

# ネットワークを介して対戦可能な ダイヤモンドゲームの試作

沼田研究室所属

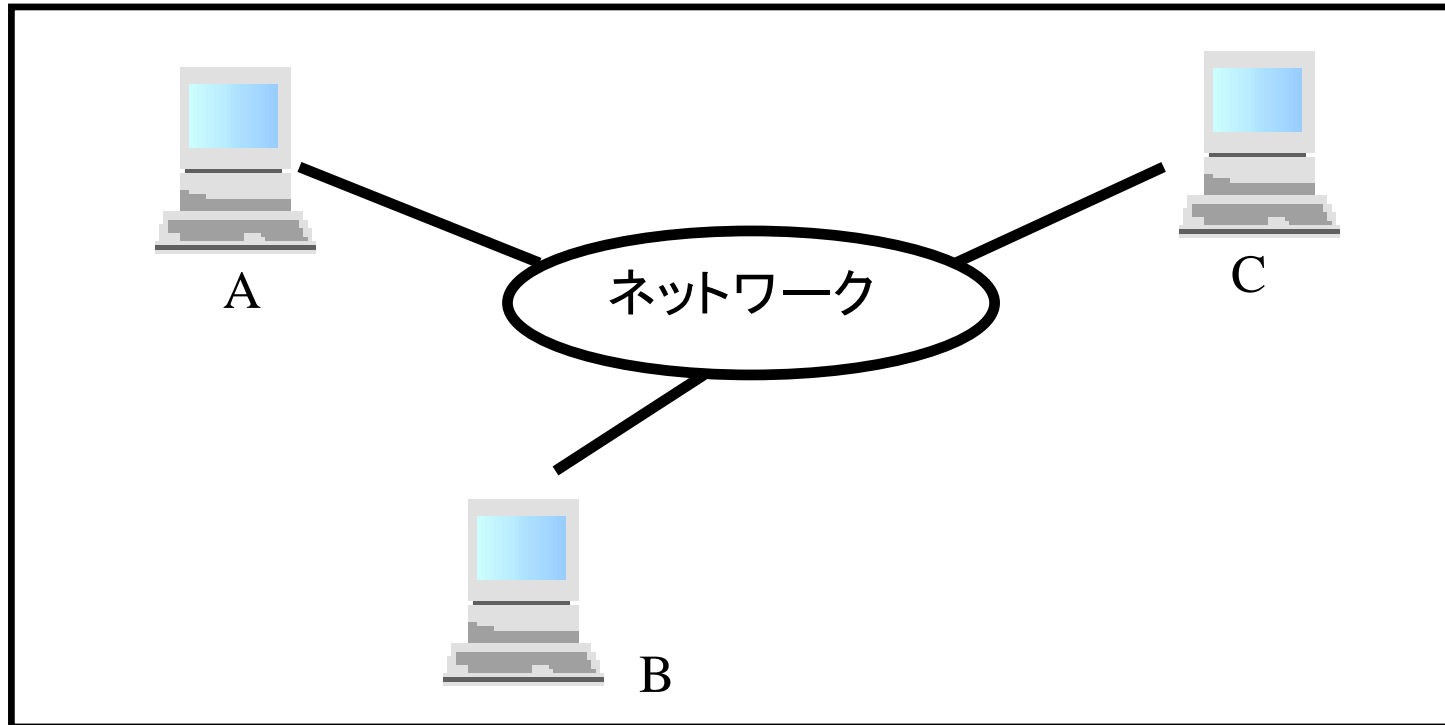
4495056 鈴木 宏史

4495095 三塚 信也

# 発表構成

- 1 はじめに
- 2 ダイヤモンドゲーム
- 3 動作環境の概要
- 4 ソフトウェアの動作概要
- 5 戦略( 4つの基本戦略プログラム)
- 6 対戦実験による考察
- 7 総括的結論

# 1 はじめに

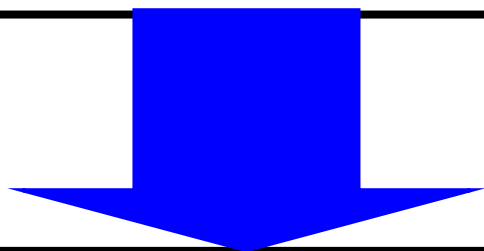


ネットワークによって

- 個々のコンピュータに格納されている情報の共有
- 遠隔地のコンピュータとの間で情報の送受信が可能

## 研究動機

計算処理を複数のコンピュータで行う  
並列計算が可能である



最適化計算の高速化  
が図れる



ネットワークを介して対戦可能な  
ダイヤモンドゲームを作成する  
ことで並列計算を行うための基  
礎的な技術の確立となる

# 卒業研究の目標

- ①本研究を通じて並列計算の技術を確立
- ②参加人数(0~3人)にフレキシビリティを持たせた  
ダイヤモンドゲームの設計と実装
- ③ダイヤモンドゲームの戦略を検討し、対戦実験により優劣を判断

## 2 ダイヤモンドゲーム

### 2.1 設定

人数 ゲームは三人で行います。  
(コンピュータを含む)

- 設定
- ①駒は全部で3種類(赤、青、黄)あり、それぞれについて10(個/色)あります。
  - ②順番はどの色から開始しても良い。その後は...→赤→青→黄→赤→...で進んでいきます。

勝敗 向かい側の自分の陣地へ、すべての駒を早く並び終えた方が勝ちです。

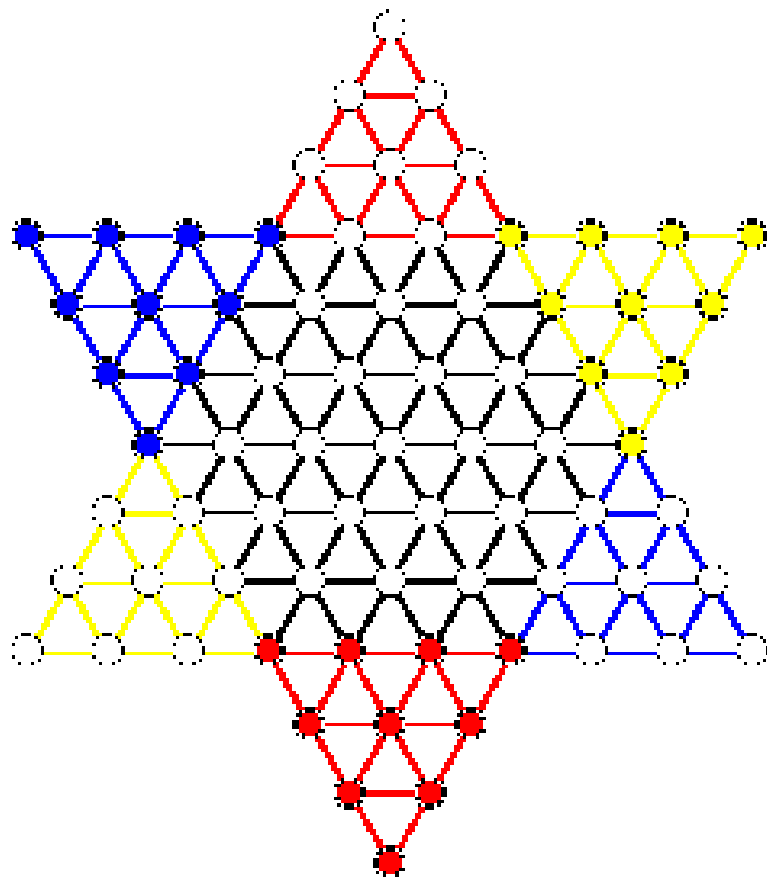


図 ダイヤモンドゲーム

## 2, 2 駒移動のルール

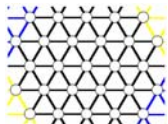
①駒は現在点と線で結ばれた点へ1つだけ進むことができます。

(駒がすでに存在する場合には、その点上には進めません。)

