

deadline **24**

EDYCJA **2015**



ZASADY I ZESTAW ZADAŃ

Gliwice, 7 – 8 kwietnia 2015

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. Energetyczny kocioł	5
5.1. Wprowadzenie	5
5.2. Zadanie	5
5.3. Model rozgrywki	5
5.4. Świat	6
5.4.1. Rodzaje terenu	6
5.4.2. Punkty Strategiczne	6
5.5. Jednostki	6
5.6. Specjalne funkcje jednostek	8
5.6.1. Działo	8
5.6.2. Sztab	8
5.7. Szczegóły dot. widoczności	9
5.8. Rozpoczęcie i zakończenie	10
5.9. Cel rozgrywki i rywalizacja	10
5.10. Komendy	12
5.11. Błędy	15
5.12. Serwery	15
5.13. Przykład	16
5.14. Dodatek A	18
6. Roboty	22
6.1. Wstęp	22
6.2. Zadanie	22
6.3. Model rozgrywki	22
6.4. Części do robotów	23
6.5. Zakupy	23
6.5.1. Planowanie	23
6.5.2. Aukcje	23
6.5.3. Wyprzedaż	24
6.6. Budowa	24
6.6.1. Robot	24
6.7. Walka	25
6.7.1. Walki na wspólnej arenie	25
6.7.2. Wyniki walk	25
6.8. Punktacja	25
6.9. Komendy	27
6.10. Błędy	32
6.11. Serwery	32
6.12. Przykład	33

7. Wszystko płynie	37
7.1. Wprowadzenie	37
7.2. Zadanie	37
7.3. Model rozgrywki	37
7.4. Świat	37
7.4.1. Obiekty	37
7.4.2. Fala powodziowa	38
7.4.3. Prognoza	38
7.5. Żukoskoczki	38
7.6. Rozpoczęcie i zakończenie	39
7.7. Cel rozgrywki i rywalizacja	40
7.8. Punktacja	40
7.9. Komendy	41
7.10. Błędy	45
7.11. Serwery	45
7.12. Przykład	46

1. Wstępniak

To już siódmy finał maratonu programistycznego Deadline24, organizowanego przez firmę Future Processing pod patronatem czterech uczelni wyższych oraz Ministerstwa Administracji i Cyfryzacji. Po raz pierwszy znajdujemy się w Galerii Sztuki Współczesnej „Elektrownia” w Czeladzi, której wnętrza doskonale nawiązują do postindustrialnego dziedzictwa województwa śląskiego.

Podobnie jak w ubiegłych latach oddajemy w Wasze ręce trzy zadania, tradycyjnie osadzone w świecie intrygujących stworzeń – żukoskoczków. Tym razem ich perypetie w odkrywaniu tajemnic świata dotyczą poszukiwania cennych surowców (*Energetyczny kociół*) oraz automatyzowania swoich podbojów (*Roboty*). Ponadto, los postawi żukoskoczki wobec konieczności walki z żywiołem (*Wszystko płynie*). Będziecie mogli im w tym wszystkim pomóc. Z jakim efektem? Przekonamy się niedługo.

Mamy nadzieję, że rozgrywka będzie emocjonująca, a 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: 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:

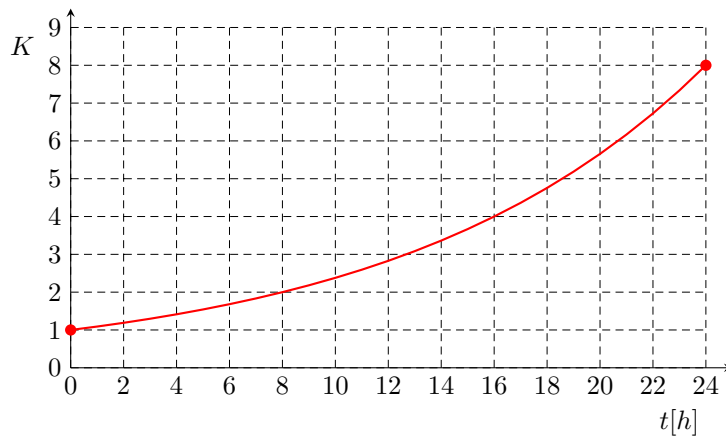
- Każda linia zakończona jest pojedynczym znakiem o kodzie ASCII 10 ('\n'). Znak powrotu karetki ('\r'; kod 13) towarzyszący mu na niektórych systemach operacyjnych będzie traktowany jako biały znak.

- 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. Energetyczny kocioł

5.1. Wprowadzenie

Posiadanie licznej armii umożliwiło Żukoskoczkom szybką i intensywną eksplorację odległych zakątków Universum. Jednak ostatnie potyczki w rejonie galaktyki Żukocieni zdziętkowały oddziały żołnierzy i zmusiły dowództwo do zweryfikowania dotychczasowych strategii.

Pułkownik Kretschmer długo analizował zebrane dane i doszedł do wniosku, że konieczne będzie utworzenie niezależnej, wydajnej i elitarnej jednostki (*Independent Efficient Elitist Entity* — IEEE). Niewielki zespół złożony z komandosów „gotowych na wszystko” miałby skutecznie przeprowadzać zwiad, infiltrować nieznanne środowiska, a w razie potrzeby organizować doraźne akcje sabotażowe. Jak jednak znaleźć fachowców o właściwych predyspozycjach do takiej grupy specjalnej?

„*Upieczemy dwie pieczenie na jednym ogniu. Już ja się o to postaram!*” – powiedział Generał Żukk, gdy wysłuchał pomysłu pułkownika. Od dłuższego czasu przygotowywał bowiem plan pojmania i postawienia przed sądem wojennym kilku grup dezertersów i renegatów. Starannie przygotowana intryga, której częścią był kontrolowany wyciek informacji, miała się niedługo ziścić.

Generał dopilnował, aby „utrzymywana w tajemnicy” wiadomość o odkryciu nowej planety Pow^er Plan_et z picartium – wysokoenergetycznym surowcem – dotarła do poszukiwanych. To miejsce stanowiło idealną pułapkę dla wszystkich wyjętych spod prawa i poszukujących szybkiego zysku. „*Nie pojmiemy ich od razu. Poczekamy, przyjrzymy się dokładnie jak walczą między sobą, a najlepszym złożymy ofertę nie do odrzucenia...*” – dodał Generał Żukk uśmiechając się szyderczo.

Kilka dni później obaj wojskowi zacierali ręce wpatrując się w olbrzymie monitory, które wyświetliły pierwsze informacje o aktywności na obserwowanej planecie.

5.2. Zadanie

Działanie na korzyść Imperium nie było nigdy dla Waszej grupy priorytetem. Korzystając z okazji „oddaliliście się”, by prowadzić niezależne interesy. Teraz wiedzeni chęcią zysku przybywacie na Pow^er Plan_et. Jesteście przygotowani na drobne starcia z innymi grupami dezertersów. Liczycie także na zdobycie jak największej liczby punktów pozyskiwania picartium, które można intratnie odsprzedać wszystkim elektrowniom w Universum. Niestety, pułapka głównodowodzącego jest bezbłędna. Wkrótce okaże się, że jesteście otoczeni przez liczną armię Generała Żukka i los Waszego zespołu jest niemal przesądzony.

Jedyną szansą na ocalenie jest wykazanie się wysokimi umiejętnościami na polu walki, by zostać wytypowanym do IEEE. W tym celu musicie jak najbardziej osłabić pozycje przeciwników i jak najdłużej utrzymywać kontrolę w zadanych punktach strategicznych.

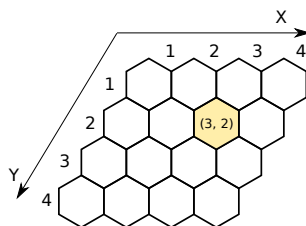
5.3. Model rozgrywki

Rozgrywka odbywa się w turach o równej długości. W trakcie trwania każdej tury drużyny komunikują się z serwerem i wydają polecenia swoim jednostkom.

5.4. Świat

Rozgrywka toczy się na powierzchni pewnego obszaru, na którym możemy wyróżnić kilka rodzajów terenu. Obszar jest reprezentowany przez siatkę heksagonalną składającą się z $N \times N$ 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 .

Na Rysunku 2 przedstawiono sposób indeksowania pól, za pomocą współrzędnych X i Y . Poza skrajnymi polami każde ma dokładnie sześć pól sąsiednich. Wyróżnione pole ma współrzędne $(3, 2)$.



Rysunek 2: Sposób indeksowania pól na obszarze rozgrywki dla $N = 4$.

Jednostki, którymi dysponują drużyny, mogą przemieszczać się pomiędzy polami, a także atakować inne pola zgodnie z ograniczeniami dotyczącymi widoczności oraz zasięgu strzału.

5.4.1. Rodzaje terenu

Każde pole reprezentuje dokładnie jeden rodzaj terenu: teren nizinny, wzniesienie, las, zarośla lub akwen wodny. Rodzaj terenu modyfikuje podstawowe parametry jednostek dotyczące ich poruszania się, widoczności oraz prowadzenia ataku – wszystkie te modyfikacje przedstawiono w Tabeli 1.

Przykłady tego, w jaki sposób rodzaj terenu wpływa na widoczność i możliwość atakowania, przedstawiono w Dodatku A.

5.4.2. Punkty Strategiczne

Na niektórych polach występują bogate złoża poszukiwanego surowca. Miejsca takie z uwagi na ich szczególne znaczenie nazywamy Punktami Strategicznymi. Mogą one znajdować się na dowolnym polu, niezależnie od rodzaju terenu. Zespoły rywalizują między sobą o kontrolę nad takimi miejscami.

Aby przejąć kontrolę nad Punktem Strategicznym, dowolna jednostka drużyny musi znaleźć się na jego polu. Kontrola jest utrzymywana niezależnie od dalszej obecności na tym polu jednostek drużyny aktualnie sprawującej kontrolę. Utrata kontroli może nastąpić tylko po wkroczeniu na pole z Punktem Strategicznym jednostek innej drużyny. Punkty za kontrolowanie każdego Punktu Strategicznego naliczane są za każdą turę, o ile po jej zakończeniu dany Punkt Strategiczny był pod kontrolą drużyny.

5.5. Jednostki

Każda drużyna rozporządza pewną liczbą jednostek czterech różnych typów: sztab, wóz pancerny, czołg oraz działo. Typy jednostek różnią się od siebie własnościami: maksymalną wytrzymałością, siłą ataku, zasięgiem strzału i szybkością poruszania. Podstawowym mechanizmem walki między drużynami jest ostrzeliwanie wrogich jednostek, co zmniejsza ich bieżącą wytrzymałość. Jednostka zostaje bezpowrotnie zniszczona, jeśli jej bieżąca wytrzymałość spadnie do zera.

Ruch jednostek Każda jednostka zajmuje dokładnie jedno pole obszaru i nie może go dzielić z żadną inną jednostką. Jednostki mogą wykonać ruch i przemieścić się o jedno pole w ramach obszaru. Ruch wykonywany jest w jednym z sześciu dopuszczalnych kierunków, o ile pole docelowe nie jest zajmowane przez inną jednostkę, nie jest akwenem wodnym i nie leży poza obszarem rozgrywki.

Ruch do sąsiedniego pola trwa od jednej do trzech tur. W przypadku ruchu trwającego jedną turę, zostanie on zakończony na początku następnej tury po wydaniu rozkazu. Analogicznie będzie dla ruchu trwającego więcej tur.

Próba wykonania ruchu na pole zajęte przez inną jednostkę nie powiedzie się – jednostka dokonująca próby pozostanie na bieżącym polu. Wyznaczenie potencjalnej kolizji następuje na początku tury, w której

Tabela 1: Wpływ rodzaju terenu na akcje podejmowane przez jednostki.

Rodzaj terenu	Wpływ na poruszanie	Wpływ na widoczność*	Wpływ na prowadzenie ataku
Teren nizinny	• brak	• brak	• brak
Wzniesienie	<ul style="list-style-type: none"> • wejście na wzniesienie z innego rodzaju pola trwa jedną turę dłużej • wzniesienie nie wpływa na czas zejścia z niego na inne pole 	<ul style="list-style-type: none"> • ogranicza widoczność między dwoma polami, poza przypadkiem, gdy oba te pola są wzniesieniami • dla jednostki na wzniesieniu: pole widzenia dwa razy większe w odniesieniu do terenów nizinnych, lasów, zarośli i akwenów wodnych (niezasłoniętych przez inne wzniesienia lub lasy) 	<ul style="list-style-type: none"> • ostrzeliwując pole inne niż wzniesienie: +1 pole do górnej granicy zasięgu strzału • jednostki, które nie są na wzniesieniach nie blokują linii ognia jednostkom będącym na wzniesieniu
Las	<ul style="list-style-type: none"> • wejście w las z dowolnego pola trwa jedną turę dłużej 	<ul style="list-style-type: none"> • ogranicza widoczność, poza przypadkiem widoczności między dwoma wzniesieniami 	<ul style="list-style-type: none"> • obrażenia zadane jednostce będącej w lesie są mniejsze (patrz: sekcja 5.5.)
Zarośla	<ul style="list-style-type: none"> • jak dla lasu 	<ul style="list-style-type: none"> • ogranicza widoczność, poza przypadkiem widoczności ze wzniesienia sięgającej poza pole z zaroślami 	<ul style="list-style-type: none"> • jak dla lasu
Akwen wodny	<ul style="list-style-type: none"> • nie można wejść na pole 	<ul style="list-style-type: none"> • brak 	<ul style="list-style-type: none"> • brak

* jednostki znajdujące się na bezpośrednio sąsiadujących polach **zawsze** widzą się wzajemnie, niezależnie od rodzaju terenu, na którym się znajdują

ruch powinien się zakończyć. Jedynym wyjątkiem jest pole zajmowane przez sztab, które może zostać zajęte przez wrogą jednostkę (nie można natomiast wejść własną jednostką na pole zajmowane przez własny sztab).

