



PPHU METEOREX

Jak powszechnie wiadomo, wszechświat składa się z wielu alternatywnych rzeczywistości. Nazywasz się Janusz Mordechaj Bizneski i prowadzisz firmę wydobywczą w kosmosie. Twoim odwiecznym wrogiem są 34 inne firmy z planety Bhall-Que'rr↓ ◊vitch. W garażu swojego dziadka Ryszarda znajdujesz maszynę służącą się do przenoszenia się między rzeczywistościami.

Dzięki wielu miesiącom kreatywnej transwymiarowej księgowości i podkładania świń udało się doprowadzić do tego, że wszystkie rzeczywistości poza 595 wybranymi zostały zniszczone. Co więcej, w każdej z rzeczywistości przetrwały tylko dwie parami różne firmy. Brawo!

Zostań wszechwymiarowym asteroidkowym potentatem!

Opis rozgrywki

Każda rozgrywka toczy się na jednej asteroidzie. Mapa asteroidy to plansza w kształcie prostokąta wypełniona w poziomie i w pionie. Pola planszy są kwadratami. Jeśli R to liczba wierszy, a C to liczba kolumn, gra akceptuje całkowite współrzędne z zakresu $[0, R - 1] \times [0, C - 1]$. Każde pole może być jednego z pięciu typów:

- bez dużych ilości złóż,
- z dużymi złożami aluminium,
- z dużymi złożami węgla,
- z dużymi złożami krzemu,
- z dużymi złożami uranu.

Na każdym polu można postawić jeden z 11 rodzajów budowli:

1. Centrum – dostarcza 10 energii i 2 załogi, nie zużywa energii,
2. Korytarz-skrzyżowanie – łączy budynki, zużywa 1 energii,
3. Korytarz poziomy – łączy budynki, zużywa 1 energii,
4. Korytarz pionowy – łączy budynki, zużywa 1 energii,
5. Kopalnia aluminium – dostarcza aluminium, zużywa 1 załogę i 3 energii,
6. Kopalnia węgla – dostarcza węgiel, zużywa 1 załogę i 3 energii,
7. Kopalnia krzemu – dostarcza krzem, zużywa 1 załogę i 3 energii,
8. Kopalnia uranu – dostarcza uran, zużywa 1 załogę i 4 energii,
9. Elektrownia węglowa – dostarcza 16 energii, zużywa 1 jednostkę węgla co turę, w przypadku braku węgla zużywa uran,
10. Elektrownia słoneczna – dostarcza 10 energii
11. Habitat – dostarcza 3 załogi, zużywa 2 energii.

Koszty budynków zależą od meczu i można je poznać za pomocą odpowiedniego polecenia.

Na start gracz ma pewną ilość zasobów. Można ją poznać za pomocą odpowiedniego polecenia. Zasoby znajdują się w magazynach pierwszej postawionej przez gracza bazy.

Opis ruchów

Rozgrywka jest podzielona na tury. W trakcie tury gracz może łącznie 500 poleceń. Polecenia są wykonywane w kolejności wpływania. Gracz może budować budynki (w tym nowe bazy), przysyłać zasoby między bazami i sprzedawać zasoby.

Budowanie

Na mapie można budować budynki należące do jakiejś bazy. W każdej bazie musi być dokładnie jedno centrum (stawiane przy okazji tworzenia bazy) – w szczególności niedopuszczalne jest połączenie korytarzami dwóch centrów. każdy budynek należący do bazy musi być podłączony do bazy poprzez korytarze i inne budynki tak, żeby dało się z niego dojść do centrum. Budynki nie będące korytarzami traktujemy na potrzeby połączeń jak korytarze-skrzyżowania.

Budynki nie mogą sąsiadować ze sobą bokami. Każdy korytarz sąsiadujący z budynkiem musi być do niego podłączony. Dwa korytarze mogą sąsiadować bokami pod warunkiem, że stykają się pasującymi do siebie bokami (albo tworząc połączenie lub prowadząc dwie równoległe trasy). Nie można budować na polach sąsiadujących bokiem z polami zajętyymi przez innego gracza.

