

deadline **24**

EDYCJA **2013**



ZASADY I ZESTAW ZADAŃ

Gliwice, 19-20 kwietnia 2013

Spis treści

1. Wstępniak	2
2. Komunikacja z serwerem	3
2.1. Logowanie	3
2.2. Komendy	3
2.3. Konwencje	3
3. Punktacja	4
3.1. Tempo rozgrywki	4
3.2. Punkty rankingowe	4
4. Sytuacje awaryjne	4
5. Zadanie Ognisko	5
5.1. Wprowadzenie	5
5.2. Model rozgrywki	5
5.3. Świat	5
5.4. Żukoskoczki	6
5.4.1. Operacje	6
5.4.2. Tratwy	6
5.5. Początkowy stan rozgrywki	7
5.6. Zasady rywalizacji	7
5.6.1. Znakowanie feromonowe	7
5.6.2. Zabieranie patyków – szczegóły	7
5.6.3. Nadmiar patyków na wyspie	8
5.6.4. Nadmiar zapotrzebowania żukoskoczków	8
5.7. Punktacja	8
5.8. Komendy	9
5.9. Błędy	13
5.10. Serwery	13
5.11. Przykład	14
6. Zadanie Sieć	15
6.1. Wprowadzenie	15
6.2. Budowa sieci	15
6.3. Ostateczny kształt sieci	16
6.4. Punktacja	16
6.5. Komendy	17
6.6. Błędy	19
6.7. Serwery	19
6.8. Przykład	20
7. Zadanie Mur	22
7.1. Wprowadzenie	22
7.2. Budowa	22
7.3. Model rozgrywki	22
7.4. Model opisu	23
7.4.1. Model opisu sześcianu	23
7.4.2. Opis cegły	23
7.4.3. Rotacje	23
7.5. Punktacja	24
7.6. Komendy	25
7.7. Błędy	28
7.8. Serwery	28
7.9. Przykład	29

1. Wstępniak

To już piąty finał maratonu programistycznego Deadline24 organizowany przez firmę Future Processing, tym razem także pod zaszczytnym patronatem rektorów, dziekanów uniwersytetów i uczelni technicznych. Po raz wtóry też znajdujemy się w Zabytkowej Kopalni Węgla Kamiennego Guido, która jest współorganizatorem tego wydarzenia. Z pewnością unikalna sceneria tego miejsca będzie niezapomnianym przeżyciem dla wszystkich drużyn walczących o każdy punkt do ostatniej sekundy konkursu.

Podobnie jak w ubiegłych latach oddajemy w Wasze ręce trzy zadania, tradycyjnie związane ze światem intrygujących stworzeń – żukoskoczków. Ich ambicja w podbijaniu kolejnych planet postawi ich wobec konieczności tworzenia podstawowych budowli obronnych (zadanie **Mur**), a także wobec budowania sieci komunikacyjnej pomiędzy ważnymi lokalizacjami (zadanie **Sieć**). Będziecie mieli też okazję przekonać się jak żukoskoczki radzą sobie w sytuacjach kryzysowych (zadanie **Ognisko**). Czy trudne położenie obudzi w nich najbardziej zwierzęce instynkty? Zobaczmy już za moment.

Mamy nadzieję, że rozgrywka będzie emocjonująca, a system konkursowy i wszyscy uczestnicy wytrzymają napięcie przez całe 24 godziny. Niech wygra najlepszy!

Zespół Deadline24

2. Komunikacja z serwerem

Uzyskiwanie aktualnych informacji o wirtualnym świecie oraz wydawanie rozkazów jest możliwe za pomocą protokołu TCP/IP. Drużyna łączy się jako klient do odpowiedniego serwera konkursowego. Adres IP oraz port, z którym należy się połączyć są podane w sekcjach *Serwery* specyfikacji poszczególnych zadań. Można nawiązać wiele połączeń jednocześnie, jednak sumaryczny transfer przypadający na każdy komputer jest ograniczony. Maksymalna liczba połączeń i maksymalny transfer podane są w *Ustaleniach Technicznych*. Komunikacja odbywa się w trybie tekstowym. Bezpośrednio po połączeniu należy się zalogować, następnie sesja przechodzi w tryb poleceń.

2.1. Logowanie

Bezpośrednio po nawiązaniu połączenia serwer wysyła prośbę o login zakończoną znakiem końca linii: LOGIN. Należy wysłać swój login, a następnie znak końca linii. Następnie serwer zapyta o hasło (PASS), na co należy analogicznie odpowiedzieć hasłem. Jeśli autoryzacja przebiegła pomyślnie, serwer odpowie ciągiem znaków: OK i przejdzie w stan oczekiwania na komendy. W przeciwnym wypadku dostaniemy odpowiedź FAILED 1 bad login or password, po czym nastąpi zamknięcie połączenia.

Poniżej znajduje się przykładowy zapis komunikacji w czasie logowania.

klient → serwer	serwer → klient
	LOGIN
login1	
	PASS
secret	
	OK

2.2. Komendy

Każde polecenie składa się z nazwy komendy, argumentów (liczba zależna od polecenia) oraz znaku końca linii. Parametry powinny być oddzielone co najmniej jednym białym znakiem.

Na każdą komendę serwer odpowiada jednym z poniższych ciągów:

- 'OK' — w przypadku zaakceptowania komendy,
- 'FAILED *e msg*' — w przypadku błędu; gdzie *e* to kod błędu, a *msg* — komunikat błędu.

Następnie, zależnie od komendy, serwer może opcjonalnie wysłać lub odebrać dodatkowe dane. Jeśli dodatkowe dane są wysyłane od klienta do serwera, to po odebraniu tych danych serwer ponownie odpowie w sposób opisany powyżej. Przykładowe zapisy komunikacji z serwerem oraz zestawienia możliwych błędów dla poszczególnych zadań znajdują się w opisie każdego z nich.

Ograniczenie liczby komend Na każdym serwerze każdego z zadań obowiązuje limit na maksymalną liczbę komend wydawanych w czasie jednej tury. Osiągnięcie limitu zostanie zasygnalizowane odpowiednim błędem: FAILED 6 commands limit reached, forced waiting activated. Po wystąpieniu błędu, serwer prześle dodatkowy komunikat: FORCED_WAITING *x* — gdzie *x* ($x \in \mathbb{R}$) oznacza liczbę sekund do zakończenia oczekiwania, tj. do końca bieżącej tury.

2.3. Konwencje

Jeśli nie jest zaznaczone inaczej, to przyjmujemy, że:

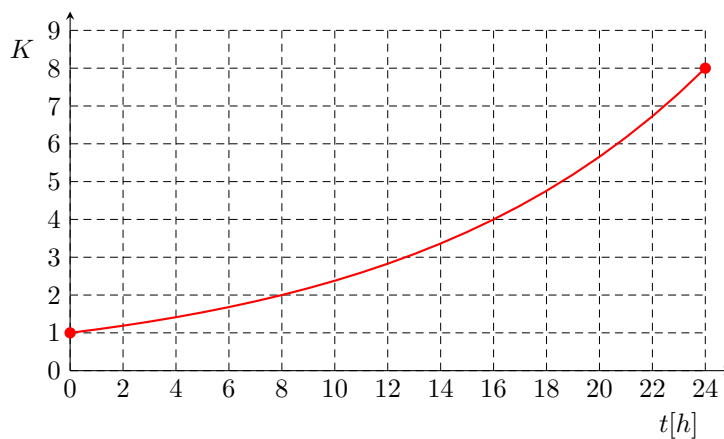
- Każda linia zakończona jest pojedynczym znakiem o kodzie ASCII 10 ('\n'). Znak powrotu karetki ('\r'; kod 13) towarzyszący mu na niektórych systemach operacyjnych będzie traktowany jako biały znak.

- W przypadku danych przesyłanych przez serwer, liczby oraz wyrazy oddzielone są pojedynczą spacją.
- W przypadku danych przesyłanych od klienta do serwera (np. parametry komendy) dozwolona jest dowolna (niezerowa) ilość białych znaków pomiędzy danymi, a także na końcu oraz na początku linii.
- Za białe znaki uznajemy: spację, powrót karetki (`'\r'`) oraz tabulator (`'\t'`).

3. Punktacja

3.1. Tempo rozgrywki

Współczynnik K Dla każdego serwera wynik wyznaczony dla drużyny jest dodatkowo mnożony przez K – współczynnik tempa rozgrywki. Wartość tego współczynnika wzrasta wykładniczo przez cały czas trwania konkursu od wartości 1 do wartości 8. Oznacza to, że w ostatnich godzinach konkursu można zdobyć kilkakrotnie więcej punktów niż na początku, a co za tym idzie warto ciągle udoskonalać swoje rozwiązania, by nie dać się wyprzedzić innym drużynom. Bieżąca wartość K jest dostępna w odpowiednim poleceniu każdego z zadań.



Rysunek 1: Zmiana współczynnika tempa rozgrywki K w czasie.