②自分の駒から6方向にのびた線上で最初の点に駒があり、かつ、2つ目の点に駒がない場合には飛び越えて2つ目の点に進むことができます。

(何色の駒でも飛び越すことができます。)

③1つ置きに駒が並んでいる場合は、連続して6方向自分の好きな所まで進めます。



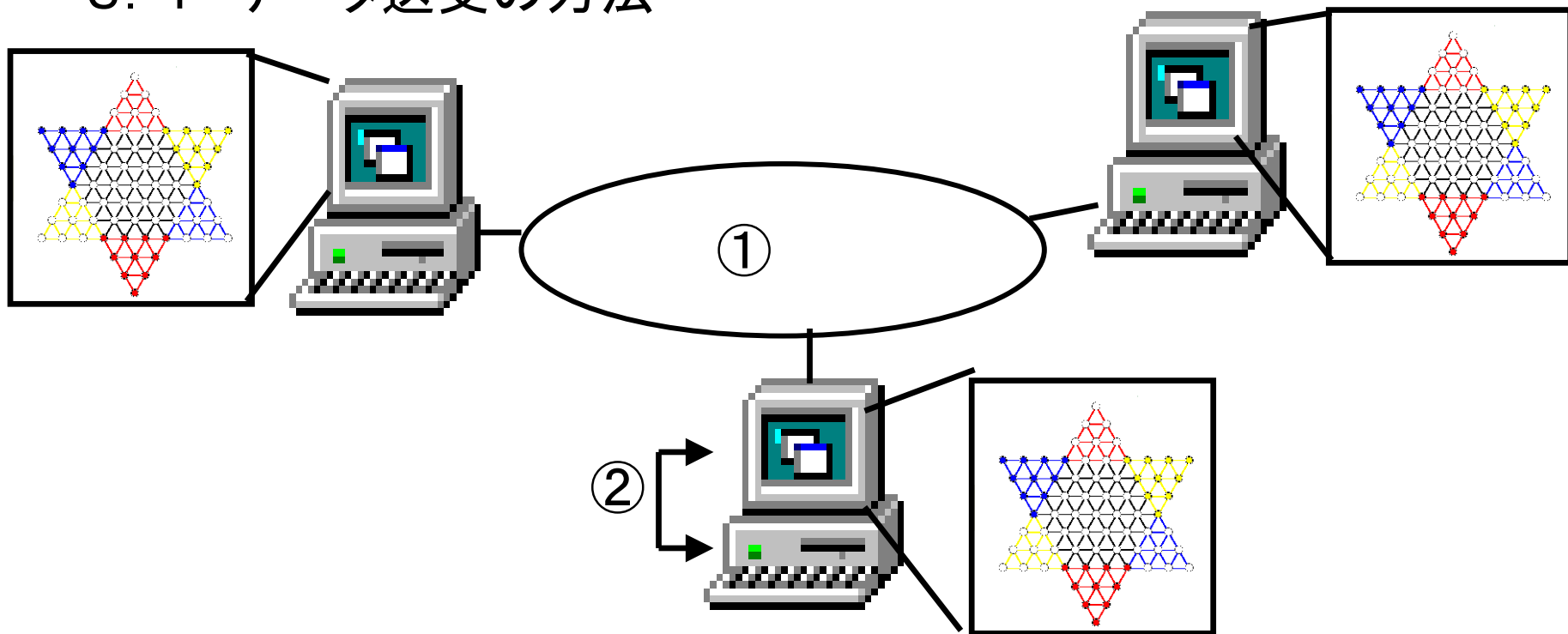
● 動かす駒(現在点)

● 移動可能点

● 既存の駒(赤・青・  
黄のうち何色でも  
良い)

