

J-11 NW 麻雀ゲームの製作

05 番 内山弘樹

22 番 藤澤 倫

担当教員 小笠原祐治

1. はじめに

4 人で遊ぶことができるネットワーク対戦型の麻雀ゲームプログラムを作成する。

麻雀とは、4 人のプレイヤーがテーブルを囲み牌を引いて役を揃えるゲームである。

2. 研究概要

2.1 目的

コンピュータ間でのデータ通信の仕組みについて詳しく知り，プログラミングの知識を深める。

2.2 実行形態

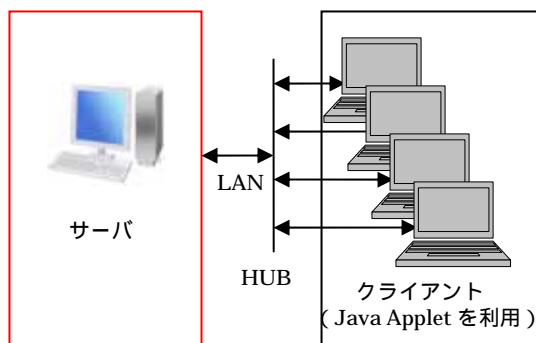


図 1. データの通信

図 1 のようにサーバとクライアントはネットワークで接続されており，Web サイトから参戦ボタンを押し，参戦する。麻雀クライアント JavaApplet が実行されゲームが始まる。

2.3 開発環境

Java 言語を使用し，開発環境は eclipse である。

サーバとクライアントの動作確認は開発している同一の PC で行っている。

3. 操作方法及び機能

プレイヤーのゲーム操作はマウスでおこなう。プレイヤーは参戦ボタン押下後，名前を入力し参加受付をおこなう。参加者が揃うとゲームが開始され，自分の番になるとツモ牌をもらい手牌の中から牌を選んで捨てることができる。

