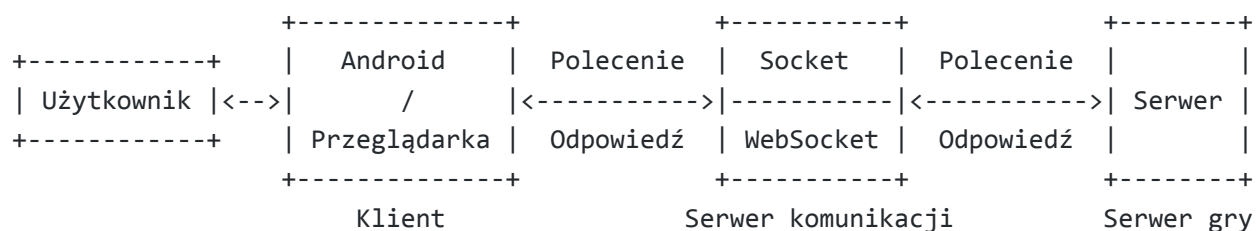


Model komunikacji

Poniższy diagram przedstawia uproszczony model komunikacji:



Gdy klient chce połączyć się z serwerem gry, ustanawia dwukierunkowy kanał transmisji do s gry. W zależności od tego, czy klient jest aplikacją na platformie Android [1] czy aplikacją w przeglądarce [2] jako sposób komunikacji wybiera Socket [1] lub WebSocket [2].

Zarówno polecenia (żądania) jak i odpowiedzi są w formacie JSON.

Polecenia

Format poleceń

Polecenia są w formacie JSON. Polecenie składa się z nazwy polecenia i argumentów polecenia.

Format polecenia:

```
{
  "command": NAZWA_POLECENIA,
  "args": ARGUMENTY_POLECENIA
}
```

Przykładowe polecenie:

```
{
  "command": "put_card",
  "args": {
    "card_id": 123456,
    "row": 1
  }
}
```

Lista poleceń

Pong

Odpowiedź klienta na żądanie [Ping](#) .

set_nickname

Ustawia nazwę użytkownika

Argumenty

nazwa	typ	opis
nickname	string	nazwa użytkownika

Odpowiedź: [SetNicknameResponse](#)

get_players

Pobiera listę graczy

Argumenty Brak

Odpowiedź: [PlayerList](#)

request_game

Prośba o zagranie partii.

Argumenty

nazwa	typ	opis
nickname	string	nazwa użytkownika, do którego wysyłana jest prośba

Odpowiedź: [RequestGameResponse](#)

reject_request_game

Odrzucenie prośby o zagranie partii.

Argumenty

nazwa	typ	opis
nickname	string	nazwa użytkownika, który wysłał prośbę

Odpowiedź: brak

put_card

Wykłada kartę na stół

Argumenty

nazwa	typ	opis
-------	-----	------

nazwa	typ	opis
cardNumber	int	numer karty z ręki (indeks w tablicy, baza 0)
row	int	numer rzędu do którego kładzie się kartę

Odpowiedź: [PutCardResponse](#)

pass

Pass

Argumenty Brak

Odpowiedź: [PassResponse](#)

Odpowiedzi

Format odpowiedzi

Odpowiedzi są w formacie JSON. Polecenie składa się z typu odpowiedzi i właściwej odpowiedzi.

Format odpowiedzi:

```
{
  "type":          TYP_ODPOWIEDZI,
  TYP_ODPOWIEDZI: ODPOWIEDŹ
}
```

Przykładowa odpowiedź:

```
{
  "type": "SetNicknameResponse",
  "SetNicknameResponse": {
    "success": true
  }
}
```

Typy odpowiedzi

Ping

Serwer wysyła `ping` aby sprawdzić czy klient jest podłączony. Klient musi odpowiedzieć na to żądanie (zalecane [Pong](#)).

BooleanResponse

Pola

nazwa	typ	opis
success	boolean	true jeśli się powiodło, false w przeciwnym razie

SetNicknameResponse

Alias [BooleanResponse](#)

Odpowiedź na ustawienie nazwy użytkownika.