### 3 動作環境の概要

#### 3.1 データ送受の方法



①コンピュータ間でのデータの送受

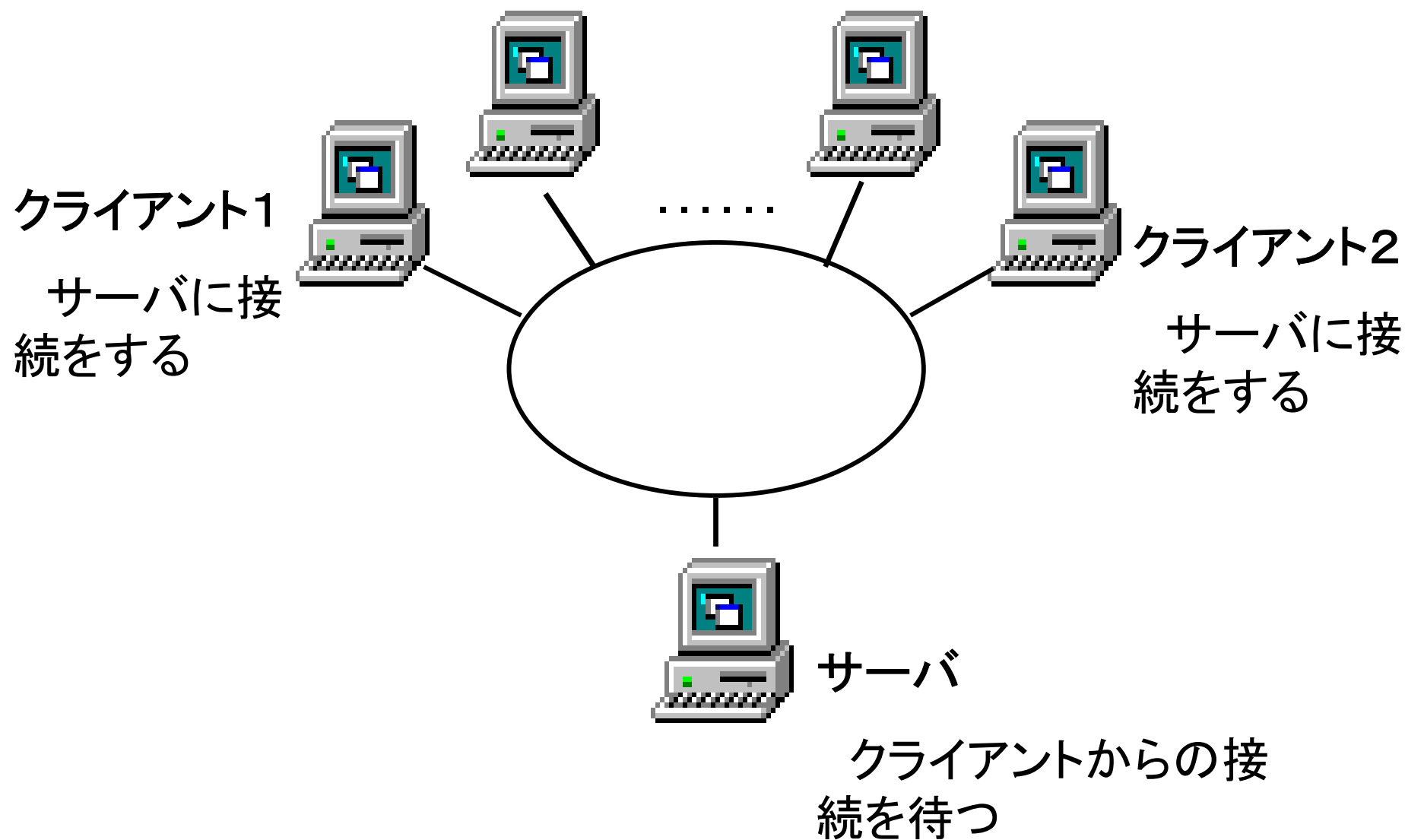
⇒TCP/IP プロトコルを使用

②コンピュータ内のプログラムと転送制御部でのデータの送受

⇒Delphi3.1のネットワークコンポーネントを使用

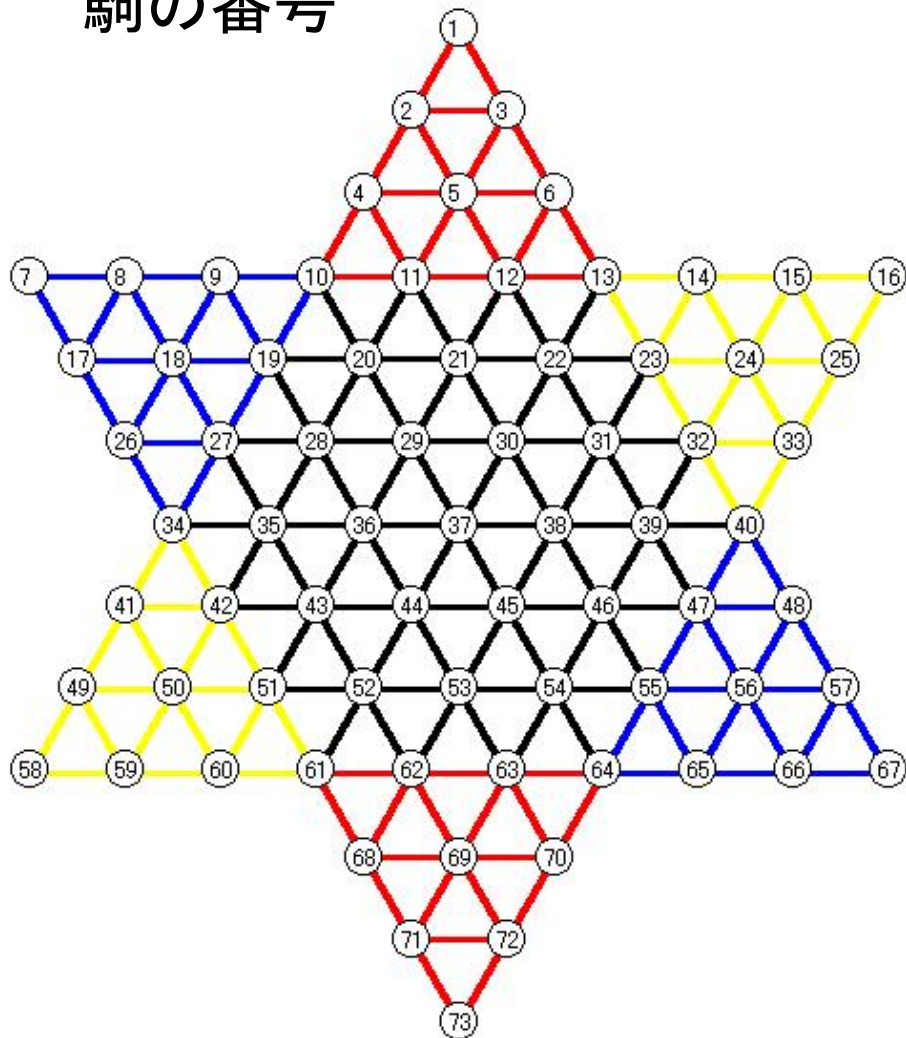


### 3. 2 コンピュータ同士の接続



### 3.3 ダイヤモンドゲームで送受されるデータ

駒の番号



