17. java 言語によるネットワーク麻雀の作成

18　湊　雄太郎

指導教員　小笠原祐治

1.　はじめに

java 言語を使用し，ネットワーク通信を用いた対戦麻雀を作成する．

2.　研究概要

2.1　目的

麻雀の政策を通して，Java 言語を用いたネットワーク通信とプログラミングについての理解と知識を深める

2.2　実行形態

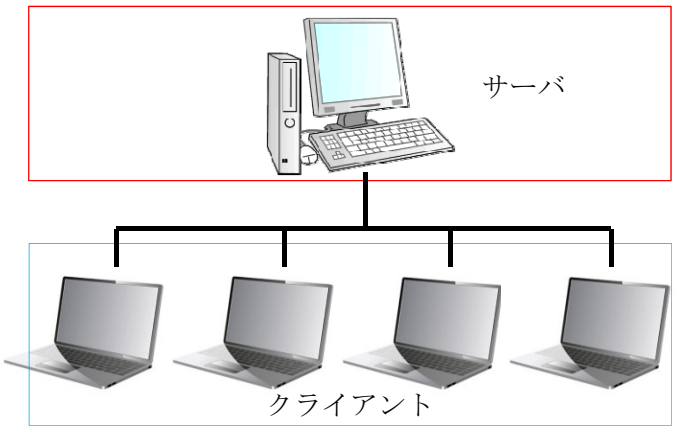


図 1，通信について

図 1 のようにクライアントはネットワークでサーバに接続されており，HP を開くとアプレットが起動しサーバとの通信を開始する．

2.3　開発環境

言語は java, 開発環境は eclipse を使用している．またサーバとクライアントの通信の動作確認は同一 pc で行っている．

3.　ゲームの進行と通信

3.1　全体の流れ

接続からゲーム終了までの流れは図 2 のとおりである．ゲームは半荘(親を二回ずつ行うこと)で終了とする．

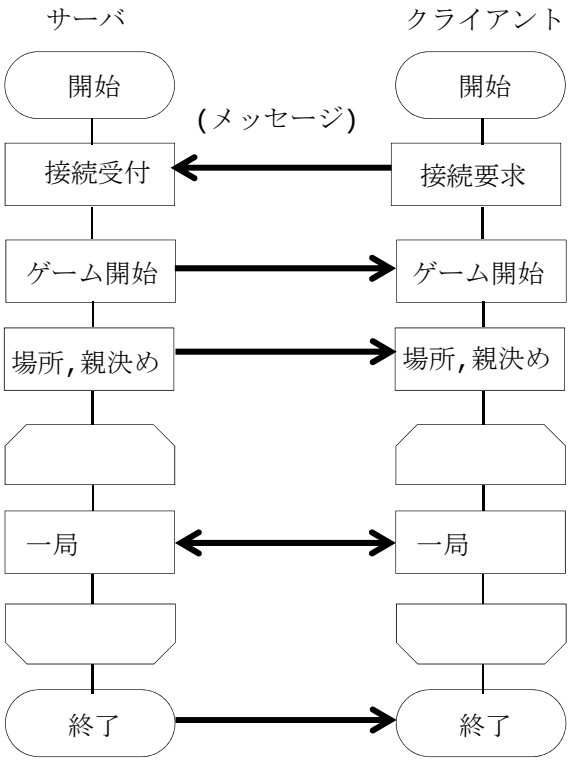


図 2，全体の流れ

3.2　接続からゲーム開始まで

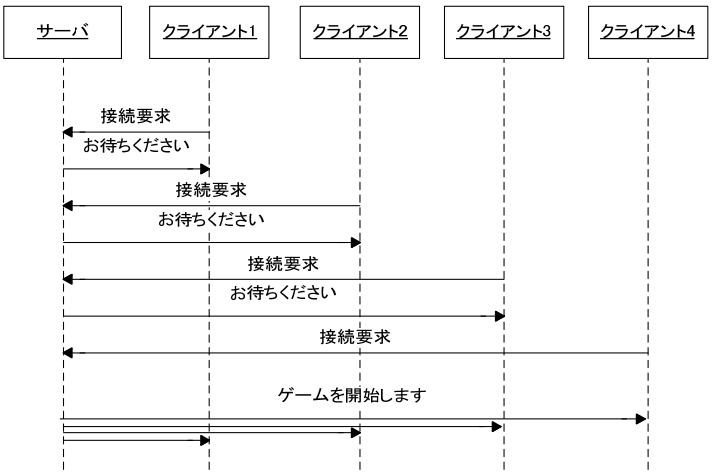


図 3，開始までの流れ図

クライアントはアプレット起動後，接続ボタンを押してサーバに接続要求を行う．サーバは，図 3 のように接続数が 4 人に達したら全クライアントにゲームの開始を通知する．

