

Napisz program w języku C lub C++ który będzie uczestnikiem gry reprezentującym gracza.

SPIS TREŚCI:

Zasady gry.....	1
Uczestnicy gry:.....	2
Mediator.....	2
Program gracza.....	2
Plik mapa.txt.....	2
Plik status.txt.....	3
Format linii określającej jednostkę na planszy:.....	3
Jednostki:.....	4
Plik rozkazy.txt.....	4
Rozkazy w zależności od jednostki oraz akcji i parametry.....	4
Zasady ruchu oraz potyczki.....	5
Ograniczenia dotyczące ruchu jednostek:.....	5
Tabela ataku:.....	6
Informacje dodatkowe.....	7

ZASADY GRY

Każdy program gracza ma swoją bazę zdolną do produkowania jednostek, celem jest zniszczenie bazy przeciwnika. Maksymalny czas trwania gry to 2000 tur (1000 tur dla każdego programu gracza),

Po przekroczeniu maksymalnego czasu gry jeżeli nie uda się zniszczyć bazy, zwyciężą będzie program mający najwięcej jednostek na planszy, w przypadku tej samej liczby jednostek u obu graczy będzie remis.

Rozgrywka toczy się na planszy o rozmiarach określonych w pliku mapa.txt (**np** 32x32 kratki). Każda jednostka, budynek, przeszkoda zajmuje dokładnie 1 kratkę. Wygląd planszy określony będzie w pliku tekstowym podawanym na wejście programów graczy.

Mediator jest programem który uruchamia programy graczy z lini poleceń w sposób następujący:

```
./<nazwa programu gracza> mapa.txt status.txt rozkazy.txt [limit_czasowy]
```

gdzie:

- mapa.txt: plik tekstowy z mapą planszy
- status.txt – plik tekstowy ze statusem i położeniem poszczególnych jednostek
- rozkazy.txt – plik do którego program gracza zobowiązany jest zapisać swoje dane wynikowe (rozkazy dla poszczególnych jednostek)
- limit_czasowy: maksymalny czas określony w sekundach na ruch gracza, przekroczenie tego czasu przez program gracza kończy grę, limit czasowy jest parametrem opcjonalnym, brak tego parametru oznacza maksymalny czas 5 sekund.

UCZESTNICY GRY:

- Mediator
- Program reprezentujący gracza nr 1
- Program reprezentujący gracza nr 2

MEDIATOR

Zadania mediatora:

- wywoływanie programów graczy z odpowiednimi parametrami
- wczytywanie wygenerowanych przez programy graczy danych
- sprawdzenie poprawności danych przygotowanych przez programy graczy
- aktualizacja stanu gry
- określenie wyniku rozgrywki

PROGRAM GRACZA

Zadania programu gracza:

- Na podstawie wczytanych informacji zaplanowanie akcji dla swoich jednostek i przekazanie ich w pliku wynikowym „rozkazy.txt”.
- Zapisanie pliku „rozkazy.txt” w zadanym limicie czasowym
- Zakończenie działania natychmiast po zapisaniu pliku „rozkazy.txt”
- Zapisanie pliku „rozkazy.txt” zawierające tylko poprawne akcje
- Program gracza może zapisać dane w prywatnym pliku o nazwie takiej samej jak nazwa programu + rozszerzenie .txt, plik nie może przekraczać maksymalnego rozmiaru 10MB.
- Program nie może komunikować się z otoczeniem w sposób inny niż wczytywanie/zapisywanie zdefiniowanych plików.

PLIK MAPA.TXT

Format pliku składa się z wierszy o określonej liczbie (parametr Y) oraz określonej liczbie znaków char (parametr X).

Każda linia składa się z tej samej liczby znaków o ilości X.

Parametry X oraz Y określają rozmiar mapy.