色の番号

赤:10 青:20

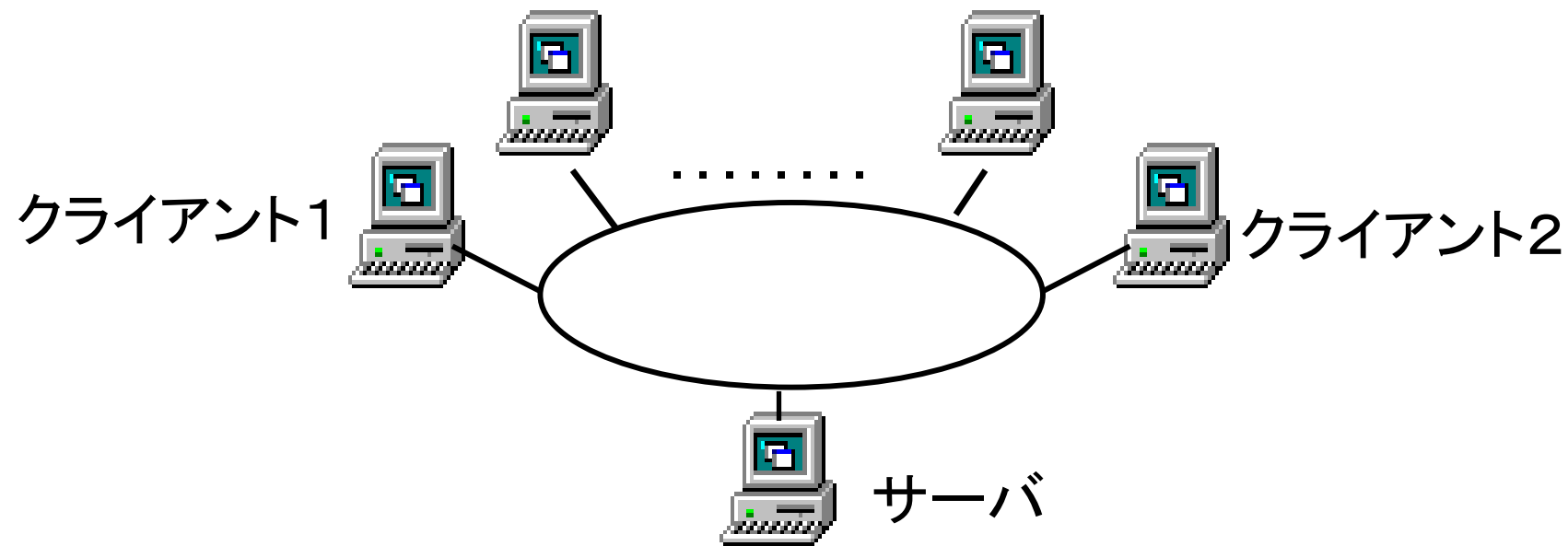
黄:30 緑:40

白:50

送られるデータ例

1023=赤の23の位置

### 3.4 ソフトウェアの機能



・ネットワークを介して0～3人でゲームをすることができる

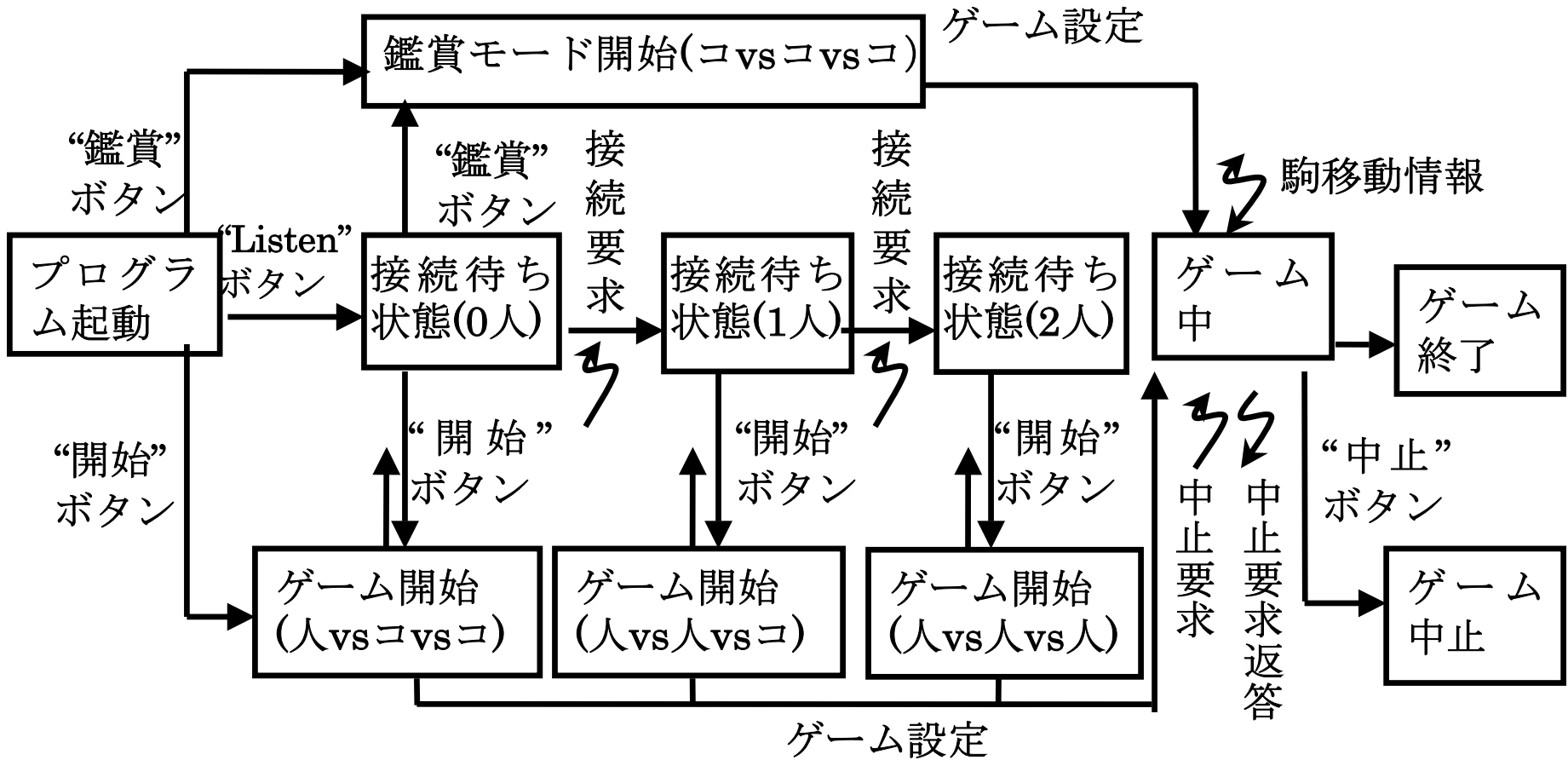
(0人: コvsコvsコ、 1人: 人vsコvsコ )

