

じゃんけん 2.0 プロトコル

LL Ring 2006 実行委員会

1. ラウンド

このじゃんけんには 2 種類の通信者によって構成されます。

- エージェント … 手を出して勝負をするもの
- コーディネイタ … 対戦表を組み双方の手を仲介し勝敗を決めるもの

じゃんけんは手の繰り返しからなるラウンドを複数回実施することで行われます。1 つのラウンドは 3 つの状態、READY, FIGHT, MATCH で表現されます。

READY

ラウンド開始前。ルールと繰り返しの回数がコーディネイタからプレイヤーに通知されます。

FIGHT

ラウンド実施中。コーディネイタから次の手の催促が直前の相手の手とともに渡され、プレイヤーは次の手を返します。

MATCH

ラウンド終了後。ラウンドの終わった後。

これらの 3 つの状態間の遷移規則は（遷移前の状態, コーディネイタのコマンド, 遷移後の状態）で表すと次の 3 つの規則で表されます。

- (READY, 手の交換, FIGHT)
- (FIGHT, 手の交換, FIGHT)
- (FIGHT, ラウンド終結宣言, MATCH)

ラウンドの終了条件

- 1 ラウンドには全体時間制約があります。
- 1 ラウンドにはじゃんけん回数制約があります。
- コーディネイタは、1 ラウンドの状態遷移を制御中に、上記 2 つの制約のどちらか一方の制約を満たさなくなった場合、ラウンド終了と判断します。

- 全体時間制約によりラウンド終了と判断された場合、プロトコル中で指示された予定じゃんけん回数より少ない回数のじゃんけん実施でありながら、ラウンド終了のための状態遷移がはじまります。エージェントはこれに対応する必要があります。

2. プロトコル

2.1. セッション

TCP 接続一つに対しセッションという単位を割り当てます。セッションはラウンドを複数回実行するための容れもののような働きをします。セッション内では一定の手順にしたがって複数回のラウンドが繰返し実行されます。

2.2. 対戦の概要

コーディネイタを介した対戦は次の順に進みます。

- 接続の確認
- セッションの開始
- ラウンド
- 次のラウンド
- セッションの終了→接続の切断

じゃんけんの状態は上記のラウンドの中の状態で表現されます。

2.2.1. 接続確認

セッションが開始するまでのシナリオには、コーディネイタが TCP 接続を開く場合と、エージェントが TCP 接続を開く場合の 2 つの場合があります。違いは最初にメッセージを送るのがどちらかということです。

コーディネイタが TCP 接続を開く場合

1. コーディネイタが TCP 接続を開きます。
2. コーディネイタが HELLO を送ります。

エージェントが TCP 接続を開く場合

1. エージェントが TCP 接続を開きます。
2. エージェントが HELLO を送ります。

2.2.2. セッションの開始

1. コーディネイタが INITIATE を送り、セッションの開始を要求します。
2. エージェントが INITIATE を送り、セッションの開始が成立します。

2.2.3. ラウンド

1. コーディネイタが READY を送り、ラウンドの準備を要求します。
2. エージェントが READY を送り、ラウンドの準備が成立します。
3. コーディネイタが CALL を送り、じゃんけんの手を要求します。
4. エージェントが MOVE を送り、じゃんけんの手を返します。
5. コーディネイタが RESULT を送り、相手の出した手を知らせます。
6. コーディネイタから READY で知らされた回数だけ 3-5 をこなすと、コーディネイタは MATCH を送り、ラウンドの終了が成立します。

2.2.4. 次のラウンド

再びコーディネイタがラウンドを送れば、一つのセッションのなかで複数のラウンドを行なうことができます。

2.2.5. セッションの終了

コーディネイタは CLOSE を送ることでエージェントにセッションの終了を通知できません。

エージェント、コーディネイタは、TCP 切断によるセッション終了を検出する必要があります。セッションは終了とされ、ルールに従ってコーディネイタが勝敗判断をします (再開はしません)。

実際にはコーディネイタは、全体時間制約に達したら、いきなり TCP セッションを切るかもしれません。:-)

2.3. 状態

TCP 接続が確立してから消失するまでをセッションと呼びます。セッションには 14 の状態があります。

ESTABLISHED

TCP 接続が確立

C_HELLO

コーディネイタが HELLO を送った

A_HELLO

エージェントが HELLO を送った

C_INITIATION

コーディネイタが INITIATE を送った

INITIATED

エージェントが INITIATE に答えた

C_ROUND_READY

コーディネイタがラウンドの開始を要求した

ROUND_READY

エージェントがラウンドの開始を受け入れた

CALL

コーディネイタがじゃんけんの手をエージェントに求めた

MOVE

エージェントがじゃんけんの手を出した

RESULT_UPDATED

コーディネイタが相手の出した手を知らせた

MATCH

コーディネイタがラウンドの終了を通知した

C_CLOSE

コーディネイタがセッションの終了を通知した

セッションの状態はどちらかが送るコマンドによって遷移します。セッションの状態規則を（遷移前の状態, コマンド, 遷移後の状態）で表すと、以下の 16 個の規則になります。

- (ESTABLISHED, Coordinator HELLO, C_HELLO)
- (ESTABLISHED, Agent HELLO, A_HELLO)
- (C_HELLO, Coordinator INITIATE, C_INITIATION)
- (A_HELLO, Coordinator INITIATE, C_INITIATION)
- (C_INITIATION, Agent INITIATE, INITIATED)
- (INITIATED, Coordinator CLOSE, C_CLOSE)
- (C_ROUND_READY, Agent READY, ROUND_READY)
- (INITIATED, Coordinator READY, C_ROUND_READY)