RequestGameResponse

Odpowiedź na prośbę o zagarnie partii.

Pola

nazwa	typ	opis
success	boolean	true jeśli się powiodło, false w przeciwnym razie
nickname	string	nazwa gracza do którego wysłano prośbę

RequestGame

Prośba o zagarnie partii.

Pola

nazwa	typ	opis
nickname	string	nazwa gracza, który wysłał prośbę

PlayerList

Lista graczy

Pola

nazwa	typ	opis
players	Player[]	Lista graczy

Player

(typ)

Gracz

Pola

nazwa	typ	opis
name	string	unikalna nazwa gracza
state	PlayerState	status gracza

PlayerState

(typ wyliczeniowy)

Stan gracza.

Wartości

nazwa	opis
FREE	wolny
PLAYING	w trakcie gry

GameStartedResponse

Rozpoczęcie gry. Alias dla [GameState](#)

OpponentActionResponse

Przeciwnik wykonał ruch. Alias dla [GameState](#)

GameState

(abstract / typ)

Stan gry.

Pola

nazwa	typ	opis
opponent	Player	przeciwnik

nazwa	typ	opis
self	Player	gracz
opponentState	OpponentState	stan gry przeciwnika
selfState	SelfState	stan gry gracza
turn	Turn	wskazuje czyja tura

OpponentState

typ

Stan gry przeciwnika.

Pola

nazwa	typ	opis
passed	boolean	czy gracz spasował
score	int	ilość wygranych rund
points	int	ilość punktów gracza
frontRow	Row	pierwszy rząd
middleRow	Row	środkowy rząd
rearRow	Row	tylny rząd
handLength	int	ilość kart w ręce

SelfState

typ

Stan gry gracza. Dziedziczy po [OpponentState](#) .

Pola Dodatkowo:

nazwa	typ	opis
hand	Card[]	karty w ręce

nazwa	typ	opis
graveyard	<code>Card[]</code>	karty na stosie kart odrzuconych

Row

typ

Rząd kart.

Pola

nazwa	typ	opis
elements	<code>Card[]</code>	karty w ręce
effects	<code>Effect[]</code>	efekty działające na rząd
points	<code>int</code>	suma punktów w rzędzie

Card

typ

Karta.

Pola

nazwa	typ	opis
name	<code>enum</code>	nazwa karty (patrz <code>model/Cards.java</code>)
strength	<code>int</code>	aktualna siła / punkty karty

Turn

(typ wyliczeniowy)

Tura.

Wartości

nazwa	opis
-------	------

nazwa	opis
YOUR	tura gracza
OPPONENT	tura przeciwnika

Effect

(typ wyliczeniowy)

Efekt.

Wartości

nazwa	opis
COMMANDERS_HORN	róg dowódcy - podwaja punkty w rzędzie
BAD_WEATHER	zła pogoda - zmniejsza bazową ilość punktów do 1

ActionResponse

Odpowiedź na ruch/akcję gracza. Dziedziczy po [BooleanResponse](#) .

Pola Dodatkowo:

nazwa	typ	opis
game	GameState	stan gry po akcji gracza

PutCardResponse

Położenie karty. Alias dla [ActionResponse](#)

PassResponse

Spasowanie. Alias dla [ActionResponse](#)

GameEndedResponse

Informacja o zakończeniu rozgrywki.

Pola

nazwa	typ	opis
reason	GameEndReason	powód zakończenia rozgrywki

GameEndReason

(typ wyliczeniowy)

Powód zakończenia rozgrywki.

Wartości

nazwa	opis
OPPONENT_DISCONNECTED	przeciwnik się rozłączył
WON	gra zakończona, gracz wygrał
LOST	gra zakończona, gracz przegrał

1 Model

Model jest częścią systemu odpowiadającą zasady rozgrywki. Zawiera klasy reprezentujące karty, ich cechy, oraz elementy planszy gry.

