

PROJEKT **WARCABY**

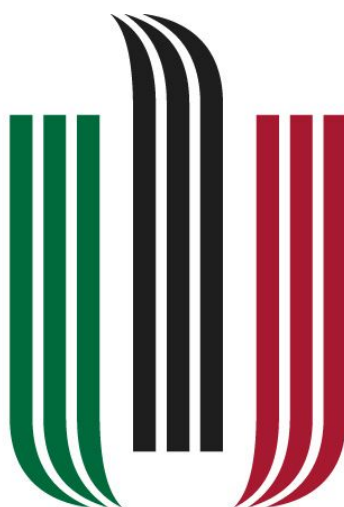
PROGRAMOWANIE WSPÓŁBIEŻNE I ROZPROSZONE

Informatyka - rok III, gr. 3

Oskar Pawica

Piotr Seemann

22.01.2020r.



AGH

**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE**

Spis treści

Opis projektu	3
Cel projektu	3
Wyróżnione metody	3
Instrukcja obsługi	4
Informacje ogólne	4
Rozgrywka	4
Zakończenie gry	4
Możliwe rozszerzenia	5

Opis projektu

Cel projektu

Celem projektu jest zaprezentowanie idei współbieżności przy wykorzystaniu języka funkcyjnego *Erlang*. Aby to osiągnąć, napisano wirtualną wersją gry planszowej *Warcaby*. Aspekt współbieżności realizowany jest przez odpowiadające graczom procesy przekazujące sobie informacje o aktualnym stanie planszy i pionków graczy.

W projekcie użyte są wyłącznie wbudowane moduły *Erlanga*.

Wyróżnione metody

W celu implementacji zasad gry utworzono bardzo wiele metod sprawdzających poszczególne warunki rozgrywki, całość zaś podzielono na trzy moduły: moduł planszy, moduł ruchów i główny moduł gry.

Niemniej jednak wśród składających się na projekt metod możemy wyróżnić następujące:

- *main()* - tworzy procesu obu graczy i rozpoczyna rozgrywkę. Odbiera również sygnał o końcu gry i wyświetla zwycięskiego gracza,
- *player()* - odpowiada za kolejne tury graczy. Pod koniec wywołuję proces oponenta,
- *showBoard(BoardState)* - wyświetla na ekranie aktualny stan rozgrywki,
- *getPlayerMoveIfSquaresAreValid()* - sprawdza wpisane przez gracza instrukcje ruchu pionka i dokonuje ich ewentualnej realizacji po wywołaniu metod sprawdzających warunki ruchu,
- *makeMove([X1, Y1, X2, Y2],BoardState,Color,PieceCount)* - przemieszcza pionek gracza, dokonując ewentualnego zbitia pionka przeciwnika, i nanosi odpowiednie zmiany na aktualną planszę.

Instrukcja obsługi

Informacje ogólne

Program należy uruchomić w środowisku *Erlanga*. Interfejs opiera się na komunikatach wysyłanych przez standardowe wejście.

Aby uruchomić program w środowisku *Erlanga* należy wpisać następujące komendy:

- `c(board)`.
- `c(moves)`.
- `c(checkers)`.
- `checkers:main()`.

Rozgrywka

Gra rozpoczyna się od ruchu białego gracza. Chcąc wykonać swój ruch, należy wpisać komendę ruchu w postaci `[A-H][1-8][A-H][1-8]`, wpisując odpowiednio najpierw współrzędne kafelka, z którego chce się rozpocząć ruch, a następnie kafelka, na który chce się przestawić pionek.

Przykładowo, chcąc przesunąć swój pionek z pola A3 na B4, wpisujemy: `A3B4`.

Zakończenie gry

Rozgrywka kończy się w momencie, gdy jeden z graczy straci swoje pionki bądź nie będzie w stanie wykonać już żadnego ruchu.

```
      A   B   C   D   E   F   G   H
-----
8 | "  " | "BP" | "  " | "BP" | "  " | "BP" | "  " | "BP" |
-----
7 | "BP" | "  " | "BP" | "  " | "BP" | "  " | "BP" | "  " |
-----
6 | "  " | "WP" | "  " | "  " | "  " | "BP" | "  " | "  " |
-----
5 | "  " | "  " | "BP" | "  " | "WP" | "  " | "  " | "  " |
-----
4 | "  " | "WP" | "  " | "  " | "  " | "BP" | "  " | "  " |
-----
3 | "  " | "  " | "  " | "  " | "  " | "  " | "WP" | "  " |
-----
2 | "  " | "WP" | "  " | "WP" | "  " | "WP" | "  " | "WP" |
-----
1 | "WP" | "  " | "WP" | "  " | "WP" | "  " | "WP" | "  " |
-----
black to move
Enter your move [format [A-H][1-8][A-H][1-8]] : 
```

Rys 1: Przykładowy zrzut ekranu trwającej rozgrywki.

Możliwe rozszerzenia

Dla urozmaicenia rozgrywki do gry można dodać tryb jednoosobowy, gdzie gracz mierzyłby swoje siły ze sztuczną inteligencją. Poza tym nie przewidziano dalszych rozszerzeń, które nie wykraczałyby poza ramy zasad podstawowej wersji gry.