



はじめての Chess

早稲田大学

たかみね

高峰

はるか

悠

目次

- ・プログラム概要 p2
- ・動作説明 p3~p7
- ・工夫した点 p8~p14
- ・このゲームの強みとは p15
- ・実装解説(追加資料)

プログラム概要

1回のプレイで初心者を卒業!?

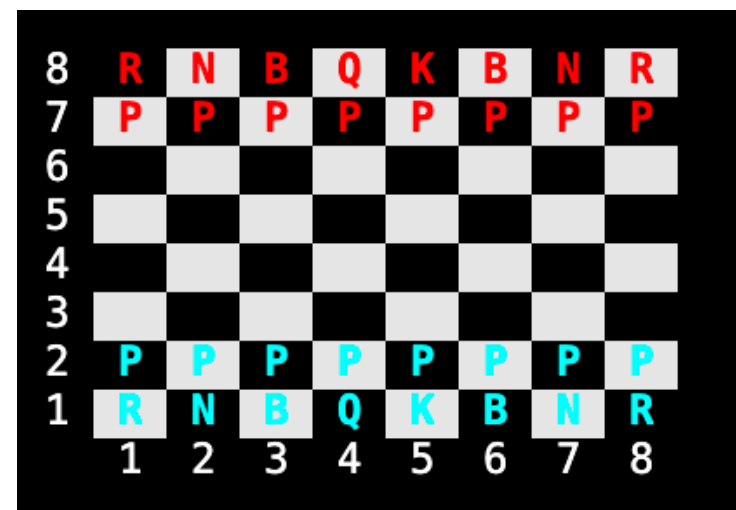
P2P型の「チェスゲーム」

—

- ・チェス初心者をターゲット
- ・プレイヤーは交互にコマを動かしゲーム進行
- ・ターミナル上にカラーで盤面を表示する

—

- ・ ChessServer.javaとChessClient.javaの2つのコードのみ使用



動作説明 0

基本動作の流れ

1.起動

2.Clientのターン

-駒の選択

-移動先の座標

3.Serverのターン

-駒の選択

-移動先の座標

2.3.を繰り返しゲーム進行

通常のチェスではKingの動ける範囲が無くなったらゲーム終了

※本実験ではKingをプレイヤーが取った場合にゲーム終了

動作説明 1

1.起動

ChessServerを先に、次にChessClientを起動

```
haruka@takamineharukaMacBook-Pro ソフト課題 % java ChessServer
- Original Chess Game - / 1 ▶ start
1
- game start - wait for answer

 8 R N B Q K B N R
 7 P P P P P P P P
 6
 5
 4
 3
 2 P P P P P P P P
 1 R N B Q K B N R
 1 2 3 4 5 6 7 8

- 相手のターン - wait for answer
```

Server側

お互いに1を入力する事でゲーム開始。
先行はClient側で固定

駒の色合い



赤色がclient側



シアンがserver側

動作説明 2

チェス初心者向けとして、
選択した駒の基本動作を説明

2.Clientのターン(駒の選択)

```
haruka@takamineharukaMacBook-Pro ソフト課題(client) % java ChessClient
```

```
- Original Chess Game - / 1 ▶ start
```

```
1
```

```
- game start - wait for answer
```

8	R	N	B	K	Q	B	N	R
7	P	P	P	P	P	P	P	P
6								
5								
4								
3								
2	P	P	P	P	P	P	P	P
1	R	N	B	K	Q	B	N	R
	1	2	3	4	5	6	7	8

```
- 自分のターン - [ 座標を入力 ]
```

```
2 2
```

- ・ pawn 基本的に前へ1マス進む。敵を倒すのは斜め前方への移動のみ。
※それぞれのPは、初期位置からの移動のみ2マス動ける。
また2マス移動の直後1ターンは1マス後方に残像を残し、
この残像が攻撃されると本体も消えてしまう。
※Pが最奥へ到達するとR,N,B,Qのうち好きなコマへ昇格できる。

移動させる駒は(縦軸,縦軸)で選択。
自分の駒以外を選択すると聞き返す。

Client側

動作説明 3

2.Clientのターン(移動先の座標)