3.2. Punkty rankingowe

Punkty Na każdym serwerze drużyny zdobywają punkty w sposób zdefiniowany w specyfikacji zadania rozgrywanego na danym serwerze. Punkty zdobyte na poszczególnych serwerach są przeliczane na **punkty rankingowe**, które definiujemy jako iloczyn liczby 100 oraz stosunku wyniku drużyny (na danym serwerze) do średniej arytmetycznej trzech najlepszych wyników (na danym serwerze). Punkty przyznawane drużynie za konkretne zadanie to średnia punktów rankingowych z wszystkich serwerów danego zadania.

Ranking Ranking zawodów jest oparty o sumę punktów rankingowych wszystkich zadań.

4. Sytuacje awaryjne

W razie wystąpienia sytuacji, w której nie wszystkie drużyny będą w stanie brać udział w konkursie na ustalonych zasadach (np. brak prądu, awaria sieci lokalnej lub jej części, problemy z systemem konkursowym, itp.) oraz wina nie leży po stronie tych drużyn ani ich sprzętu, organizatorzy wstrzymają działanie systemu konkursowego, a o jego ponownym uruchomieniu uczestnicy zostaną poinformowani w lokalnym serwisie WWW konkursu. W tym czasie punkty nie będą naliczane. W takiej sytuacji mogą zostać zerwane wszystkie połączenia z serwerem konkursowym.

5. Zadanie Ognisko

5.1. Wprowadzenie

Przemierzając galaktykę jednym ze swych okazałych statków transportowych żukoskoczki doświadczyły niemiłej niespodzianki. Pewien odłamek meteorytu uszkodził układ sterujący i statek skierował się wprost na ogromną planetę. Kapsuły ewakuacyjne szybko wypełniły się pasażerami, jednak spora część załogi rozbiła się na powierzchni planety. Ta zaś, na szczęście dla zdrowia i życia żukoskoczków, składa się głównie z wielkich oceanów, na których występują tylko drobne wysepki, zaledwie z kępami drzew.

Każdy z rozbitków po mokrym lądowaniu szczęśliwie dopłynął też do takiej wysepki. Są oni jednak wszyscy rozstrzeleni zupełnie przypadkowo po okolicznych łądach, a chcieliby jak najszybciej wrócić do swoich typowych działań związanych z podbojem Universum.

Wiść o katastrofie szybko się rozeszła. Planeta jest jednak bardzo duża, więc przeczesywanie jej całej przez statek ratunkowy potrwa bardzo długo. Jediną opcją na przyspieszenie jego przybycia jest rozpalenie olbrzymiego ogniska na jednej z wysepek, bo chociaż nad samą powierzchnią wody unosi się dość gęsta mgła, uniemożliwiająca dokładną obserwację tego, co ma kontakt z wodą, to powyżej - na wyspach - atmosfera jest bardzo przejrzysta, przez co dym i ogień widać będzie jak na dłoni. Ognisko musi być jednak odpowiednio duże, by z pewnością zostało dostrzeżone. Należy więc zebrać w jednym miejscu odpowiednią ilość drewna...

5.2. Model rozgrywki

Rozgrzywka odbywa się w turach równej długości. Pomiędzy turami drużyny komunikują się z serwerem i wydają polecenia swoim jednostkom. Każda operacja podejmowana przez jednostkę trwa określoną, całkowitą liczbę tur, po których upływie można wydać kolejne polecenie danej jednostce.

Zakończenie rozgrywki Od początku rozgrywki znany jest czas po jakim statek ratunkowy będzie przelatywał nad danym sektorem i odnajdzie rozbitków. Rozpalenie ogniska może jednak przyspieszyć jego przylot, podając mu dokładną lokalizację zaginionych. Po udanym roznieceniu ognia do końca rozgrywki zostanie nie więcej niż 15 tur.

5.3. Świat

Rozgrzywka toczy się na powierzchni pewnego obszaru, na którym wyróżnić możemy łąd oraz wodę. Obszar jest reprezentowany przez regularną siatkę składającą się z $N \times N$ kwadratowych elementów, które nazywamy **polami**. Pola rozmieszczone są w N rzędach po N pól. Każde pole jest identyfikowane przez parę liczb (P_x, P_y) , które przyjmują wartości całkowite z przedziału od 1 do N . Każde z pól może być wyłącznie albo wodą, albo wyspą (łądem).

Wyspy Na każdej wyspie początkowo jest dostępna pewna ilość nieoznakowanego drewna – patyków, które można zabierać i przenosić w inne miejsca. Każda wyspa zajmuje dokładnie jedno pole i z wszystkich dostępnych podczas rozgrywki stron otoczona jest wodą. Wyspy wznoszą się ponad mgłę, która snuje się po wodzie. Z każdej z nich doskonale widać też drewno dostępne na innych wyspach w pobliżu.

Rozpalenie ogniska Ze względu na niedużą ilość dostępnych materiałów palnych, żukoskoczki nie mogą sobie pozwolić na ich marnotrawienie. Niedopuszczalne jest zatem podpalenie stosu patyków, dla którego nie ma pewności, że zostałyby zauważony ze statku ratunkowego.

5.4. Żukoskoczki

Każda drużyna rozporządza pewną liczbą żukoskoczków, które mogą wykonywać pewne czynności. Na małych odległościach – jak w przypadku obszaru rozgrywki – żukoskoczki potrafią porozumiewać się między sobą korzystając z przekazu telepatycznego, stąd wiedza jednego z nich jest ogólnodostępna w ramach wszystkich dostępnych żukoskoczków danej drużyny.

5.4.1. Operacje

Poruszanie się Żukoskoczek może przemieścić się o jedno pole mapy w każdej turze. Ruch musi mieścić się w obszarze mapy. Żywiołem żukoskoczków są jednak skoki na lądzie, a nie woda, stąd pływanie wprawd jest dla nich nie lada wyzwaniem i każda tura spędzona w wodzie oznacza dla każdego z nich utonięcie z prawdopodobieństwem 0.1%.

Zabranie drewna Oczywiście każdy żukoskoczek przemieszczając się może także przenosić drewno. Zanim jednak będzie je przenosić – musi je zabrać z jakiejś wyspy, co jest możliwe tylko jeśli drewno nie jest odpowiednio pilnowane (patrz poniżej) oraz nie znajduje się na już palącym się stosie. Indywidualny żukoskoczek przenosić może co najwyżej **pięć** patyków naraz.

Odłożenie drewna Jako, że patyki są bardzo cennym materiałem ratunkowym, nie wolno ich upuszczać nigdzie indziej poza lądem. Operacja ta wykonywana jest natychmiast.

Pilnowanie Po wykonaniu tej operacji żukoskoczek staje się strażnikiem drewna na danej wyspie. Strażnik nie może się przemieszczać, ani wykonywać innych operacji dopóki nie zostanie zwolniony ze służby (operacja wykonywana natychmiast). Podjęcie patyków na danej wyspie jest możliwe tylko wtedy, gdy liczba strażników pod koniec danej tury jest mniejsza od liczby żukoskoczków próbujących zabrać drewno w tym miejscu.

5.4.2. Tratwy

Żukoskoczki są może mało wodolubne, ale przecież bardzo sprytnie. Potrafią więc z zebranego drewna zbudować tratwę, która znacząco poprawia możliwości transportu drewna. Każda tratwa może oprócz jej kapitana pomieścić 40 patyków drewna. Pływając na tratwie żukoskoczki czują się dużo pewniej, gdyż nie ma wtedy dla nich ryzyka utonięcia.

Budowanie Każdy żukoskoczek nie będący aktualnie kapitanem tratwy może podjąć się jej budowy, o ile będzie on miał możliwość zabrania na danej wyspie 100 patyków (i znów – nie mogą one być odpowiednio pilnowane). Budowa trwa 20 tur i nie może zostać przerwana. Po jej zakończeniu budowniczy automatycznie staje się kapitanem zbudowanej jednostki wodnej. Każda drużyna może zbudować maksymalnie 12 tratw. Kapitan tratwy poświęca jej całą swą uwagę – nie może pilnować patyków na wyspie.

Porzucanie W każdej chwili kapitan może zdecydować się na opuszczenie tratwy (wysiąść z niej – operacja natychmiastowa). Tratwa bez kapitana może zostać przejęta przez innego żukoskoczka.

Przejmowanie Dowolny żukoskoczek nie będący aktualnie kapitanem może przejąć porzuconą tratwę, o ile znajdzie się w polu, w którym jest ona zlokalizowana i pierwszy złapie za jej stery (wyda odpowiednie polecenie wykonywane natychmiast). Nawigowanie tratwą to odpowiedzialne i angażujące zajęcie, stąd potencjalny kapitan nie może samodzielnie nosić żadnych patyków podczas przejmowania steru. Pilnowanie patyków nie ma wpływu na możliwość przejmowania. Nie można w żaden sposób przejąć tratwy aktualnie posiadającej kapitana.