Nowe położenia jednostek ustalane są na początku każdej tury (zanim gracze podejmą jakiegokolwiek dalsze działania) i z zachowaniem kolejności w jakiej zostały wydane polecenia ruchu w poprzednich turach.

Pole widzenia Każda jednostka posiada określone pole widzenia – 4 lub 5 pól od swojego aktualnego położenia. Pole widzenia może być zwiększane lub ograniczane przez teren: wzniesienia, zarośla i lasy. Obecność innych jednostek nie ma wpływu na pole widzenia.

Dla wozu pancernego oraz czołgu, posiadanie atakowanej jednostki w swoim polu widzenia jest warunkiem koniecznym do przeprowadzania ataku.

Zasięg strzału Zasięg strzału to maksymalna odległość, wyrażona w polach, na jaką jednostka może oddać strzał. Warunkiem przeprowadzenia ostrzału innej jednostki jest:

- obecność celu w zasięgu strzału,
- obecność celu w polu widzenia (nie dotyczy działa),
- brak innych jednostek na linii ognia (nie dotyczy działa).

Jeśli na linii ognia będzie inna jednostka niż cel (dotyczy to jednostek będących na wysokości strzelającego, które znajdują się pomiędzy nim a celem), to właśnie ona zostanie trafiona, blokując ostrzał docelowego pola.

Typy jednostek Podstawowe różnice pomiędzy jednostkami poszczególnych typów zostały przedstawione w Tabeli 2.

Tabela 2: Zróżnicowanie parametrów jednostki w zależności od jej typu.

Typ jednostki	Szybkość poruszania	Maksymalna wytrzymałość	Siła ataku	Siła ataku – las*	Zasięg strzału	Pole widzenia	Funkcje specjalne
Sztab	1 pole / 1 turę	brak	brak	brak	brak	4	produkcja, naprawa, wzmacnianie jednostek
Wóz pancerny	1 pole / 1 turę	10	2	1	1 – 4	5	brak
Czołg	1 pole / 2 tury	20	5	4	1 – 5	4	brak
Działo	1 pole / 2 tury	7	12	9	3 – 8	4	ostrzeliwanie ponad przeszkodami

* dotyczy przypadku ostrzeliwania jednostek znajdujących się w lesie lub w zaroślach

5.6. Specjalne funkcje jednostek

Dwa rodzaje jednostek odznaczają się specjalnymi możliwościami, które mogą być bardzo przydatne w zdobywaniu przewagi nad pozostałymi drużynami.

5.6.1. Działo

Dzięki specjalnym właściwościom ofensywnym, ta jednostka może prowadzić ostrzał ponad przeszkodami i jednostkami. Działo zawsze ma „czystą” linię ognia – może atakować dowolne pole będące w swoim zasięgu strzału. Przygotowanie tak potężnego ładunku jakim dysponuje działo wymaga od żukoskoczków dodatkowego wysiłku, stąd każda jednostka tego typu może atakować tylko raz na dwie tury.

5.6.2. Sztab

Ta jednostka nie może atakować innych, ale też sama nie może zostać ostrzelana. Sztab blokuje linię ognia, jednak pomimo tego nie otrzymuje żadnych obrażeń. Jediną możliwością zniszczenia sztabu jest wejście wrogiej jednostki na pole zajmowane przez sztab.

Jednostka sztabu zwiększa możliwości jednostek znajdujących się w jej pobliżu, umożliwia ich naprawę oraz produkcję nowych jednostek.

Zwiększone możliwości pobliskich jednostek Obecność sztabu pozytywnie wpływa na morale walczących. Dzięki temu jednostki, które znajdują się w odległości do 3 pól od swojego sztabu, odznaczają się lepszymi właściwościami. Zmiany są przedstawione w Tabeli 5.

Naprawa jednostek Jednostka znajdująca w odległości nie większej niż 3 pola od swojego sztabu, może zostać poddana naprawie. Naprawy dokonuje się kosztem 3 punktów zwycięstwa.

Podczas naprawy można odzyskać do 50% maksymalnej wytrzymałości dowolnej jednostki, zaokrąglając w dół. Wytrzymałość jednostki po naprawie to suma wytrzymałości sprzed naprawy i 50% maksymalnej wytrzymałości tej jednostki, o ile ta suma nie przekracza wytrzymałości maksymalnej.

Naprawa trwa do początku następnej tury (jej wytrzymałość jest zaktualizowana natychmiast) – w tym czasie jednostka nie może się poruszać ani podejmować kolejnej naprawy.

Tabela 3: Wpływ bliskości sztabu na parametry jednostek.

Typ jednostki	Wzmocniona siła ataku	Wzmocniona siła ataku – las*
Wóz pancerny	3	2
Czołg	7	5
Działo	15	12

* dotyczy ostrzeliwania jednostek znajdujących się w lesie lub w zaroślach

Produkcja jednostek Sztab umożliwia produkcję jednostek. Utworzenie nowej jednostki, niezależnie od jej typu, obarczone jest tym samym kosztem – należy poświęcić 10 punktów zwycięstwa. Nie można wyprodukować kolejnego sztabu. Dodatkowym warunkiem jest możliwość ulokowania nowej jednostki wyłącznie w bezpośrednim sąsiedztwie sztabu, tzn. przynajmniej jedno z jego sąsiednich pól musi być nieobsadzone jednostką i różne od akwenu wodnego. Jeśli takich wolnych pól wokół sztabu jest więcej, nowa jednostka pojawi się na polu losowo wybranym spośród nich.

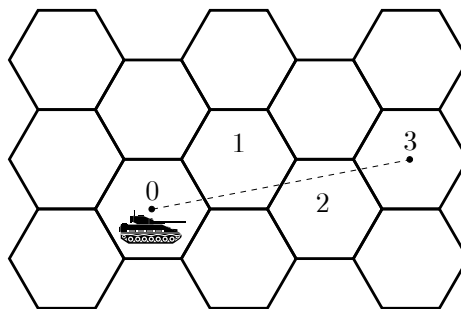
Rozszerzanie oddziałów o dodatkowe jednostki wymaga odpowiedniego planowania i rozważenia. Wasza drużyna ma ograniczone możliwości dowodzenia grupą jednostek – z góry znany jest limit liczby poleceń L , jakie można wydać w jednej turze rozgrywki. Dodatkowo drużyna nie może posiadać więcej niż 5 dział.

5.7. Szczegóły dotyczące widoczności

Pole widzenia jednostki może być ograniczone przez wniesienia, zarośla oraz lasy. Aby stwierdzić, czy konkretne pole jest widoczne dla danej jednostki, wykonuje się następujące kroki:

- wyznacza się odcinek łączący środki dwóch pól – tego, na którym stoi jednostka i docelowego,
- tworzy się listę rodzajów pól, które przecina wyznaczony odcinek,
- na podstawie listy rodzajów pól, określa się widoczność zgodnie z zasadami podanymi w sekcji 5.4.1.

Na Rysunku 3 przedstawiono przykładowy odcinek pomocny przy analizie widoczności pola z numerem 3 dla jednostki znajdującej się na polu z numerem 0. Każde pole oznaczone na rysunku cyfrą – zależy od rodzaju terenu – może mieć wpływ na wspomnianą widoczność.

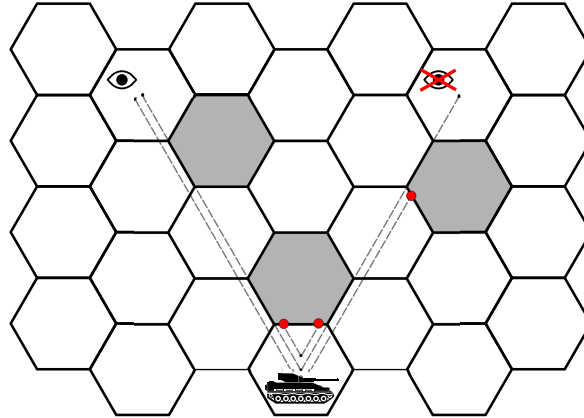


Rysunek 3: Wyznaczanie listy pól wpływających na widoczność pola z numerem 3.

Szczególny przypadek wyznaczania pola widzenia jednostki (odcinek łączący środki pól pokrywa się z krawędziami niektórych pól) został przedstawiony na Rysunku 4. Zaciemnione miejsca oznaczają pola ograniczające widoczność. Sytuacja taka rozpatrywana jest w następujący sposób:

- po obu stronach pierwotnego odcinka widoczności wyznacza się dwa dodatkowe odcinki do niego równoległe, w odległości nie większej niż połowa długości boku pola rozgrywki,
- sprawdza się, czy pole docelowe jest widoczne za pośrednictwem pól przecinanych przez dodatkowe odcinki (dla każdego oddzielnie),

- pole docelowe jest widoczne dla jednostki, jeśli jest ono widoczne za pośrednictwem któregośkolwiek z dodatkowych odcinków – w przeciwnym wypadku jest niewidoczne.



Rysunek 4: Szczególny przypadek określania widoczności pola.

Analogicznie określa się, czy dana jednostka znajduje się na linii ognia innej jednostki, z tą różnicą, że pola zajmowane przez jednostki traktuje się jako blokujące – ostrzelanie jednostki znajdującej się za inną spowoduje trafienie jednostki blokującej (nie dotyczy działa).

Jeżeli – w szczególnym przypadku – jednostka docelowa jest osłonięta przez dwie różne jednostki (po obu stronach pierwotnego odcinka służącego do analizy widoczności), zostanie trafiona jednostka położona bliżej pola docelowego lub, dla przypadku równych odległości, ta położona po prawej stronie z punktu widzenia atakującego.

5.8. Rozpoczęcie i zakończenie

Początkowy stan rozgrywki Każda drużyna na początku rozgrywki dysponuje tą samą liczbą jednostek poszczególnych typów. Jednostki każdej drużyny zgromadzone są w pobliżu należącego do niej sztabu, a bieżąca wytrzymałość każdej z nich jest maksymalna.

Zakończenie rozgrywki Od początku rozgrywki znana jest maksymalna liczba tur do jej zakończenia. Generał Żuk może jednak wcześniej stracić cierpliwość do przyglądania się walczącym i wkroczyć ze swoimi oddziałami, które uniemożliwią dalsze działania dezertorów.

Oceniając walczące zespoły, generał bierze pod uwagę następujące aktywności: trafienie we wrogą jednostkę, zajęcie sztabu oraz przejście Punktu Strategicznego. Jeśli – w rozumieniu Generała Żukka – drużyny nie będą wykazywały aktywności na polu walki w ciągu kolejnych E_I tur, spowoduje to skrócenie rozgrywki. W takim wypadku, rozpocznie się odliczanie E_T tur do przybycia armii żukoskoczków.

Procedura skrócenia rozgrywki nie będzie uruchomiona wcześniej niż po E_S turach od jej rozpoczęcia.

5.9. Cel rozgrywki i rywalizacja

Waszym celem jest osłabienie pozycji przeciwników oraz jak najdłuższe utrzymywanie kontroli w Punktach Strategicznych. Za każde z działań, które pomogą w realizacji tego zadania, zespół otrzyma odpowiednią liczbę punktów zwycięstwa.

Nadrzędnym priorytetem drużyny jest posiadanie jak największej liczby punktów zwycięstwa w chwili zakończenia rozgrywki.

Punktacja Punkty zwycięstwa przydzielane są za zniszczenie jednostki przeciwnika, za przejście sztabu oraz za kontrolowanie Punktów Strategicznych.

Niszczenie jednostek Za każdą zniszczoną jednostkę wroga, niezależnie od jej typu, drużyna otrzymuje 5 punktów zwycięstwa. Punkty te naliczane są natychmiast po dokonaniu zniszczenia.

Przejmowanie sztabów Za przejście sztabu przeciwnika drużyna otrzymuje jednorazowo określoną liczbę punktów zwycięstwa O . Wartość ta jest stała dla danej rozgrywki, a między rozgrywkami zależy wyłącznie od początkowej liczby jednostek każdego rodzaju.

Kontrolowanie Punktów Strategicznych Jeśli drużyna przejęła kontrolę nad Punktem Strategicznym, to za każdą turę utrzymywania tej kontroli otrzymuje ona określoną liczbę punktów zwycięstwa S . Wartość ta jest stała dla danej rozgrywki, a między rozgrywkami zależy od: początkowej liczby jednostek każdego rodzaju, maksymalnej długości rozgrywki oraz liczby Punktów Strategicznych na obszarze rozgrywki.

Punkty za kontrolowanie każdego Punktu Strategicznego naliczane są niezależnie.

5.10. Komendy

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

DESCRIBE_WORLD Zwraca parametry rozgrywki i wartość współczynnika skalującego wynik.

Parametry: brak

Dane (od serwera):

W pojedynczej linii serwer zwróci następujące wartości oddzielone pojedynczymi spacjami:

- N ($N \in \mathbb{N}$, $1 \leq N \leq 200$) — długość boku planszy,
- T ($T \in \mathbb{N}$, $1 \leq T \leq 3$) — czas trwania pojedynczej tury w sekundach,
- O ($O \in \mathbb{N}$, $1 \leq O \leq 200$) — liczba punktów przyznawana za przejście sztabu przeciwnika,
- S ($S \in \mathbb{N}$, $1 \leq S \leq 100$) — liczba punktów przyznawana za kontrolowanie Punktu Strategicznego,
- E_S ($E_S \in \mathbb{N}$) — okres karencji (w turach) dotyczący procedury skrócenia rozgrywki,
- E_I ($E_I \in \mathbb{N}$) — okres braku aktywności (w turach), po którym uruchamiana jest procedura skrócenia rozgrywki,
- E_T ($E_T \in \mathbb{N}$) — długość procedury skrócenia rozgrywki w turach,
- L ($L \in \mathbb{N}$, $1 \leq L \leq 100$) — maksymalna liczba poleceń, jaką można wykonać w jednej turze,
- K ($K \in \mathbb{R}$, $1 \leq K \leq 8$) — wartość współczynnika skalującego wynik.

TIME_TO_END Zwraca liczbę tur pozostałych do końca rozgrywki, wliczając aktualną turę.

