaby	17 161
byście	16 821
liście	15 910
aniami	15 585
aniach	15 585
łyście	15 405
owanie	14 674
owałby	14 437
nieprz	14 328

7-gram	Wystąpienia
owałaby	12 401
libyśmy	12 267
łybyśmy	12 094
ałyście	9 902
aliście	9 304
ibyście	8 445
libyści	8 443
ybyście	8 317
łybyści	8 314
nieprze	8 111

Najlepsze otwarcia

		x2								x2			
P	Ó	Ż	N	O	Ś	Ć				x2			126 punktów
		x2				B	Ł	Ą	D	Ż	Ż	E	124 punkty
		x2				U	B	O	D	Ż	Ż	E	124 punkty
		x2				P	Ó	J	D	Ż	K	Ę	124 punkty
		x2				G	Ł	Ó	D	Ż	Ż	E	124 punkty
		x2				U	B	Ą	D	Ż	Ż	E	124 punkty
		x2				U	G	Ó	D	Ż	Ż	E	124 punkty
		x2				B	L	U	Ż	Ń	Ż	E	124 punkty
		x2				P	Ó	J	D	Ż	K	Ą	124 punkty
		x2			U	G	R	Z	Ą	Ż	Ć		124 punkty

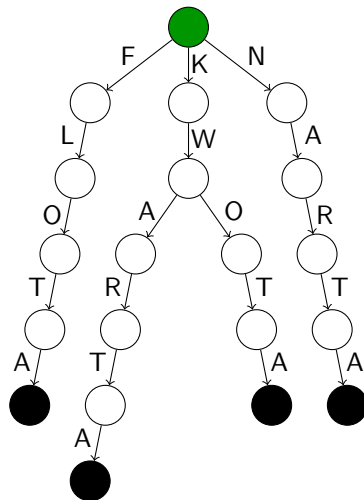
Najlepsze kombinacje liter

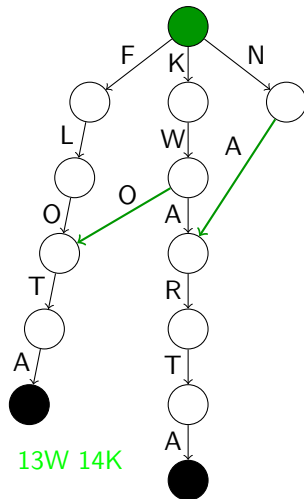
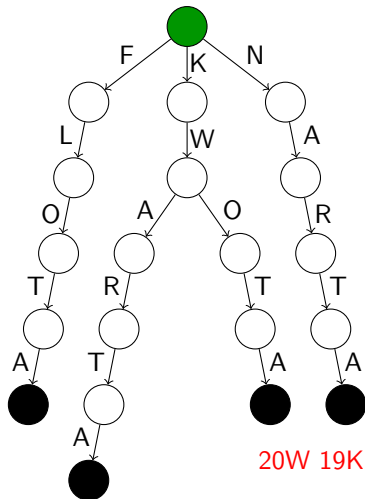
Najlepsze kombinacje liter to zawartość stojaka, która umożliwia ułożenie (niezależnie) jak największej ilości słów.

6 liter	Kombinacje
e, m, n, o, r, t	10 wyrazów
a, i, k, l, n, o	10 wyrazów
a, e, i, l, m, n	10 wyrazów
e, i, k, m, o, s	9 wyrazów
a, i, k, m, n, o	9 wyrazów
a, i, l, m, o, s	9 wyrazów
a, i, k, o, t, w	9 wyrazów
a, i, k, n, t, u	9 wyrazów
a, e, k, l, s, z	9 wyrazów
a, e, i, k, m, r	9 wyrazów

7 liter	Kombinacje
e, i, k, l, n, o, w	12 wyrazów
a, e, i, k, p, r, s	12 wyrazów
a, e, i, k, l, n, p	12 wyrazów
a, e, k, n, r, t, y	11 wyrazów
a, i, k, m, o, p, s	11 wyrazów
a, i, k, m, o, r, w	11 wyrazów
a, a, i, k, l, m, s	10 wyrazów
a, i, k, m, o, s, t	10 wyrazów
a, i, k, l, n, o, w	10 wyrazów
a, i, k, l, m, n, o	10 wyrazów

- ▶ Algorytm opisany w pracy *The World's Fastest Scrabble Program* A. W. Appela i G. J. Jacobsona.
- ▶ Algorytm z nawrotami.
- ▶ Bazuje na skompresowanej, grafowej odmianie drzewa *trie* o nazwie **DAWG** (ang. **D**irected **A**cyclic **W**ord **G**raph).