Suszenie i rozbiórka Kiedy tratwa nie jest już potrzebna drużynie, może ona ponownie skorzystać z użytego do jej budowy materiału, by dołożyć go do stosu patyków. Zanim to będzie jednak możliwe tratwa musi zostać rozładowana oraz wyciągnięta z wody, by wyschnąć (mokre drewno nie nadaje się do spalania) i dopiero wtedy można ją zdemontować. Podczas budowy, użytkowania i rozbiórki materiał tratwy traci swoje właściwości energetyczne, stąd można odzyskać z tratwy jedynie 50 drewnianych patyków. Jeśli kapitan opuści swoje stanowisko przed zakończeniem procesu wysychania i rozbiórki to tratwa może zostać przejęta przez innego żukoskoczka. Proces wysychania i rozbiórki trwa 16 tur, kapitan może go przerwać przed tym czasem wydając polecenie poruszenia się bądź załadunku drewna.

5.5. Początkowy stan rozgrywki

W momencie rozpoczęcia rozgrywki wszystkie żukoskoczki należące do drużyny znajdują się na wyspach w losowych lokalizacjach, być może każdy w innej. Drużyna nie posiada żadnych tratw. Żukoskoczki różnych drużyn mogą zajmować te same pola (na danym polu w danej turze może znajdować się wiele żukoskoczków samodzielnie, na tratwach lub w roli strażnika). Początkowy stan ocalałych z katastrofy pasażerów będzie składać się z od 3 do 10 żukoskoczków. Początkowa liczba żukoskoczków w pojedynczej rozgrywce będzie taka sama dla wszystkich drużyn.

5.6. Zasady rywalizacji

5.6.1. Znakowanie feromonowe

Podczas przenoszenia patyków z jednego miejsca do innego żukoskoczki pozostawiają na drewnie swój unikalny feromonowy znacznik, który może zostać zniszczony (zastąpiony) tylko przez znacznik innej drużyny na tej samej zasadzie (podpalenie patyka nie usuwa znacznika). Jeśli patyk zostaje odłożony w tej samej lokalizacji, z której został zabrany, pozostaje na nim znacznik poprzedniej drużyny lub żaden, jeśli wcześniej ów patyk nie był przenoszony. Zakończone suszenie i demontaż tratwy oznakowują patyki z niej uzyskane znacznikiem feromonowym jej ostatniego kapitana niezależnie od źródła pochodzenia patyków, z których tratwa była zbudowana.

5.6.2. Zabieranie patyków – szczegóły

Zabranie patyków z wyspy jest możliwe tylko wtedy, jeśli liczba pilnujących je na tej wyspie rozbitków (wliczając w to strażników danej drużyny) jest mniejsza od liczby żukoskoczków próbujących w danym miejscu zabrać drewno (wliczając w to budowniczych tratw). Jeśli próba się nie powiedzie (strażnicy dadzą wycisk potencjalnym złodziejom) wszystkie podejmujące próbę zabrania drewna żukoskoczki nie będą mogły wykonywać żadnych ruchów przez **15 kolejnych tur**. Strażnicy z kolei wobec przeważającej liczby napastników po prostu ustępują im pola (nie mają nigdy obrażeń po walce), a w kolejnej turze wracają na swoje stanowiska.

Nie można podejmować patyków z płonącego stosu. Każdy żukoskoczek próbuje zawsze zabrać maksymalną liczbę patyków (5 samodzielnie, 40 na tratwę będąc kapitanem lub 100 pod budowę tratwy), jednak otrzymać ich może mniej zależnie od dostępnej na wyspie ilości drewna.

Poza oczywistą sytuacją udanego podjęcia patyków, gdy suma patyków na wyspie pokrywa dokładnie zapotrzebowanie żukoskoczków, możliwe są dwa przypadki opisane szczegółowo poniżej.

5.6.3. Nadmiar patyków na wyspie

Całe zapotrzebowanie żukoskoczków na patyki dotyczące danej wyspy w danej turze zostanie zaspokojone. Z wyspy zostaną podjęte w pierwszej kolejności patyki nie należące do żadnej drużyny, następnie patyki należące do drużyn innych niż te, które podjęły próbę zabrania drewna. Zostaną one przyznane zbieraczom z takim samym procentowym udziałem każdej grupy (z dokładnością do zaokrąglenia do jednej sztuki) – tj. np. po 50% z każdej grupy inaczej oznakowanych patyków. W dalszej kolejności będą podnoszone patyki już oznakowane przez drużyny podejmujące drewno – podobnie z zachowaniem tego samego procentowego udziału w grupach i losowym rozdysponowaniem pomiędzy żukoskoczki.

5.6.4. Nadmiar zapotrzebowania żukoskoczków

W tym przypadku z wyspy znikną wszystkie patyki. Przyznane one zostaną w pierwszej kolejności budowniczym (zależnie od czasu zgłoszenia ich chęci budowy w danej turze – kto pierwszy ten lepszy), a w dalszej kolejności podział dostępnych jeszcze patyków odbywać się będzie na zasadzie równych udziałów procentowych zapotrzebowania każdego żukoskoczka.

5.7. Punktacja

Wszystkie żukoskoczki bardzo chcą się wykazać swoją aktywnością i przypisać sobie największy udział w szybkim sprowadzeniu statku ratunkowego, by opowiadać potomnym o swym zuchwałym bohaterstwie. Każda drużyna zatem otrzymuje punkt za każdy oznakowany jej znacznikiem feromonowym patyk na stosie którejkolwiek z wysp. Ponadto, jeśli udało się rozpalić przynajmniej jedno ognisko, to każdy składający się na płonący stos patyk jest liczony wielokrotnie z mnożnikiem F .

5.8. Komendy

Ogólne założenia protokołu komunikacji (łączenie się, logowanie, wysyłanie komend, format odpowiedzi) opisane są w rozdziale *Komunikacja z serwerem*. Poniżej znajduje się lista komend dostępnych dla zadania Ognisko.

DESCRIBE_WORLD Zwraca parametry rozgrywki i wartość współczynnika skalującego wynik.

Parametry: brak

Dane (od serwera):

W pojedynczej linii serwer zwróci sześć wartości:

- N ($N \in \mathbb{N}$, $100 \leq N \leq 500$) — długość boku planszy
- I ($I \in \mathbb{N}$, $500 \leq I \leq 8000$) — liczba wysp na planszy
- S_{min} ($S_{min} \in \mathbb{N}$, $1000 \leq S_{min} \leq 2000$) — minimalny rozmiar stosu patyków, który z pewnością będzie zauważony przez statek ratunkowy
- F ($F \in \mathbb{N}$, $10 \leq F \leq 20$) — mnożnik punktów przyznawanych za patyki będące częścią płonącego stosu
- T ($T \in \mathbb{N}$, $1 \leq T \leq 5$) — czas trwania pojedynczej tury w sekundach
- K ($K \in \mathbb{R}$, $1 \leq K \leq 8$) — wartość współczynnika skalującego wynik

TIME_TO_RESCUE Zwraca liczbę tur pozostałych do przylotu statku ratunkowego, informację o tym, czy w obszarze rozgrywki udało się już rozpałcić ognisko, a także pozycje wysp, na których zlokalizowano największe ilości drewna.

Parametry: brak

Dane (od serwera):

Serwer zwraca w pierwszej linii oddzielone spacjami wartości:

- *FireStatus* ($FireStatus \in \{NONE, BURNING\}$) — jeden z dwóch ciągów znaków: NONE, jeśli żadnego stosu patyków nie udało się jeszcze podpalić lub BURNING, jeśli któryś ze stosów płonie
- L ($L \in \mathbb{N}$) — liczba tur pozostałych do zakończenia rozgrywki

Kolejne pięć linii opisuje pięć wysp, na których na początku tury znajdowało się najwięcej patyków. W liniach podane są kolejno:

- P_X ($P_X \in \mathbb{N}$, $1 \leq P_X \leq N$) — współrzędna X pola, w którym znajduje się wyspa
- P_Y ($P_Y \in \mathbb{N}$, $1 \leq P_Y \leq N$) — współrzędna Y pola, w którym znajduje się wyspa
- *Sticks* ($Sticks \in \mathbb{N}$) — liczba patyków na danej wyspie

LIST_SURVIVORS Zwraca listę ocalałych z katastrofy żukoskoczków.

Parametry: brak

Dane (od serwera):

Pierwsza linia:

- B ($B \in \mathbb{N}$) — liczba żukoskoczków

Druga linia zawiera B oddzielonych od siebie pojedynczą spacją liczb:

- ID ($ID \in \mathbb{N}$) — identyfikator żukoskoczka

LIST_RAFTS Zwraca listę tratw danej drużyny, ich pozycje, identyfikator kapitana oraz załadunek.