Załogi i energia

Kopalnie działają, jeśli mają niezbędne załogi i energię. Wytwarzana w bazie energia najpierw jest przeznaczana na habitaty i korytarze. Następnie załogi i energia przypisywane są do kopalń zgodnie z kolejnością ich budowy. Jeśli brakuje energii na zasilenie kopalni, jest ona pomijana i podejmuje się próbę uruchomienia kolejnej z listy.

Zasoby

Zasoby można przesyłać za pomocą odpowiedniego polecenia. Zasoby można sprzedawać. Ceny zasobów są dostępne za pomocą odpowiedniego polecenia. Na początku meczu ceny są równe tym podanym w poleceniu, z upływem czasu ceny maleją liniowo, osiągając połowę początkowej ceny.

Opis serwera gry

Gra jest podzielona na rundy – runda odpowiada jednej asteroidzie. W każdej rundzie rozgrywane są jednocześnie mecze między każdą parą graczy. Runda trwa pewną liczbę tur. Punkty są podliczane pod koniec każdej rundy i odpowiadają sumie zarobionej na sprzedaży surowców.

Dokładny opis poleceń

Każde polecenie – jeśli było poprawne – zwraca natychmiast OK. W przeciwnym wypadku zwraca komunikat ERR z numerem błędu i krótkim opisem.

GET_TURNS_LEFT

Zwraca liczbę tur pozostałych do końca rundy. Przykład:
 > GET_TURNS_LEFT
 < OK
 < 28

GET_MAP

Zwraca dwie liczby – **R C**. **R** oznacza liczbę wierszy w mapie, a **C** – liczbę kolumn. Kolejno następuje **R** wierszy długości **C** każdy. Wiersz składa się ze znaków **.**, **A**, **C**, **S**, **U**. Znaki odpowiadają kolejno brakowi złóż, aluminium, węgłowi, krzemowi i uranowi. > GET_MAP
 < OK
 < 3 4
 < .S..
 < AA..
 < ..CU

GET_COSTS

Zwraca 11 wierszy postaci **A C S U**, gdzie kolejne zmienne odpowiadają ilości aluminium, węgla, krzemu i uranu potrzebne do postawienia budynku. Wiersze odpowiadają kolejnością budynkom wymienionym wyżej. Przykład:

```
> GET_COSTS
< OK
< 5 0 3 0
(i 10 podobnych wierszy)
```

GET_RESOURCE_INFO

Zwraca 4 wiersze postaci **S L P**, gdzie **S** to ilość zasobu znajdującego się w pierwszej bazie, **L** to maksymalna pojemność danego zasobu w jednej bazie, a **P** to początkowa cena za jedną jednostkę zasobu. Pierwszy wiersz odpowiada aluminium, drugi węglowi, trzeci krzemowi, a czwarty uranowi. Przykład:

```
> GET_RESOURCE_INFO
< OK
< 20 100 10
< 14 120 15
< 5 50 40
< 6 40 200
```

GET_MY_MATCHES

Zwraca **M** – liczbę meczów. Następnie zwraca wiersz z **M** numerami meczów, w których uczestniczysz. Przykład:

```
> GET_MY_MATCHES
< OK
< 4
< 5 87 2 534
```

GET_MY_BUILDINGS

Przyjmuje jedną liczbę **M** – numer meczu. Zwraca **B** – liczbę budynków należących do gracza w meczu **M**. Następnie zwraca **B** wierszy – każdy postaci **R C B T**, gdzie **R** to numer wiersza, **C** to numer kolumny, **B** to numer bazy, a **T** to rodzaj budynku. Zwraca **M** – liczbę meczów. Następnie zwraca wiersz z **M** numerami meczów, w których uczestniczysz. Przykład:

```
> GET_MY_BUILDINGS 83
< OK
< 2
< 2 1 13 1
< 2 2 13 3
```

GET_ENEMY_BUILDINGS

Przyjmuje jedną liczbę **M** – numer meczu. Działa analogicznie do polecenia GET_MY_BUILDINGS, tylko zwraca budynki przeciwnika w meczu **M**. Przykład:

```
> GET_ENEMY_BUILDINGS 83
< OK
< 2
< 4 1 13 1
< 4 2 13 3
```