(2人: 人vs人vsコ、 3人: 人vs人vs人) コ: コンピュータが駒を動かすの意味

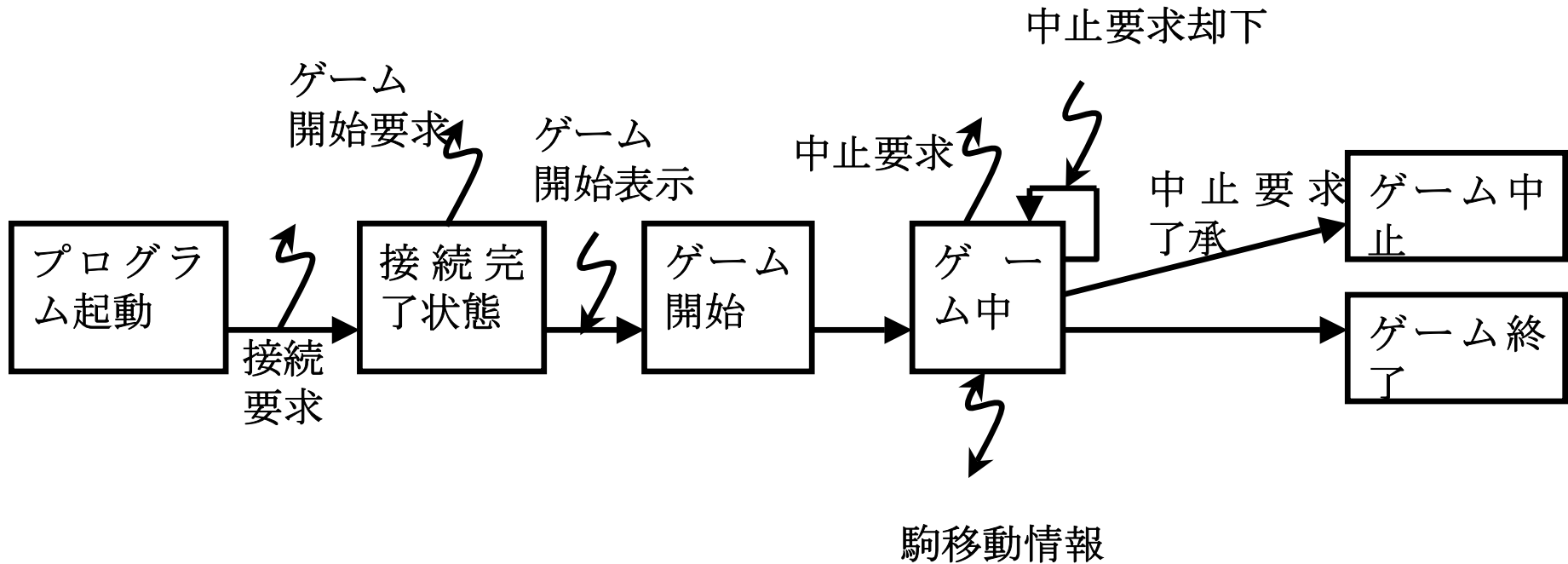
・チャットをすることができる

# 4 ソフトウェアの動作概要

## 4.1 サーバの動作

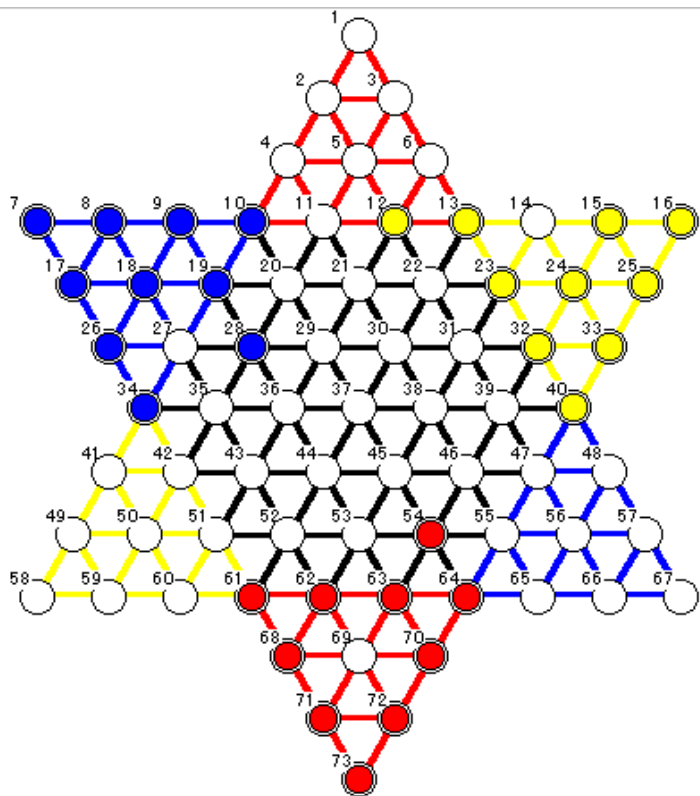


## 4.2 クライアントの動作

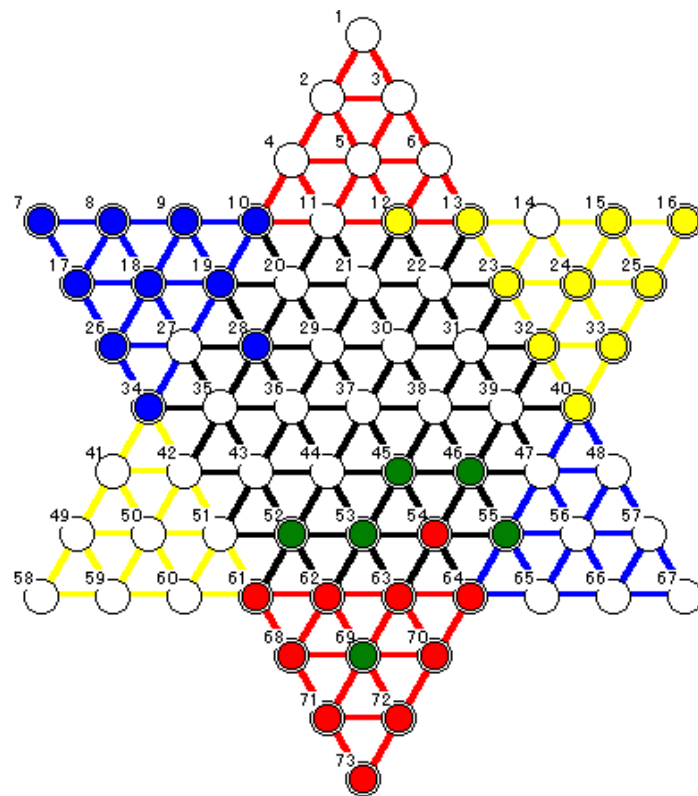
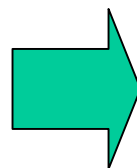


## 4.3 工夫した点

①オプションで駒の移動可能範囲を表示する機能をつけた。

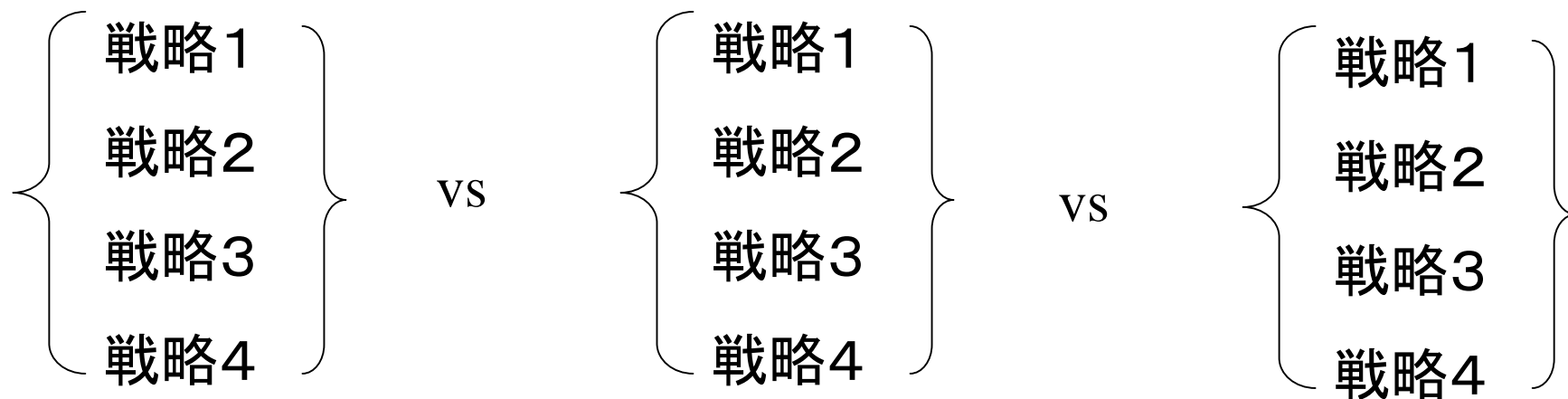


移動可能表示前



移動可能表示後

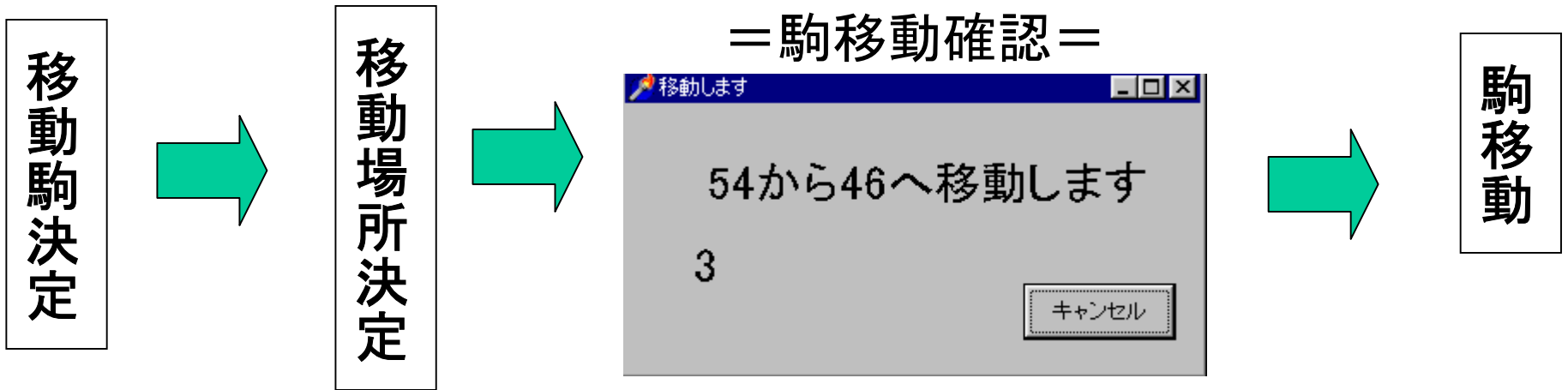
②コンピュータ同士の対戦を鑑賞できるようにした。



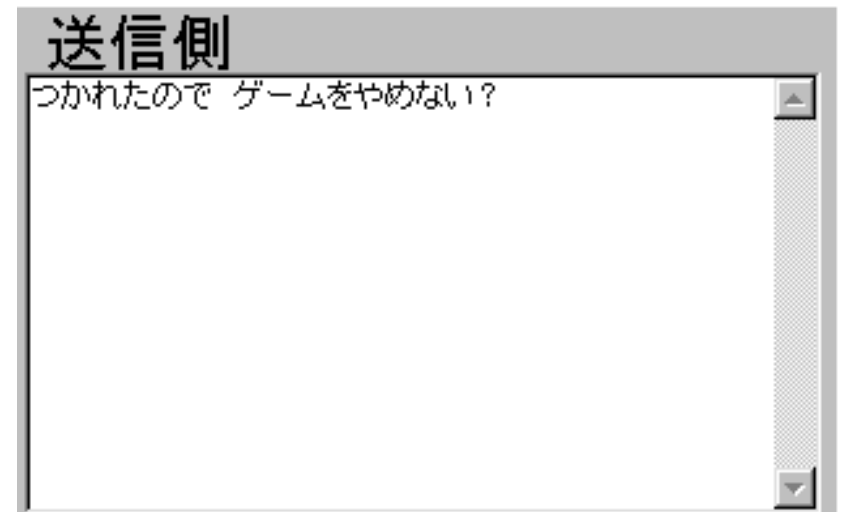
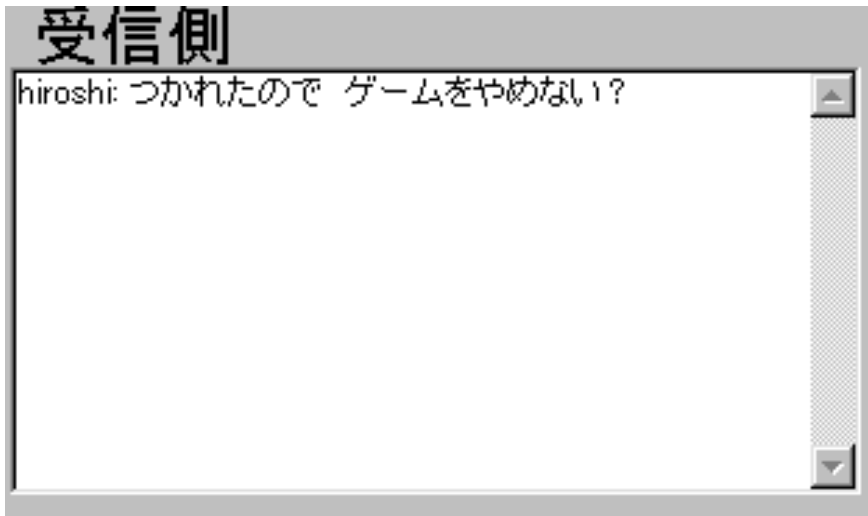
③0～3人までの状態ならどれでもゲームが出来る様に作成した