Parametry: brak

Dane (od serwera):

Serwer zwraca jedną wartość:

- Q ($Q \in \mathbb{N}$) — liczba tur pozostałych do zakończenia rozgrywki.

LIST_UNITS Zwraca listę jednostek podlegających dyspozycji drużyny.

Parametry: brak

Dane (od serwera):

Pierwsza linia:

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

Każda kolejna linia opisuje jedną jednostkę za pomocą oddzielonych pojedynczymi spacjami wartości oznaczających kolejno:

- ID ($ID \in \mathbb{N}$) — identyfikator jednostki,
- P_X ($P_X \in \mathbb{N}$, $1 \leq P_X \leq N$) — współrzędna X pola, w którym znajduje się jednostka,
- P_Y ($P_Y \in \mathbb{N}$, $1 \leq P_Y \leq N$) — współrzędna Y pola, w którym znajduje się jednostka,
- $Type$ ($Type \in \{\text{CANNON, TANK, ARMORED, HQ}\}$) — typ jednostki (kolejno: działko, czołg, wóz pancerny, sztab),
- W ($W \in \mathbb{N} \cup \{\text{INF}\}$) — aktualna wytrzymałość jednostki (dla sztabu – INF),
- C ($C \in \mathbb{N}$, $0 \leq C \leq 3$) — liczba kolejnych tur, w których jednostka będzie zajęta wykonywaniem ruchu.

MOVE Wydaje jednostce polecenie przejścia na sąsiednie pole, wskazywane przy pomocy współrzędnych względnych. Wykonanie operacji jest opóźnione o tyle tur, ile wynosi czas przemieszczenia się o jedno pole. Wartość przynajmniej jednej ze współrzędnych (D_X , D_Y) musi być różna od zera, a ich iloczyn nie może być ujemny.

Parametry:

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

CANCEL_MOVE Wycofuje rozkaz przejścia na sąsiadujące pole dla zadanej jednostki.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID jednostki, której ruch zostanie anulowany.

SHOOT Wydaje jednostce polecenie ataku wybranego pola wskazywanego przy pomocy współrzędnych względnych. Pole musi znajdować się w zasięgu strzału jednostki. Jeśli na drodze pocisku stoi inna jednostka, to ona zostanie trafiona. Można zaatakować własne jednostki. Bezpośredni ostrzał pola, na którym znajduje się sztab jest niemożliwy. Operacja wykonywana jest natychmiast. Dla działła komendę można wykonać raz na 2 tury.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID jednostki,
- D_X ($D_X \in \mathbb{N}, -9 \leq D_X \leq 9$) — przesunięcie wzdłuż osi X ,
- D_Y ($D_Y \in \mathbb{N}, -9 \leq D_Y \leq 9$) — przesunięcie wzdłuż osi Y .

Dane (od serwera):

Serwer zwraca parę współrzędnych w jednej linii, oddzielone spacją:

- H_X ($H_X \in \mathbb{N}, 1 \leq H_X \leq N$) — współrzędna X trafionego pola,
- H_Y ($H_Y \in \mathbb{N}, 1 \leq H_Y \leq N$) — współrzędna Y trafionego pola.

CREATE_UNIT Tworzy nową jednostkę, która pojawi się na losowym wolnym polu w sąsiedztwie sztabu.

Parametry:

- $UnitType$ ($UnitType \in \{\text{CANNON}, \text{TANK}, \text{ARMORED}\}$) — odpowiednio: działło, czołg, wóz pancerny.

Dane (od serwera):

- ID ($ID \in \mathbb{N}$) — numer ID jednostki,
- P_X ($P_X \in \mathbb{N}, 1 \leq P_X \leq N$) — współrzędna X pola, w którym znajduje się jednostka,
- P_Y ($P_Y \in \mathbb{N}, 1 \leq P_Y \leq N$) — współrzędna Y pola, w którym znajduje się jednostka.

REPAIR_UNIT Naprawia jednostkę, która znajduje się w zasięgu działania sztabu.

Parametry:

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

LOOK_AROUND Zwraca opis obszaru rozgrywki – informację o rodzaju terenu każdego pola. Komendę w danej rozgrywce można wykonać raz na 100 tur.

Parametry: brak

Dane (od serwera):

Zwracanych jest N linii (patrz: DESCRIBE_WORLD), z których każda zawiera N znaków. Każda kolejna linia (od 1 do N – według rosnących wartości współrzędnej Y) opisuje jeden wiersz obszaru rozgrywki w postaci listy liter odpowiadających rodzajom pól:

- $Type$ ($Type \in \{H, B, F, L, P\}$) — typ pola, gdzie H to wzniesienie, B to zarośla, F to las, L to akwen wodny, a P to teren płaski.

Litery czytane w każdej linii od lewej do prawej odpowiadają polom obszaru rozgrywki zgodnie z rosnącą wartością współrzędnej X (od 1 do N).

LIST_ENEMIES Zwraca listę wrogich jednostek w zakresie sumarycznego pola widzenia wszystkich jednostek należących do drużyny. Informacje dotyczą stanu z początku aktualnej tury.

Parametry: brak

Dane (od serwera):

Pierwsza linia:

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

Każda kolejna linia opisuje jedną jednostkę wroga za pomocą oddzielonych pojedynczymi spacjami wartości określających kolejno:

- P_X ($P_X \in \mathbb{N}$, $1 \leq P_X \leq N$) — współrzędna X pola,
- P_Y ($P_Y \in \mathbb{N}$, $1 \leq P_Y \leq N$) — współrzędna Y pola,
- $Type$ ($Type \in \{\text{CANNON, TANK, ARMORED, HQ}\}$) — typ jednostki (kolejno: działo, czołg, wóz pancerny, sztab),
- W ($W \in \mathbb{N} \cup \{\text{INF}\}$) — aktualna wytrzymałość jednostki (dla sztabu – INF),
- $Team$ ($Team \in \mathbb{N}$) — identyfikator drużyny (przypisywany jest losowo na początku każdej rozgrywki).

LIST_STRATEGIC_POINTS Zwraca listę wszystkich punktów strategicznych.

Parametry: brak

Dane (od serwera):

Pierwsza linia:

- B ($B \in \mathbb{N}$) — liczba punktów strategicznych.

Kolejne B linii zawiera opisy pól. Każda linia dotyczy jednego pola – za pomocą oddzielonych pojedynczymi spacjami wartości opisane są kolejno:

- P_X ($P_X \in \mathbb{N}$, $1 \leq P_X \leq N$) — współrzędna X pola,
- P_Y ($P_Y \in \mathbb{N}$, $1 \leq P_Y \leq N$) — współrzędna Y pola,
- $Team$ ($Team \in \mathbb{N} \cup \{\text{NONE, YOURS}\}$) — identyfikator drużyny kontrolującej dany punkt strategiczny. Gdy punkt nie należy do żadnej drużyny zwracana jest wartość NONE. Gdy punkt należy do drużyny wywołującej komendę, zwracana jest wartość YOURS.

MY_STAT Zwraca informacje dotyczące osiągnięć drużyny.

Parametry: brak

Dane (od serwera):

Serwer zwróci jedną wartość:

- $Points$ ($Points \in \mathbb{N}$) — liczba posiadanych w tej rozgrywce punktów zwycięstwa.

WAIT Czeka do momentu rozpoczęcia kolejnej tury.

Parametry: brak

Dane (od serwera):

Serwer odpowiada jedną linią postaci:

- WAITING S

gdzie S ($S \in \mathbb{R}$, $S \geq 0$) oznacza liczbę sekund do zakończenia oczekiwania. Po upływie tego czasu wysyła dodatkowo linię:

- OK

5.11. 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 Energetyczny kocioł.

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
100	incorrect identifier
101	incorrect destination terrain
102	incorrect coordinates
103	incorrect unit type
104	not enough points
105	no empty field
106	usage limit reached for this command
107	hq does not exist
108	no more shoots available
109	target out of sight
110	invalid target
111	out of shoot range
112	unit not moving
113	unit already moving
115	no more actions available for this unit

5.12. Serwery

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

Tabela 4: Adresy i parametry serwerów rozgrywek.

Nazwa	Adres:Port	Typ obszarów zielonych	Początkowa liczba jednostek
Battle1	universum.dl24:20000	zarośla	nieznana
Battle2	universum.dl24:20001	las	nieznana
Battle3	universum.dl24:20002	zarośla	mała

5.13. Przykład

Poniżej znajduje się przykładowy zapis komunikacji z serwerem.

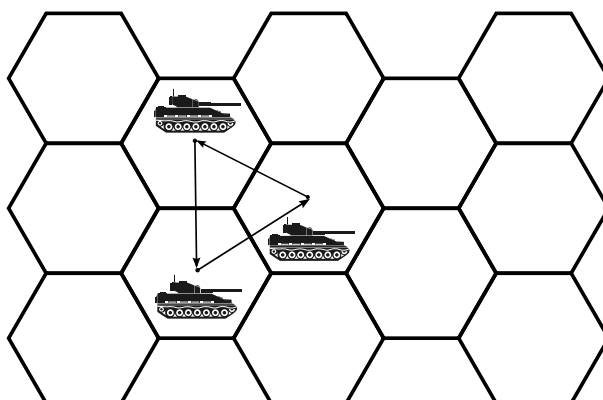
klient → serwer	serwer → klient
	LOGIN
login1	PASS
secret	OK
DESCRIBE_WORLD	OK
	150 1 60 3 20 20 30 40 7.200000
LIST_UNITS	OK
	5
	1 4 4 HQ INF 0
	5 4 3 ARMORED 10
	10 3 3 CANNON 7
	11 4 5 ARMORED 10
	12 5 4 TANK 20
TIME_TO_END	OK
	200
LIST_STRATEGIC_POINTS	OK
	5
	1 1 4
	7 7 3
	4 4 YOURS
	7 6 3
	2 5 NONE
MOVE 11 1 1	OK
MOVE 12 1 1	FAILED 101 incorrect destination terrain
LIST_ENEMIES	OK
	3
	1 1 ARMORED 5 4
	1 2 ARMORED 10 4
	7 6 CANNON 6 3
MY_STAT	OK
	17
SHOOT 5 -3 -2	OK
	1 1

```
SHOOT 12 2 3 |
               | OK
               | 7 7
LIST_ENEMIES  |
               | OK
               | 3
               | 1 2 ARMORED 10 4
               | 7 6 CANNON 1 3
MY_STAT       |
               | OK
               | 22
WAIT         |
               | OK
               | WAITING 0.420000
               | OK
LIST_UNITS   |
               | OK
               | 5
               | 1 4 4 HQ INF 0
               | 5 4 3 ARMORED 10
               | 10 3 3 CANNON 7
               | 11 5 6 ARMORED 10
               | 12 5 4 TANK 20
MY_STAT      |
               | OK
               | 25
```

5.14. Dodatek A

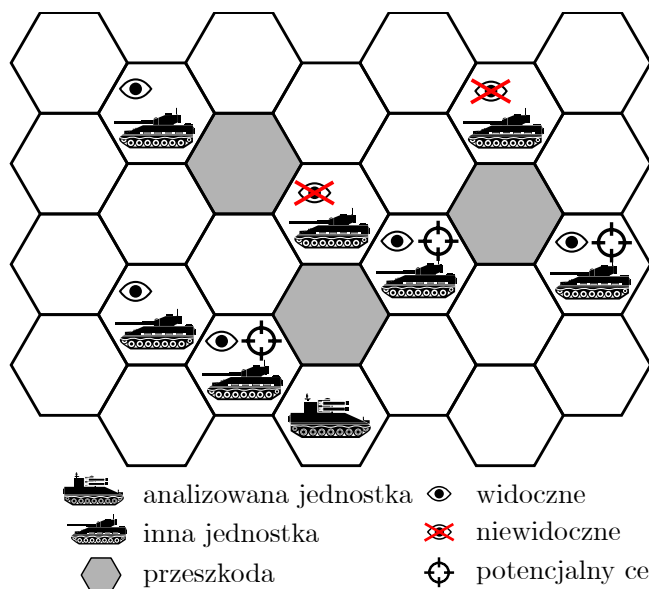
Poniżej zobrazowano różne przypadki dotyczące możliwości ruchu, widoczności oraz linii ognia jednostek w zależności od rodzaju terenu.

Impas w przemieszczaniu Na Rysunku 5 przedstawiono sytuację, w której żadna jednostka nie zostanie przemieszczona, niezależnie od kolejności wydawania rozkazów. Szczegółowy opis możliwości wykonywania ruchu znajduje się w sekcji 5.5.



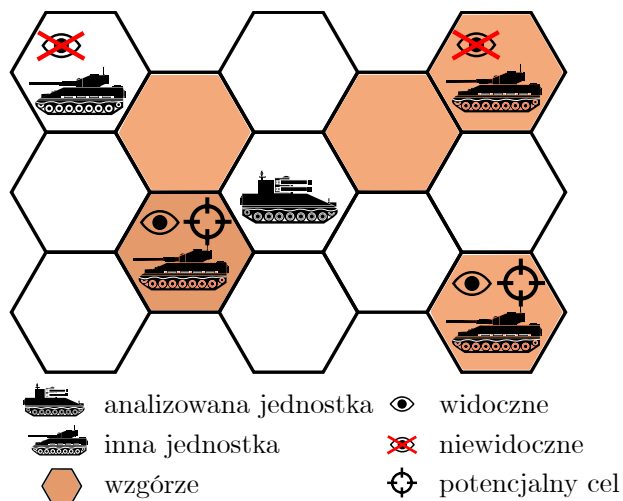
Rysunek 5: Impas w wykonywaniu ruchu.

Atakowanie jednostek Na Rysunku 6 zobrazowano zasady dotyczące widoczności i linii ognia. Opis tego zagadnienia znajduje się w sekcji 5.5. oraz 5.7.