Parametry: brak

Dane (od serwera):

Pierwsza linia:

- W ($W \in \mathbb{N}$) — liczba przypisanych drużynie tratw

Każda kolejna linia opisuje jedną tratwę za pomocą oddzielonych spacjami liczb określających kolejno:

- P_X ($P_X \in \mathbb{N}$, $1 \leq P_X \leq N$) — współrzędna X pola, w którym znajduje się tratwa
- P_Y ($P_Y \in \mathbb{N}$, $1 \leq P_Y \leq N$) — współrzędna Y pola, w którym znajduje się tratwa
- ID ($ID \in \mathbb{N} \cup \{NIL\}$) — identyfikator żukoskoczka, który jest kapitanem tej tratwy lub napis NIL jeśli została ona porzucona, a nie została jeszcze przejęta przez inną drużynę.
- $Sticks$ ($Sticks \in \mathbb{N}$) — liczba patyków przewożonych na tratwie

IGNITION Podpala stos patyków w polu, w którym znajduje się żukoskoczek.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka

MOVE Wydaje żukoskoczkowi polecenie przejścia do sąsiedniego pola, wskazywanego przy pomocy względnych współrzędnych. Dokładnie jedna ze względnych współrzędnych D_X , D_Y musi być niezerowa. Operacja trwa jedną turę.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka
- D_X ($D_X \in \{-1, 0, 1\}$) — przesunięcie wzdłuż osi X
- D_Y ($D_Y \in \{-1, 0, 1\}$) — przesunięcie wzdłuż osi Y

TAKE Próbuje podjąć patyki leżące na wyspie pod koniec bieżącej tury.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka

GIVE Odkłada na wyspie wszystkie aktualnie przetrzymywane patyki.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka

Polecenie wykonywanie jest natychmiast po zgłoszeniu. Nie można wyrzucić patyków do wody. Jeśli na wyspie pali się ognisko – po odłożeniu patyki staną się jego częścią.

GUARD Nadaje żukoskoczkowi status strażnika pilnującego drewna na wyspie.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka

Polecenie wykonywanie jest natychmiast po zgłoszeniu. Kapitan tratwy może stać się strażnikiem dopiero po jej porzuceniu.

STOP_GUARDING Cofa status strażnika dla wybranego żukoskoczka.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka

Polecenie wykonywanie jest natychmiast po zgłoszeniu.

LIST_WOOD Zwraca aktualny stan liczby patyków na okolicznych łąkach jeśli żukoskoczek znajduje się na jakiejś wyspie.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka

Dane (od serwera):

Pierwsza linia:

- W ($W \in \mathbb{N}$) — liczba wysp na obszarze 17×17 pól, w którego środku znajduje się wskazany żukoskoczek

Każda kolejna z W linii opisuje jedną wyspę za pomocą oddzielonych spacjami liczb oznaczających kolejno:

- P_X ($P_X \in \mathbb{N}$, $1 \leq P_X \leq N$) — współrzędna X pola, w którym znajduje się wyspa
- P_Y ($P_Y \in \mathbb{N}$, $1 \leq P_Y \leq N$) — współrzędna Y pola, w którym znajduje się wyspa
- $Sticks$ ($Sticks \in \mathbb{N}$) — liczba patyków na danej wyspie
- $MySticks$ ($MySticks \in \mathbb{N}$) — liczba patyków oznakowanych przez drużynę na danej wyspie

Liczone są wyłącznie patyki gotowe aktualnie do podpalenia. Materiał tratw (nawet wysychających) nie jest brany pod uwagę do czasu ich całkowitej rozbiórki.

INFO Zwraca szczegółowe informacje o żukoskoczku, o polu, w którym on się znajduje i czterech sąsiednich polach.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka

Dane (od serwera):

Operacja zwróci w pierwszej linii informacje dotyczące wybranego żukoskoczka w postaci:

- P_X ($P_X \in \mathbb{N}$, $1 \leq P_X \leq N$) — współrzędna X pola, w którym znajduje się żukoskoczek
- P_Y ($P_Y \in \mathbb{N}$, $1 \leq P_Y \leq N$) — współrzędna Y pola, w którym znajduje się żukoskoczek
- $Sticks$ ($Sticks \in \mathbb{N}$) — liczba patyków będących aktualnie w posiadaniu żukoskoczka (samodzielnie lub poprzez bycie kapitanem tratwy)
- $Busy$ ($Busy \in \mathbb{N} \cup \{UNKNOWN\}$) — liczba tur, przez które żukoskoczek będzie niedostępny wykonując wcześniejsze rozkazy lub UNKNOWN dla strażników
- $Role$ ($Role \in \{GUARD, CAPTAIN, BUILDER, NONE\}$) — informacja o tym czy rozbitek pełni aktualnie rolę (kolejno): strażnika, kapitana, budowniczego tratwy lub żadnej

Kolejne dane to opis pięciu pól – pola, w którym znajduje się wskazany żukoskoczek i jego czterech sąsiadów. Każde pole zostanie opisane w osobnej linii o jednej z postaci:

- NIL — jeśli dane pole nie istnieje (w przypadku pól poza obszarem rozgrywki)
- $Type D_X D_Y B G R_O R_A S_O S_A$ — jeśli pole istnieje

W powyższej specyfikacji zmienna $Type$ przyjmuje jedną z wartości: LAND (dla wysp), WATER (dla pól wodnych), zmienne $D_X D_Y$ oznaczają względne (względem aktualnego pola) współrzędne opisywanego pola, B to sumaryczna liczba żukoskoczków na tym polu (razem z aktualnie opisywanym rozbitkiem), G to liczba działających strażników na danym polu, R_O to liczba tratw z kapitanem, R_A to sumaryczna liczba porzuconych na danym polu tratw (również wysychających, które zostały opuszczone), S_O to sumaryczna liczba patyków na tratwach z kapitanem, a S_A to sumaryczna liczba patyków na porzuconych tratwach. Wszystkie te wartości dotyczą stanu z początku tury.

BUILD Próbuje podjąć patyki leżące na wyspie i zbudować z nich tratwę. Budowniczy nie może być już kapitanem innej jednostki.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka

Budowę można rozpocząć tylko po zabraniu 100 patyków pod budowę. Budowa rozpoczyna się w kolejnej turze jeśli pod koniec aktualnej rundy żukoskoczкови udało się podjąć odpowiednią ilość drewna. Proces budowy trwa 20 tur i nie może zostać przerwany.

ABANDON Umożliwia w dowolnym momencie tratwy jej porzucenie przez kapitana. Operacja wykonywana natychmiast.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka - kapitana

Załadunek tratwy pozostaje bez zmian po opuszczeniu jej przez kapitana.

TAKE_OVER Umożliwia przejęcie wcześniej porzuconej tratwy. Operacja wykonywana natychmiast.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka

Jeśli na polu znajduje się więcej tratw w pierwszej kolejności przejmowane są tratwy należące do danej drużyny, w drugiej kolejności przejmowane są tratwy z jak największą liczbą patyków.

DRY Rozpoczyna proces wysuszania i demontażu tratwy. Operacja nie może być wykonana dopóki na tratwie znajduje się jakiś ładunek.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka - kapitana

Jeśli w momencie ukończenia operacji na wyspie pali się ognisko, patyki pozostałe z demontażu tratwy od razu staną się częścią płonącego stosu.

MY_WOOD Zwraca informacje o oznakowanych i zebranych przez drużynę patykach.

Parametry: brak

Dane (od serwera):

W pojedynczej linii serwer zwróci trzy wartości:

- T ($T \in \mathbb{N}$) — suma punktów przyznana drużynie za patyki leżące na wyspach (bez uwzględniania współczynnika K)
- S ($S \in \mathbb{N}$) — liczba patyków oznakowanych przez drużynę i leżących na wyspach
- C ($C \in \mathbb{N}$) — suma liczby patyków transportowanych przez drużynę (samodzielnie przez rozbitków i na tratwach)

WAIT Czeka do momentu rozpoczęcia kolejnej tury.

