



小さな修正と欠落した単語



Tom van Dijk authored 3 weeks ago

commands.md 8.47 KB

## プロトコル形式

- 
- `<argument>` 必須。
- `<argument>^n` は正確に `n` 回必要な引数です。  
[引数] オプションの入力または引数です。  
[引数]\* オプションの入力または引数を 0 回以上入力する。  
[引数]+オプションの入力または引数を 1 回以上入力する。

## コマンド一覧

- 
- サーバー側コマンド

- 
- [ハロ](#)
- [一ログ](#)
- [イン](#)
- [プレジデントリスト](#)
- [ニューゲー](#)
- [ムムーブゲ](#)
- [ームオーバ](#)

- [二](#)
- 

## クライアント側コマンド

- 
- [hello](#)
- [ログイ](#)
- [ンリ](#)
- [スト](#)
- [キュー](#)
- [移動](#)

## コマンド形式

コマンドの異なる引数はチルダ(" ~ ")で区切られます。チルダは、明示的に指定された場合を除き、どの引数の一部としても許されません。その他の文字（スペースなど）はすべて使用可能です。

コマンドは改行(`\n`)で区切られます。以下の説明では、この改行が省略されています。

## 拡張機能の取り扱い

HELLO コマンドの引数では、サーバーとクライアントの両方が、サポートしている拡張子を示しています。残りの接続の間、両者はこれを尊重し、相手がサポートしていることを知っているコマンドのみを送信します。たとえば、グローバルチャットメッセージは、**CHAT** 拡張をサポートしているクライアントにのみ配信されます。同様に、クライアントもサーバが**NOISE**拡張をサポートしていないことが分かれば、**NOISE** ハンドシェイクメッセージを送信することはありません。

## エラー処理

不正な入力が提供された場合、サーバーまたはクライアントはエラーで応答することができます。エラーが発生したコマンドは、サーバーまたはクライアントの内部で破棄されます。例えば、クライアントが**MOVE**を送信したが、その手が現在のプレイヤーの合法的な手でない場合、クライアントはその手が違法であることを知ることができたので、**ERROR**が送信される。

構文**ERROR**[`~description`]（エラー

説明文は完全に任意である。デバッグの便宜のために追加することができます。別のグループが作成したソフトウェアと接続する場合、その内容は不明であるため、明示的にユーザーに表示されることはないはずです。

# 初期化シーケンス（ハンドシェイク）

ゲームをプレイする前に、サーバーとクライアントは情報を交換する必要があります。これはハンドシェイクの間に行われる。初期化シーケンスが完了するまで、他のコマンドは許可されません。ハンドシェイクはHELLOメッセージの交換と、オプションでNoiseプロトコルによる暗号化の確立、そしてLOGINによるサーバへのユーザー名の要求から構成されます。クライアントはHELLOメッセージを送信することで開始する。

## HELLO (クライアント)

接続が確立された後、クライアントが送信する最初のメッセージ。

構文HELLO~<クライアントの説明>[~拡張子]\*。

クライアントの説明は、短い文字列でなければならない。クライアントは、関連する拡張文字列を最後に含めることで、どの拡張がサポートされているかを示す。拡張は任意の順序で与えられる。

### 事例紹介

- HELLO~拡張機能がサポートされていない場合、Aliceによるクライアント
- HELLO~Bobのクライアント~CHAT~RANK(CHATとRANKがサポートされている場合)
- 同じ拡張子がサポートされていれば、HELLO~Client~RANK~CHATも合法です(順番は関係ないため)。

## HELLO (サーバー)

クライアントによる最初のHELLOに対する応答。メッセージの詳細は、クライアントと全く同じです。

構文HELLO~<サーバーの説明>[~拡張子]\*。

### 事例紹介

- HELLO~拡張機能がサポートされていない場合、Aliceによるサーバ
- HELLO~ボブの作ったサーバーへようこそ！~CHAT~RANK（CHATとRANKがサポートされている場合
- HELLO~Server~RANK~CHATで同じ拡張子のリストが表示されます。

## LOGIN (クライアント)

クライアントがサーバにユーザ名を要求するために送信します。このユーザ名でログインしているクライアントが既に存在する場合、サーバは以下のように応答する必要があります。

ALREADYLOGGEDIN 、およびクライアントは別のユーザー名を試す必要があります。そうでなければ、サーバはLOGINで応答します。

構文LOGIN~<ユーザー名

### 事例紹介

- LOGIN~Alice
- LOGIN~Johnny Flodder  
(ジョニー・フロッター

## LOGIN (サーバー)

クライアントがログインに成功したときの応答として送信されます。これは、ハンドシェイクの終了を意味する。

構文LOGIN

## ALREADYLOGGEDIN (サーバー)

このユーザ名でサーバに接続しているクライアントが既に存在する場合、クライアントからのLOGINコマンドに対する応答として送信されます。クライアントは別のユーザ名で再試行する必要があります。