Znaki char są ograniczone do podanych w poniższej tabelce:

0	Droga po której mogą przemieszczać się jednostki
9	Przeszkoda, jednostki nie mogą zatrzymać się na tym polu, w przypadku wystarczającej ilości punktów ruchu przeszkody można przeskakiwać
1	Baza pierwszego programu gracza
2	Baza drugiego programu gracza
6	kopalnia

[illegible]

PLIK STATUS.TXT

Linia określająca ilość złota (liczba typu long)
Linia określająca jednostkę na planszy (wg formatu opisanego niżej)
...
Linia określająca jednostkę na planszy (wg formatu opisanego niżej)

FORMAT LINII OKREŚLAJĄCEJ JEDNOSTKĘ NA PLANSZY:

Przynależność jednostki	Typ jednostki	ID jednostki	Współrzędna x jednostki	Współrzędna y jednostki	Wytrzymałość	Jednostka produkowana przez bazę
(czy należy do nas czy do przeciwnika)	(K/S/A/P/C/R/W/B)	0-1000				(K/S/A/P/C/R/W)
P – należy do nas	Patrz typy jednostek niżej					Lub 0 gdy nic nie produkuje.
E – należy do przeciwnika						Parametr obowiązuje tylko dla bazy

Przykład pliku wraz z komentarzem:

2500 // ilość złota

P B 10 0 0 98 A // baza gracza w pozycji 0,0, wytrzymałość 98, w trakcie produkowania jednostki archer

E B 12 31 31 100 0 // baza przeciwnika w pozycji 31,31, wytrzymałość 100 jednostek niczego nie produkuje

P K 13 20 10 5 // Knight gracza na pozycji 20,10, wytrzymałość 5 jednostek

E S 16 12 30 1 //Swordsman przeciwnika na pozycji 12,30, wytrzymałość 1 jednostka

JEDNOSTKI:

Typ jednostki	Wytrzymałość	Prędkość	Koszt zakupu	Zasięg ataku	Czas budowania
Knight „K”	70	5	400	1	5
Swordsman „S”	60	2	250	1	3
Archer „A”	40	2	250	5	3
Pikeman „P”	50	2	200	2	3
Ram „R”	90	2	500	1	4
Catapult „C”	50	2	800	7	6
Worker „W”	20	2	100	1	2
Base „B”	200	0	-	-	-

PLIK ROZKAZY.TXT

Plik składa się z linii wg schematu poniżej

Id jednostki	Akcja	Parametry
...		
Id jednostki	Akcja	Parametry

ROZKAZY W ZALEŻNOŚCI OD JEDNOSTKI ORAZ AKCJI I PARAMETRY

Rozkaz dla jednostki:	Możliwa akcja	Parametry	Przykład
K/S/A/P/C/R/W	M Ruch	X Y Współrzędne	12 M 10 10 Ruch jednostki o id=12 na pole (10,10)
K/S/A/P/C/R/W	A Atak	ID ID jednostki atakowanej	13 A 25 Jednostka 13 atakuje jednostkę 25
B	B Budowanie	K/S/A/P/C/R/W Typ budowanej jednostki	10 B A Nakazuje bazie o id=10 rozpocząć produkcję Archer'a

Ilość rozkazów oraz ich kolejność jest dowolna, poszczególne wiersze pliku wynikowego powinny być oddzielone tylko znakami nowej linii. W przypadku gdy nie chcemy wydawać żadnych rozkazów należy plik pozostawić całkowicie pustym.

W przypadku niezgodności żadanego rozkazu z zasadami lub gdy format pliku wynikowego będzie błędny (nie da się sparsować) program gracza zostaje zdyskwalifikowany

Przykłady nieprawidłowych ruchów (dyskwalifikacja)

- Jednostka o podanym ID nie istnieje
- Niewłaściwa akcja dla danej jednostki
- Współrzędne docelowe niezgodne z punktami ruchu
- W miejscu docelowym obca jednostka lub przeszkoda
- Próba zaatakowania własnej jednostki
- Polecenie budowy jednostki podczas gdy baza jest już zajęta produkcją

ZASADY RUCHU ORAZ POTYCZKI

W przypadku wygranej potyczki (zniszczeniu jednostki wroga) jednostka atakująca pozostaje na swoim miejscu, zniszczona zostaje usunięta.

Jeśli punkty ruchu danej jednostki na to pozwolą może ona podjąć kilka akcji w jednej turze np dla jednostki Knight możliwy jest następujący zapis akcji:

10 M 10 11

10 A 12

Niedopuszczalne jest jednak dokonanie więcej niż jednego ataku w turze.