1.1 Klasa Attribute

Klasa `Attribute` to klasa wyliczeniowa, której każdy element reprezentuje specjalną cechę, jaką może posiadać karta. Cechy specjalne wpływają na zachowanie karty w czasie gry – przykładowo, karta bohatera nie jest podatna na żadne efekty, a zagranie karty medyka przywraca graczowi do ręki jedną z kart wcześniej odrzuconych.

1.1.1 Attribute.HERO

Karta bohatera. Nie jest podatna na żadne efekty innych kart. W szczególności, w następnych sekcjach, gdy jest mowa o "wszystkich kartach", należy dodać "z wyłączeniem kart bohaterów".

1.1.2 Attribute.BOND

Karta więzi. Gdy w jednym rzędzie znajduje się n kart o tej samej klasie więzi (patrz: klasa `BondClass`), siła każdej z nich jest mnożona przez n .

1.1.3 Attribute.SUPPLY

Karta wsparcia. Zwiększa o 1 siłę każdej innej karty w swoim rzędzie.

1.1.4 Attribute.COMMANDERS_HORN

Karta morale. Podwaja siłę wszystkich kart w swoim rzędzie.

1.1.5 Attribute.SCOURGE

Karta pożogi. Usuwa z planszy przeciwnika wszystkie karty o sile (efektywnej) równej sile (efektywnej) jego najmocniejszej karty (nie licząc kart bohaterów).

1.1.6 Attribute.COLD

Karta zimna. Zmniejsza do 1 siłę wszystkich kart w pierwszym rzędzie po stronie gracza zagrywającego, jak i przeciwnika.

1.1.7 Attribute.RAIN

Karta deszczu. Zmniejsza do 1 siłę wszystkich kart w drugim rzędzie po stronie gracza zagrywającego, jak i przeciwnika.

1.1.8 Attribute.FOG

Karta deszczu. Zmniejsza do 1 siłę wszystkich kart w trzecim rzędzie po stronie gracza zagrywającego, jak i przeciwnika.

1.1.9 Attribute.GOOD_WEATHER

Karta dobrej pogody. Usuwa efekty zimna, deszczu i mgły.

1.1.10 Attribute.MEDIC

Karta medyka. Przywraca do ręki gracza jedną kartę ze stosu kart odrzuconych (o ile jakaś tam jest).

1.1.11 Attribute.SPY

Karta szpiega. Zostaje umieszczona na planszy przeciwnika, wspomagając go, dodaje jednak zagrywającemu 2 karty.

1.1.12 Attribute.SCORCH

Karta spalenia. Usuwa z pierwszego rzędu przeciwnika wszystkie karty o sile (efektywnej) równej sile (efektywnej) jego najmocniejszej znajdującej się w tym rzędzie karty (nie licząc kart bohaterów).

1.1.13 Attribute.MUSTER

Karta braterstwa. Przy zagranie jednej karty o danym atrybucie, zagrywane są wszystkie posiadane przez gracza karty o tej samej klasie braterstwa (patrz: klasa `MusterClass`).

1.2 Klasa BondClass

Klasa `BondClass` to klasa wyliczeniowa, reprezentująca klasy więzi (patrz: `Attribute.BOND`). Każda karta może należeć do jednej z klas więzi i w takiej sytuacji posiada obiekt klasy `BondClass`.

1.3 Klasa Card

Klasa `Card` reprezentuje kartę w rozgrywce. Jest klasą wyliczeniową; każdy z jej elementów to jedna z możliwych kart, jakie mogą wystąpić w talii gracza.

Pola i metody klasy Card:

`int maxInDeck`

Maksymalna ilość kart danego typu, jakie mogą jednocześnie znaleźć się w talii gracza.

`List<Attribute> attributes`

Lista atrybutów karty (patrz: klasa `Attribute`).

`List<Integer> rowsAllowed`

Lista numerów rzędów, do których można zagrać daną kartę.

`int basicStrength`

Bazowa siła karty, używana do obliczania siły efektywnej (pod działaniem efektów kart specjalnych).