Parametry: brak

Dane (od serwera):

Serwer odpowiada jedną linią postaci:

- `WAITING S (THIS WORKS DIFFERENTLY)`

gdzie S ($S \in \mathbb{R}, S \geq 0$) oznacza ilość sekund do zakończenia oczekiwania

5.9. Błędy

Zgodnie z opisem w rozdziale *Komunikacja z serwerem*, w wypadku podania nieprawidłowego polecenia, serwer odpowiada komunikatem:

- 'FAILED *e msg*',

gdzie *e* to kod błędu, a *msg* — komunikat błędu. W tabeli znajduje się zestawienie błędów, które mogą wystąpić przy wydawaniu poleceń w zadaniu Ognisko.

kod błędu	komunikat błędu
1	bad login or password
2	unknown command
3	bad format
4	too many arguments
5	internal error, sorry...
6	commands limit reached, forced waiting activated
101	incorrect survivor identifier
102	destination is not neighbour
103	destination is outside the world
104	unavailable - survivor is not on land
105	incorrect identifier, the survivor has drown
106	captain cannot build raft
107	not enough sticks to set wood on fire
108	nothing to give
109	guard cannot do such operation
110	survivor is already a guard
111	survivor is busy
112	survivor is not a guard
113	survivor is already a captain
114	survivor is not a captain
115	captain cannot guard wood
116	cannot take over a raft with a captain
117	rafts count limit reached
118	no rafts to take over
119	you cannot carry wood to become a captain
120	wood is burning - taking not possible
121	unable to dry non empty raft
122	already drying

5.10. Serwery

Rozgrywki będą odbywać się na trzech serwerach, różniących się między sobą parametrami.

nazwa	adres:port	charakter akwenu
F1	universum.dl24:20000	losowy
F2	universum.dl24:20001	pierścieniowy

5.11. Przykład

Poniżej znajduje się przykładowy zapis komunikacji z serwerem.

klient → serwer	serwer → klient
	LOGIN
login1	PASS
secret	OK
DESCRIBE_WORLD	OK
	100 4000 1500 15 2 1.000000
LIST_SURVIVORS	OK
	4
	1 2 3 4
INFO 1	OK
	7 12 20 0 CAPTAIN
	WATER 0 0 2 0 2 0 25 0
	LAND 0 1 5 3 1 0 12 0
	WATER 1 0 0 0 0 0 0 0
	WATER 0 -1 0 0 0 1 0 0
	WATER -1 0 0 0 0 1 0 19
MOVE 1 0 1	OK
GIVE 1	FAILED 111 survivor is busy
WAIT	OK
	WAITING 0.310000
	OK
GIVE 1	OK
DRY 1	OK
GIVE 3	FAILED 104 unavailable - survivor is not on land
GIVE 2	OK
DRY 2	OK
ABANDON 2	OK
IGNITE 1	OK
TIME_TO_RESCUE	OK
	BURNING 12
	7 13 1517
	97 43 920
	74 11 920
	23 56 817
	87 24 799

6. Zadanie Sieć

6.1. Wprowadzenie

Strategia podboju Universum przez żukoskoczki zakłada maksymalne wykorzystanie wszelkich zasobów dostępnych w nowo zdobytych lokalizacjach. Gdy wszystkie przydatne surowce zostaną już wydobyte, miejsce takie jest porzucane, a dotychczasowi mieszkańcy przesiedlani w nowe miejsca – taki właśnie los spotyka zdecydowaną większość planet, na której żukoskoczki postawiły swoje odnoża. Tylko nieliczne planety posiadające szczególnie sprzyjający dla żukoskoczków klimat zasiedlane są na stałe, służąc za bazy wypadowe dla kolejnych ekspedycji.

Permanenta kolonizacja planety wiąże się z problemami natury niemilitarnej, takimi jak utworzenie odpowiedniej infrastruktury. Przed takim właśnie problemem stoją mieszkańcy zasiedlonej już jakiś czas temu planety T_{HEX} . Aktualnie szczególnie zależy im na utworzeniu sieci komunikacyjnej, umożliwiającej telefoniczne połączenia międzymiastowe. Wietrząc możliwość dużego zarobku, na T_{HEX} zjawili się wysoko postawieni przedstawiciele firm operatorów żukoskoczkowej sieci komunikacyjnej.

To właśnie Wy, główni planiści różnych operatorów sieci, zostaliście oddelegowani na tę planetę. Waszym zadaniem jest budowa infrastruktury przekaźników, która umożliwi mieszkańcom telefoniczną komunikację w Waszych sieciach.

6.2. Budowa sieci

Cały zamieszkiwany przez żukoskoczki teren planety T_{HEX} został podzielony na L^2 kawałków należących do najbogatszych właścicieli ziemskich. Tereny te mają bardzo regularne kształty i są oznaczone przez $F_{i,j}$ dla $1 \leq i, j \leq L$ w taki sposób, że wieża przekaźnikowa postawiona na kawałku $F_{x,y}$ może przekazać sygnał do każdej z sześciu wież stojących w: $F_{x,y-1}, F_{x-1,y-1}, F_{x-1,y}, F_{x,y+1}, F_{x+1,y}, F_{x+1,y-1}$ o ile x jest nieparzyste, lub każdej z sześciu wież stojących w: $F_{x,y-1}, F_{x-1,y}, F_{x-1,y+1}, F_{x,y+1}, F_{x+1,y+1}, F_{x+1,y}$ jeśli x jest parzyste (przy czym gdy $x \in \{1, L\}$ lub $y \in \{1, L\}$ niektóre z tych pól nie istnieją). Dokładnie N spośród terenów $F_{i,j}$ są zajmowane przez miasta, które muszą być połączone wynikową siecią wież przekaźnikowych.

Aby postawić na danym terenie wieżę przekaźnikową, trzeba liczyć się z pewną coroczną opłatą dzierżawczą, ustalaną przez właściciela terenu. Co więcej, na danym obszarze może stać tylko ograniczona liczba wież – jest to uwarunkowane rzeźbą terenu oraz widzimisją właściciela.

Zgodnie z lokalnym prawem antymonopolowym, w danej jednostce czasu można uzyskać tylko jedno pozwolenie na postawienie wieży przekaźnikowej. Ponadto takie pozwolenie może być uzyskane tylko pod warunkiem, że operator posiada pozwolenie na budowę innej wieży w odległości co najwyżej D (odległość jest mierzona jako minimalna liczba przejść między sąsiednimi terenami potrzebna, aby dojść z jednego terenu do drugiego). Każdy operator ma z urzędu pozwolenie na budowę wież przekaźnikowych w każdym z miast.

6.3. Ostateczny kształt sieci

W każdym momencie drużyna może zgłosić ostateczny plan sieci wież do wybudowania. Dany plan musi umożliwiać połączenia telefoniczne między dowolnymi dwoma miastami i jest to jedyny nakładany na niego wymóg. W szczególności nie trzeba budować wież wszędzie, gdzie uzyskane zostały na to pozwolenia oraz można uwzględnić wieże na terenie, na którym nie posiada się pozwolenia na budowę – będzie to jednak wiązało się z wykorzystaniem stojącego tam prywatnego przekaźnika należącego do właściciela danego terenu i coroczną opłatą w wysokości C -krotności normalnej opłaty za dzierżawę.

Koszt danego planu jest obliczany jako suma corocznych opłat z każdego terenu należącego do tego planu (nie uwzględnia się więc kosztów tych terenów, na których uzyskano pozwolenie na budowę, ale z niej nie skorzystano).

6.4. Punktacja

Wszystkie drużyny, które zgłoszą prawidłowy plan ostatecznej sieci, zostaną uszeregowane w rankingu wg kosztu rocznego utrzymania zgłoszonej sieci (im koszt mniejszy, tym wyżej dana drużyna w rankingu). Drużyna, która zajmie w tym rankingu m -te miejsce, otrzyma $W \cdot \left(\frac{31}{32}\right)^{m-1}$ punktów. Dla każdej drużyny liczy się tylko ostatni zgłoszony przez nią prawidłowy plan.

6.5. Komendy

Ogólne założenia protokołu komunikacji (łączenie się, logowanie, wysyłanie komend, format odpowiedzi) opisane są w rozdziale *Komunikacja z serwerem*. Poniżej znajduje się lista komend dostępnych dla zadania Sieć.

DESCRIBE_WORLD Zwraca wszystkie parametry rozgrywki oraz wartość współczynnika skalującego wynik. Komenda ta może zostać użyta w tej samej rozgrywce kilka razy, ale nie częściej niż raz na 60 rund.

Parametry: brak

Dane (od serwera):

Pierwsza linia zawiera oddzielone pojedynczymi spacjami wartości:

- L ($L \in \mathbb{N}$, $1 \leq L \leq 100$) — z L^2 kawałków terenu składa się zamieszкана część planety $\text{T}_{\text{H}}\text{E}\text{X}$
- N ($N \in \mathbb{N}$, $1 \leq N \leq 500$) — liczba miast, które trzeba połączyć siecią
- D ($D \in \mathbb{N}$, $1 \leq D \leq 1000$) — dopuszczalna odległość terenu, na którym można uzyskać pozwolenie na budowę wieży, od terenu, na którym takie pozwolenie już się posiada
- C ($C \in \mathbb{N}$, $1 \leq C \leq 1000000$) — mnożnik corocznej opłaty dzierżawnej za użytkowanie cudzego przekąźnika
- W ($W \in \mathbb{N}$) — wynik punktowy zwycięzcy rankingu (wartość stała dla danej rozgrywki)
- T ($T \in \mathbb{N}$, $1 \leq T \leq 3$) — czas trwania jednej tury w sekundach; nie dotyczy to pierwszej tury każdej rozgrywki, która trwa 20 sekund
- K ($K \in \mathbb{R}$, $1 \leq K \leq 8$) — wartość współczynnika skalującego wynik

Następnie znajduje się opis wysokości opłat dzierżawczych poszczególnych terenów, czyli L linii, w każdej z nich jest L oddzielonych pojedynczymi spacjami liczb naturalnych mniejszych niż 1000. Liczby te określają opłaty poszczególnych terenów; j -ta liczba w i -tym wierszu opisuje opłatę związaną z terenem $F_{i,j}$. Opłata wysokości 0 oznacza, że znajduje się tam miasto. Gwarantowane jest, że w opisie opłat będzie dokładnie N zer. Następnie znajduje się L linii, w każdej z nich jest L oddzielonych pojedynczymi spacjami liczb naturalnych nie mniejszych niż 0 i nie większych niż 30, oznaczających, ile różnych pozwoleń może wydać właściciel odpowiedniego terenu; j -ta liczba w i -tym wierszu opisuje wartość związaną z terenem $F_{i,j}$.