3.3　ゲーム開始からの動作

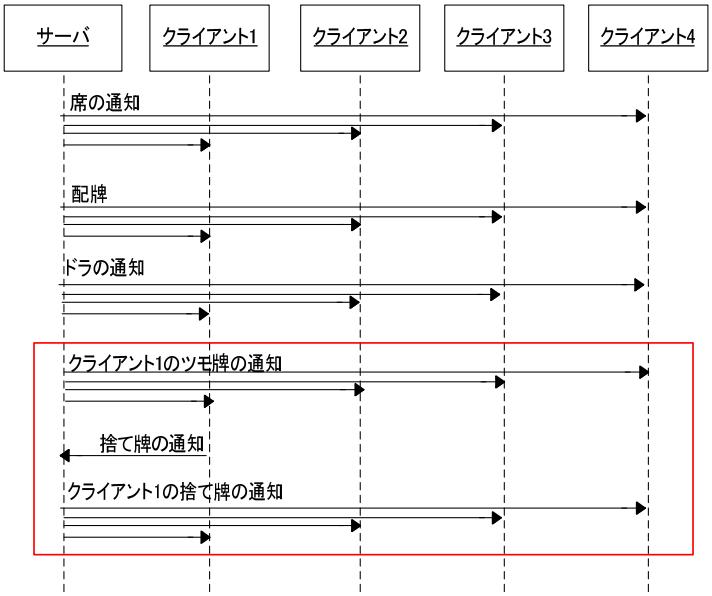


図 4，開始からの流れ図

図 4 は，ゲーム開始から一人目が牌を捨てるまでの流れである．これ以降の動作については一局の終了条件(配牌を含めた 122 枚を配り終わる，誰かがあがる，四開槓，四風連打，九種九牌，四家立直)を満たすまで赤枠部分を繰り返していく．

3.4　鳴きの動作

図 6，鳴き動作について

図 7，鳴き実行時の手牌

図 5 は鳴き発生時の流れである．クライアント側の動作は，サーバ側から捨て牌の通知を受け取った時点で，この捨て牌に対して鳴きを行えるかどうか判断し，鳴けない場合は自動で鳴き無しの通知をサーバ側に送信する．鳴ける場合は，可能な鳴きのボタンを押し鳴きの宣言をサーバに送るか，キャンセルボタンを押し，鳴き無しの通知をサーバに送るかを選択することができる．

サーバ側の動作は，捨て牌の通知をした時点で全員からの鳴きについての通知を待つ．鳴きを宣言するプレイヤーがいた場合は，全員に鳴きの発生を通知する．鳴き無しの場合，つまり全員から鳴き無しの通知を受け取った場合は手番を次に回し，ツモ通知を全員に送信する．

3.5　通信メッセージ

サーバ，クライアント間で送受信するメッセージは，送信元，メッセージの種類，データ部で構成された一つの文字列である．受け取った側はその文字列を各要素に分割し，メッセージの種類に応じて動作を決定する．

送信元	:	種類	:	データ	,	データ…

分割には String.split メソッドを用いる．分割文字は“:”，送信するデータが多い場合，データ部内の分割に“,”を使用する．

送受信するメッセージの種類を設定しその種類に応じて処理を決定する．以下は現時点で採用しているメッセージの種類とその目的である．

(1)サーバから送信

種類	目的
ACC	通信待ちを通知
START	開始の通知,席の通知
HAIPAI	配牌する
TUMO	各クライアントのツモ牌を通知
SUTE(中継)	各クライアントの捨て牌を通知
PON(中継)	ポン宣言の通知
KAN(中継)	カン宣言の通知
DORA	ドラを通知
END	終了の通知

(2)クライアントから送信

種類	目的
CON	接続要求
SUTEHAI	捨て牌を通知
PON	ポン要求を通知
KAN	カン要求を通知
CANCEL	鳴き無しの通知

接続待ちや接続要求をイベント処理にするために以下の二つのクラスとインターフェースを作成した．これはサーバとクライアント両方が共通して使用している．（PortCon，PortCom クラス）