Algorytm Appela-Jacobsona (1)

1. Redukcja złożoności problemu do jednego wymiaru:

- ▶ rozpatrywanie ruchów wyłącznie poziomo,
- ▶ ograniczenie zbioru wyłącznie do jednego wiersza.

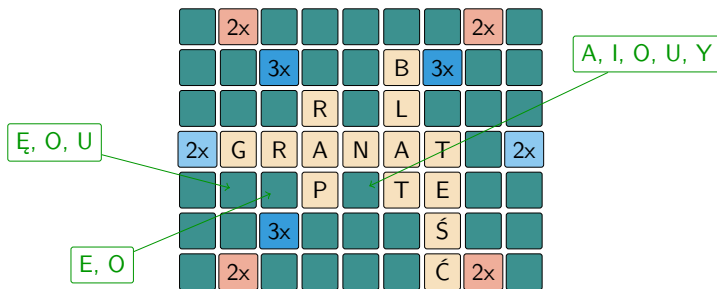
Rozumowanie należy powtórzyć dla wszystkich wierszy, a następnie transponować planszę i zastosować do ruchów w pionie.



Algorytm Appela-Jacobsona (2)

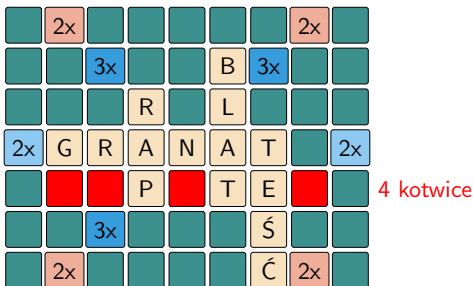
2. Ograniczenie zbioru znaków możliwych do wstawienia w miejsce pustych płytek:

- ▶ ruch w danym kierunku może skutkować tworzeniem nowych słów w kierunku przeciwnym,
- ▶ słowa utworzone w kierunku przeciwnym powstają zawsze poprzez dodanie jednego znaku.



3. Wyznaczenie kotwic (ang. *anchors*):

- ▶ kotwica to najbardziej wysunięta na lewo płytką nowego słowa, która jest przyległa do płytki istniejącego już na planszy słowa,
- ▶ kotwicą może być każde puste miejsce przyległe do płytki znajdującej się na planszy.



Algorytm Appela-Jacobsona (4)

4. Rozwinięcie słów, wychodząc od wyznaczonych kotwic, z uwzględnieniem ograniczeń.

Lewa strona

- ▶ Obejmuje wszystkie płytki na lewo od kotwicy.
- ▶ Może:
 - ▶ Składać się wyłącznie z płytek już znajdujących się na planszy - przypadek trywialny.
 - ▶ Składać się wyłącznie z płytek znajdujących się na stojaku. Wymaga wyznaczenia wszystkich możliwych kombinacji płytek.
 - ▶ Być pusta.

Prawa strona

- ▶ Obejmuje kotwicę oraz wszystkie płytki na prawo od niej.
- ▶ Wyznaczana poprzez dopełnianie lewej strony wyrazami ze słownika.
- ▶ Poszczególne litery muszą być dostępne na stojaku, a także spełniać ograniczenia wyznaczone dla poszczególnych pól planszy.

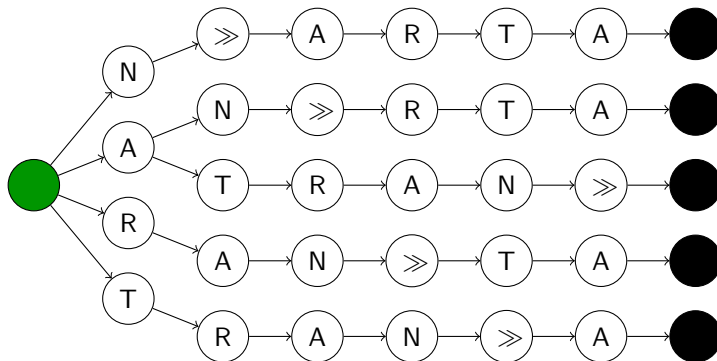
Algorytm Appela-Jacobsona - wydajność

Potencjalnym problemem wydajnościowym jest wyznaczanie wszystkich możliwych kombinacji prefiksów:

- ▶ W pesymistycznym przypadku kotwica może być skrajnie prawą płytką wyrazu.
- ▶ Dla określonych liter na stojaku może istnieć do $6! = 720$ lewostronnych kombinacji do zbadania.
- ▶ W przypadku, gdy na stojaku znajdują się dwa *blanki*, liczba kombinacji rośnie do $\frac{4! \times 32^2}{2} = 12288$.
- ▶ Nadmiarowość obliczeń - duża część badanych kombinacji może nie istnieć (lub nie posiadać rozwinięcia) w słowniku.