`MusterClass musterClass`

Klasa braterstwa, do której należy karta; w wypadku, w którym nie należy do żadnej, wartość pol jest równa `null`.

`BondClass bondClass`

Klasa więzi, do której należy karta; w wypadku, w którym nie należy do żadnej, wartość pol jest równa `null`.

`int getStrength(Row row)`

Metoda obliczająca efektywną siłę danej karty w kontekście określonego rzędu.

`void specialActions(Player player, int row)`

Metoda uruchamiająca akcje specjalne danej karty, wywoływana przy jej zagranie.

`boolean is(Attribute a)`

Predykat określający, czy dana karta posiada określony atrybut.

`List<Card> VALUES`

Tablica maksymalnej talii; służy do losowania kart z uwzględnieniem częstości ich występowania.

`Card getRandom()`

Metoda zwracająca losową kartę z dostępnych.

1.4 Klasa Effect

Klasa `Effect` to klasa wyliczeniowa, której każdy element przedstawia któryś z efektów, pod których działaniem może znaleźć się rząd kart na planszy. W przypadku wystąpienia obu efektów naraz, siłę kart znajdu-

jących się w danym rzędzie oblicza się, najpierw aplikując skutki działania `Effect.BAD_WEATHER`, a następnie `Effect.COMMANDERS_HORN`.

1.4.1 `Effect.COMMANDERS_HORN`

Efekt morale; podwaja siłę kart.

1.4.2 `Effect.BAD_WEATHER`

Efekt złej pogody; ustawia siłę kart na 1.

1.5 Klasa `Game`

Klasa reprezentująca stan pojedynczej rozgrywki. Gra odbywa się poprzez wywoływanie przez serwer metod tej klasy.

Pola i metody klasy `Game`:

`Player player1, player2, currentPlayer`

Gracze; `currentPlayer` jest równy `player1` lub `player2` w zależności od tego, czyja jest obecnie tura.

`void chooseStartingPlayer()`

Metoda losująca gracza rozpoczynającego.

`Player determineRoundWinner()`

Wywoływana wraz ze spasowaniem obu graczy metoda określająca, który z graczy wygrał rundę.

`void clearTable()`

Metoda opróżniająca stół gry przed następną rundą.

`void switchPlayers()`

Metoda zamieniająca obecnego gracza na jego przeciwnika.

`void pass(Player p)`

Metoda implementująca pasowanie przez gracza `p`.

`void getNewCard(Player p)`

Metoda implementująca pobieranie nowej karty przez gracza `p`.

`void revive(Player p, Card c)`

Metoda implementująca przywracanie karty `c` do ręki przez gracza `p`.

`void putCard(Player p, Card c, int row)`

Metoda implementująca zagrywanie karty `c` przez gracza `p` do rzędu o numerze `row`.

`boolean roundIsOver()`

Predykat określający, czy runda zakończyła się już.

`boolean gameIsOver()`

Predykat określający, czy gra zakończyła się już.

1.6 Klasa `MusterClass`

Klasa `BondClass` to klasa wyliczeniowa, reprezentująca klasy braterstwa (patrz: `Attribute.MUSTER`). Każda karta może należeć do jednej z klas braterstwa i w takiej sytuacji posiada obiekt klasy `MusterClass`.

1.7 Klasa `Player`

Klasa `Player` reprezentuje stan gracza wewnątrz gry. Metody tej klasy są wywoływane przez metody klasy `Game`, reprezentując ruchy graczy.

Pola i metody klasy Player:

`int gameScore`

Ilość dotychczas wygranych rund w danej grze

`Boolean passed`

Odpowiedź na pytanie: czy gracz spasował już w tej rundzie?

`Player opponent`

Przeciwnik gracza

`Card lastPlayedCard`

Ostatnio zagrana karta

`Game game`

Gra, w której uczestniczy gracz

`List<Card> deckInHands, graveyard`

Talie, odpowiednio: kart w ręku i kart odrzuconych

`Row frontRow, middleRow, rearRow`

Rzędy kart: każdy gracz posiada trzy rzędy kart, będące zasięgami działania części efektów specjalnych.