TIME_TO_REQUEST Zwraca liczbę tur pozostałych do zakończenia rozgrywki.

Parametry: brak

Dane (od serwera):

Serwer zwraca pojedynczą wartość:

- T ($T \in \mathbb{N}$, $T < 2000$) — liczba tur pozostałych do zakończenia rozgrywki

DECLARE_PLAN Wydaje polecenie zgłoszenia ostatecznego planu sieci. W danej turze można zgłosić tylko jeden plan.

Parametry:

Należy przekazać dokładnie dwie linie. W pierwszej z nich znajduje się tylko liczba $R < 2000$, oznaczająca liczbę terenów należących do planu. W następnej linii musi znajdować się dokładnie $2R$ pooddzielanych pojedynczymi spacjami liczb naturalnych z przedziału $[1, L]$, oznaczających współrzędne różnych pól należących do planu. W poprawnym planie muszą znaleźć się między innymi wszystkie miasta. Para współrzędnych (x, y) opisuje teren $F_{x,y}$.

REQUEST Wydaje polecenie złożenia podania o pozwolenie na budowę wieży przekąźnikowej. W danej turze można uzyskać tylko jedno pozwolenie na budowę.

Parametry:

Należy przekazać dokładnie dwa oddzielone pojedynczą spacją parametry z zakresu $[1, L]$, oznaczające współrzędne kawałka terenu. Para współrzędnych (x, y) opisuje teren $F_{x,y}$.

MY_REQUESTS Zwraca listę wszystkich uzyskanych przez drużynę pozwoleń, z wyjątkiem tych dotyczących miast. W danej turze można wydać to polecenie tylko jeden raz.

Parametry: brak

Dane (od serwera):

Serwer zwraca dokładnie jeden ciąg liczb naturalnych, który składa się najpierw z liczby S , określającej długość ciągu, a następnie z $2S$ oddzielonych spacją liczb z przedziału $[1, L]$ opisujących S par współrzędnych. Para współrzędnych (x, y) opisuje teren $F_{x,y}$.

LAST_OBTAINED Zwraca listę pozwoleń ostatnio uzyskanych przez wszystkie drużyny w ciągu ostatnich pięciu rund.

Parametry: brak

Dane (od serwera):

Serwer zwraca dokładnie pięć ciągów par liczb, każdy ciąg w osobnej linii. k -ty ciąg opisuje (w przypadkowej kolejności) wszystkie tereny, na które zostały uzyskane pozwolenia przez drużyny k rund temu. Każdy jeden ciąg składa się najpierw z liczby S , określającej długość ciągu, a następnie z $2S$ oddzielonych spacją liczb z przedziału $[1, L]$ opisujących S par współrzędnych. Para współrzędnych (x, y) opisuje teren $F_{x,y}$. Każdy teren będzie zawarty w danej liście dokładnie tyle razy, ile drużyny uzyskało w danej rundzie nań pozwolenie.

GET_RANKING Zwraca aktualny ranking planów sieci zadeklarowanych przez drużyny, w kolejności od najlepszego (czyli najtańszego).

Parametry: brak

Dane (od serwera):

Serwer zwraca dokładnie jeden ciąg liczb naturalnych, który składa się najpierw z liczby S , określającej długość ciągu (S jest liczbą tych drużyn, które zgłosiły jakikolwiek poprawny plan sieci), a następnie z S oddzielonych spacją liczb naturalnych w kolejności niemalejącej, oznaczających koszty (ostatnich poprawnych) zgłoszonych przez drużyny planów w kolejności od najtańszego.

WAIT Czeka do momentu rozpoczęcia kolejnej tury.

Parametry: brak

Dane (od serwera):

Serwer odpowiada jedną linią postaci:

- WAITING S

gdzie S ($S \in \mathbb{R}, 0 \leq S$) oznacza ilość sekund do zakończenia oczekiwania

6.6. Błędy

Zgodnie z opisem w rozdziale *Komunikacja z serwerem*, w wypadku podania nieprawidłowego polecenia, serwer odpowiada komunikatem:

- 'FAILED *e msg*',

gdzie *e* to kod błędu, a *msg* — komunikat błędu. W tabeli znajduje się zestawienie błędów, które mogą wystąpić przy wydawaniu poleceń w zadaniu Sieć.

kod błędu	komunikat błędu
1	bad login or password
2	unknown command
3	bad format
4	too many arguments
5	internal error, sorry...
6	commands limit reached, forced waiting activated
101	usage limit reached for this command
102	invalid field coordinates
103	too many fields in declared plan
104	duplicate field in declared plan
105	invalid plan
106	permission already granted for this field
107	permissions limit reached for this field
108	permission already granted in this turn
109	permission cannot be granted for this field

6.7. Serwery

Rozgrywki będą odbywać się na dwóch serwerach, różniących się między sobą parametrami.

nazwa	adres:port	ograniczenia
N1	universum.dl24:20002	–
N2	universum.dl24:20003	$D = 1000$

6.8. Przykład

Poniżej znajduje się przykładowy zapis komunikacji z serwerem.

klient → serwer	serwer → klient
	LOGIN
login1	
	PASS
secret	
	OK
DESCRIBE_WORLD	
	OK
	3 4 1 5 30 1 1.000000
	0 8 0
	4 5 2
	0 10 0
	0 10 0
	4 6 2
	0 0 0
REQUEST 1 1	
	FAILED 106 permission already granted for this field
REQUEST 3 2	
	FAILED 107 permissions limit reached for this field
REQUEST 2 2	
	OK
WAIT	
	OK
	WAITING 0.73000
	OK
REQUEST 2 3	
	OK
REQUEST 1 2	
	FAILED 108 permission already granted in this turn
DECLARE_PLAN 6	
1 1 1 3 2 2 2 3 3 1 3 3	
	FAILED 106 invalid plan
DECLARE_PLAN 6	
1 1 1 3 2 1 2 2 3 1 3 3	
	FAILED 101 usage limit reached for this command
WAIT	
	OK
	WAITING 0.12000
	OK
DECLARE_PLAN 6	
1 1 1 3 2 1 2 2 3 1 3 3	
	OK
GET_RANKING	
	OK
	1 25
REQUEST 2 1	
	OK

klient → serwer	serwer → klient
GET_RANKING	OK 1 9
MY_REQUESTS	OK 3 2 2 2 3 2 1

7. Zadanie Mur

7.1. Wprowadzenie

Huczne świętowanie zwycięstwa nad krwiożerczymi stworzeniami nie wiele pomogły propagandzie w zatartciu wspomnień i komentarzy dotyczących złej taktyki obieranej przez dowództwo przy kolonizowaniu nowych lokalizacji. Bolesne dla żukoskoczków wydarzenia z planety CFK odbiły się szerokim echem we wszystkich znanych zakątkach Universum.

Najwyższa Rada Kolonizacyjna słusznie zauważyła, że tamtejszą sytuację udało się opanować jedynie dzięki bohaterskiej postawie obronnej osadników, a straty byłyby znacznie mniejsze, gdyby wcześniej pomyślano o choćby podstawowych budowlach obronnych. Wniosek ten zaowocował zmianą w oficjalnych wytycznych dla kolonizatorów nowych planet – od tej pory każda pierwsza osada na dotąd nieskolonizowanej planecie musi zostać otoczona murem.

Z racji pozyskania nowych wysokoenergetycznych surowców tempo ekspansji żukoskoczków zaskoczyło je same. Pierwsze oddziały zajęły właśnie kolejną planetę, dla której nie wymyślono jeszcze nawet nazwy. Wy także bierzecie udział w akcji kolonizacyjnej tego miejsca i to właśnie Wam przypadła w udziale budowa obowiązkowego, zgodnego z nowymi przepisami muru obronnego.

7.2. Budowa

Surowcem do budowy muru będzie rzadki materiał zwany Skałą, którego znaczne pokłady odkryto niedaleko nowo powstałej osady. Skała wykazuje się pewnymi specyficznymi własnościami fizycznymi, które powodują, że mniejsze fragmenty złożone z tego materiału mogą być łączone ze sobą w większe bez użycia jakiegokolwiek materiału łączącego. To nasunęło żukoskoczkom następujący proces budowy:

- Przygotowywana jest prostopadłościenna forma, której podstawa będzie miała wymiary $X \times Y$, a wysokość będzie równa Z .
- Cegły są wrzucane z góry i spadają bezwładnie do momentu, aż zatrzymają się na wcześniej postawionych ceglach lub podstawie formy.
- Ewentualnie powstałe dziury zostaną na koniec wypełnione Uniwersalnym Wypełniaczem do Murów w Sprayu.