4.　サーバについて

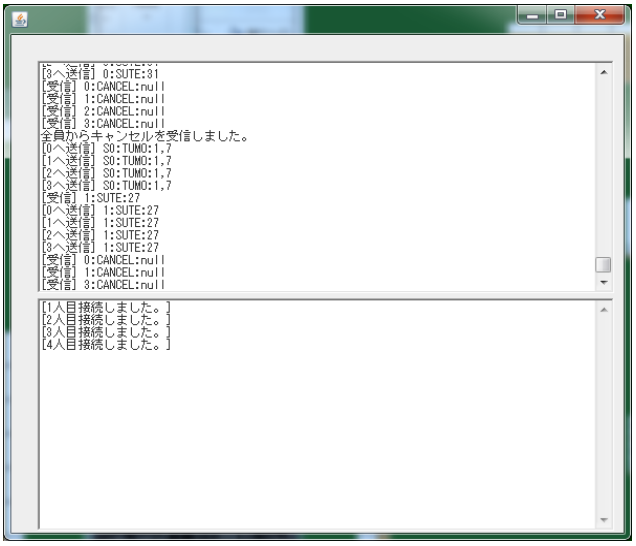


図 8，サーバ画面

4.1　サーバ機能

・　メッセージの送受信を行う
・　送受信したメッセージを表示する
・　クライアントの接続を受け付ける
・　乱数による場所決め
・　ゲーム開始時の山のシャッフル
・　ドラの通知
・　配牌を行う
・　ツモ牌を通知する
・　捨て牌を通知する
・　終了宣言をする

4.2　クラス

サーバクラス(Mahjong_server.class)

参加者が 4 人確認されたら山をシャッフル(yama_shuffle メソッド)し，場所，ドラ，配牌の情報をクライアントに送る．それ以外はメッセージの中継を主に行う．

5.　クライアントについて

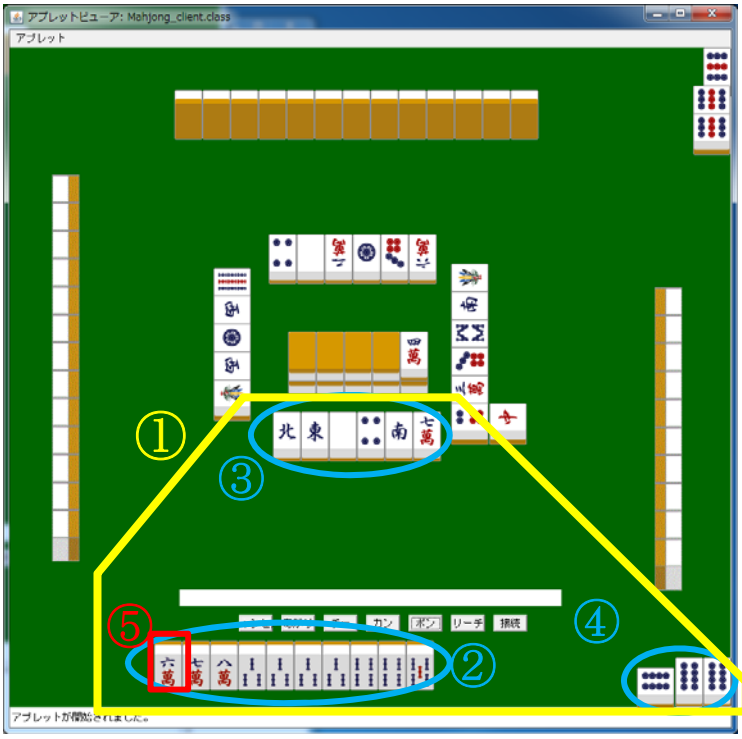


図 9，クライアント画面

5.1　クライアント機能

・　起動後，接続ボタンを押すと接続要求を行う．
・　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手牌から河へ送られる．
・　鳴き可能時に，可能な鳴きに対応したボタンを押すことで鳴きを実行，表示．その後捨て牌選択へ移る．

5.2　クラス

クライアントクラス(Mahjong_client.class)

接続，描画，メッセージの種類に応じた処理，マウスクリック時の処理，ボタン押下時の処理等，ゲームを行うためのクライアント側の処理を行う．4 人分のプレイヤークラスの情報を持つ