また，相手が捨てた牌が鳴ける場合，ボタンが表示され鳴きを実行することができる。

4. ゲームの進行と通信

4.1 ゲームの進行

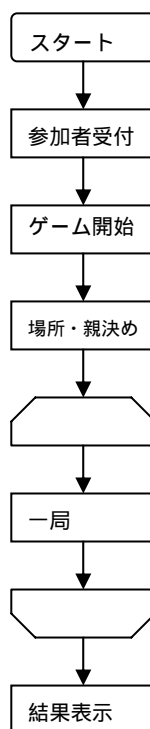


図 2. 全体の流れ

図 2 にある「一局」とは牌を配ってから誰かがあがるか，場が流れるまでのことを言う。

参加受付をしてからゲーム終了までの全体の流れを示している。場所・親は乱数によって決定する。半荘（親が 2 周すること）でゲームを終了するように設定する。

4.2 ゲーム開始までの流れ

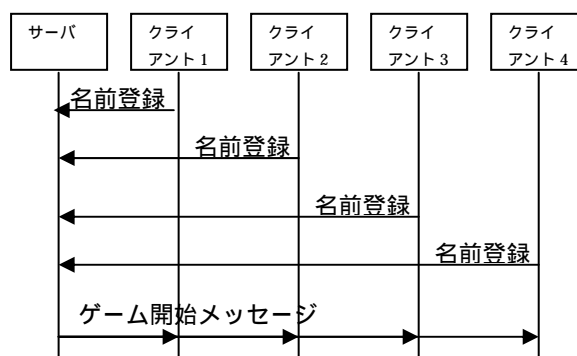


図 3. ゲーム開始まで

参加するには Web サイトから名前を入力して参加ボタンを押す．図 3 のように参加者が 4 人揃ったらゲーム開始メッセージを送る．

4.3 一局の流れについて

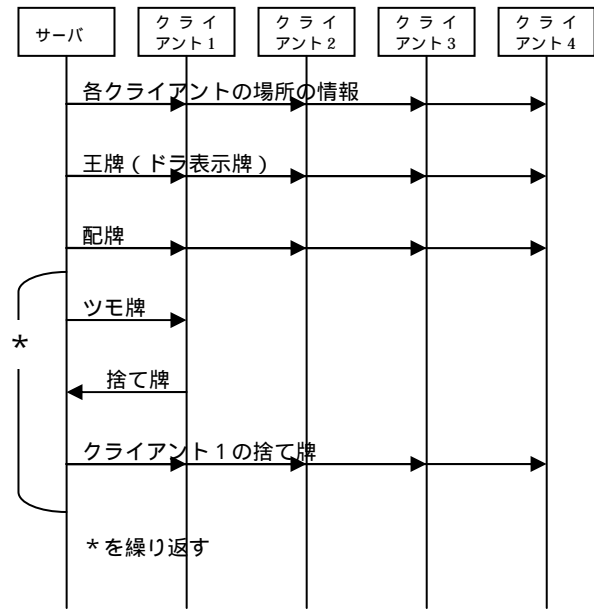


図 4. 一局の開始

まず， 場所と親の情報をクライアントに通知する．その後， ドラを表示して 配牌を配る．親から順番に ツモ牌を配っていく．ツモ牌をもらったプレイヤーは 手牌の中から牌を選択して捨て牌としてサーバに送る．サーバは全クライアントに通知する．以降，*の部分を繰り返す．ポンのときの処理を図 5 に示す．チー・カンの場合も同様にツモの順番が変わる．誰かがあがるか山牌が無くなったら局を終了する．

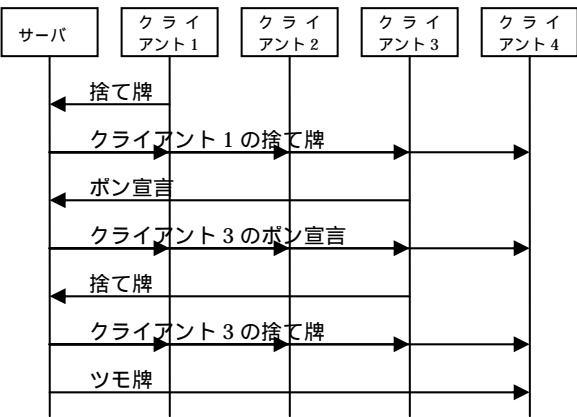


図 5. ポンのときの処理

4.4 通信

メッセージの形式

サーバ，クライアント間で送受信するデータのメッセージ形式を次に示す．

データの種類：プレイヤー-id：データの内容

メッセージの例を表 1 に示す．

表 1 メッセージの例

データの種類	例	説明
Thai	Thai:1:m2,m4,.....	手牌
Thai	Thai:3:j2	山から引いた牌
Sute	Sute:4:j1	捨て牌
Whai	Whai:0:s4,p6,.....	王牌 (ドラ表示牌)
Chi	Chi:3:p5,p3,p4	鳴き牌 (チー)
Pon	Pon:2:j4	鳴き牌 (ポン)
Kan	Kan:1:j2	鳴き牌 (カン)

表 1 は一例なので ,他にもデータの种类がある .

h: m1 ~ m9(一萬 ~ 九萬).

s1 ~ s9 (一索 ~ 九索).

p1 ~ p9(一筒 ~ 九筒).

j1 ~ j7(東,南,西,北,白,發,中).