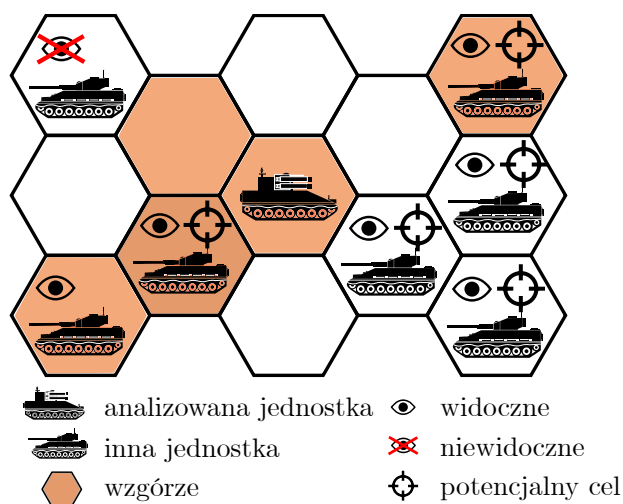


Rysunek 6: Widoczność i możliwość przeprowadzenia ataku.

Wpływ wzniesienia na widoczność Poniższe rysunki przedstawiają możliwe sytuacje dotyczące widoczności i linii ognia związane z usytuowaniem jednostki względem wzniesienia.

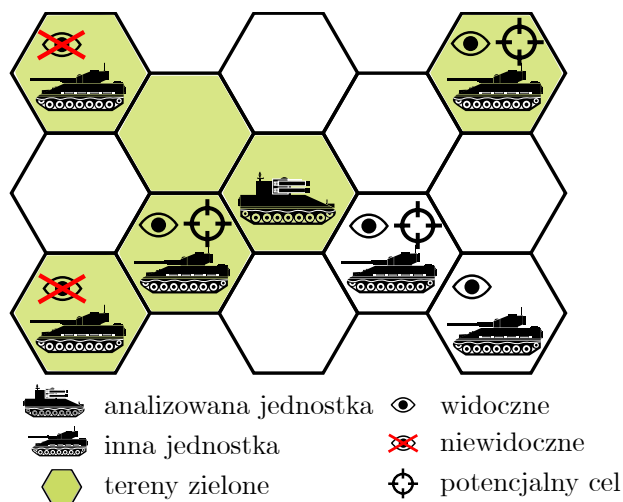


Rysunek 7: Widoczność i możliwość ataku – jednostka znajduje się poniżej wzniesień.

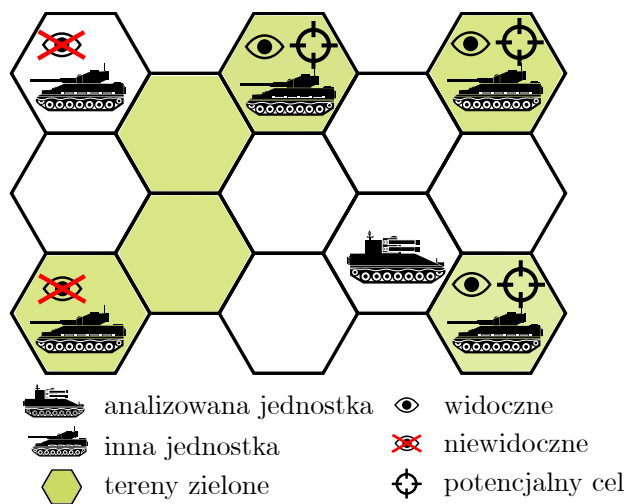


Rysunek 8: Widoczność i możliwość ataku – jednostka znajduje się na wzniesieniu.

Wpływ lasu i zarośli na widoczność Poniższe rysunki przedstawiają możliwe sytuacje dotyczące widoczności i linii ognia związane z usytuowaniem jednostki względem lasu lub zarośli. Sytuacje niezależne od wysokości przeszkody (las czy zarośla).

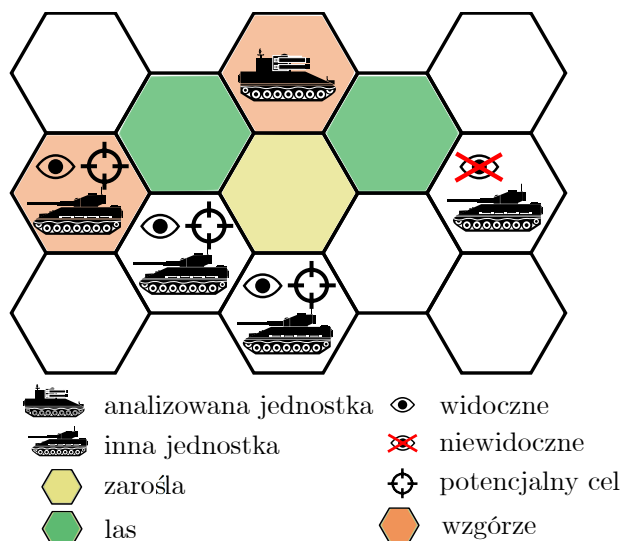


Rysunek 9: Widoczność i możliwość ataku – jednostka znajduje się w lesie (w zaroślach).

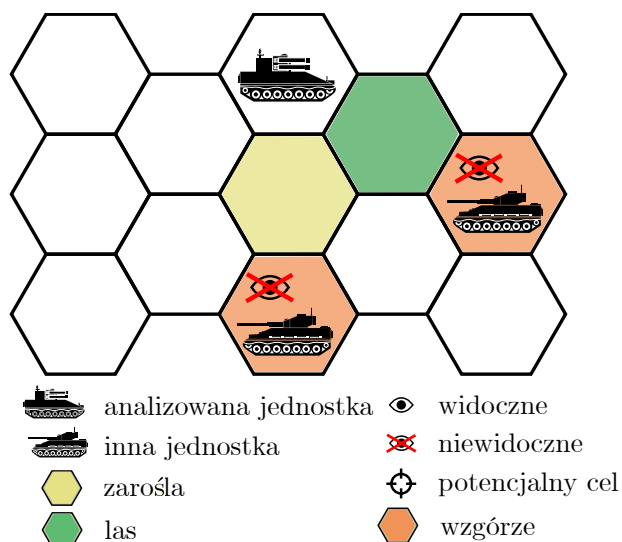


Rysunek 10: Widoczność i możliwość ataku – jednostka znajduje się poza lasem (zaroślami).

Wpływ wysokości przeszkody terenowej na widoczność Poniższe rysunki przedstawiają możliwe sytuacje dotyczące widoczności i linii ognia związane z usytuowaniem jednostek względem lasu i zarośli. Sytuacje przedstawione na rysunkach są hipotetyczne – w rozgrywce występuje zawsze tylko jeden rodzaj przeszkody: las lub zarośla.



Rysunek 11: Różnice pomiędzy lasem a zaroślami – jednostka na wzniesieniu.



Rysunek 12: Różnice pomiędzy lasem a zaroślami – jednostka poniżej wzniesień.

6. Roboty

6.1. Wstęp

Konsekwentnie realizowany Wielki Plan Podboju Universum sprawiał wrażenie doskonale przemyślanego. Dzięki wsparciu ekspertów, oddziały żukoskoczków mogły precyzyjnie atakować i sprawnie przejmować kontrolę na kolejnych planetach. Niestety, nawet tak znakomicie zorganizowane działania mogą czasem utknąć w martwym punkcie i wymagać wprowadzenia poprawek do Planu. Właśnie taka sytuacja miała miejsce na planecie X_X^X .

Armia żukoskoczków natknęła się tam na Rzeczy Których Żukoskoczek Nie Powinien Znać. Miało to straszliwe konsekwencje dla dalszej ekspansji. W kontakcie z Rzeczami niemal wszystkie żukoskoczki całkowicie utraciły zainteresowanie podbojem i odczuły nagły pociąg do ogrodnictwa, rolnictwa oraz innej, w ogóle niezwiązanej z walką, działalności na łonie natury. Nawet groźba terapii współczesną muzyką popularną nie zmusiła ich do zmiany postawy. Nieliczni trzeźwo myślący dowódcy uznali swe wojska za nieprzydatne do dalszego podboju i spisali żołnierzy na straty, odsyłając ich do pracy na farmach.

Sztab realizujący Plan miał twardy orzech do zgryzienia. Nic nie wskazywało na zmianę sytuacji. Kolejne armie skierowane na X_X^X mógł spotkać ten sam straszliwy los: psychiczna demobilizacja, przekwalifikowanie i odesłanie na farmę. Aby przeciwdziałać tak poważnemu zagrożeniu, dowódcy postanowili podjąć drastyczne kroki: X_X^X podbiją całkowicie autonomiczne roboty bojowe. Plan, na pozór genialny, miał jeden słaby punkt – sztab nie dysponował robotami bojowymi.

Produkcja takich maszyn wojennych na odpowiednio dużą skalę jest możliwa tylko dzięki szerokiemu zaangażowaniu sił cywilnych. Aby zachęcić naukowców i wszelakich domowych majsterkowiczów do konstruowania robotów, ogłoszono konkurs z atrakcyjnymi nagrodami. Jego najważniejszym elementem mają być walki robotów. Pod przykrywką tego wydarzenia, wojskowi zbierają „zmechanizowany materiał bojowy” i mobilizują siły przed kolejnym najazdem na feralną planetę.

6.2. Zadanie

Podejmujecie wyzwanie zbudowania najlepszego robota, który będzie świetnie spisywał się na polu walki. Jednak nie tylko umiejętności bojowe robota mogą mieć znaczenie. Nad całością procesu budowania i porównywania robotów czuwa Komisja Kontroli Gier i Zawodów. Od jej decyzji zależy w dużej mierze to, jak zostaną ukierunkowane prace nad mechaniczną armią.

6.3. Model rozgrywki

Rozgrywka odbywa się w turach o równej długości. W trakcie trwania każdej tury drużyny komunikują się z serwerem i wydają polecenia, które mają doprowadzić do stworzenia robota i zwycięstwa w walce z innymi maszynami.

Rozgrywka dzieli się na kilka etapów, które następują po sobie w ściśle określonej kolejności. Każdy z etapów dotyczy jednego z trzech niżej wymienionych obszarów działań:

- **Zakupy**, w czasie których pozyskuje się części na drodze składania ofert w aukcji (etapy: PLANNING, AUCTION i SALE),

- **Budowa**, podczas której drużyny pracują nad ostatecznym kształtem swoich robotów (etap: **BUILDING**),
- **Walka**, podczas której wyżej wspomniane roboty próbują roznieść w pył konkurencję – oczywiście pod czujnym okiem Komisji (etapy: **FIGHTING**, **RESULTS**).

By zachować obiektywność ocen, na początku każdej rozgrywki drużynom przydzielane są losowe numery używane do rozróżniania ich między sobą.

6.4. Części do robotów

Obiektem zainteresowania drużyn podczas zakupów są części do robotów. Każda część posiada kilka parametrów, które bezpośrednio wpływają na końcowe możliwości bojowe robota zawierającego tę część:

- **HP** — wpływa na wytrzymałość robota, tj. ilość obrażeń, które może on przyjąć przed wyeliminowaniem z gry,
- **ATK** — wpływa na siłę ataku robota – ilość obrażeń jaką można zadać przy pojedynczym ataku,
- **DEF** — wpływa na możliwości obronne robota (patrz sekcja 6.7.),
- **SPD** — wpływa na prędkość robota – priorytet zadawania obrażeń w walce.

Ponadto każda część posiada zestaw (od 1 do 6) interfejsów, które umożliwiają łączenie części między sobą.

6.5. Zakupy

Każda drużyna na początku rozgrywki otrzymuje pewną ustaloną liczbę żukomonet – taką samą dla wszystkich drużyn. Waluta ta ma zastosowanie w domach aukcyjnych, gdzie można nabyć za nią odpowiednie części do konstrukcji robotów.

Każdy dom aukcyjny w danej rozgrywce dysponuje tym samym asortymentem i może pomieścić tylko trzy drużyny. W związku z tym rywalizacja o części między drużynami prowadzona jest w **grupach** po trzy zespoły. Wszelkie dokonane zakupy w domach aukcyjnych są ostateczne – nie ma możliwości zwrotu nabytych towarów bądź też handlowania nimi między drużynami.

Zakupy składają się z trzech etapów: planowania, aukcji oraz wyprzedazy.

6.5.1. Planowanie

W czasie tego etapu można przeglądać cały asortyment domu aukcyjnego i przygotowywać strategię zakupów. Zgodnie z polityką transparentności, Komisja udostępnia listę całego towaru, a także informację o kolejności w jakiej będzie on sprzedawany.

Żukoskoczki-sprzedawcy działają bardzo metodycznie, co znajduje odzwierciedlenie w formie sprzedaży: przedmioty są numerowane kolejnymi liczbami naturalnymi, zaczynając od 1. W takiej też kolejności są one wystawiane do sprzedaży w kolejnych dwóch etapach związanych z zakupami. Z uwagi na przepisy podatkowe, numery raz nadane nie ulegają zmianie przez całą rozgrywkę.

6.5.2. Aukcje

Podczas tego etapu każda z drużyn ma możliwość złożenia oferty na interesujące ją części. W każdej turze trwania aukcji dostępnych jest kolejnych 5 przedmiotów z wcześniej udostępnionej listy (tj. w turze pierwszej dostępne są przedmioty o numerach 1 – 5, w drugiej o numerach 6 – 10, itd.). Każdy przedmiot posiada cenę wywoławczą oraz szczegółowy opis swoich parametrów.

Warunki licytacji określają poniższe zasady:

- każda drużyna może przedstawić swoją ofertę za dany przedmiot. Oferowana cena musi jednak być nie mniejsza niż cena wywoławcza przedmiotu,
- suma wartości ofert złożonych przez drużynę nie może przekraczać liczby aktualnie posiadanych przez nią żukomonet,
- jeśli drużyna zaliczytuje kilkakrotnie ten sam przedmiot, uwzględniana jest najwyższa złożona oferta,

- przedmiot trafia do tej drużyny, która zaoferowała za niego najwyższą cenę,
- jeśli ofert o tej samej, najwyższej wartości było więcej niż jedna, aukcja pozostaje nierozstrzygnięta,
- towar, na który nie złożono ofert oraz ten z aukcji nierozstrzygniętych jest przekazywany do zbioru przedmiotów niezakupionych.

