

TheGame: Era botów

Zadania kwalifikacyjne

1. Kim Ty jesteś? (2p. ☺)

Napisz kilka zdań o sobie. Nie oczekuję CV czy życiorysu, ale krótkiej wypowiedź o Twoich zainteresowaniach (nie tylko tych związanych bezpośrednio z tematyką zajęć) i dlaczego chcesz być na moich warsztatach. Chętnie też przeczytam, czego się po nich spodziewasz i w jakiej formie byś je najchętniej widział(a).

2. Programowanie (10p.)

Pod [linkiem](#) znajduje się kod nowej wersji zadania programistycznego. Wydaje mi się, że jest sporo trudniejsze niż poprzednie. Uznałem, że tak powinno być, bo strasznie dużo się Was zgłosiło. Nie będę brał pod uwagę rozwiązań poprzedniej wersji tego zadania, bo bardzo wcześnie padła.

Scenariusz: Utknąłeś w mieście podczas nalotu bombowego (np. takiego jak na [Drezno](#))

Zadanie: przeżyć jak najdłużej.

Na samym początku serwer nasłuchuje na porcie **31415** i po połączeniu wysyła wymiary mapy (możliwe że w wersji na której będę testował będą zmienne) w postaci: **MAP:11x11**

Jest to format **SZEROKOŚĆxWYSOKOŚĆ**

Początek rundy oznajmia: **ROUND:X** gdzie **X** to numer rundy. Na każdą rundę składa się jakaś liczba kroków.

Każdy kolejny krok serwer oznajmia przez: **STEP:X** gdzie **X** to numer kroku.

Następnie wysyła stan planszy w formacie [JSON](#). Są tam podane współrzędne x i y każdego miejsca poprzedniego zrzucenia bomby jak i położenie gracza.

Założenie jest takie, że jeżeli samolot zrzucił bombę na jakieś pole to w kroku następnym może zrzucić tylko na pole sąsiadujące (lub to samo), czyli ma 9 pól do wyboru – wybiera losowo.

W odpowiedzi należy odesłać serwerowi ruch gracza także w formacie JSON jako słownik:

{'x': dx, 'y': dy} gdzie dx i dy to przesunięcie gracza w następnym kroku. Wartość ta nie może być większa od 2 ani mniejsza od -2 (domyślnie 2 – przy sprawdzaniu ta wartość może ulec zmianie). Plansza jest zapętłona: wychodząc z lewej pojawia się z prawej. Tak samo góra-dół.

Masz przetrwać jak największą liczbę pełnych rund. Z każdą rundą zwiększa się liczba miejsc, w które będą spadały bomby (początkowo 5). Z każdą rundą cała mapa generowana jest ponownie. Początkowe ustawienie gracza i miejsc zrzucenia bomb jest losowe, ale gracz nie może zacząć martwy.

Z uwagi na otwartość (i delikatną losowość) problemu będę puszczał Wasze skrypty u siebie kilkanaście razy i brał średnią. Oceny przyznam w zależności od wyników: najlepszy program dostanie 10 punktów, a reszta odpowiednio (niekoniecznie liniowo) przeskalowane. Jako rozwiązanie prześlij (jako oddzielny plik) w pełni funkcjonalny skrypt Pyhona, który będę mógł u siebie odpalić.

3. Geometria analityczna/pseudofizyka (10p)

- Mając punkt $P = (P_x, P_y)$ i prostą o równaniu $ax + by + c = 0$ wyprowadź wzór na odległość tego punktu od prostej. (2p.)
- Ciało ma stałą prędkość $\vec{v} = (v_x, v_y)$ i znajduje się w początku układu współrzędnych. Pod jakim kątem α należy ustawić armatę, znajdującą się w punkcie $(L, 0)$, wystrzelującą pocisk z prędkością u , (strzela w tym momencie, w którym ciało rusza ze swojego punktu startowego) tak by trafić w to ciało? Możesz dać odpowiedź w formie funkcji trygonometrycznej (np. $\cos \alpha$, $\tan \alpha$ itp.). Pomiń grawitację i inne tego typu realistyczne rzeczy. (3p.)
- Masz punkt $P = (P_x, P_y)$ i trzy wierzchołki trójkąta A, B, C (oraz ich odpowiednie współrzędne). Znajdź jak najwygodniejszy sposób by określić czy ten punkt znajduje się wewnątrz tego trójkąta. (5p.)

4. Algorytmika (10p.)

Masz prostą grę 2D, w której poruszają się 2 punkty. Maksymalna składowa prędkość w kierunku osi x lub y wynosi V (czyli maksymalna prędkość wypadkowa to $V\sqrt{2}$).

W każdej rundzie:

- Dostajesz współrzędne drugiego punktu (swoje znasz)
- Wydajesz komendę by Twój punkt zmienił (o jedną jednostkę) prędkości składowe w następnej rundzie - np. zwiększył w kierunku x i zmniejszył w y

Założmy, że wyliczasz na bieżąco (a więc znasz) prędkości swoje i przeciwnika. W formie pseudokodu przedstaw algorytm pozwalający na śledzenie drugiego punktu (on zmienia swoją prędkość losowo). Plansza jest nieograniczona w każdym kierunku. Drugi punkt porusza się na tych samych zasadach, co Twój (może zmieniać prędkości o jednostkę). Zaczynacie w losowej odległości z losowymi prędkościami. Jako śledzenie rozumiem dążenie do utrzymania jak najmniejszej odległości między tymi dwoma punktami. Poza pseudokodem mile widziane jest wyjaśnienie, dlaczego to działa 😊

To jest delikatnie otwarte zadanie i jestem ciekaw, jakie przedstawicie pomysły na doganianie takiego przeciwnika.

5. Podsumowanie

W razie jakichkolwiek problemów pisz na kalmar2718@gmail.com

Jeżeli nie uda Ci się zrobić jakiegoś zadania do końca to wyślij tyle, ile zrobił(e/a)ś.

Rozwiązania przyjmuję raczej tylko w formie pliku *.pdf* poza specjalnymi przypadkami – jeżeli nie możesz tego dać w tej formie to się ze mną skontaktuj.