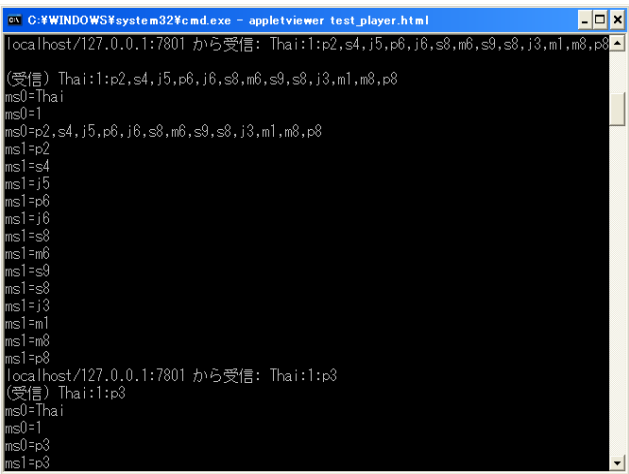


図 6. データ分割処理

処理の仕方は，データを「:」で区切られたところで分割し，それぞれ ms0[0]:ms0[1]:ms0[2]に格納する (図 6 の 4,5,6 行目) . ms0[0]の部分には表 1 で挙げた「Thai」「Sute」などの文字列が格納されている．データを受け取る側がその文字列を判別して処理する 図 6 のように ms0[2]の内容に「,」が付いているときはさらに分割し ,ms1[]の配列に格納する .

5. サーバの機能と構造

Server

```

server
sPort[] comm=new sPort[USER_MAX]
void run()
void receive(sPort p, String mes)
void disconn

```

sPort

```

void setCB(sPortCB cb)
void poll()
void disable()
void send(String sdata)
void run()
void wait_ms(int t)

```

```

aliveCKTimer
void run()

```

図 7. サーバのクラス

5.1 機能

- ・ゲームの進行を管理する .
- ・参加を受け付ける .
- ・それぞれに場所 (東南西北) を与える .
- ・牌をシャッフルする .
- ・王牌の情報を送る .
- ・配牌を配る .
- ・順番にツモ牌を配る .
- ・クライアントから送られてきた捨て牌の情報を全員に送る .

5.2 クラス

server クラス

参加者が 4 人確認されたら山を積み、場所・ドラ・配牌などの情報をそれぞれのクライアントに送る

sPort クラス

スレッドが独立して動作しており、クライアントからメッセージが来たときは Socket が受信処理をする .

6. クライアントの機能と構造

6.1 機能

- ・サーバから送られた配牌などの情報を定められた位置に表示させる .
- ・参加受付の画面で、名前を入力し、参戦ボタンを押すと参加受付される .
- ・自分の番にマウスで手牌をクリックすると、その場所に対応した牌が選択され捨て場に出る .
- ・相手の捨て牌を鳴くときはポンカンチーの対応

したボタンを押すと、相手の牌をもらう .その後、捨て牌を選択する .

Player

```

test_player
mPress
void mousePressed(MouseEvent me)

Thai
Thai(String s)
void paint(int x, int y, int d, int m, ImageObserver io)

Hairetu
Hairetu(int player, int x, int y, int kaeshi)
void add(String s, int dar)
void remove(int i)
void sort()
void paint(ImageObserver io)
cPort comm
void init()
void conn_server(String na)
void receive(cPort p, String msg)
void disconn(cPort p)
void stop()
void paint(Graphics g)

cPort
cPort(Socket socket, String na)
void setCB(cPortCB cb)
void send(String sdata)
void run()
void wait_ms(int t)

```

図 8. クライアントのクラス

6.2 クラス

mousePressed クラス

マウスでクリックしたところの x 座標と y 座標の値を検出する.

Thai クラス

画面に牌 (手牌,捨て牌,鳴き牌,ドラを含む山) を表示させる.

Hairetu クラス

複数の牌を記録するクラスである.牌の表示のさせ方と牌の情報,例えば各プレイヤーの手牌,鳴き牌,捨て牌などを登録する.

主なメソッド

- ・メソッド add
手牌,捨て牌,鳴き牌の配列に牌を追加する.
- ・メソッド remove
手牌,捨て牌の配列から牌を削除する.
- ・メソッド get
データをコピーする.
- ・メソッド sort
手牌の牌を並び変える.