0人:	コンピュータ1	VS	コンピュータ2	VS	コンピュータ3
1人:	人	VS	コンピュータ1	VS	コンピュータ2
2人:	人	VS	人	VS	コンピュータ1
3人:	人	VS	人	VS	人

④駒の移動決定後 3 秒以内ならばキャンセル出来るようにした。

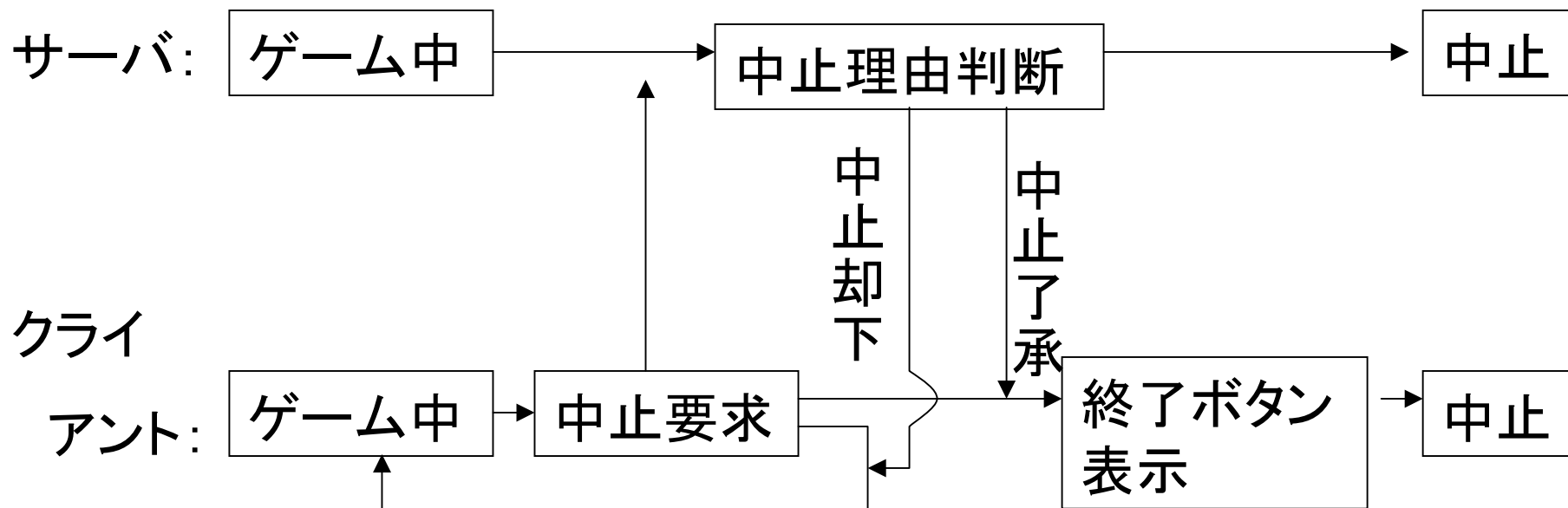


⑤お互い会話ができるようにチャット機能を設けた。





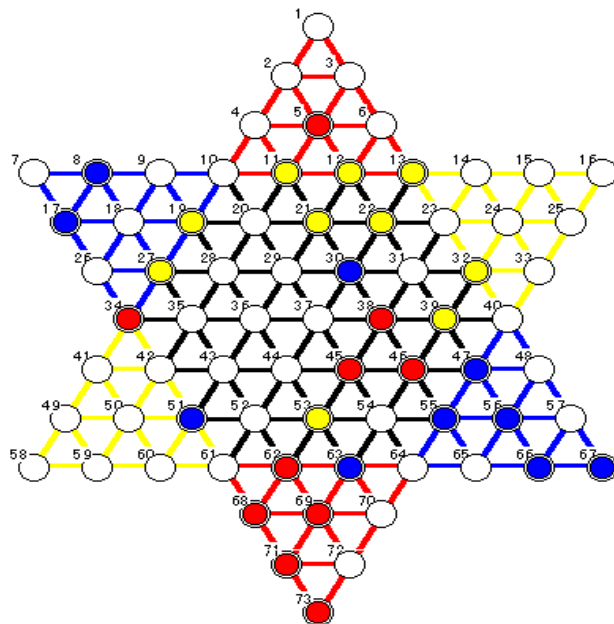
⑥中止する時の手順の踏み方をしっかりと作成した。



## 5 戦略(4つの基本戦略プログラム)

### 戦略1 ランダムに駒を選ぶ

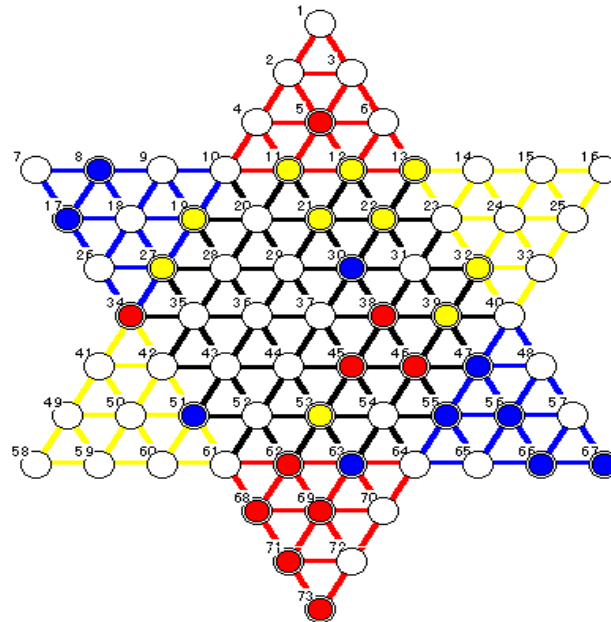
- Step 1 自駒の位置をすべて把握する
- Step 2 自駒の中から動かす駒をランダムに選択する
- Step 3 選択された駒の移動範囲を計算する
- 移動範囲内にゴールへ近づく位置が有るとき Step 4-1
- 移動範囲内にゴールへ近づく位置が無いとき Step 4-2
- Step 4-1 移動範囲内で最もゴールに近い位置へ駒を移動する
- End
- Step 4-2 選択された駒以外の自駒の中から新たに動かす駒をランダムに選択する
- Step 3へ