`int getRoundScore()`

Metoda zwracająca wynik gracza w danej turze

`void clear()`

Metoda opróżniająca stół gracza i przygotowująca go do następnej rundy

`void pass()`

Metoda implementująca pasowanie przez gracza

`void scourge()`

Metoda implementująca efekt pożogi (patrz: `Attribute.SCOURGE`)

`void scorch(int row)`

Metoda implementująca efekt spalenia zastosowany do rzędu o numerze `row` (patrz: `Attribute.SCORCH`)

`void play(Card c, int row)`

Metoda implementująca zagranie karty `c` przez gracza do rzędu o numerze `row`

`void playAll(Card c, int row)`

Metoda implementująca zagranie karty `c` przez gracza do rzędu o numerze `row`, a następnie zagranie wszystkich jego kart o tej samej klasie braterstwa (patrz: `Attribute.MUSTER`)

1.8 Klasa Row

Klasa `Row` reprezentuje pojedynczy rząd kart na stole gracza: każdy gracz posiada 3 rzędy (przedni, środkowy i tylni), działanie pewnych atrybutów kart jest ograniczone właśnie do rzędów kart, a każdą kartę można zagrać tylko do określonych rzędów.

Pola i metody klasy Row:

`List<Card> elements`

Karty znajdujące się aktualnie w danym rzędzie

`List<Effect> effects`

Efekty działające aktualnie na dany rząd

```
int getScore()
```

Metoda zwracająca sumaryczną siłę kart w danym rzędzie

```
int getScourgePeak()
```

Metoda zwracająca siłę efektywną najsilniejszej karty w rzędzie, podatnej na efekt pożogi lub spalenia (patrz: `Attribute.SCOURGE`, `Attribute.SCORCH`)

```
void affect(Effect e)
```

Poddaje rząd działaniu efektu `e` (patrz: klasa `Effect`)

```
void disaffect(Effect e)
```

Usuwa z rzędu działanie efektu `e` (patrz: klasa `Effect`)

```
boolean isAffectedBy(Effect e)
```

Predykat określający, czy rząd jest pod działaniem efektu `e` (patrz: klasa `Effect`)

```
void clear()
```

Metoda opróżniająca rząd i przygotowująca go do następnej rundy

```
Row add(Card c)
```

Metoda implementująca zagrywanie do rzędu karty `c`; zwraca ten sam rząd celem chainowania wywołań

```
void sort()
```

Metoda sortująca elementy rzędu celem ułatwienia graczowi rozeznania się na planszy

```
Row scourge(List<Card> graveyard, int value)
```

Metoda implementująca poddanie rząd efektom pożogi lub spalenia (patrz: `Attribute.SCOURGE`, `Attribute.SCORCH`), gdzie karty o sile efektywnej `value` są przemieszczane z rzędu do talii `graveyard`.

Nilfgaard	Punkty	Rząd	Specjalna umiejetnos c	Atrybuty	Opis	Nasza nazwa
Menno Coehorn	10	1	Medyk	Wybierz jedną kartę ze stosu swoich kart odrzucony ch. Odporność na działanie kart specjalnyc h i efektów pogodowy ch	Baldur jest znany z dobrej walki ciężkim mieczem.	Baldur
Wielki Ognisty Skorpion	10	3	-		Taki wąż potrafi zabić jednym ukąszenie m.	Deszcz Węży
Letho z Gulety	10	1	-	Odporność na działanie kart specjalnyc h i efektów pogodowy ch	Loki zjednoczył wiele plemion w jeden kraj.	Loki