Po każdej turze aukcji publikowana jest informacja o zwycięskich ofertach za oferowane przedmioty.

6.5.3. Wyprzedaż

W czasie tego etapu wszystkie części, które trafiły do zbioru przedmiotów niezakupionych, można ponownie zalicytować. Opisy i ceny wywoławcze pozostają bez zmian. Podobnie, każda drużyna ma prawo przedstawić swoją ofertę za konkretny przedmiot, a proponowana cena musi być nie mniejsza niż cena wywoławcza. Pozostałe warunki licytowania podczas wyprzedaży różnią się jednak nieco od tych, które dotyczą aukcji:

- wszystkie przedmioty dostępne są jednocześnie,
- oferty licytujących porównywane są jednokrotnie, na sam koniec wyprzedaży (która trwa więcej niż jedną turę).

Każdy przedmiot trafia do tej drużyny, która zaoferowała za niego najwięcej żukomonet. Towar bez ofert lub bez zwycięzcy (oferty remisowe) nie jest sprzedawany nikomu – zostaje bezpowrotnie zełomowany. Po zakończonym etapie wyprzedaży publikowana będzie także informacja o zwycięskich ofertach w ramach domu aukcyjnego.

6.6. Budowa

Ten etap umożliwia drużynom skonstruowanie działającego robota z posiadanych części.

6.6.1. Robot

Każdy robot musi składać się z przynajmniej dwóch części. Każda część posiada pewien zestaw interfejsów i tylko za ich pośrednictwem może być połączona z innymi częściami. Rodzaj interfejsu opisany jest zawsze liczbą naturalną, a połączenia mogą być wykonane wyłącznie między interfejsami tego samego rodzaju. Nie można łączyć części samej ze sobą. Zbudowany robot musi stanowić spójną całość – jeden mechaniczny organizm. Każda jego część musi mieć łączność z dowolną inną częścią tego robota, być może za pośrednictwem innych części.

Parametry wynikowe zbudowanej maszyny (HP, ATK, DEF, SPD) zależą od sumy parametrów poszczególnych części oraz dodatkowych premii za:

- każde połączenie dwóch części robota (niezależnie dla każdego interfejsu),
 - jeśli jedna z nich ma wartość $ATK = A_1$, a druga $ATK = A_2$, to wartość premii dla parametru ATK wynosi: $W_1 \cdot \min(A_1, A_2)$,
 - wartość premii dla parametru DEF wynosi W_2 ,
- za każdą część użytą do budowy robota,
 - wartość premii dla parametru SPD wynosi W_3 .

Jeśli którykolwiek z wynikowych parametrów robota miałby na skutek użytych części oraz premii osiągnąć wartość ujemną, przyjmuje się dla tego parametru wartość równą 0.

Budowa robota jest zajęciem bardzo czasochłonnym. W czasie trwania etapu BUILDING każda drużyna może wykonać to zadanie najwyżej dwa razy. Ewentualna druga próba spowoduje wcześniejsze całkowite rozmontowanie pierwotnej wersji maszyny.

6.7. Walka

6.7.1. Walki na wspólnej arenie

Podczas tego etapu roboty wszystkich drużyn walczą na wspólnej arenie. Przez cały okres walk drużyny mają pełną informację o bieżących statystykach wszystkich rywalizujących ze sobą maszyn.

Wytrzymałość Każdy robot posiada na początku każdej walki określoną wytrzymałość – parametr HP, którego wartość jest sumą analogicznej wielkości jego elementów składowych. Walki pomiędzy robotami polegają na zadawaniu obrażeń przeciwnikom, co skutkuje obniżeniem ich bieżącej wartości HP.

Robot, którego bieżąca wytrzymałość spadnie do zera zostaje wyeliminowany z dalszej walki – nie może atakować pozostałych jednostek.

Obrona Parametr DEF przekłada się na możliwości obronne danej jednostki. Im większa wartość liczbową parametru DEF, tym większa procentowo część ataku może zostać odparta (D_P). Wyścig zbrojeń zawsze wpływa zarówno na możliwości ofensywne jak i defensywne. Z tego powodu nie można odeprzeć nigdy więcej niż 75% ataku.

Atak Wartość parametru ATK robota oznacza maksymalną wartość obrażeń jakie może on zadać wykonując atak. Faktyczna wartość zadanych obrażeń zależy od możliwości obronnych jednostki atakowanej i wyniesie: $ATK \cdot (100\% - D_P)$, w zaokrągleniu do najbliższej wartości całkowitej.

Kolejność wykonywania ataków W każdej turze etapu FIGHTING każda drużyna może zaplanować jeden atak na jednego konkretnego robota należącego do innej drużyny. Wszystkie ataki zostaną wykonane po zakończeniu danej tury i w kolejności, która wynika z parametrów SPD atakujących. Im większa wartość SPD, tym szybciej atak zostanie wykonany. Dla tych samych wartości parametrów SPD atak jest wykonywany jednocześnie.

Drużyny mają możliwość celowego opóźnienia ataku swoim robotem. Efektem tego jest traktowanie rozkazu tak, jakby został wydany przez jednostkę, której parametr prędkości ataku ma wartość $-SPD$ (ta sama wartość bezwzględna, ale przeciwny znak).

Cykliczność walki W ramach każdej rozgrywki przeprowadzonych może zostać wiele bezpośrednio następujących po sobie etapów FIGHTING. Każdorazowo do nowej walki wystawiany jest nowy egzemplarz zbudowanego wcześniej robota (robot jest całkowicie naprawiany).

Ogólny ranking za walkę Po zakończeniu ostatniej tury każdego etapu FIGHTING robotom przyznawane są miejsca w **ogólnym rankingu za walkę**. O zajętych miejscach decyduje liczba punktów HP, które pozostały robotom (im większa tym wyższa lokata) lub – w przypadku $HP = 0$ – kolejność wyeliminowania robota z walki (im później tym wyższa lokata).

6.7.2. Wyniki walk

W ramach walk wszystkie statystyki przeprowadzonych ataków są dostępne dla wszystkich graczy w kolejnej turze. Dodatkowo po przeprowadzeniu wszystkich Q etapów FIGHTING następuje ostatni etap rozgrywki: RESULTS. Podczas tego etapu dostępne są statystyki z ostatniej tury ostatniego etapu FIGHTING.

6.8. Punktacja

Wszystkim działaniom drużyn przygląda się Komisja Kontroli Gier i Zawodów. Celem wyłonienia zwycięzcy, Komisja przyznaje drużynom nagrody punktowe w różnych kategoriach.

Działania oceniane – zgodnie z obowiązującymi przepisami – są absolutnie transparentne, stąd też już przed rozpoczęciem danej rozgrywki wiadomo, jaką wagą odznaczają się poszczególne działania i w jaki sposób punkty są przyznawane. Przed rozgrywką, dla każdej kategorii C ustala się liczbę punktów bazowych B_C , a następnie tworzy się oddzielny ranking wszystkich rywalizujących drużyn. Drużyna, która w danej kategorii zajmuje i -tą lokatę otrzymuje zaokrągloną do wartości całkowitej liczbę punktów

wyrażoną wzorem: $B_C \cdot (1 - \frac{\ln(i)}{\ln(32)})$. Za zajęcie ostatniej lokaty dla każdego rankingu nie są przyznawane żadne punkty.

Punkty będą przyznawane w kategoriach opisanych w Tabeli 5. Na skutek zmian personalnych w Komisji, wartości punktów bazowych dla każdej kategorii mogą zmieniać się pomiędzy poszczególnymi rozgrywkami (są stałe w ramach jednej rozgrywki).

O ostatecznym wyniku drużyny decyduje suma punktów z wszystkich kategorii. Jeśli ta suma miałaby być ujemna, przyjmuje się zamiast niej wartość 0. Wszystkie statystyki (jeśli nie wskazano inaczej) są zaokrąglane w dół do najbliższej liczby całkowitej.

Tabela 5: Przyznawanie punktów – opis kategorii.

Nazwa kategorii	Kryterium tworzenia rankingu*	Etap**
<i>Parts</i>	Liczba nabytych podczas aukcji części do robotów	SALE
<i>Cheap</i>	Stosunek liczby zakupionych przez drużynę części do ich sumarycznego kosztu	SALE
<i>Attack</i>	Wartość ATK zbudowanego robota	BUILDING
<i>Defence</i>	Wartość DEF zbudowanego robota	BUILDING
<i>Speed</i>	Wartość SPD zbudowanego robota	BUILDING
<i>Health</i>	Wartość HP zbudowanego robota	BUILDING
<i>Total</i>	Suma liczby części oraz połączeń między nimi dla zbudowanego robota	BUILDING
<i>Unused</i>	Liczba zakupionych części, które nie zostały użyte do zbudowania robota	BUILDING
<i>Hits</i>	Suma zadanych obrażeń innym robotom	FIGHTING
<i>Frag</i>	Liczba wyeliminowanych robotów	FIGHTING
<i>Condition</i>	Miejsce w ogólnym rankingu za walkę – liczba HP robota po zakończeniu walki lub liczba tur przez które robot nie został wyeliminowany	FIGHTING

* ranking w każdej kategorii wyznacza się według malejącej wartości opisanego wskaźnika

** nazwa etapu, po zakończeniu którego ustalany jest ranking drużyn

6.9. Komendy

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

DESCRIBE_GAME Zwraca parametry rozgrywki i wartość współczynnika skalującego wynik.

Parametry: brak

Dane (od serwera):

W pierwszej linii serwer zwróci pięć wartości oddzielonych pojedynczą spacją:

- W_1 ($W_1 \in \mathbb{R}$, $-10 \leq W_1 \leq 10$) — mnożnik premii ataku (patrz sekcja 6.6.1.),
- W_2 ($W_2 \in \mathbb{R}$, $-10 \leq W_2 \leq 10$) — mnożnik premii obrony (patrz sekcja 6.6.1.),
- W_3 ($W_3 \in \mathbb{R}$, $-10 \leq W_3 \leq 10$) — mnożnik premii szybkości ataku (patrz sekcja 6.6.1.),
- T ($T \in \mathbb{N}$, $1 \leq T \leq 3$) — czas trwania pojedynczej tury w sekundach,
- K ($K \in \mathbb{R}$, $1 \leq K \leq 8$) — wartość współczynnika skalującego wynik.

W drugiej linii serwer zwróci oddzielone pojedynczymi spacjami 11 wartości punktów bazowych (patrz sekcja 6.8.):

- B_C ($B_C \in \mathbb{Z}$, $-1000 \leq B_C \leq 1000$),

gdzie C należy kolejno do kategorii: *Parts*, *Cheap*, *Attack*, *Defence*, *Speed*, *Health*, *Total*, *Unused*, *Hits*, *Frag*s oraz *Condition*.

MY_CASH Zwraca liczbę żukomonet posiadanych przez daną drużynę.

Parametry: brak

Dane (od serwera):

- C ($C \in \mathbb{N}$, $0 \leq C \leq 10^9$) — suma posiadanych żukomonet.

MY_ID Zwraca identyfikator drużyny obowiązujący w trakcie aktualnej rozgrywki.

Parametry: brak

Dane (od serwera):

- ID ($ID \in \mathbb{N}$, $1 \leq ID \leq 100$) — obowiązujący w tej rozgrywce identyfikator drużyny.

CURRENT_STAGE Podaje nazwę aktualnego etapu rozgrywki oraz liczbę tur do jego zakończenia. Etap **FIGHTING** może zakończyć się przed upływem L tur, jeśli na arenie pozostanie co najwyżej jeden robot z dodatnią wytrzymałością.

Parametry: brak

Dane (od serwera):

Serwer zwraca cztery wartości oddzielone pojedynczą spacją:

- $Type$ ($Type \in \{\text{PLANNING, AUCTION, SALE, BUILDING, FIGHTING, RESULTS}\}$) — nazwa aktualnie trwającego etapu, odpowiednio: planowanie aukcji, aukcja, wyprzedaż, budowanie robota, walki oraz wyniki,
- L ($L \in \mathbb{N}$) — liczba pełnych tur do zakończenia aktualnego etapu, dla **FIGHTING** oznacza maksymalną liczbę tur, po której etap zostanie zakończony,
- F ($F \in \mathbb{N}$, $0 \leq F \leq Q$) — numer porządkowy trwającego właśnie etapu **FIGHTING**, 0 w przypadku trwania innego etapu,
- Q ($Q \in \mathbb{N}$, $1 \leq Q \leq 10$) — liczba wszystkich etapów **FIGHTING**, które wystąpią w rozgrywce.

ALL_PARTS Zwraca opis wszystkich przedmiotów, które będzie można nabyć w ramach aukcji. To polecenie można wykonać tylko podczas trwającego etapu **PLANNING** i wyłącznie raz podczas całej rozgrywki.

Parametry: brak

Dane (od serwera):

Pierwsza linia odpowiedzi zawiera:

- C ($C \in \mathbb{N}, 1 \leq C \leq 5000$) — liczba części dostępnych ogółem na aukcji.

Dalsze C linii to opisy poszczególnych części do robotów. W każdej linii podano oddzielone pojedynczymi spacjami wartości, które oznaczają kolejno:

- ID ($ID \in \mathbb{N}, 1 \leq ID \leq C$) — unikalny w ramach rozgrywki identyfikator części,
- Hp ($Hp \in \mathbb{N}, 1 \leq Hp \leq 1000$) — wytrzymałość danej części,
- Atk ($Atk \in \mathbb{N}, 1 \leq Atk \leq 1000$) — wyrażony liczbowo wpływ części na siłę ataku robota,
- Def ($Def \in \mathbb{N}, 1 \leq Def \leq 1000$) — wyrażony liczbowo wpływ części na możliwości obronne robota,
- Spd ($Spd \in \mathbb{N}, 1 \leq Spd \leq 1000$) — wyrażony liczbowo wpływ części na prędkość robota,
- K ($K \in \mathbb{N}, 1 \leq K \leq 6$) — liczba interfejsów dostępnych w ramach opisywanej części,
- K wartości I_N ($N \in \mathbb{N}, 1 \leq N \leq K, I_N \in \mathbb{N}, 1 \leq I_N \leq 1000$) — lista konkretnych typów interfejsów występujących w opisywanej części,
- $Price$ ($Price \in \mathbb{N}, 1 \leq Price \leq 10^9$) — cena wywoławcza za część.

