

Metody SI w grach komputerowych

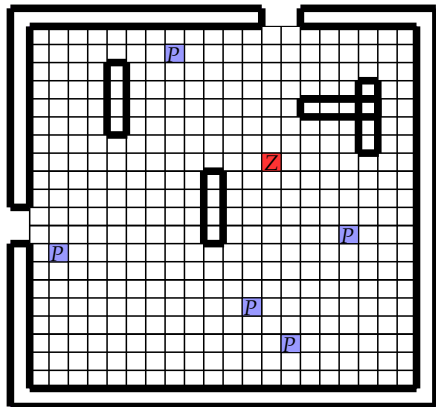
Gra „Policjanci i złodziej”

(Algorytmy przeszukiwania grafów)

Przemysław Kłesk
pklesk@wi.zut.edu.pl

Katedra Metod Sztucznej Inteligencji i Matematyki Stosowanej

„Policjanci i złodziej” — ilustracja gry



Na dwuwymiarowej planszy z kwadratową siatką pól mamy 1 złodzieja i 5 policjantów. Zadaniem policjantów jest złapać złodzieja. Zadaniem złodzieja jest jak najdłużej pozostać niezłapanym lub uciec poza planszę.

Szczegóły (1)

- 1 Plansza w ogólności $n \times n$. Domyślnie 20×20 .
- 2 Możliwość poruszania się tylko w 4 kierunkach lub braku ruchu: $\{\leftarrow, \rightarrow, \uparrow, \downarrow, \circ\}$.
- 3 Liczba policjantów może być parametrem. Domyślnie 5 policjantów.
- 4 W brzegach planszy znajdują się furtki. Parametry: *liczba furtek* (domyślnie 2) i *szerokość furtki* (domyślnie 2).
- 5 Trafienie złodzieja na dowolne pole furtki oznacza jego ucieczkę poza planszę (zwycięstwo złodzieja).
- 6 Na planszy znajdują się przeszkody prostokątne — ściany. Parametry: *liczba ścian* (domyślnie 4), *długość ściany* (domyślnie 4). Każda ściana ma wymiary $1 \times \text{długość ściany}$. Przeszkoda może być skierowana poziomo lub pionowo.
- 7 Zarówno furtki jak i przeszkody mogą poruszać się. Dla wszystkich furtek obowiązują 2 parametry: *prawdopodobieństwo ruchu* P_{FR} , *prawdopodobieństwo zmiany kierunku ruchu* P_{FZ} . Możliwe ruchy furtki: zgodny lub przeciwny do ruchu wskazówek zegara. Domyślnie: $P_{FR} = 0.5$, $P_{FZ} = 0.01$. Analogiczne parametry dla przeszkód, wynoszą domyślnie $P_{SR} = 0.75$, $P_{SZ} = 0.05$, przy czym ściany mogą poruszać się we wszystkich 4 kierunkach. Częstość zmian można kojarzyć z prędkościami.
- 8 Wykonywana będzie pętla $t = 0, 1, 2, \dots, T - 1$ (dyskretny zegar). Można myśleć, że ruchy wykonywane są momentach przeskoków zegara.
- 9 Gracze muszą zaplanować k swoich ruchów, co k chwil zegarowych. Domyślnie $k = 5$. Obliczenia planujące ruchy muszą być wykonane w limicie czasowym $\Delta = 500 \text{ ms}$.

Szczegóły (2)

- 1 Ściany są dla siebie przezroczyste — mogą nachodzić na siebie (przenikać się).
- 2 Furtki są dla siebie przezroczyste.
- 3 Ściany nie zmieniają swojej orientacji (pozioma, pionowa) w trakcie gry.
- 4 Ściany nie są przezroczyste dla graczy. Jeżeli jako decyzję gracz wybierze ruch, który skieruje go na pole zajęte przez ścianę, to ruch nie zostanie wykonany (tożsame z wykonaniem ruchu pustego ◦).
- 5 Ściany odbijają się od brzegów planszy.
- 6 Ruchy wykonywane są „jednocześnie” przez wszystkie obiekty (gracze, furtki, ściany) wraz z przeskokami dyskretnego zegara, ale pierwszeństwo zajmowania pól mają ściany i furtki.
- 7 Policjanci nie mogą wchodzić na pola furtki (czychać w furtce).
- 8 W rogach planszy, furtki o szerokości 2 pozwalają przedostać się tylko przez 1 swoje pole.
- 9 Złodziej jest uznany za złapanego jeżeli jego pozycja pokryje się z pozycją dowolnego policjanta.

Punktacja — wypłaty w grze

Wypłaty

- 1 Wypłata policjantów równa jest ujemnej wypłacie złodzieja.
- 2 Złodziej złapany w chwili t dostaje t punktów.
- 3 W szczególności: jeżeli pętla się wykona i złodziej nie zostanie złapany, to złodziej dostaje T punktów.
- 4 Jeżeli złodziej ucieka furtką w chwili t to otrzymuje $2T - t - 1$ punktów. Ucieczka w ostatnim kroku (gdy $t = T - 1$) jest równowarta przetrwaniu.