- (ROUND_READY, Coordinator CALL, CALL)
- (ROUND_READY, Coordinator MATCH, MATCH)
- (CALL, Agent MOVE, MOVE)
- (MOVE, Coordinator RESULT, RESULT_UPDATED)
- (RESULT_UPDATED, Coordinator CALL, CALL)
- (RESULT_UPDATED, Coordinator MATCH, MATCH)
- (MATCH, Coordinator READY, C_ROUND_READY)
- (MATCH, Coordinator CLOSE, C_CLOSE)

3. コマンド

個々のコマンドについて説明します。コマンドは 1 行のテキストとして送られます。行末は CR LF (0x0d 0x0a) です。

3.1. HELLO

```
HELLO
```

TCP 接続を開いた側が送ります。コーディネイタが開いた場合もエージェントが開いた場合も書式は共通です。

引数

引数はありません。

3.2. Coordinator INITIATE

```
INITIATE session-id
```

コーディネイタがセッションの開始を送ります。

引数

session-id

一つの接続で複数のセッションを扱うためのセッション ID です。Coordinator INITIATE においてコーディネイタが発番します。これ以降の同一セッションないでのコマンドでは第一引数として同じセッション ID を用います。

3.3. Agent INITIATE

```
INITIATE session-id agent-name capacity
```

Coordinator INITIATE に対するエージェントの返答です。

引数

session-id

Coordinator INITIATE で指定されたセッション ID です。

agent-name

エージェントが名乗る名前です。プロトコルでは用いられませんが表示などのために送ります。

capacity

参加可能なルールの集合の識別子です。現時点で有効な値は 1 のみです。

3.4. Coordinator READY

```
READY session-id round-id iteration rule-id
```

ready はひとつのセッションで複数のラウンドを扱うためのメソッドです。コーディネイタから ready が通知されると ready 状態に戻ります。

引数

session-id

Coordinator INITIATE で指定されたセッション ID です。

round-id

ラウンドを区別するための ID です。

iteration

1 ラウンドで一つのエージェントが何回手を出すかを指定する数値です。通信状況によってこの回数じゃんけんを行えない場合もあるので、ラウンドの終了はこの iteration の値に依存することなく必ず Coordinator MATCH に従ってください。

rule-id

このラウンドで用いるルールを表します。現時点で有効な値は 1 のみです。

3.5. Agent READY

```
READY session-id round-id
```

Coordinator READY に応えて、エージェントがラウンドの準備完了を宣言するコマンドです。

引数

session-id

Coordinator INITIATE で指定されたセッション ID です。

round-id

Coordinator READY で指定されたラウンド ID です。

3.6. Coordinator CALL

```
CALL session-id round-id
```

コーディネイタがエージェントに手の送信を要求するコマンドです。

引数

session-id

Coordinator INITIATE で指定されたセッション ID です。

round-id

Coordinator READY で指定されたラウンド ID です。

3.7. Agent MOVE

```
MOVE session-id round-id move
```

Coordinator CALL に応えて、エージェントがコーディネイタに手を送信するコマンドです。

手は数字に符号化され、ぐー、ちょき、ぱー、はそれぞれ 1, 2, 3 に対応します。
move は 1, 2, 3 のいずれかでなければなりません。

引数

session-id

Coordinator INITIATE で指定されたセッション ID です。

round-id

Coordinator READY で指定されたラウンド ID です。

move

手です。

3.8. Coordinator RESULT update

```
RESULT session-id round-id previous-opposite-move
```

引数

previous-opposite-move

対戦相手からコーディネイタに Agent MOVE で送られた直前の相手の手です。
値はぐー、ちよき、ぱーを表す 1, 2, 3 のほかに無効な手を意味する 0 の場合があります。

3.9. coordinator called MATCH

```
MATCH session-id round-id
```

引数

session-id

Coordinator INITIATE で指定されたセッション ID です。

round-id

Coordinator READY で指定されたラウンド ID です。

3.10. coordinator CLOSE

```
CLOSE session-id
```

セッションの終了を促します。

4. 字句規則

それぞれの識別子は次の字句規則に従います。

```
session-id = word
round-id = word
rule-id = numeric
capacity = numeric
agent-name = word
move = DIGIT
previous-opposite-move = DIGIT

word = 1*32(ALPHA / DIGIT / "-" / "_" / ".")
numeric = 1*DIGIT
DIGIT = "0" / "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9"
```

5. タイムアウト

プレイヤーからのレスポンスが一定時間内に返ってこない場合、タイムアウトとして扱われることがあります。タイムアウトがコーディネイタによって検出されるとコーディネイタは TCP 接続を切ります。タイムアウトは 5 秒とします。

6. TCP 接続

スループットをあげるために、プレイヤーは複数のセッションを同時に張れることが推奨されます。具体的には 5 個以上の接続ができることが期待されます。

Coordinator からの接続を待つサーバー型のエージェントはコーディネイタから来る接続要求を、5 個は受け入れられるのが望ましい挙動です。

Coordinator への接続を開始するクライアント型のエージェントは接続が 5 個に満たない場合、コーディネイタへの接続を開くことが期待されます。

サーバー型、クライアント型いずれのエージェントの場合もタイムアウトなどの理由で、すべての接続が切れた場合に再接続するための方法を用意しておかないと、じゃんけんに参加できないために負けることがあります。

最終更新: 2006-08-14