Tibbor Egebracht	10	2	-	Odporność na karty specjalne i efekty pogodowe.	Hajmdal strzela z łuku zrobionego z wierzby.	Hajmdal
Morvran Voorhis	10	3	-	Odporność na działanie kart specjalnych i efektów pogodowych	Magni podczas bitwy pomaga z dystansu.	Magni
Nilfgaardzki Łucznik	10	2	-		Z Thrudheim pochodzą najlepsi łucznicy.	Łucznicy z Thrudheim
Stefan Skellen	9	1	Szpieg	Umieść kartę na polu wroga (dodaje siłę do szeregu) i weź dwie losowe karty ze swojej talii.	Aegir potrafi bezszelestnie chodzić	Aegir
Fitz-Oesterlen	7	1	Szpieg	Umieść kartę na polu wroga (dodaje siłę do szeregu) i weź dwie losowe karty ze swojej talii.	Hodur nie przegrał nigdy żadnego pojedynku	Hodur

Assire van Anahid	6	2	-		Frigg podobno potrafi latać na miotle.	Frigg
Fringilla Vigo	6	2	-		Fulla jest bardzo znaną czarodziejką.	Fulla
Cahir	6	1	-		Widar w starym języku oznacza "milczący".	Widar
Renuald aep Matsen	5	2	-		Forseti mieszka w pałacu ze złotymi kolumnami .	Forseti
Zerikkansk i Olbrzymi skorpion	5	3	-		Strzały spadające z nieba mogą roztrzygnąć wynik potyczki.	Deszcz Strzał
	5	1	Więź	Umieść kartę obok podobnie nazwanej karty, żeby	Nawet młody olbrzym jest większy od	Młody olbrzym

Młody emisariusz 2				podwoić ich siły.	dorostego człowieka.	
Zerrikański Ognisty Skorpion	5	3	-		Kamienie mogą rozwalić czyjąś czaszkę.	Deszcz Kamieni
Vattier de Rideaux	4	1	Szpieg	Umieść kartę na polu wroga (dodaje sił do szeregu) i weź dwie losowe karty ze swojej talii.	Powiadają, że Hermod został posłany na świat przez Bogów.	Hermod
Vanhemar	4	2	-		Kvaser zna odpowiedź na każde pytanie.	Kvaser
Rainfar	4	1	-		Sigyn jest synem olbrzyma i kobiety.	Sigyn
					Freja w czasie bitwy	

Cynthia	4	2	-		jeździ na swoim dziku	Freja
Brygada Impera 4	3	1	wież	Umieść kartę obok karty o tej samej nazwie, żeby podwoić siły obu kart	Przed głodnymi dzikami za daleko nie uciekniesz .	Horda dzików
Zdezelowana mangonela	3	3	-		Katapulta w nienajlepszym stanie nadal spełnia swoje zadanie	Podniszczona katapulta
Puttkammer	3	3	-		Hel jest jednym z największych wojowników.	Hel
Morteisen	3	3	-		Mimir pomimo swojego wieku bardzo dobrze walczy	Mimir
Kawaleria Nauzicaa 3	2	1	Wież	Umieść kartę obok karty o tej samej nazwie, żeby	Nikt nie chciałby dostać takim toporem w głowę.	Wyrzutnia toporów

				podwoić siły obu kart		
Sveers	2	2	-		Legenda mówi, że Sol urodził się na słońcu.	Sol
Albrich	2	2	-		Mani bardzo szybko biega.	Mani
Vreemde	2	1	-		Gullweig podobno potrafi używać magii.	Gullweig
Wsparcie łuczników2	1	2	medyk	Wybierz jedną kartę ze stosu swoich kart odrzucony ch (poza Bohateram i i Specjalny mi Kartami).	Wilki przestrasz ą nawet najbardziej odważnyc h wojownikó w.	Horda wilków
Wsparcie oblężnicze	0	3	SUPPLY	Dodaje+1 do wszystkich jednostek poza sobą .	Modlitwa do Bogów czyni cuda	Wsparcie od Bogów