Wypełniacz jest oczywiście trochę słabszy niż sama Skała, zależy nam więc na uzyskaniu jak najbardziej szczelnego muru.

7.3. Model rozgrywki

Rozgrzywka toczy się na wspólnej dla wszystkich drużyn formie do murów i odbywa się w turach równej długości. W każdej z nich dostępnych jest B rodzajów cegieł. Każda z drużyn w każdej z tur ma prawo wybrać jeden z dostępnych rodzajów cegły, odpowiednio wybraną cegłę obrócić i zrzucić ją w głąb formy.

Po każdej turze co najwyżej jeden z dostępnych rodzajów cegieł zmienia się na inny. Dany rodzaj cegieł może być w każdej turze użyty przez dowolną liczbę drużyn. Rozgrywka kończy się, gdy przez 10 kolejnych tur żadna z drużyn nie zrzuci cegły w prawidłowy sposób.

7.4. Model opisu

Przyjmujemy, że forma podzielona jest na $X \cdot Y \cdot Z$ sześcianików jednostkowych oraz związujemy z nimi układ współrzędnych w taki sposób, że jego osie są równoległe do boków formy i współrzędne (x, y, z) każdego z sześcianików spełniają nierówności:

$$1 \leq x \leq X \quad \wedge \quad 1 \leq y \leq Y \quad \wedge \quad 1 \leq z \leq Z$$

oraz sześcianiki przylegające do dna formy mają współrzędną $z = 1$.

7.4.1. Model opisu sześcianu

Na wejściu przekazywanych jest wiele opisów sześcianów, ale sposób opisu każdego z nich jest taki sam. Opis sześcianu o boku długości L składa się z L^2 linii, po L znaków każda. Znak '#' oznacza pole (sześcianik) zajęte, znak '.' pole wolne. Linia numer l zawiera opisy sześcianików o współrzędnych $(x, L - [(l - 1) \bmod L], \lceil \frac{l}{L} \rceil)$ dla rosnących od 1 do L wartości x . Przykładowo, dla $L = 2$, współrzędne kolejno podawanych sześcianików są następujące:

$$\begin{array}{ll} (1, 2, 1) & (2, 2, 1) \\ (1, 1, 1) & (2, 1, 1) \\ (1, 2, 2) & (2, 2, 2) \\ (1, 1, 2) & (2, 1, 2) \end{array}$$

7.4.2. Opis cegły

Cegła każdego rodzaju składa się z dodatniej liczby sześcianików jednostkowych i jest opisywana w sposób przedstawiony w *Modelu opisu sześcianu* na planie sześcianu o boku długości D , która to wartość jest wspólna dla wszystkich cegieł w obrębie danej rozgrywki. Każdy taki opis reprezentuje spójną bryłę (dwa sześcianiki sąsiadują ze sobą wtedy i tylko wtedy, gdy mają jedną wspólną ścianę).

Każdy rodzaj cegły jest ponadto opisywany przez kilka wartości: jego unikalną liczbę porządkową, wagę pojedynczej cegły danego rodzaju oraz wartość minimalnego dopasowania. Wartość minimalnego dopasowania P^{min} danej cegły oznacza, że po jej zrzuceniu co najmniej P^{min} z jej ścian musi sąsiadować z wcześniej zrzuconymi cegłami lub dnem formy. W przeciwnym przypadku cegła taka jest usuwana z muru tak, jakby nigdy nie została zrzuciona.

7.4.3. Rotacje

Pojedyncza rotacja odbywa się względem jednej z płaszczyzn xy, xz, yz o 90° . Niech dana będzie następująca cegła (w tym przypadku $D = 2$):

$$\begin{array}{ll} a & b \\ c & d \\ e & f \\ g & h \end{array}$$

Pojedynczy obrót w płaszczyźnie xy przekształciłby ją na cegłę:

$$\begin{array}{ll} b & d \\ a & c \\ f & h \\ e & g \end{array}$$

Pojedynczy obrót w płaszczyźnie xz przekształciłby ją zaś na cegłę:

$$\begin{array}{cc} e & a \\ g & c \\ f & b \\ h & d \end{array}$$

Natomiast pojedynczy obrót w płaszczyźnie yz przekształciłby ją na cegłę:

$$\begin{array}{cc} c & d \\ g & h \\ a & b \\ e & f \end{array}$$

7.5. Punktacja

Niech W_b oznacza wagę cegły, V_b jej objętość (czyli liczbę sześcianików, z których się składa), a P_b liczbę ścian tej cegły sąsiadujących z wcześniej już zrzuconymi cegłami lub dnem formy. Jeżeli cegła mieści się w całości w formie oraz $P_b \geq P_b^{min}$, to cegła jest zrzucana prawidłowo i pozostaje w jednym i tym samym miejscu do końca rozgrywki, a drużyna ją zrzucająca otrzymuje za nią: $W_b \cdot (V_b \cdot C_V + P_b \cdot C_P)$ punktów (C_V i C_P to stałe wspólne dla wszystkich cegieł w obrębie danej rozgrywki). W przeciwnym przypadku cegła jest całkowicie usuwana z muru i otrzymuje się za nią 0 punktów.

7.6. Komendy

Ogólne założenia protokołu komunikacji (łączenie się, logowanie, wysyłanie komend, format odpowiedzi) opisane są w rozdziale *Komunikacja z serwerem*. Poniżej znajduje się lista komend dostępnych dla zadania Mur.

DESCRIBE_WORLD Zwraca wszystkie parametry rozgrywki oraz wartość współczynnika skalującego wynik.

Parametry: brak

Dane (od serwera):

Jedyna linia zawiera oddzielone pojedynczymi spacjami wartości:

- X ($X \in \mathbb{N}$) — długość podstawy formy
- Y ($Y \in \mathbb{N}$) — szerokość podstawy formy
- Z ($Z \in \mathbb{N}$) — wysokość formy
- D ($D \in \mathbb{N}$, $D \geq 2$) — długość boku sześcianu służącego do opisu pojedynczej cegły
- B ($B \in \mathbb{N}$, $1 \leq B \leq 30$) — liczba rodzajów cegieł dostępnych w każdej z tur
- R ($R \in \mathbb{N}$, $1 \leq R \leq 10$) — sumaryczna dopuszczalna liczba zrzuceń cegły na jedną turę
- C_V ($C_V \in \mathbb{R}$, $0 \leq C_V \leq 2$) — współczynnik punktowy skalujący objętość zrzuconej cegły
- C_P ($C_P \in \mathbb{R}$, $1 \leq C_P \leq 10$) — współczynnik punktowy skalujący liczbę ścian wspólnych zrzuconej cegły oraz cegieł wcześniej zrzuczonych i dna formy
- T ($T \in \mathbb{N}$, $1 \leq T \leq 3$) — czas trwania jednej tury w sekundach
- K ($K \in \mathbb{R}$, $1 \leq K \leq 8$) — wartość współczynnika skalującego wynik

LIST_BRICKS Zwraca numery wszystkich aktualnie dostępnych rodzajów cegieł.

Parametry: brak

Dane (od serwera):

Zwracanych jest w jednej linii B różnych liczb całkowitych z przedziału $[1, 10^6]$ oddzielonych pojedynczymi spacjami. Są to numery porządkowe rodzajów cegieł dostępnych w aktualnej turze. Ten sam numer opisuje tę samą cegłę w ciągu całych 24h trwania konkursu na konkretnym serwerze.

SHOW_BRICK Zwraca opis jednej z dostępnych cegieł.

Parametry:

Należy przekazać dokładnie jedną liczbę całkowitą będącą numerem porządkowym jednego z aktualnie dostępnych rodzajów cegieł.

Dane (od serwera):

Zwracany jest opis pojedynczego, aktualnie dostępnego rodzaju cegły, który zaczyna się od jednej linii zawierającej dwie liczby

- W ($W \in \mathbb{R}$, $1 \leq W \leq 10$) — waga cegły
- P^{min} ($P^{min} \in \mathbb{N}$, $P^{min} \geq 0$) — minimalna liczba ścian cegły, które muszą sąsiadować z wcześniej zrzucenymi cegłami lub dnem formy

Następnie podanych jest D^3 znaków w D^2 liniach po D znaków każda, opisujących cegłę tak, jak zostało to opisane w *Modelu opisu sześcianu*.

TIME_TO_BUILD Zwraca liczbę tur pozostałych do zakończenia rozgrywki.