①プレイヤークラス(Mahjong_player.class)

一人分のプレイヤーの情報を定義する．Mahjong_list 型の②手牌，③捨て牌，④鳴き牌リストを持つ．

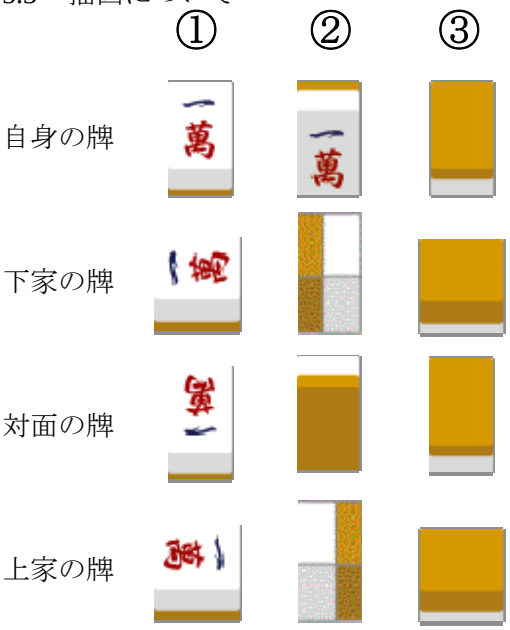
②～④リストクラス(Mahjong_list.class)

複数の牌を格納するための arraylist を用意する．処理はリストのソート，要素の追加，削除，鳴きの判定等を行う．

⑤牌クラス(Mahjong_hai.class)

牌一枚が持つ情報(牌の種類，牌の向く方向，牌の名前)の定義，表示処理をする．

5.3　描画について



プレイヤークラスが持っている手牌，捨て牌，鳴き牌リストを，指定した位置・方向で描画する必要がある．

ゲームの進行に合わせて，各リストへ牌を追加，削除していくが，リストに追加する牌は，牌クラスで定義されたもので，牌の種類，状態，牌の名前の情報を持っている．

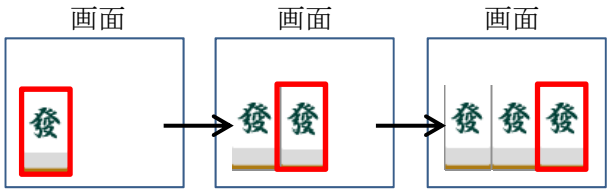
牌の状態は，プレイヤーごとに，①倒れていて表になっている牌，②起きている牌，③倒れていて裏になっている牌の 3 種類．計 12 種類である．

状態の定義は，各状態すべての牌画像を読み込んでいる Image 配列の配列番号に対応させている．

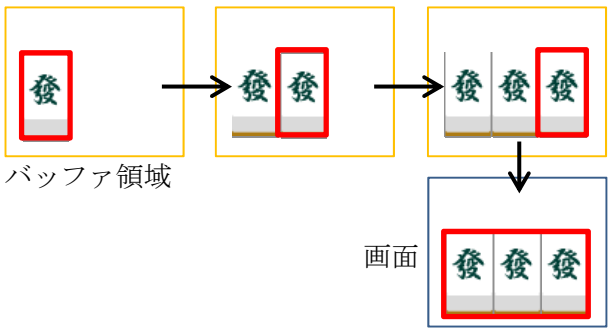
描画の方法は、リストに格納されている牌を一つ一つずつ重ならないように描画位置をずらして表示していく。ずらす幅は、牌自身が持っている幅を参照しているため、リストに格納されている牌の状態が変わってもずれが発生しない。

また、アプレットの描画の際、画面に描画する情報が多いと描きかけの画面が表示され、画面がちらついたように見えることがあるため(リフレッシュレート)、ダブルバッファリングという手法を用いてちらつきを抑えている。

・　通常



・　ダブルバッファリング



6.　最後に

6.1　問題点

重複する判定のパターンの多さから，チー判定，テンパイ判定，上がり判定，役判定の機能が完成していない．それに伴い点数計算も実装ができていない状況である．また，4 人目以降の接続者に対する処理や，全プレイヤーの手牌を見ることができる観戦機能も未実装である．

6.2　まとめ

NW 麻雀の練習として製作した NW オセロの作成に時間をかけすぎてしまった．実装できていない機能が多いので，産技短展までには完成させたい．