GET_MY_RESOURCES

Przyjmuje jedną liczbę **M** – numer meczu. Zwraca **B** – liczbę baz należących do gracza w meczu **M**. Następnie zwraca **B** wierszy postaci **N A C S U**, gdzie **N** to numer bazy, **A** to ilość aluminium w bazie, **C** to ilość węgla w bazie, **S** to ilość krzemu w bazie, a **U** to ilość uranu w bazie. Przykład:

```
> GET_MY_RESOURCES 83
```

```
< OK
< 1
< 13 20 5 43 2
```

GET_POINTS

Przyjmuje jedną liczbę **M** – numer meczu. Zwraca dwie liczby **M E** – odpowiednio liczbę punktów gracza i przeciwnika w meczu **M**. Przykład:

```
> GET_POINTS 83
< OK
< 40 36
```

WAIT

Natychmiast zwraca OK (jak każda inna komenda), po czym zwraca dodatkowe OK. Przykład:

```
> WAIT
< OK
(po pewnym czasie...)
< OK
```

SET_BASE

Przyjmuje cztery argumenty **M R C B** – odpowiednio numer meczu, numer wiersza, numer kolumny, numer bazy, która finansuje budowę nowej bazy. Ostatni argument jest ignorowany w przypadku umieszczania pierwszej bazy. Zwraca jedną liczbę **N** – numer nowo postawionej bazy. Przykład:

```
> SET_BASE 40 3 1 5
< OK
< 12
```

BUILD

Przyjmuje cztery argumenty **M B R C** – odpowiednio numer meczu, typ budynku, numer wiersza, numer kolumny. **Uwaga:** centrum budujemy poleceniem SET_BASE. Przykład:

```
> BUILD 40 1 5 6
< OK
```

TRANSFER

Przyjmuje 7 liczb **M F T A C S U**, gdzie **M** to numer meczu, **F** to numer bazy, z której przesyłamy zasoby, **T** to numer bazy, do której przesyłamy zasoby, **A** to ilość aluminium, którą chcesz przesłać, **C** to ilość węgla, którą chcesz przesłać, **S** to ilość krzemu, którą chcesz przesłać, **U** to ilość uranu, którą chcesz przesłać. Przykład:

```
> TRANSFER 143 2 4 10 0 0 2
< OK
```

SELL

Przyjmuje 6 liczb **M B A C S U**, gdzie **M** to numer meczu, **B** to numer bazy, **A** to ilość aluminium, którą chcesz sprzedać, **C** to ilość węgla, którą chcesz sprzedać, **S** to ilość krzemu, którą chcesz sprzedać, **U** to ilość uranu, którą chcesz sprzedać. Przykład:

```
> SELL 143 2 10 0 0 2
< OK
```

Możliwe błędy

- 898 No such building type – nie ma budynku o podanym typie
- 899 No base with given ID – nie ma bazy o takim numerze
- 900 Tile not owned – to pole nie należy do gracza

- 901 Tile already owned – to pole już jest zabudowane
- 902 Base belongs to someone else – baza o tym numerze należy do innego gracza
- 903 Tile is not connected to any base – wybrane pole nie przylega do żadnej bazy
- 904 No such tile – współrzędne pola są poza dopuszczalnym zakresem
- 905 Tile not empty – pole nie jest już puste
- 906 Tile adjacent to enemy base – pole sąsiaduje z bazą przeciwnika
- 907 The building would connect bases – budynek połączyłby dwie bazy
- 908 Corridors aligned improperly – korytarze nie pasują do siebie
- 909 Tile adjacent to building – wybrane pole sąsiaduje z budynkiem
- 910 Not enough resources – brakuje zasobów do wykonania operacji
- 911 Not enough storage – brak miejsca w magazynie do wykonania operacji
- 912 Invalid amount – podane ilości zasobów są niepoprawne
- 920 Not your match – nie grasz w tym meczu
- 921 No such match – mecz o podanym numerze nie istnieje