Parametry: brak

Dane (od serwera):

Serwer zwraca pojedynczą wartość:

- L ($L \in \mathbb{N}$, $1 \leq L \leq 10$) — liczba tur pozostałych do zakończenia rozgrywki; w każdej turze, w której zostanie zrzuciona w sposób prawidłowy choć jedna cegła, wartość ta resetowana jest z powrotem do wartości 10

GET_SCORE Zwraca aktualną liczbę punktów zdobytych przez drużynę w danej rozgrywce (nie przemnożoną przez K).

Parametry: brak

Dane (od serwera):

- P ($P \in \mathbb{R}$) — liczba punktów

VIEW_FROM_ABOVE Opisuje stan muru jako widok z góry.

Parametry: brak

Dane (od serwera):

Serwer zwraca Y linii po X liczb naturalnych oddzielonych pojedynczymi spacjami w każdej z nich, reprezentujących wysokości, do jakich dosięgają zrzucone cegły. Wysokości zwracane są kolejno rzędami zaczynając od rzędu numer Y , a w obrębie rzędu kolumnami, zaczynając od kolumny numer 1 w następującej kolejności:

$$\begin{array}{cccc} (1, Y) & (2, Y) & \dots & (X, Y) \\ (1, Y - 1) & (2, Y - 1) & \dots & (X, Y - 1) \\ \vdots & \vdots & \ddots & \vdots \\ (1, 1) & (2, 1) & \dots & (X, 1) \end{array}$$

DESCRIBE_WALL Zwraca wygląd fragmentu muru.

Parametry:

Należy przekazać dokładnie cztery liczby naturalne: pierwsza z nich to długość boku sześcianu, którego opis zostanie zwrócony, dalsze trzy opisują współrzędne środka żądanego fragmentu muru.

- L ($L \in \{3, 5, 7\}$) — długość boku żądanego fragmentu muru
- S_x ($S_x \in \mathbb{N}$, $1 \leq S_x \leq X$) — współrzędna na osi x środka żądanego fragmentu muru
- S_y ($S_y \in \mathbb{N}$, $1 \leq S_y \leq Y$) — współrzędna na osi y środka żądanego fragmentu muru
- S_z ($S_z \in \mathbb{N}$, $1 \leq S_z \leq Z$) — współrzędna na osi z środka żądanego fragmentu muru

Dane (od serwera): Serwer zwraca sześcian o boku L pół w sposób opisany w *Modelu opisu sześcianu*, przy czym pola poza formą oznaczone są jako zajęte.

DROP_BRICK Wydaje polecenie zrzucenia odpowiednio obróconej cegły nad wskazanymi polami formy.

Parametry:

Należy przekazać dokładnie 6 parametrów oddzielonych pojedynczymi spacjami: numer porządkowy wybranej cegły, trzy liczby określające o jaką całkowitą wielokrotność 90° ją kolejno obrócić w płaszczyźnie, odpowiednio, xy , xz i yz , oraz dwie liczby naturalne określające współrzędne w osiach x i y , nad którymi ma znajdować się sześcianik $(1, 1, 1)$ danej cegły. Uwaga: obracany jest cały sześcian o długości boku D użyty do zdefiniowania cegły. Sześcian ten nie musi w całości zmieścić się w formie - mieścić musi się w niej tylko sama cegła, czyli zajęte sześcianiki ją definiujące. W danej turze można wykonać tylko jeden poprawny zrzut, sumaryczna liczba zrzuceń na turę (z winy niepoprawnych zrzutów) nie może przekroczyć R (patrz: DESCRIBE_WORLD).

- ID ($ID \in \mathbb{N}$) — numer porządkowy cegły przeznaczonej do upuszczenia
- ROT_{xy} ($ROT_{xy} \in \mathbb{N}$, $0 \leq ROT_{xy} \leq 3$) — ile razy należy obrócić cegłę o 90° w płaszczyźnie xy
- ROT_{xz} ($ROT_{xz} \in \mathbb{N}$, $0 \leq ROT_{xz} \leq 3$) — ile razy należy obrócić cegłę o 90° w płaszczyźnie xz
- ROT_{yz} ($ROT_{yz} \in \mathbb{N}$, $0 \leq ROT_{yz} \leq 3$) — ile razy należy obrócić cegłę o 90° w płaszczyźnie yz
- x_1 ($x_1 \in \mathbb{Z}$, $-D + 2 \leq x_1 \leq X$) — wartość współrzędnej na osi x sześcianiku, na który trzeba zrzucić sześcianik $(1, 1, 1)$ zrucanej cegły
- y_1 ($y_1 \in \mathbb{Z}$, $-D + 2 \leq y_1 \leq Y$) — wartość współrzędnej na osi y sześcianiku, na który trzeba zrzucić sześcianik $(1, 1, 1)$ zrucanej cegły

Dane (od serwera):

Po przyjęciu zgłoszenia o prawidłowo upuszczonej (a więc mieszczącej się w całości w obrębie formy oraz pasującej do wcześniej upuszczonych cegieł) cegle serwer zwróci pojedynczą linię zawierającą następujące, oddzielone pojedynczymi spacjami wartości:

- ACCEPTED
- P ($P \in \mathbb{R}$) — liczba punktów uzyskana za zrzucenie cegły, zależna od jej wagi W_b , objętości V_b oraz pola powierzchni P_b wspólnej z już zrzuconymi cegłami i dnem formy, wyrażona wzorem:
$$P = W_b \cdot (V_b \cdot C_V + P_b \cdot C_P)$$

Jeżeli cegła nie zmieściła się w obrębie formy z powodu wartości jej współrzędnej wysokości z , serwer zwróci pojedynczą linię z napisem:

- REJECTED_HEIGHT

Jeśli cegła nie pasuje do wcześniej upuszczonych cegieł (tj. $P_b < P_b^{min}$) serwer zwróci pojedynczą linię z napisem:

- REJECTED_MATCH

WAIT Czeka do momentu rozpoczęcia kolejnej tury.

Parametry: brak

Dane (od serwera):

Serwer odpowiada jedną linią postaci:

- WAITING S

gdzie S ($S \in \mathbb{R}, 0 \leq S$) oznacza ilość sekund do zakończenia oczekiwania

7.7. Błędy

Zgodnie z opisem w rozdziale *Komunikacja z serwerem*, w wypadku podania nieprawidłowego polecenia, serwer odpowiada komunikatem:

- `'FAILED e msg'`,

gdzie e to kod błędu, a msg — komunikat błędu. W tabeli znajduje się zestawienie błędów, które mogą wystąpić przy wydawaniu poleceń w zadaniu Mur.

kod błędu	komunikat błędu
1	bad login or password
2	unknown command
3	bad format
4	too many arguments
5	internal error, sorry...
6	commands limit reached, forced waiting activated
101	incorrect identifier of brick
102	incorrect length of cube to describe
103	incorrect position to describe
104	incorrect number of rotations
105	incorrect position to drop
106	brick already dropped
107	drop attempts limit reached

7.8. Serwery

Rozgrywki będą odbywać się na trzech serwerach, różniących się między sobą parametrami, przede wszystkim maksymalnym rozmiarem bloków.

nazwa	adres:port	ograniczenia
W1	universum.dl24:20004	$D = 2, X \leq 30, Y \leq 30$
W2	universum.dl24:20005	$D = 3, X \leq 50, Y \leq 50$
W3	universum.dl24:20006	$D = 5, X \leq 30, Y \leq 30$

7.9. Przykład

Poniżej znajduje się przykładowy zapis komunikacji z serwerem.

klient → serwer	serwer → klient
	LOGIN
login1	
	PASS
secret	
	OK
DESCRIBE_WORLD	
	OK
	4 3 10 3 2 3 1.000000 3.000000 1 1.000000
LIST_BRICKS	
	OK
	7 11
SHOW_BRICK 1	
	FAILED 101 incorrect identifier of brick
SHOW_BRICK 11	
	OK
	3.500000 2
	.#.
	...
	...
	###
	#..
	...
	...
	...
	...
VIEW_FROM_ABOVE	
	OK
	0 0 0 0
	0 0 0 0
	0 0 0 0
DROP_BRICK 11 1 0 0 1 2	
	FAILED 105 incorrect position to drop
DROP_BRICK 11 1 0 0 1 1	
	OK
	REJECTED_MATCH
DROP_BRICK 11 1 0 1 0 1	
	OK
	ACCEPTED 59.500000
VIEW_FROM_ABOVE	
	OK
	1 0 0 0
	1 1 0 0
	2 0 0 0

klient → serwer	serwer → klient
DESCRIBE_WALL 3 2 2 2	OK #.. ##. #.. #..
WAIT	OK WAITING 0.21000 OK
LIST_BRICKS	OK 2 11
SHOW_BRICK 2	OK 9.00000 1 #.. #.. #..
VIEW_FROM_ABOVE	OK 1 0 0 1 1 1 0 1 2 0 1 1
DROP_BRICK 2 2 0 0 -1 1	OK ACCEPTED 54.000000
VIEW_FROM_ABOVE	OK 3 0 0 1 3 1 0 1 3 0 1 1