CURRENT_PARTS Zwraca opis przedmiotów, które można zaliczyć w aktualnej turze etapu **AUCTION**.

Parametry: brak

Dane (od serwera):

Pierwsza linia odpowiedzi zawiera cyfrę 5. Każda z kolejnych 5 linii zawiera opis jednej części do robotów. Opisy są identyczne jak te zawarte w poleceniu **ALL_PARTS**.

SALE_PARTS Zwraca opis przedmiotów, które można zaliczyć w aktualnej turze etapu **SALE**. Podczas trwania całego etapu **SALE** polecenie można wykonać tylko raz.

Parametry: brak

Dane (od serwera):

Pierwsza linia odpowiedzi zawiera:

- C ($C \in \mathbb{N}, 0 \leq C \leq 5000$) — liczba części dostępnych do zaliczenia w bieżącej turze.

Dalsze C linii to opisy poszczególnych części do robotów. Opisy są identyczne jak te zawarte w poleceniu **ALL_PARTS**.

MY_PARTS Zwraca listę posiadanych przez drużynę części do robotów. Polecenie można wykonać raz na 10 tur.

Parametry: brak

Dane (od serwera):

Pierwsza linia:

- C ($C \in \mathbb{N}, 0 \leq C \leq 5000$) — liczba posiadanych części.

Dalsze C linii to opisy poszczególnych części do robotów. Opisy są podobne do tych zawartych w poleceniu **ALL_PARTS**, nie zawierają jedynie ostatniego elementu: *Price*.

LAST_WINNING_BIDS Podaje listę transakcji z poprzedniej tury – zakończone sukcesem aukcje z domu aukcyjnego, do którego należy drużyna.

Parametry: brak

Dane (od serwera):

Pierwsza linia odpowiedzi zawiera:

- C ($C \in \mathbb{N}, 0 \leq C \leq 5000$) — liczba zakupionych w poprzedniej turze części (podczas etapu AUCTION górne ograniczenie ma wartość 5).

Kolejne wiersze zawierają po 3 oddzielone pojedynczymi spacjami wartości, oznaczające kolejno:

- $PartId$ ($PartId \in \mathbb{N}$) — ID zakupionej części,
- $Price$ ($Price \in \mathbb{N}$) — cena końcowa zakupionego przedmiotu,
- $TeamId$ ($TeamId \in \mathbb{N}$) — ID zespołu, który dokonał zakupu.

BID Składa ofertę zakupu części robota. Jeśli drużyna zaliczytuje kilkakrotnie ten sam przedmiot w danej turze, uwzględniana jest najwyższa złożona oferta.

Parametry:

- ID ($ID \in \mathbb{N}, 1 \leq ID \leq 5000$) — identyfikator części, której dotyczy składana oferta,
- $Price$ ($Price \in \mathbb{N}$) — proponowana cena za przedmiot aukcji.

MY_ROBOT Wypisuje właściwości robota zbudowanego przez drużynę.

Parametry: brak

Dane (od serwera):

Serwer odpowiada jedną linią, która zawiera oddzielone pojedynczymi spacjami wartości:

- Hp ($Hp \in \mathbb{N}$) — wytrzymałość robota,
- Atk ($Atk \in \mathbb{N}$) — siła ataku robota,
- Def ($Def \in \mathbb{N}$) — możliwości obronne robota (wyrażone liczbowo),
- Spd ($Spd \in \mathbb{N}$) — prędkość robota,

BUILD_ROBOT Buduje robota składającego się z podanych części. Polecenie może zostać wykonane wyłącznie podczas etapu budowy (BUILDING) – tylko dwa poprawne roboty mogą zostać zbudowane podczas każdej rozgrywki. W czasie jednej tury etapu BUILDING można podjąć tylko jedną próbę zbudowania poprawnego robota.

Parametry: Należy przekazać dokładnie dwie linie. W pierwszej z nich (razem z nazwą polecenia) znajduje się tylko liczba $0 < R < 5000$, oznaczająca liczbę połączeń interfejsów z jakich zostanie skonstruowany robot.

W następnej linii musi znajdować się dokładnie $3 \cdot R$ oddzielonych pojedynczymi spacjami liczb naturalnych, oznaczających w ramach każdej trójki kolejno:

- ID_A ($ID_A \in \mathbb{N}, 1 \leq ID_A \leq 5000$) — numer ID pierwszej części,
- ID_B ($ID_B \in \mathbb{N}, 1 \leq ID_B \leq 5000$) — numer ID drugiej części,
- I ($I \in \mathbb{N}, 1 \leq I \leq 1000$) — interfejs, który umożliwi połączenie podanych dwóch części.

LIST_ROBOTS Wypisuje szczegółowe informacje o wszystkich robotach dostępnych na arenie walki (łącznie z robotem drużyny).

Parametry: brak

Dane (od serwera):

Pierwsza linia odpowiedzi zawiera:

- C ($C \in \mathbb{N}, 1 \leq C \leq 30$) — liczba robotów dostępnych na arenie.

Kolejne C linii opisuje poszczególne roboty. Każda linia zawiera oddzielone pojedynczymi spacjami wartości:

- ID ($ID \in \mathbb{N}$) — numer ID drużyny, której robot jest własnością,
- Hp ($Hp \in \mathbb{N}$) — aktualna wytrzymałość robota,
- Atk ($Atk \in \mathbb{N}$) — siła ataku robota,
- Def ($Def \in \mathbb{N}$) — możliwości obronne robota (wyrażone liczbowo),
- Spd ($Spd \in \mathbb{N}$) — prędkość robota,
- D_P ($D_P \in \mathbb{N}, 0 \leq D_P \leq 75$) — możliwości obronne robota wyrażone w procentach unikanego ataku.

LAST_ATTACKS Zwraca listę ataków wykonanych w poprzedniej turze. Informacje są posortowane w kolejności zadawania obrażeń.

Parametry: brak

Dane (od serwera):

Pierwsza linia odpowiedzi zawiera:

- C ($C \in \mathbb{N}, 0 \leq C$) — liczba poprawnie wykonanych w poprzedniej turze ataków.

Każdy kolejny spośród C wierszy zawiera oddzielone spacjami:

- ID_A ($ID_A \in \mathbb{N}$) — numer ID robota atakującego,
- ID_D ($ID_D \in \mathbb{N}$) — numer ID robota atakowanego,
- $Damage$ ($Damage \in \mathbb{N}$) — liczba zadanych obrażeń.

ATTACK Wydaje rozkaz ataku jednostki o podanym identyfikatorze. Wykonanie rozkazu może zostać opóźnione, co oznacza zmianę znaku statystyki SPD robota, używanej do określania kolejności podejmowanych na arenie ataków. Jeśli cel został zniszczony we wcześniejszych atakach, polecenie zostanie wykonane, jednak nie zostaną przyznane żadne punkty za zadane obrażenia. Robot nie może zaatakować sam siebie.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID atakowanego robota,
- $Delayed$ ($Delayed \in \{Y, N\}$) — zmienia wartość SPD robota na przeciwną (Y) lub pozostawia bez zmian (N).

MY_STATS Zwraca listę aktualnych osiągnięć drużyny z bieżącej rozgrywki. Po zakończeniu każdego z etapów rozgrywki lista uzupełniana jest o kolejne punkty zdobyte przez drużyny. To polecenie można wykonać tylko podczas trwającego etapu RESULTS.

Parametry: brak

Dane (od serwera):

Pierwsza linia odpowiedzi zawiera jedną wartość:

- Sum ($Sum \in \mathbb{N}$) — liczba sumarycznie zdobytych punktów w aktualnej rozgrywce.

Każda z kolejnych $8 + 3 \cdot Q$ linii zawiera dwie wartości oddzielone spacją (Q – patrz polecenie CURRENT_STAGE):

- $Rank$ ($Rank \in \mathbb{N}$) — pozycja rankingowa drużyny w danej kategorii,
- $Points$ ($Points \in \mathbb{Z}$) — liczba punktów przyznana drużynie za daną kategorię.

Każda z tych $8 + 3 \cdot Q$ linii odpowiada osiągnięciom drużyny w jednej kategorii punktowej, odpowiednio w podanej kolejności: *Parts*, *Cheap*, *Attack*, *Defence*, *Speed*, *Health*, *Total*, *Unused*, $Q \cdot Hits$, $Q \cdot Frags$ oraz $Q \cdot Condition$. Dla kategorii *Hits*, *Frags* oraz *Condition* kolejność informacji odpowiada kolejnym etapom FIGHTING.

WAIT Czeka do momentu rozpoczęcia kolejnej tury.

Parametry: brak

Dane (od serwera):

Serwer odpowiada jedną linią postaci:

- WAITING S

gdzie S ($S \in \mathbb{R}, S \geq 0$) oznacza liczbę sekund do zakończenia oczekiwania. Po upływie tego czasu wysyła dodatkowo linię:

- OK

6.10. 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 Roboty.

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	improper current turn stage
102	improper previous turn stage
103	too many calls within a specific turns period
104	part with given ID is not currently available
105	the price is too low
106	you have not enough beetcoins
107	you do not own part with given ID
108	improper number of connections of the robot
109	improper connection between parts
110	given interface is no more available for one of the parts
111	not all parts are connected
112	a part cannot be connected to itself
113	improper target ID
114	improper delayed value
115	only one attack is available for a single turn
116	your robot has no stamina to attack

6.11. Serwery

Rozgrywki będą odbywać się na serwerach, różniących się między sobą liczbą etapów walki.

Tabela 6: Adresy i parametry serwerów rozgrywek.

Nazwa	Adres:Port	Liczba etapów FIGHTING
Robots1	universum.d124:20003	mała
Robots2	universum.d124:20004	duża

6.12. Przykład

Poniżej znajduje się przykładowy zapis komunikacji z serwerem.

klient → serwer	serwer → klient
	LOGIN
login1	PASS
secret	OK
DESCRIBE_GAME	OK 4.20000 2.20000 -1.20000 1 1.20000 111 500 301 226 683 744 808 -613 711 720 813
CURRENT_STAGE	OK PLANNING 2 0 5
MY_CASH	OK 9800
MY_ID	OK 44
ALL_PARTS	OK 10 1 2 4 2 1 6 3 5 4 1 7 9 394 2 9 8 1 5 5 1 5 4 7 4 413 3 2 2 4 4 5 4 2 1 2 6 589 4 5 8 2 3 5 3 8 8 8 7 890 5 7 5 8 9 4 9 3 3 2 394 6 1 4 4 6 3 6 5 3 819 7 5 8 8 4 3 6 2 6 248 8 4 1 9 1 3 3 6 7 846 9 3 4 4 5 3 4 2 7 556 10 3 8 6 4 2 7 1 104
WAIT	OK WAITING 0.55000 OK
CURRENT_STAGE	OK AUCTION 1 0 5
CURRENT_PARTS	OK 5 1 2 4 2 1 6 3 5 4 1 7 9 394 2 9 8 1 5 5 1 5 4 7 4 413 3 2 2 4 4 5 4 2 1 2 6 589 4 5 8 2 3 5 3 8 8 8 7 890 5 7 5 8 9 4 9 3 3 2 394

BID 2 400	FAILED 105 the price is too low
BID 2 600	OK
BID 4 895	OK
WAIT	OK
	WAITING 0.25000
	OK
CURRENT_STAGE	OK
	AUCTION 0 0 5
LAST_WINNING_BIDS	OK
	1 550 70
	2 600 44
	4 990 70
	5 1000 70
CURRENT_PARTS	OK
	5
	6 1 4 4 6 3 6 5 3 819
	7 5 8 8 4 3 6 2 6 248
	8 4 1 9 1 3 3 6 7 846
	9 3 4 4 5 3 4 2 7 556
	10 3 8 6 4 2 7 1 104
MY_CASH	OK
	9200
BID 2 600	FAILED 104 part with given ID is not currently available
BID 6 900	OK
BID 7 900	OK
BID 7 1900	OK
WAIT	OK
	WAITING 0.42000
	OK
CURRENT_STAGE	OK
	SALE 0 0 5
LAST_WINNING_BIDS	OK
	6 900 44
	7 1900 44
	8 850 70

```

MY_PARTS
OK
3
2 9 8 1 5 5 1 5 4 7 4
6 1 4 4 6 3 6 5 3
7 5 8 8 4 3 6 2 6

SALE_PARTS
OK
3
3 2 2 4 4 5 4 2 1 2 6 589
9 3 4 4 5 3 4 2 7 556
10 3 8 6 4 2 7 1 104

MY_PARTS
FAILED 103 too many calls within a specific turns period

BID 3 590
OK

BID 9 560
OK

LAST_WINNING_BIDS
OK
3 600 70
9 800 70
10 700 70

CURRENT_STAGE
OK
BUILDING 0 0 5

BUILD_ROBOT 2
OK

6 7 6 2 6 5
OK

MY_ROBOT
OK
15 53 17 11

CURRENT_STAGE
OK
FIGHTING 1 1 5

LIST_ROBOTS
OK
5
1 21 20 5 25 60
5 12 2 1 3 55
15 41 11 8 1 65
44 15 53 17 11 70
70 14 18 25 6 75

```

```
ATTACK 5 N |
            | OK
MY_PARTS   |
            | FAILED 101 improper current turn stage
LAST_ATTACKS |
            | OK
            | 4
            | 1 5 9
            | 44 5 3
            | 70 5 0
            | 15 1 4
```

7. Wszystko płynie

7.1. Wprowadzenie

Mieszkającym na *Rozległej równinie* wielu mogło zazdrościć spokojnej egzystencji. Ten mało zaludniony teren odznacza się bardzo malowniczym krajobrazem, a sąsiedzi – żyjący w dostatku – często wzajemnie sobie pomagali, m.in. podczas rozbudowy swoich domostw.