Środowisko — silnik wykonujący grę

Algorytm

- ❶ **Inicjalizacja.** Wylosuj bezkonfliktowo pozycje początkowe ścian i graczy.
- ❷ **Instancjonowanie graczy.** Utwórz w pamięci obiekty dla złodzieja i policjantów.
- ❸ **Pierwsze wyświetlenie.** Wyświetl początkowy stan gry na ekran.
- ❹ **Główna pętla.** Dla $t = 0, 1, \dots, T - 1$ wykonuj:
 - ❶ **Przesunięcia furtek i ścian.** Dla każdej furki i ściany sprawdź, czy zaszło prawdopodobieństwo zmiany kierunku i/lub ruchu. Wykonaj odpowiednie ruchy. Zaniechaj ruchu ściany, jeżeli ta wchodzi na starą pozycję gracza.
 - ❷ **Przesunięcia graczy.**
 - ❶ Jeżeli $t \% k = 0$, to: dla każdego gracza utwórz wątek opakowujący obiekt gracza; uruchom wątek z limitem czasu Δ ; w wątku wykonaj na obiekcie metodę liczącą plany k ruchów; gdy upłynie czas Δ , odbierz plany.
 - ❷ Przesuń graczy zgodnie z zaplanowanym ruchem. Jeżeli nowa pozycja gracza pokrywa się ze ścianą, to zaniechaj ruchu.
 - ❸ **Wyświetlenie.** Wyświetl stan gry na ekran.
 - ❹ **Sprawdzenie złapania.** Jeżeli pozycja złodzieja pokrywa się z pozycją dowolnego policjanta, to przerwij pętlę — złodziej jest złapany.
 - ❺ **Sprawdzenie ucieczki.** Jeżeli pozycja złodzieja pokrywa się z pozycją furki, to przerwij pętlę — ucieczka złodzieja.

Świadomość graczy

- Gracze są świadomi nastaw wszystkich parametrów gry. Otrzymują je w momencie konstrukcji i mogą dokonać dowolnych obliczeń oraz inicjalizacji pomocniczych struktur danych bez limitu czasu.
- Gracze są świadomi stanów planszy w poszczególnych chwilach t . Informację o k ostatnich stanach gracze otrzymują od silnika środowiska w momencie tworzenia wątku opakowującego, tuż przed wykonaniem metody liczącej plany.
- Gracze we własnym zakresie mają prawo przechowywać historię stanów (np. żeby zaobserwować aktualny kierunek ruchów furtek/ścian), przy czym muszą to robić w ramach limitów czasu przeznaczonych na ruch.

Sugestie rozwiązania (SI)

Heurystyczny zmienny cel + algorytm A^*

- W każdym kroku wykonywać A^* wyznaczając trasę do pewnego docelowego pola. Pole powinno być wybrane w każdym kroku wg opracowanej przez siebie heurystyki (nie mylić z heurystyką w ramach A^*).
- Złodziej powinien ocenić, czy atrakcyjniejszym w danej chwili polem docelowym jest furтка czy miejsce odległe od policjantów.
- Policjanci mogą ocenić czy należy zastawić furtkę czy osaczać złodzieja.
- Dobre rozwiązanie powinno też próbować przewidywać/odgadywać plany przeciwnika (jego docelowe pola).

Sugestie rozwiązania (SI)

- **Nagradzanie policjantów za lokalne rozproszenie.** W przypadku osaczania złodzieja, warto, aby policjanci „celowali w różne pola”. Można też rozważyć zaburzanie ich ścieżek, gdy idą z podobnego kierunku, tak aby ścieżki nie pokrywały się kawałkami.
- **Unikanie policjantów.**
 - Złodziej może starać się wyznaczać trasy omijające policjantów, poprzez potraktowanie ich pozycji jako przeszkód wraz z pewną otoczką (dobór promienia otoczki).
 - Otoczkę wokół policjanta można potraktować jako *miękką przeszkodę*, zezwalając na przejście przez nią, ale większym kosztem (składnik g w A^*). Można stopniować koszty w otoczce, obniżając w kierunku rosnącego promienia.

Sugestie rozwiązania (SI)

- **Cudze A^* .** Gracze w ramach limitu czasu mogą wykonywać dodatkowe A^* dla hipotetycznych tras przeciwnika (odgadując ich cel) i np. patrzeć, czy trasy te mogą przeciąć lub przebiec blisko własnej.
- **Szybkie struktury danych.** W ramach algorytmu A^* warto zbiór *Open* oprócz kolejki priorytetowej przechować dodatkowo w formie mapy haszującej — szybkie sprawdzenie czy stan jest w zbiorze *Open* — $O(1)$ i ewentualne usunięcie. Niestety włożenie po aktualizacji znów $O(\log_2 n)$.