## 戦略2 貪欲に進める

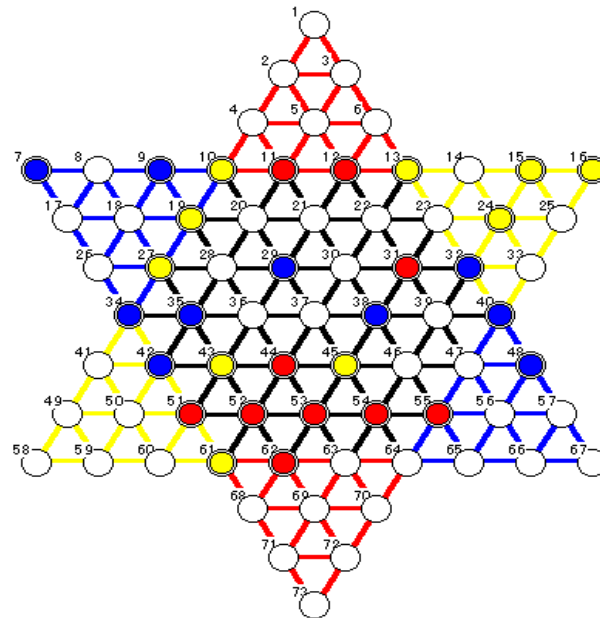
- Step 1 自駒の位置をすべて把握する
- Step 2 自駒すべての移動範囲を計算する
- Step 3 自駒すべてについて、移動範囲内で最もゴールに近い位置と元の位置との差を計算する
- Step 4 差が最も大きくなる駒を移動範囲内で最もゴールに近い位置まで移動する  
(差が等しくなるときはゴールまで遠い駒を選択する)

End



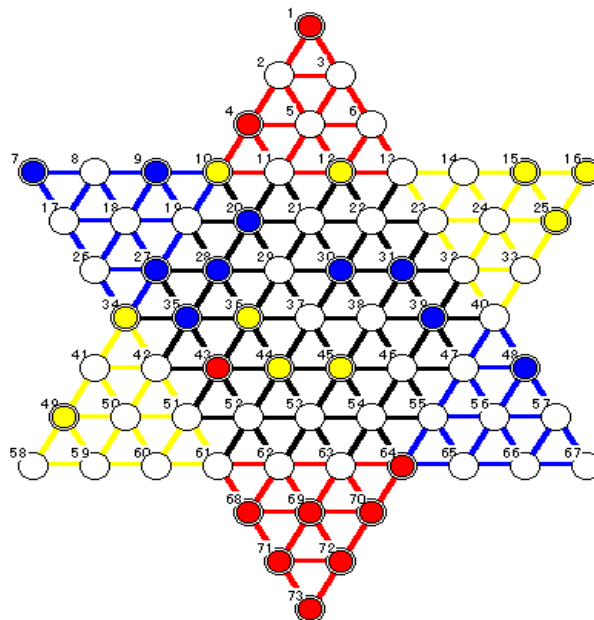
### 戦略3 後ろから進める

- Step 1 自駒の位置をすべて把握する
- Step 2 自駒の中で最もゴールまで遠い位置の駒を選択する
- Step 3 選択された駒の移動範囲を計算する
  - 移動範囲内にゴールへ近づく位置が有るとき Step 4-1
  - 移動範囲内にゴールへ近づく位置が無いとき Step 4-2
- Step 4-1 選択された駒を移動範囲内で最もゴールに近い位置まで移動する **End**
- Step 4-2 選択された駒の次にゴールまで遠い位置の駒を新たに選択する Step 3へ



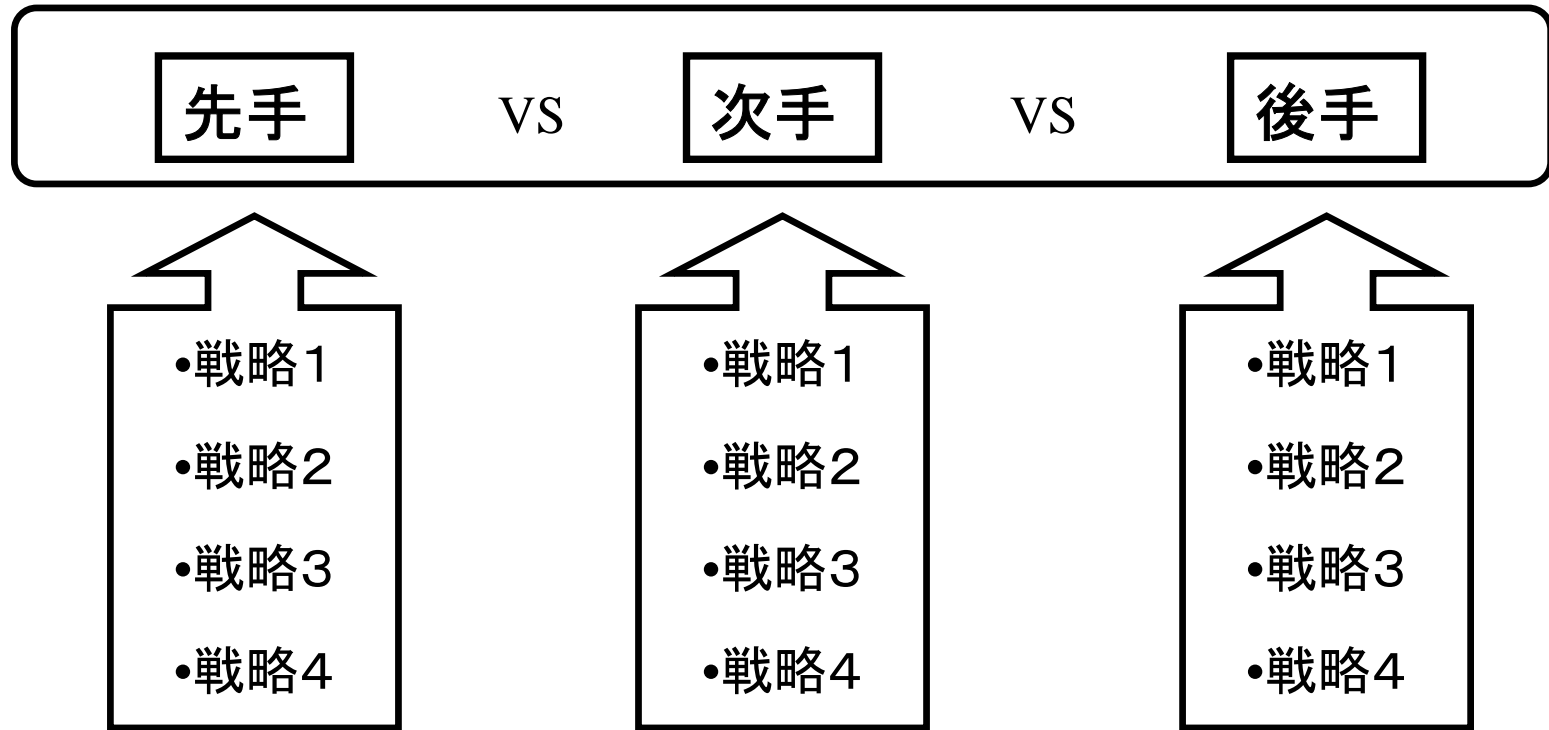
## 戦略4 前から進める

- Step 1 自駒の位置をすべて把握する
- Step 2 自駒の中で最もゴールまで近い位置の駒を選択する
- Step 3 選択された駒の移動範囲を計算する
- 移動範囲内にゴールへ近づく位置が有るとき Step 4-1
- 移動範囲内にゴールへ近づく位置が無いとき Step 4-2
- Step 4-1 選択された駒を移動範囲内で最もゴールに近い位置まで移動する End
- Step 4-2 選択された駒の次にゴールまで近い位置の駒を新たに選択する Step 3へ