Przyszła jednak czas próby dla wszystkich przebywających w tym rejonie. Do planety, na której umiejscowiona jest *Rozległa równina*, zbliża się ogromna kometa. Z jednej strony można mówić o sporym szczęściu sprzyjającym mieszkańcom – obiekt niebieski nie uderzy w planetę, lecz ominie ją w pewnej bezpiecznej odległości. Z innej strony jednak kataklizm zbliża się nieuchronnie. Tak wielka masa wywoła spore zakłócenia w ekosystemie planety i spowoduje niebezpieczne fale powodziowe. Żaden specjalista od wydawania zezwoleń budowlanych nie przewidział niestety takich kłopotów...

Ratuj się kto może!

7.2. Zadanie

Jako zgrana grupa mieszkańców próbujecie ocalić swój dotychczasowy dobytek przed spodziewaną serią fal powodziowych. Do dyspozycji, poza własnymi siłami, macie jedynie magazyn wypełniony workami z piaskiem.

7.3. Model rozgrywki

Rozgrywka odbywa się w turach o równej długości. W trakcie trwania każdej tury drużyny komunikują się z serwerem i wydają polecenia swoim jednostkom (żukoskoczkom).

7.4. Świat

Rozgrywka toczy się na obszarze, który reprezentowany jest przez regularną siatkę składającą się z $N \times N$ kwadratowych elementów, nazywanych **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 . Poza polami będącymi na skraju obszaru, każde pole ma osiem pól sąsiednich.

W różnych miejscach obszaru znajdują się Wasze posiadłości – **obiekty**, które pragniecie ocalić przed zbliżającą się wysoką falą. Aby je odpowiednio zabezpieczyć, należy obudować je murem z worków z piaskiem.

7.4.1. Obiekty

Każdy obiekt posiada określoną wartość, lokalizację oraz rozmiar. Obiekty mogą z sobą sąsiadować, natomiast nie mogą zajmować tego samego pola. Niektóre obiekty (magazyn i skokowozy) mogą odznaczać się dodatkowymi własnościami.

Magazyn na worki z piaskiem Jeden z obiektów należących do drużyny to magazyn na worki z piaskiem. To bardzo wysoki budynek, którego podstawa zajmuje zawsze 2×2 pola. Żukoskoczek – będąc w magazynie – może podnieść każdy dostępny tam worek z piaskiem.

Dzięki temu, że magazyn jest wodoszczelny, żukoskoczki mogą w nim bezpiecznie przeczekać powódź. Z tego samego powodu może on być wykorzystany jako fragment muru przeciwpowodziowego – nie będzie przepuszczał wody niezależnie od wysokości fali powodziowej. Wszystkie pozostałe obiekty są wrażliwe na działanie wody i nie stanowią dla niej żadnej bariery.

Skokowozy Wszystkie obiekty o rozmiarze 1×1 pole to pojazdy używane na *Rozległej równinie* – skokowozy. Te nieduże samochody mogą pomieścić jednocześnie do czterech żukoskoczków. Żaden z nich nie może jednak zabrać ze sobą worka z piaskiem – pojazdy te nie są przystosowane do ich przewożenia.

Ciekawa własność skokowozów to możliwość skokowego przemieszczenia o 3 pola w dowolnym kierunku w czasie trwania jednej tury. Ruch ten może zostać wykonany mimo przeszkód znajdujących się na pośrednich polach między punktem startu a punktem docelowym (o ile na polu docelowym nie znajduje się inny obiekt lub worek z piaskiem). Pierwszy żukoskoczek, który wsiądzie do skokowozu staje się jego kierowcą. Ze względów bezpieczeństwa nie jest możliwe wsiadanie i wysiadanie ze skokowozu, który się porusza.

7.4.2. Fala powodziowa

Podstawowym parametrem fali powodziowej jest jej wysokość. Budując mury zabezpieczające przed zalaniem należy brać ją pod uwagę, gdyż tylko odpowiednio wysoki mur ochroni dobytek przed niszczącą siłą wody. Wszystkie obiekty, które znajdują się na terenie niezabezpieczonym wystarczająco wysokim murem zostaną zalane i utracą 30% swojej wartości. Nawet niewielki ubytek w murze obronnym powoduje wdarcie się wody do środka i zniszczenie mienia.

Na skutek anomalii pogodowej, fala powodziowa nadciąga na *Rozległą równinę* z losowo wybranej krawędzi obszaru, a jej wysokość w danej turze jest równa na całym obszarze.

Przejście fali powodziowej trwa 7 tur. Na skutek traumatycznych (i nie tak dawnych) wydarzeń, żukoskoczki nienawidzą wody. Do tego stopnia, że kontakt z nią wywołuje u nich żukatak, na który nie znaleziono lekarstwa w całym Universum. Niestety żukatak jest śmiertelny... Żukoskoczki mogą schronić się przed wodą w jednym z poniższych miejsc:

- na terenie, który został obudowany odpowiednio wysokim murem,
- na stosie worków z piaskiem o wysokości równej co najmniej wysokości fali powodziowej,
- w magazynie.

Po ustąpieniu fali powodziowej poziom wody natychmiast opada i ponownie można bezpiecznie przemieszczać się także po terenach nieosłoniętych murem.

7.4.3. Prognoza

Sztab Kryzysowy publikuje co pewien czas aktualne informacje dotyczące spodziewanych fal powodziowych, aby usprawnić przygotowania przed nadejściem zagrożenia. Dzięki skomplikowanym modelom matematycznym podawane do publicznej wiadomości prognozy powodziowe są zawsze prawdziwe, ale... niestety często są także mało dokładne lub niekompletne.

Dla każdej spodziewanej fali powodziowej podaje się jej czas wystąpienia oraz wysokość, określone zakresami wartości minimalnej i maksymalnej. Sztab Kryzysowy po uzyskaniu nowszych informacji może aktualizować poprzednie prognozy.

Co najmniej na *E* tur przed nadejściem fali powodziowej, informacja o niej jest kompletna i jest natychmiast publikowana przez Sztab Kryzysowy.

7.5. Żukoskoczki

Każda drużyna dysponuje pewną liczbą żukoskoczków, które mogą podejmować różne czynności. W celu zbudowania muru przeciwpowodziowego i ochrony posiadanych dóbr należy odpowiednio skoordynować ich działania.

Pole widzenia i poruszanie Każdy żukoskoczek dostrzega tylko to co znajduje się w jego bliskim sąsiedztwie – widzi na odległość do 3 pól (w każdym kierunku) od pola, w którym się znajduje.

W czasie gdy fala powodziowa nie stanowi bezpośredniego zagrożenia, każdy żukoskoczek może swobodnie przemieszczać się po całym obszarze. Może chodzić po terenach niezabudowanych, wspinać się na mur o dowolnej wysokości, a także wchodzić do obiektów (o ile właśnie nie przenosi worka z piaskiem). Przejście do sąsiedniego pola zajmuje zawsze jedną turę. Kilka żukoskoczków może znajdować się na jednym polu obszaru rozgrywki jednocześnie.

Kontakt z wodą jest dla żukoskoczków śmiertelny. W czasie trwania fali powodziowej wyjście poza magazyn lub poza obszar otoczony i chroniony murem worków z piaskiem kończy się utratą życia.

Przenoszenie worków i budowa muru Każdy żukoskoczek może samodzielnie podnosić i przenosić jeden worek z piaskiem. Worków nie można stawiać na obiektach. Próba wejścia z workiem do jakiegokolwiek obiektu (także do magazynu) zawsze kończy się niepowodzeniem. Worki można układać jeden na drugim, tworząc dzięki temu odpowiednio wysoki mur. Próba zabrania wcześniej ułożonego worka (lub worków) dotyczy zawsze elementów znajdujących się najwyżej na stosie w danym polu – nie można wyciągać worków ze środka czy dołu stosu.

Taczka Dzięki taczkom, żukoskoczki mogą przenosić worki bardziej efektywnie. Tackę można nabyć drogą wymiany towarowej, a przed powodzią oczywiście najcenniejszym materiałem są... worki z piaskiem. Aby wymiana mogła mieć miejsce, zarówno kupujący jak i odpowiednia liczba worków z piaskiem musi znajdować się w środku magazynu danej drużyny (nie są zaliczane worki wcześniej pobrane z magazynu).

Taczka umożliwia posiadaczowi podnoszenie i przenoszenie do 5 worków jednocześnie. Z tacką nie można wejść nigdy do skokowozu. Po nabyciu tacki dany żukoskoczek nie ma już możliwości jej zbycia.

Jazda skokowozem Jeśli żukoskoczek znajduje się na tym samym polu co skokowóz, może spróbować do niego wsiąść. Można wsiadać jedynie do skokowozów własnej drużyny. Pierwszy żukoskoczek, który wsiądzie do danego skokowozu zostaje automatycznie jego kierowcą. Kolejne żukoskoczki, które wsiądą do pojazdu, są pasażerami.

Przemieszczanie skokowozu odbywa się tylko dzięki działaniom jego kierowcy – pasażerowie są skazani na jego wolę. Pojazd może wykonać jeden ruch o maksymalnie trzy pola w każdej turze, nie zważając na przeszkody pomiędzy swoim polem a polem docelowym (o ile pole docelowe jest dostępne). Nie dopuszcza się wjechania samochodem na pola zajęte przez inne obiekty lub worki z piaskiem.

Jeśli kierowca opuści pojazd, automatycznie opuszczają go także wszyscy pasażerowie.

Własność obiektów i worków z piaskiem Każdy obiekt występujący na obszarze rozgrywki ma przypisanego właściciela, który ma niezbywalne prawo do dysponowania nim. Podobnie każdy worek z piaskiem należy do konkretnej drużyny.

Niemniej jednak, jeśli worek nie jest aktualnie obserwowany przez swojego właściciela (w polu widzenia chociaż jednego żukoskoczka danej drużyny), może być on przenoszony przez innych. Jeśli „złodziej” zostanie przyłapany, tj. wejdzie z nieswoim workiem w zakres widoczności właściciela, automatycznie upuszcza niesiony worek.

7.6. Rozpoczęcie i zakończenie

Początkowy stan rozgrywki Każda drużyna na początku rozgrywki dysponuje taką samą liczbą żukoskoczków oraz taką samą liczbą obiektów, o takich samych parametrach i rozkładzie względem siebie. W pierwszej turze rozgrywki wszystkie żukoskoczki znajdują się w swoich magazynach na worki z piaskiem.

Zakończenie rozgrywki Od początku rozgrywki znana jest najbardziej prawdopodobna liczba fal powodziowych, na które należy się przygotować. Czas do ich nadejścia oraz wysokość fali są prognozowane – podane w pewnym przybliżeniu. Tuż przed nadejściem fali powodziowej (*E* tur) informacja o niej staje się pełna i niezmienna. Rozgrzywka kończy się z zakończeniem wszystkich zagrożeń – tuż po ostatniej fali powodziowej.

Możliwe jest zakończenie rozgrywki przed czasem w przypadku, gdy wszystkie żukoskoczki wszystkich drużyn zginą na skutek kontaktu z falą powodziową. W tym wypadku, rozgrywka zostanie zakończona wraz z ustaniem fali powodziowej.

7.7. Cel rozgrywki i rywalizacja

Głównym celem drużyn jest ochrona mienia przed powodzią. Należy zatem zabezpieczyć możliwie największą liczbę obiektów. Dodatkowo, aby umożliwić niezależne funkcjonowanie żukoskoczków na zabezpieczonym terenie w trakcie powodzi, pożądana jest możliwość przemieszczania między obiektami „suchą nogą” – im więcej obiektów zostanie objętych wspólnym murem, tym lepiej.

7.8. Punktacja

Przenoszenie worków Podczas całej rozgrywki przyznawane są punkty za przenoszenie worków. Drużyna otrzymuje 1 punkt za każdy postawiony worek, jeśli spełniony zostanie jeden z poniższych warunków:

- worek przed postawieniem znajdował się w magazynie,
- worek przed postawieniem był ustawiony przez inną drużynę.

Punkty karne W trakcie przechodzenia fali powodziowej, dodatkowo naliczane będą punkty karne za worki, które nie zostały poprawnie ulokowane. Worek uznaje się za poprawnie postawiony, gdy przynajmniej jedno z jego pól sąsiednich spełnia wszystkie poniższe warunki:

- na polu nie znajdują się worki z piaskiem ani magazyn na worki,
- pole nie zostało zalane falą powodziową.

W każdej turze przechodzenia fali powodziowej, za każdy worek, który nie został poprawnie ulokowany, drużyna, która go tam umieściła otrzymuje 1 punkt karny. W tej definicji mieszczą się także worki porzucone przez żukoskoczki na skutek swej nagłej śmierci po kontakcie z wodą (worki pozostają w miejscu utonięcia).

Zabezpieczony obszar Po przejściu fali powodziowej naliczane są punkty za ocalone obiekty. Wartość pól zabezpieczonych przed powodzią obliczamy w **suchych grupach**. Każda sucha grupa składa się z sąsiadujących ze sobą pól, do których nie dotarła fala powodziowa z pominięciem pól, na których znajdują się niepoprawnie postawione worki.

Każde ocalone pole jest cenne dla żukoskoczków, dlatego **wartość suchej grupy** to suma aktualnych wartości występujących w tej grupie obiektów oraz liczby pól w tej grupie nie zajmowanych przez żadne obiekty.

Po przejściu każdej fali powodziowej, każda z drużyn otrzymuje sumę punktów za każdą występującą na obszarze rozgrywki suchą grupę. Punkty te są iloczynem:

- wartości danej suchej grupy z ostatniej tury występowania fali powodziowej,
- liczby obiektów w tej suchej grupie należących do danej drużyny.

Końcowa punktacja Suma punktów za przenoszenie worków oraz zabezpieczony obszar pomniejszona o punkty karne decyduje o końcowym wyniku drużyny. W przypadku gdy liczba punktów karnych jest większa od sumy pozostałych punktów, za wynik końcowy uznaje się wartość 0.

7.9. Komendy

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

DESCRIBE_WORLD Zwraca parametry rozgrywki i wartość współczynnika skalującego wynik.

