

Dokumentacja do projektu AgentChess

Michał Bobowski, Zofia Abramowska, Jakub Meller

9 kwietnia 2014

1 Założenia wstępne

Niniejszy projekt jest realizowany w ramach przedmiotu SAG. Jego przedmiotem jest stworzenie prostego, bazującego na agentowym podejściu, modelu sztucznej inteligencji służącej do gry w szachy. Silnik gry i warstwa graficzna zostaną oparte na istniejących rozwiązaniach. Całość zostanie zrealizowana w języku programowania Scala.

1.1 Figury

Dla uporządkowania wiedzy dziedzinowej posługujemy się nazwami figur w wersji polskiej i angielskiej. Są to kolejno:

- Pion (ang. Pawn) - porusza się o jedno pole naprzód (ew. o dwa pola na początku), bije tylko po skosie.
- Skoczek (ang. Knight) - porusza się systemem dwa pola naprzód i jedno do boku w dowolnym kierunku.
- Goniec (ang. Bishop) - porusza się po skosie.
- Wieża (ang. Rook) - porusza się w pionie lub poziomie.
- Hetman (ang. Queen) - łączy ruchy wieży i gońca.
- Król (ang. King) - porusza się o jedno pole w dowolnym kierunku.

Na potrzeby projektu każda konkretna figura będzie posiadała unikalny identyfikator, składający się z jednego lub dwóch znaków. Pierwszym znakiem identyfikatora jest pierwsza litera angielskiej nazwy, za wyjątkiem skoczka, który z przyczyn oczywistych przyjmie drugą literę tj. 'n'. Dla rozróżnienia kolorów, figury czarne będą w przeciwieństwie do figur białych używały wielkich liter. Drugi znak identyfikatora jest po prostu liczbą i nie występuje dla króla i hetmana.

2 Opis algorytmu

Główną cechą niniejszego algorytmu jest jego zorientowanie na agentowość. Bezcelowe jest porównywanie go do algorytmów budujących drzewo rozwiązań, gdyż na pewno są one wydajniejsze. Sukcesem projektu będzie stworzenie grywalnego, działającego w rozsądnym czasie programu.

2.1 Architektura agentów

Agenci są umieszczeni w architekturze wertykalnej. Głównym mózgiem przedsięwzięcia jest super-agent, którego zdaniem jest wysyłanie żądań do agentów podrzędnych i przeprowadzanie wnioskowania dotyczącego wyboru ruchu.

Na niższym poziomie w hierarchii znajdują się agenci reprezentujący figury obsługiwane przez komputer. Można rozważyć również stworzenie agentów dla figur obsługiwanych przez gracza, ale nie wydaje się to w tym momencie zasadne. W kolejnych częściach opisany jest szczegółowy protokół komunikacji między agentami.

2.2 Struktury pomocnicze algorytmu

W tej sekcji zawarty jest opis pewnych abstrakcyjnych bytów, na podstawie których przeprowadzane jest wnioskowanie.

2.2.1 Ocena heurystyczna

Każda z figur biorących udział w grze posiada heurystyczną ocenę swojej przydatności, oznaczaną dalej jako H . Nie czuję się specjalistą, ale wikipedia proponuje następującą wycenę:

- $H=1$ dla piona.
- $H=3$ dla skoczka i gońca.
- $H=5$ dla wieży.
- $H=9$ dla hetmana.

Do kompletu dodany zostaje król z wyceną równą np. $H=20$.

2.2.2 Mapa ruchu wroga

Mapa ruchu wroga ma za zadanie pokazać w łatwo dostępnej formie, gdzie i jakimi figurami może ruszyć się przeciwnik w pewnej konkretnej sytuacji na planszy. Zakładam, że bicie jest traktowane w tym przypadku jako normalny ruch. Od strony programistycznej mapa ruchu wroga jest mapą, w której kluczem jest pole planszy, a wartością lista identyfikatorów figur, które mogą wykonać ruch na dane pole.

2.2.3 Mapa wsparcia wroga

Wprowadzam relację wsparcia zdefiniowaną w następujący sposób: figura A wspiera figurę B, jeśli w przypadku zbicia B, A może w kolejnym ruchu dokonać bicia na polu zajmowanym wcześniej przez B. Mapa wsparcia ma za zadanie zagregować informacje dotyczące wzajemnego wsparcia figur wroga. Kluczem mapy jest identyfikator figury wspieranej, a wartością lista identyfikatorów figur wspierających.