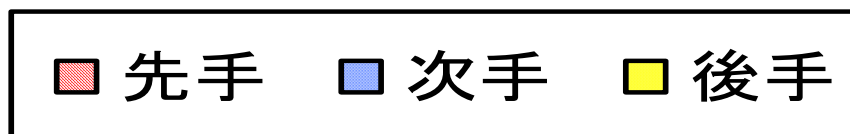
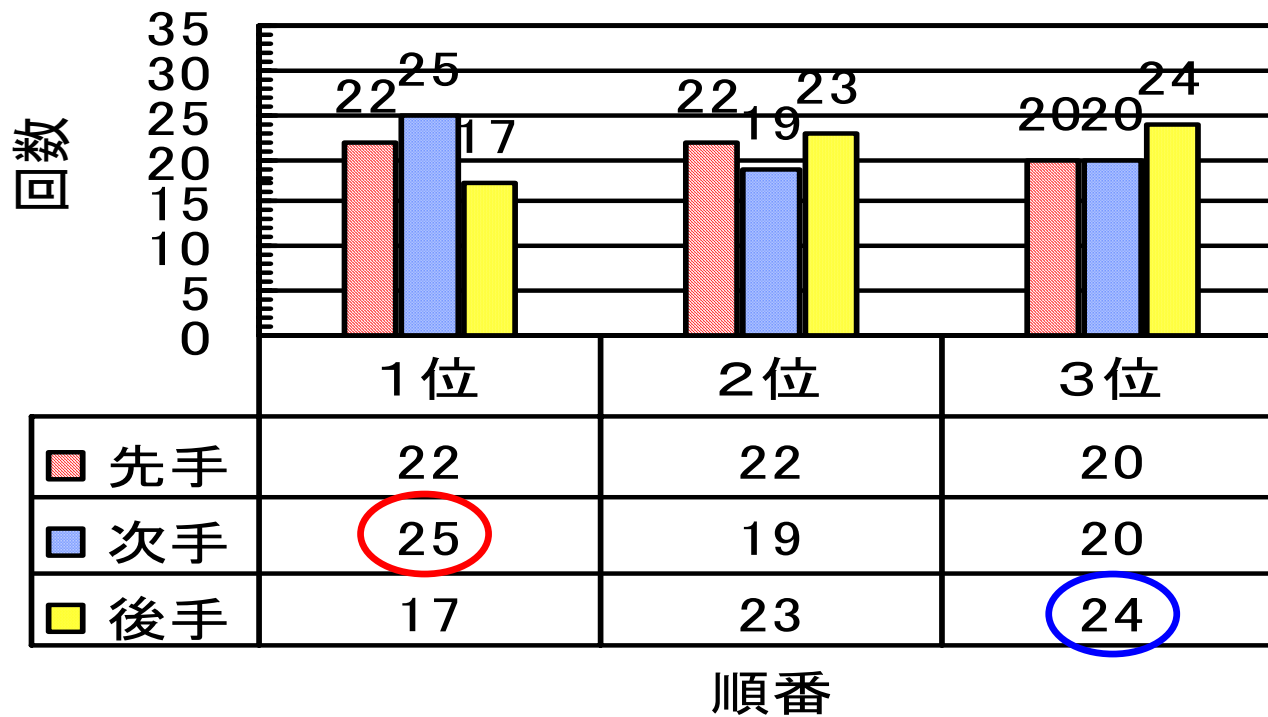
## 6 対戦実験による考察

### 6, 1 対戦実験



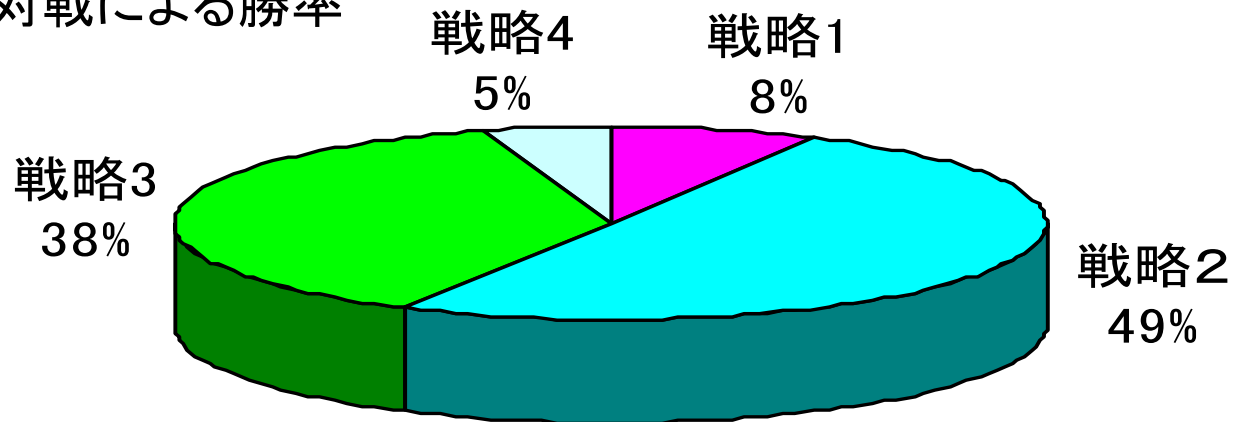
順番が3通り、戦略が4種類なので  
64(4の3乗)通りの対戦が考えられる

## 6, 2 順番に関する考察

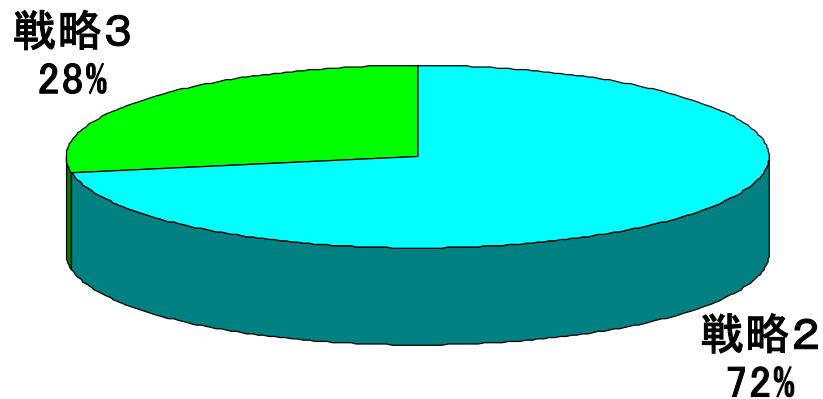


## 6, 3 戦略に関する考察

4つの戦略の対戦による勝率



戦略2、戦略3の両方を含む対戦  
についての戦略2、戦略3の勝率





## 6, 4 総合的考察

各戦略における順番についての勝利数をみると、

戦略1では  $\{ 2 \cdot 2 \cdot 1 \}$

戦略2では  $\{ 1 \cdot 2 \cdot 1 \cdot 1 \cdot 9 \}$

戦略3では  $\{ 7 \cdot 1 \cdot 0 \cdot 7 \}$

戦略4では  $\{ 1 \cdot 2 \cdot 0 \}$

{先手の勝利数・次手の勝利数・後手の勝利数}

## 7 総括的結論

### 研究目標

- ①本研究を通じて並列計算の為の基礎的な通信技術を確立
- ②参加人数(0~3人)にフレキシビリティを持たせた  
ダイヤモンドゲームの設計と実装
- ③ダイヤモンドゲームの戦略を検討し、対戦実験により優劣を判断

### 結論

- ①Delphiのコンポーネントの利用によりスムーズに行うことが出来た
- ②中断機能や会話の出来るチャット機能の追加により満足できるレベルに仕上がったと考えている。
- ③戦略については、自分本位の戦略だけとなってしまったので相手の動きを考慮するような順応性のある戦略があった方がいいと考えています。