--	--	--	--	--	--	--

Krolestwa Polnocy	Punkty	Rząd	Specjalna umiejetnos c	Atrybuty	Opis	Nasza nazwa
Filippa Eilhart	10	3		Odpornoś ć na karty specjalne i efekty pogodowe .	Mędrzec, który podobno potrafi rozmawiać z Bogiem	Morrigan
Esterad Thyssen	10	1		Odpornoś ć na karty specjalne i efekty pogodowe .	Najdzielni ejszy rycerz w kraju.	Dagda
Jan Natalis	10	1		Odpornoś ć na karty specjalne i efekty pogodowe .	Belenus walczy zakrzywio nym mieczem	Belenus
Vernon Roche	10	1		Odpornoś ć na karty specjalne i efekty pogodowe .	Lug walczy ramię w ramię ze swoim wilkiem	Lug
Katapult a 2	8	3	Więż	Umieść kartę obok podobnie nazwanej karty, żeby podwoić ich siły.	Katapult a pozwala atakować z daleka	Katapult a

Detmold	6	2			Lukan strzela z długiego łuku	Lukan
Balista	6	3			Balista trafia do celu nawet z 800 metrów.	Balista
Trebusz 2	6	3			Kule gradu mogą zabić wrogą armię	Gradobicie
Wieża oblężnicza	6	3			Hulajgród jest nowoczesną machiną oblężniczą	hulajgród
Medyczka burej chorągwi	5	3	medyk	Wybierz jedną kartę ze stosu kart odrzuconych (poza bohaterami i specjalnymi kartami).	Jonsi to druid, który potrafi wyleczyć każdą chorobę.	Jonsi
Rębacze z Crinfrid 3	5	2	wież	Umieść kartę obok podobnie nazwanej karty, żeby podwoić ich siły.	Druidzi nie rozstają się ze swoimi cisowymi laskami.	Druidzi z lbar
Keira Metz	5	2			Brigit jest największym wojownikiem-kobietą.	Brigit
Ves	5	1			Taranis	Taranis

					pije tylko wino	
Zygryd z Denesle	5	1			Esus przed walką odprawia nabożeńst wo dla całej armii.	Esus
Stennis	5	1	szpieg	Umieść kartę na polu wroga (dodaje sił do szeregu) i weź dwie losowe karty ze swojej talii.	Galahad już od małego chciał zostać wielkim rycerzem	Galahad
Sheala de Tancerville	5	2			Aibell została wychowan a przez druidów	Aibell
Sheldon Skaggs	4	2			Uther jest znanym w kraju wynalazcą	Uther
Sabrina Glevissig	4	2			Talitia bardzo dobrze strzela z sosnoweg o łuku.	Talitia
Sigismund Dijkstra	4	1	szpieg	Umieść kartę na polu wroga (dodaje sił do szeregu) i weź dwie losowe karty ze	Arthwys napisał już wiele książek na temat sztuki wojennej.	Arthwys

				swojej talii.		
Komandos Niebieskich Pasów3	4	1	wież	Umieść kartę obok podobnie nazwanej karty, żeby podwoić ich siły.	Szare Szeregi to kilkusobowe grupy skrytobójców.	Szare Szeregi
Yarpen Zigrin	2	1			Gwynedd jest przywódcą krasnolodów z pobliskiej wsi.	Gwynedd
Redański piechur 2	1	1			Akademia Wojskowa w Ostendzie kształci najlepszych żołnierzy	Piechurzy z Ostendy
Biedna Piechota 3	1	1	wież	Umieść kartę obok podobnie nazwanej karty, żeby podwoić ich siły.	Zostali zwerbowani prosto z pola. Walczą przyrządami rolniczymi	Uzbrojeni wieśniacy
Mistrz oblezeń z kaedwen	1	3	Wysokie morale	Dodaje+1 do wszystkich jednostek poza sobą	Powys jest w armii mistrzem obleżeń.	Powys
Talar	1	3	szpieg	Umieść kartę na polu wroga (dodaje sił do szeregu) i weź dwie losowe karty ze	Owain potrafi się bezgłośnie przedostać na tyły wroga.	Owain

				swojej talii.		
--	--	--	--	---------------	--	--