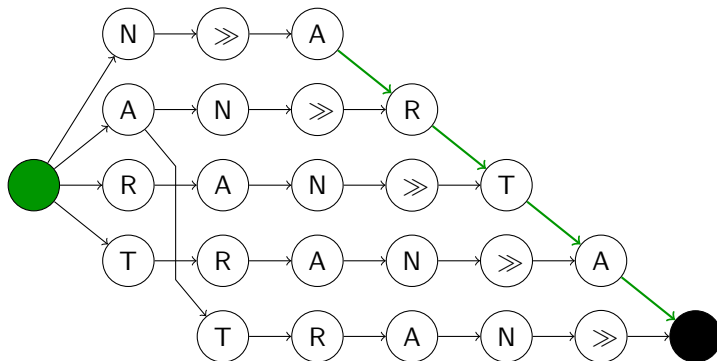
GADDAG

- ▶ S. A. Gordon, *A Faster Scrabble Move Generation Algorithm*.
- ▶ Struktura nastawiona na szybkie prefiksowanie wyrazów.



GADDAG - wady

- ▶ Duża złożoność pamięciowa.
- ▶ Można próbować minimalizować graf po węzłach zawierających \gg .



Maven i Quackle - porównanie

	Maven	Quackle
Autorzy	Brian Sheppard	Jason Katz-Brown, John O'Laughlin
Źródło	Zamknięte	Otwarte (C++, Qt)
Struktura słownika	DAWG	GADDAG
Strategia	Zależna od fazy gry. Wykorzystanie heurystyk i symulacji do ewaluacji najbardziej korzystnych ruchów.	Zależna od fazy gry. Wykorzystanie heurystyk i symulacji do ewaluacji najbardziej korzystnych ruchów.
Wyniki przeciwko ludziom	▶ 9-5 vs Adam Logan (1997) ▶ 6-3 vs Joel Sherman (2006)	▶ 3-2 vs David Boys (2006)
„Bezpośrednie” starcie	▶ 30-6	▶ 32-4

„It's still better to be a human than to be a computer” - David Boys

Fazy gry

1. **MG** - mid-game:

- ▶ Trwa od momentu rozpoczęcia gry, aż do osiągnięcia fazy pre-endgame.

2. **PEG** - pre-endgame.

- ▶ Dzieli się na dwa etapy - PEG-1 oraz PEG-2.
- ▶ Występuje, gdy do pobrania pozostają odpowiednio jedna lub dwie płytki.
- ▶ Przez PEG przechodzi ponad połowa gier.

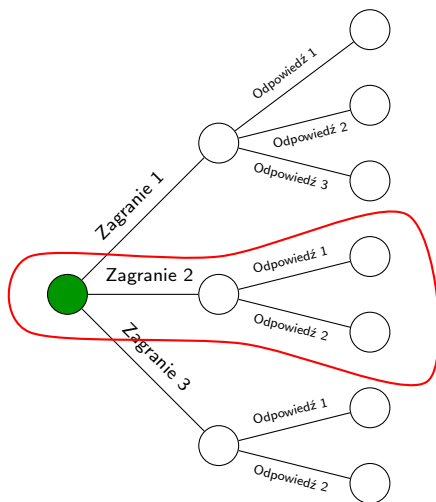
3. **EG** - endgame.

- ▶ Rozpoczyna się, gdy pobrane zostaną wszystkie płytki.
- ▶ Wiadomo jakimi literami dysponuje przeciwnik.

Mid-game (Quackle)

(Pre-)Endgame

- ▶ W fazach PEG, EG możliwe jest zastosowanie wyszukiwania wyczerpującego przestrzeni stanów.
- ▶ Algorytmy przeszukiwania $\alpha - \beta$, A^* , B^* .
- ▶ Obliczenia progresywne - przeszukiwanie rozpoczynane w miejscu, w którym można podjąć szybką i pewną decyzję.



Dziękuję za uwagę!