注意：サーバーがNOISE拡張をサポートしており、ユーザー名がすでに（別の）公開静的鍵を使用して主張されている場合、このコマンドは返信としても使用されます。詳しくは [ドキュメント](#) を参照してください。

構文alldyloggedin

## 非初期化コマンド

ハンドシェイクが終了すると、サーバーとクライアントが他のコマンドを利用できるようになる。コマンドはまだ時機を逸している可能性があることに注意してください。

サーバーが、すでにゲームに参加しているクライアントに送信するNEWGAME。このようなタイムリーでないコマンドに対しては、受信側でERRORを送信する必要がある。

以下の説明は、各コマンドの詳細を列挙したものです。各セクションのタイトルには、そのコマンドの送信を許可する相手を示しています。

## リスト (クライアント)

クライアントが送信し、現在サーバーにログインしているクライアントのリストを要求します。ゲーム中も含め、クライアントがログインしていればいつでも許可されます。

構文リスト

## LIST (サーバー)

クライアントからのLISTコマンドに対する応答として、サーバーから送信されます。要求しているクライアントを含め、現在サーバにログインしている異なるユーザ名をリストアップします。ユーザ名の順番は任意である。

構文LIST[~username]\*

### 事例紹介

- アリスが唯一のアクティブなクライアントである場合。LIST~Alice
- アリス、ボブ、チャーリーの3人がサーバにいるとき。LIST~Charlie~Alice~Bob

## キュー

クライアントがゲームに参加したいことを示すために送信する。サーバーはそのクライアントをプレイヤーの待ち行列の最後尾に配置する。このコマンドが2回目に発行されると、クライアントは待ち行列から外される。サーバーは応答を送信しない。

構文QUEUE

## NEWGAME (サーバー)

新しく開始されたゲームに投入されたすべてのプレイヤーにサーバーから送信される。キューに入れられたプレイヤー([QUEUE](#)参照)のみがゲームに投入されることができる。プレイヤーは同時に最大1つのゲームにしか参加できない。

このコマンドの最初のパラメータは、1人目のプレーヤーの名前です。第2パラメータは第2プレイヤーの名前である。先攻の

プレーヤーは先手を打つことができる。

構文NEWGAME~<プレイヤー1名>~<プレイヤー2名>とする。

### 事例紹介

アリスとボブの新しいゲームに。NEWGAME~Alice~Bob

## ムーヴ (クライアント)

クライアントから送信され、プレイヤーがどの行または列を押したいかを示す。プレイヤーの順番が回ってきたときのみ許可される。

シンタックスMOVE~<N>

一手は1つの整数Nとして符号化される。

- 0 <= N <= 63の場合、ゲームボードの左上から順に、ディスクをマスに置くことを表す。
  - A1が0
  - H1が7
  - A8は56
  - H8は63
- N==64の場合、通過する手を表します。

ゲームボード上のNの符号化を可視化したのが次の図である。

	A	B	C	D	E	F	G	H
1	0	1	2	3	4	5	6	7
2	8	9	10	11	12	13	14	15
3	16	17	18	19	20	21	22	23
4	24	25	26	27	28	29	30	31
5	32	33	34	35	36	37	38	39
6	40	41	42	43	44	45	46	47
7	48	49	50	51	52	53	54	55
8	56	57	58	59	60	61	62	63

その手が合法であれば、サーバーは[MOVE\(server\)](#)で定義されているように、その手をすべてのプレイヤーに送信する。なお、サーバーが不正な移動と判断した場合は代わりにERRORが返信されます。また、フィールドD4, E4, D5, E5は最初から占有されているので、27 , 28 , 35 , 36の手は常に不正である。

## MOVE（サーバー）

サーバーから送信され、次に打たれる手を示す。これは、その手を実行したプレイヤーを含む、ゲーム内のすべてのプレイヤーに送信される。

シンタックスMOVE～<move>とする。

[サーバーに送信するときと同じように、](#)手をエンコードします。

### 事例紹介

- move~15
- move~64

## GAMEOVER（サーバー）

対局の終了を示すためにサーバーから送信される。サーバーは、勝利したプレーヤーと対局終了の理由を提供する。

構文GAMEOVER～<理由>[~勝者]である。

ゲームが引き分けに終わらなかった場合、勝者は常に含まれる。理由は以下の文字列のいずれかである。

- どちらかのプレイヤーとの接続が切れてゲームが終了した場合、DISCONNECTとなる。接続が続いているプレイヤーを勝者とする。
- どちらかのプレイヤーの勝利でゲームが終了した場合はVICTORY。
- ゲームが引き分けに終わった場合はDRAW。

両プレイヤーはサーバーに接続したままですが、待ち行列にはもう入っていないはずです。キューに入った後、彼らは新しいゲームを、おそらく別の対戦相手とプレイできるはずです。

### 例

- アリスが勝利したゲームにGAMEOVER～VICTORY～アリス
- アリスとサーバとの接続が切れたためにボブが勝ったゲームの場合。GAMEOVER～DISCONNECT～Bob
- 引き分けに終わった試合の場合。GAMEOVER～DRAW