Parametry: brak

Dane (od serwera):

W pojedynczej linii serwer zwróci sześć wartości:

- N ($N \in \mathbb{N}$, $20 \leq N \leq 300$) — długość boku planszy,
- P ($P \in \mathbb{N}$, $1 \leq P \leq 10000$) — cena taczki wyrażona w liczbie worków z piaskiem,
- E ($E \in \mathbb{N}$, $1 \leq E \leq 100$) — czas (wyrażony w liczbie tur przed falą powodziową) ujawnienia dokładnych danych o nadchodzącej fali,
- T ($T \in \mathbb{N}$, $1 \leq T \leq 5$) — czas trwania pojedynczej tury w sekundach,
- L ($L \in \mathbb{N}$, $1 \leq L \leq 100$) — maksymalna liczba poleceń, jaką można wykonać w jednej turze,
- K ($K \in \mathbb{R}$, $1 \leq K \leq 8$) — wartość współczynnika skalującego wynik.

LIST_WORKERS Zwraca listę dostępnych dla drużyny żukoskoczków.

Parametry: brak

Dane (od serwera):

Pierwsza linia:

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

Każda kolejna linia opisuje jednego żukoskoczka za pomocą oddzielonych pojedynczymi spacjami wartości oznaczających kolejno:

- ID ($ID \in \mathbb{N}$) — identyfikator żukoskoczka,
- 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,
- $Moving$ ($Moving \in \{Y, N\}$) — informacja o tym czy żukoskoczek jest już w trakcie wykonywania operacji przemieszczania się (Y) czy też nie (N),
- $Capacity$ ($Capacity \in \mathbb{N}$) — maksymalna liczba worków, które żukoskoczek może przetranszować jednocześnie,
- $Status$ ($Status \in \mathbb{N} \cup \{D, P\}$) — liczba aktualnie przemieszczanych worków, D w przypadku kierowania pojazdem lub P w przypadku bycia pasażerem skokowozu.

LIST_OBJECTS Zwraca listę obiektów będących w posiadaniu drużyny. Każdy obiekt ma określony punkt zaczepienia, który jest polem zajmowanym przez ten obiekt o najmniejszej sumie współrzędnych.

Parametry: brak

Dane (od serwera):

Pierwsza linia:

- C ($C \in \mathbb{N}$) — liczba obiektów

Każda kolejna linia opisuje jeden obiekt za pomocą oddzielonych pojedynczymi spacjami wartości oznaczających kolejno:

- P_X ($P_X \in \mathbb{N}$, $1 \leq P_X \leq N$) — współrzędna X punktu zaczepienia,
- P_Y ($P_Y \in \mathbb{N}$, $1 \leq P_Y \leq N$) — współrzędna Y punktu zaczepienia,
- S_X ($S_X \in \mathbb{N}$, $1 \leq S_X \leq 5$) — rozmiar obiektu wzdłuż osi X ,

- S_Y ($S_Y \in \mathbb{N}$, $1 \leq S_Y \leq 5$) — rozmiar obiektu wzdłuż osi Y ,
- $Value$ ($Value \in \mathbb{N}$, $Value \leq 2000$) — aktualna wartość obiektu,
- $Bags$ ($Bags \in \mathbb{N} \cup \{\mathbf{NA}\}$) — dla magazynu to liczba dostępnych w nim worków, dla pozostałych obiektów wartość \mathbf{NA} .

BUY_WHEELBARROW Dokonuje zakupu taczki dla żukoskoczka o numerze ID .

Parametry:

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

MOVE Wydaje polecenie przejścia do sąsiedniego pola, wskazywanego przy pomocy współrzędnych względnych. Przynajmniej jedna ze współrzędnych względnych (D_X lub D_Y) musi być niezerowa. Efekt wykonania operacji będzie widoczny w kolejnej turze.

Wykonanie może zostać anulowane (automatycznie) jeśli doszłoby do kolizji z innym obiektem (np. do zderzenia dwóch skokowozów). W danej rundzie można wydać tylko jedno poprawne polecenie **MOVE** dla danego żukoskoczka. W przypadku kierowania skokowozem jako miejsce docelowe można wskazać pole w odległości do 3 pól od bieżącego.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka,
- D_X ($D_X \in \{-3, -2, -1, 0, 1, 2, 3\}$) — przesunięcie wzdłuż osi X ,
- D_Y ($D_Y \in \{-3, -2, -1, 0, 1, 2, 3\}$) — przesunięcie wzdłuż osi Y .

TAKE Próbuje podnieść worki lub próbuje wsiąść do skokowozu. Dotyczy pola, na którym jest dany żukoskoczek. Polecenie wykonywane jest natychmiast po zgłoszeniu.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka,
- $Repeat$ ($Repeat \in \mathbb{N}$, $1 \leq Repeat \leq 5$) — maksymalna liczbę worków do podniesienia lub 1 w przypadku wsiadania do pojazdu.

Dane (od serwera):

W pojedynczej linii serwer zwróci jedną wartość:

- C ($C \in \mathbb{N} \cup \{\mathbf{NA}\}$ ($\mathbb{N} \in \mathbb{N}$, $1 \leq N \leq 5$)) — liczbę worków, które udało się podnieść lub \mathbf{NA} w przypadku wsiadania do samochodu.

Podczas przechodzenia fali powodziowej, próba podniesienia worka (lub worków) z pola, które stanowi bezpośrednią barierę dla wody (będącego częścią aktywnej zapory), zakończy się natychmiastową śmiercią żukoskoczka (worki pozostaną na swoim miejscu).

LEAVE Odkłada worki z piaskiem lub wysiada ze skokowozu. Dotyczy pola, na którym jest dany żukoskoczek. Polecenie wykonywane jest natychmiast po zgłoszeniu. Jeśli żukoskoczek jest kierowcą lub pasażerem, to nie może on wysiąść z pojazdu w danej turze, jeśli zostało w niej wykonane już polecenie **MOVE** dotyczące tego pojazdu.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka,
- $Repeat$ ($Repeat \in \mathbb{N}$, $1 \leq Repeat \leq 5$) — należy podać liczbę worków do odłożenia lub 1 w przypadku wysiadania z pojazdu.

LOOK_AROUND Zwraca opis pól znajdujących się wokół żukoskoczka. Dotyczy siatki 7×7 pól, pośrodku której znajduje się żukoskoczek.

Parametry:

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

Dane (od serwera):

Serwer zwróci dwie grupy linii – po siedem linii w grupie. Każda z tych linii składa się z siedmiu znaków, opisujących poszczególne pola wokół żukoskoczka w taki sposób, że i -ty znak w j -tej linii (w rozważanej grupie) opisuje pole o współrzędnych $(X+i-4, Y+j-4)$, gdzie X i Y to współrzędne opisujące położenie żukoskoczka, i oraz $j \in \{1, 2, 3, 4, 5, 6, 7\}$. Każdy taki znak może być jednym niżej wymienionych. Dla pierwszych siedmiu linii:

- # — pole niedostępne (poza obszarem rozgrywki),
- . — pole puste,
- W — pole zajmowane przez magazyn,
- X — pole zajmowane przez obiekt (budynek lub samochód),
- 1-9A-U — pole zajmowane przez worki z piaskiem, odpowiednio od 1 do 30 worków,
- Z — pole zajmowane przez więcej niż 30 worków z piaskiem.

Dla kolejnych siedmiu linii:

- # — pole niedostępne (poza obszarem rozgrywki),
- . — pole puste,
- b — pole z przynajmniej jednym żukoskoczkiem,
- B — pole z przynajmniej jednym żukoskoczkiem, który niesie przynajmniej jeden worek z piaskiem,
- C — pole z żukoskoczkiem, który prowadzi samochód.

SHOW_STACK Zwraca opis worków znajdujących się na konkretnym polu w pobliżu żukoskoczka. Współrzędne pola podaje się jako wartości względne w odniesieniu do pola zajmowanego przez żukoskoczka.

Parametry:

- ID ($ID \in \mathbb{N}$) — numer ID żukoskoczka,
- D_X ($D_X \in \{-3, -1, -1, 0, 1, 2, 3\}$) — przesunięcie wzdłuż osi X ,
- D_Y ($D_Y \in \{-3, -1, -1, 0, 1, 2, 3\}$) — przesunięcie wzdłuż osi Y .

Dane (od serwera):

Pierwsza linia odpowiedzi zawiera jedną wartość:

- C ($C \in \mathbb{N}$) — liczba worków na danym polu.

Gdy $C > 0$ serwer zwróci drugą linię odpowiedzi. Linia ta zawiera zestaw C znaków, które opisują poszczególne worki na danym polu. Od lewej strony znajdują się opisy worków będących najbliżej poziomemu gruntu. Znaki w tej linii oznaczają:

- M — worek należący do drużyny,
- 0 — worek należący do innej z drużyn.

FORECAST Zwraca listę przewidywanych fal powodziowych.

Parametry: brak

Dane (od serwera):

Pierwsza linia:

- C ($C \in \mathbb{N}$) — liczba podawanych prognoz

Każda kolejna linia opisuje jedną prognozę za pomocą oddzielonych spacjami wartości oznaczających kolejno:

- Age ($Age \in \mathbb{N}$) — liczba tur od opublikowania prognozy,

- P_{min} ($P_{min} \in \mathbb{N}$, $1 \leq P_{min} \leq 2000$) — dolna granica czasowa (liczba tur od opublikowania nadejścia fali,
- P_{max} ($P_{max} \in \mathbb{N}$, $1 \leq P_{max} \leq 2000$) — górna granica czasowa (liczba tur od opublikowania nadejścia fali,
- H_{min} ($H_{min} \in \mathbb{N}$, $1 \leq H_{min} \leq 30$) — dolna granica wysokości fali,
- H_{max} ($H_{max} \in \mathbb{N}$, $1 \leq H_{max} \leq 30$) — górna granica wysokości fali.

FLOOD_STATUS Zwraca informacje o bieżącej fali powodziowej.

Parametry: brak

Dane (od serwera):

Serwer odpowiada jedną linią postaci:

- H ($H \in \mathbb{N}$) — aktualna wysokość fali powodziowej,
- E ($E \in \mathbb{N} \cup \{\text{NA}\}$) — liczba tur do zakończenia fali powodziowej lub NA jeśli podtopienie aktualnie nie występuje.

MY_STAT Zwraca informacje o osiągnięciach drużyny w aktualnej rozgrywce.

Parametry: brak

Dane (od serwera):

Serwer odpowiada jedną linią postaci:

- G ($G \in \mathbb{N}$) — liczba punktów uzyskana za przemieszczanie worków,
- B ($B \in \mathbb{N}$) — liczba karnych punktów za błędnie ustawione worki,
- S ($S \in \mathbb{N}$) — liczba punktów za zabezpieczony obszar,
- $Total$ ($Total \in \mathbb{N}$) — suma punktów.

WAIT Czeka do momentu rozpoczęcia kolejnej tury.

Parametry: brak

Dane (od serwera):

Serwer odpowiada jedną linią postaci:

- WAITING S

gdzie S ($S \in \mathbb{R}$, $S \geq 0$) oznacza liczbę sekund do zakończenia oczekiwania. Po upływie tego czasu wysyła dodatkowo linię:

- OK

7.10. 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 Wszystko płynie.

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 explorer identifier
102	explorer is busy
103	destination is not a neighbor
104	there are no sandbags to leave
105	unable to put sandbag here
106	unable to buy wheelbarrow outside warehouse
107	not enough sandbags to buy wheelbarrow
108	cannot take another sandbag
109	unable to take sandbag here
110	explorer already has wheelbarrow
111	invalid repeat value
112	explorer cannot move that far
113	cannot get into a jumpcar with a wheelbarrow
114	driver cannot do that
115	passenger cannot do that
116	explorer cannot enter jumpcar while he carries sandbags
117	this is not your jumpcar
118	this jumpcar is full
119	this jumpcar is moving
120	explorer with sandbags cannot go there
121	driver cannot go there
122	explorer cannot see that far
123	unable to see warehouse contents

7.11. Serwery

Rozgrywki będą odbywać się na serwerach, różniących się między sobą ilością i rozkładem poszczególnych zasobów.

Tabela 7: Adresy serwerów rozgrywek.

Nazwa	Adres:Port
Flood1	universum.d124:20005
Flood2	universum.d124:20006

7.12. Przykład

Poniżej znajduje się przykładowy zapis komunikacji z serwerem.

klient → serwer	serwer → klient
	LOGIN
login1	PASS
secret	OK
DESCRIBE_WORLD	OK
	150 200 30 2 50 5.50000
FORECAST	OK
	1
	100 200 400 2 3
LIST_WORKERS	OK
	5
	1 2 2 N 5 4
	2 3 2 N 1 0
	3 5 1 N 1 1
	4 5 5 N 1 0
	5 4 1 N 5 0
FLOOD_STATUS	OK
	0 NA
LOOK_AROUND 1	OK
	#####
	#####
	##..2..
	##.....
	##..X..
	##X...X
	##.WW..
	#####
	#####
	##...bB
	##.Bb..
	##.....
	##.....
	##....b
SHOW_STACK 1 1 -1	OK
	2
	MM
SHOW_STACK 1 1 1	OK
	0


```

LEAVE 5 1 | FAILED 104 there are no sandbags to leave
           |
LEAVE 1 4 | OK
           |
MOVE 2 0 -1 | OK
           |
WAIT       | OK
           | WAITING 0.20000
           | OK
LOOK_AROUND 1 | OK
           | #####
           | #####
           | ##..2..
           | ##.4...
           | ##..X..
           | ##X...X
           | ##.WW..
           | #####
           | #####
           | ##..bbB
           | ##.b...
           | ##.....
           | ##.....
           | ##....b
TAKE 2 1  | OK
           | 1
LIST_WORKERS | OK
           | 5
           | 1 2 2 N 5 0
           | 2 3 1 N 1 1
           | 3 5 1 N 1 1
           | 4 5 5 N 1 0
           | 5 4 1 N 5 0

```