cPort クラス

スレッドが独立して動作しており，サーバからメッセージが来たときは Socket が受信処理をする．

7. メッセージ通信と処理

- ・サーバからクライアントに送信するとき

sPort の `comm[0].send("Thai:1:m3")` でメッセージを送る．

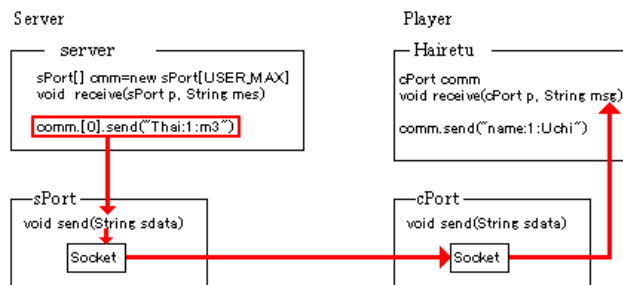


図 9.サーバからクライアントへの処理

図 9 のように server クラスにある“Thai:1:m3”という内容のメッセージをクライアント 1 に送る．

その内容は sPort クラスの send メソッドに行き，そこから Socket へ送られる．その後 Server の Socket からネットワークにデータが送信される．Player の Socket へ受信される．Socket は常に動作していて，独立してスレッドとして動いているのでメッセージが来たときすぐに受け取ることができる．Socket で受け取ったあとは receive メソッドに送られ，メッセージ毎に異なる処理を実行する．

- ・クライアントからサーバに送信されるとき

cPort クラスの `comm.send("name:1:Uchi")` メッセージを送る．

クライアント 1

クライアント 2

クライアント 3

クライアント 4

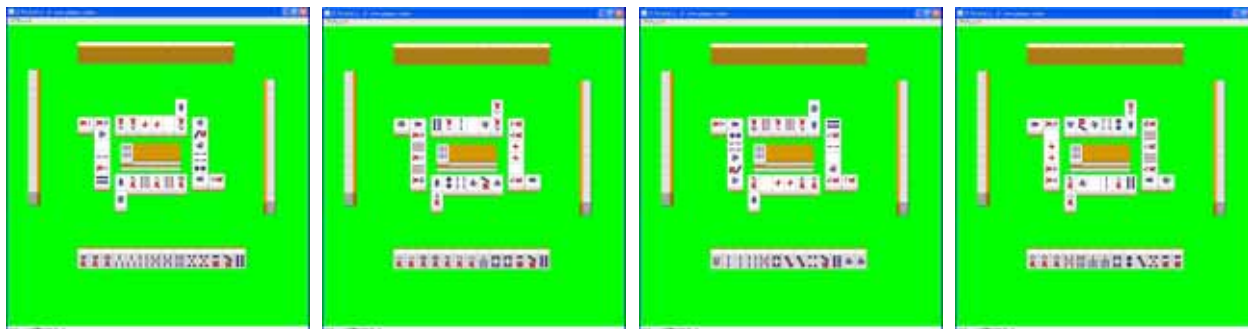


図 11. ゲーム画面

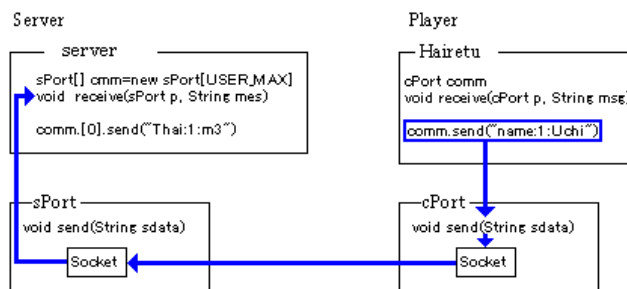


図 10.クライアントからサーバへの処理

サーバからクライアントにデータが送られたときと同様，メッセージは Player の send メソッドから Socket に送られ，Server の Socket に送られる．受け取ったサーバは receive メソッドでメッセージによって異なる処理を実行する．

現在，ゲームを開始するとドラ牌の表示，各プレイヤーの場所，配牌の情報を送信する．その後，親から順番にツモ牌を配る．プレイヤーは自分の番になったら牌を切ることができる．各プレイヤーのゲーム画面を図 11 に示す．

8. まとめ

ネットワーク麻雀の製作に挑戦した．現状は，ツモ切りすることができるが鳴くことはできない．

今後は鳴いたときの動作・役の判定を進め，麻雀らしい機能を実装していく予定である．