OGRANICZENIA DOTYCZĄCE RUCHU JEDNOSTEK:

- Jednostka nie może zatrzymać się na polu oznaczającym przeszkodę lub jednostce / bazie przeciwnika
- Jednostka nie może pokonać większego dystansu niż pozwalają jej na to aktualne wartości ruchu
- Jednostka nie może zatrzymać się na polu na którym znajduje się aktualnie jednostka przeciwnika lub jego baza
- Jednostka nie może wyjść poza planszę

Programy graczy rozpoczynają grę na planszy bez jednostek,

Jednostki mogą być produkowane w trakcie gry i po wybudowaniu pojawiają się na polu własnej bazy.

Jednostki mogą stać na tym samym polu, wyjątkiem są przeszkody oraz jednostki i baza przeciwnika.

W przypadku parametru „Zasięg ataku” oraz „Prędkość” odległości w grze liczone są w następujący sposób:

$$D = |x_1 - x_2| + |y_1 - y_2|$$

Gdzie (x_1, y_1) oraz (x_2, y_2) to współrzędne pól między którymi liczona jest odległość.

Układ współrzędnych planszy zaczyna się w jej lewym górnym rogu (0,0).

Atak jest możliwy jedynie gdy wroga jednostka znajduje się w zasięgu. Atak kosztuje tyle jednostek co ruch o jedno pole.

Przykładowo Knight może maksymalnie przesunąć się o 4 pola i wykonać atak w trakcie jednego ruchu.

Siła ataku danej jednostki przeciwko każdej ujęta jest w poniższej tabeli. W trakcie ataku podana wartość odejmowana jest od parametru „Wytrzymałość” jednostki atakowanej. Po osiągnięciu przez jednostkę wytrzymałość = 0 jednostka jest usuwana z planszy.

TABELA ATAKU:

	Knight	Swordsman	Archer	Piekeman	Catapult	Ram	Worker	Base
Knight	35	35	35	35	35	50	35	35
Swordsman	30	30	30	20	20	30	30	30
Archer	15	15	15	15	10	10	15	15
Pikeman	35	15	15	15	15	10	15	10
Catapult	40	40	40	40	40	40	40	50
Ram	10	10	10	10	10	10	10	50
Worker	5	5	5	5	5	5	5	1

Baza uczestnika nie podlega ruchom i nie może atakować jednostek a jej parametr „Wytrzymałość” = 200.

Zniszczenie bazy kończy rozgrywkę.

W bazie może być produkowana jedynie jedna jednostka na raz.

Początkowo do dyspozycji gracza jest 2000 jednostek złota, za które może on budować jednostki. Złoto może być pozyskiwane w trakcie rozgrywki poprzez dojście jednostki „Worker” na pole kopalni (oznaczone na mapie znakiem „6”). Prędkość wydobywania złota przez jednego robotnika wynosi 50 jednostek na turę.

W kopalni może znajdować się dowolna ilość robotników.

Informacje o aktualnym statusie i położeniach wszystkich jednostek na planszy są przekazywane programowi gracza w pliku tekstowym.

Kolejność wpisów w pliku opisującym jednostki jest przypadkowa. Parametr ID jest dla każdej jednostki stały i unikalny w trakcie całej rozgrywki.

Program po przeanalizowaniu mapy, stanów wszystkich jednostek oraz ewentualnie historii gry musi utworzyć wynikowy plik tekstowy pod nazwą podaną w **linii poleceń** oraz zapisać do niego podjęte akcje.

INFORMACJE DODATKOWE

- Należy dołączyć pliki do automatycznego budowania, jak np Makefile lub inny.
- Można korzystać z dowolnych bibliotek ale trzeba dostarczyć opis instalacji, kompilacji oraz przykłady użycia
- Program gracza (i ewentualnie mediatora) należy umieścić na <https://github.com/>, z dostępem publicznym
- Program mediatora nie jest obowiązkowy ale jest zalecany (język dowolny), będzie dodatkowo oceniany jako atut
- Testy jednostkowe są opcjonalne (ale dodatkowo punktowane)
- Program nie może zostać zdyskwalifikowany (przez nieprawidłowy ruch itp) a także powodować crasha, wycieki pamięci itp
- Oceniany będzie styl kodowania, komentarze itp
- Oceniana będzie skuteczność programu (wygrywalność z przeciętnym programem)
- Program będzie sprawdzany pod systemem linux (ubuntu)