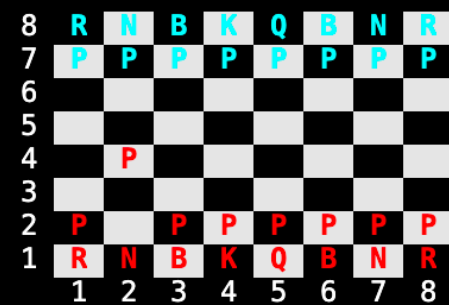
- 自分のターン - [座標を入力]

2 2

- ・ pawn 基本的に前へ1マス進む。敵を倒すのは斜め前方への移動のみ。
※それぞれのPは、初期位置からの移動のみ2マス動ける。
また2マス移動の直後1ターンは1マス後方に残像を残し、
この残像が攻撃されると本体も消えてしまう。
※Pが最奥へ到達するとR,N,B,Qのうち好きなコマへ昇格できる。

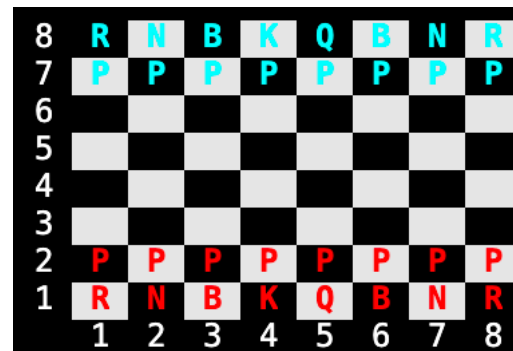
- 自分のターン - P [移動座標を入力]

2 4



- 相手のターン - wait for answer

移動させる座標も(縦軸,縦軸)で選択。
移動先に自分の駒がある時は聞き返す。



Client側

元の盤面

動作説明 4

3.Serverのターン(駒の選択)

Client側の駒の動きがServerに！

Server側

haruka@takamineharukaMacBook-Pro ソフト課題 % java ChessServer

```
- Original Chess Game - / 1 ▶ start
1
- game start - wait for answer

 8 R N B Q K B N R
 7 P P P P P P P P
 6
 5
 4
 3
 2 P P P P P P P P
 1 R N B Q K B N R
   1 2 3 4 5 6 7 8

- 相手のターン - wait for answer

 8 R N B Q K B N R
 7 P P P P P P P P
 6
 5
 4
 3
 2 P P P P P P P P
 1 R N B Q K B N R
   1 2 3 4 5 6 7 8

- 相手のターン - Pを動かしてきた
- 自分のターン - [ 座標を入力 ]
```


工夫した点 1

- ・ターミナル上でのカラフルさ

目的:プレイヤーの視認性を改善

エスケープシーケンスを使いターミナルの標準出力に色付け

→入力が必要なところのみ明るい色をつけて通知

- 相手のターン - wait for answer

白背景:入力をせずに待つ。

- 自分のターン - [座標を入力]

青背景:入力が必要！

実装はとても簡単→実装説明

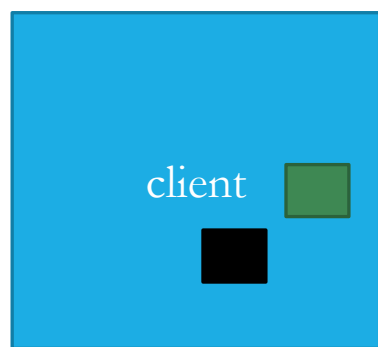
工夫した点 2

・通信に関する工夫

レスポンスを速くするため通信回数を減らす

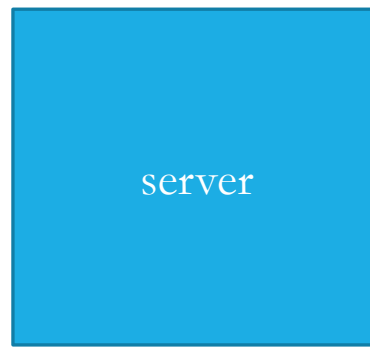
1. 移動する駒
2. 移動後の座標

この2つの整数情報を送受信しあうだけで、
ゲームを進めるようにした



それぞれが盤面を格納

→
ここを変更しました



←
OK

※プレイヤーは対面する。
よって、お互いの盤面は上下反転。

工夫した点 3

初学者には難しいので、
熟練者向け

・チェスの特殊ルール

チェスは1ターンに駒を2回動かすような特殊ルールがある.....

キャスリング 自分のkingを動かし、横にrookを付ける (簡単に説明)

→kingを動かし、rookも動かす。In 1ターン

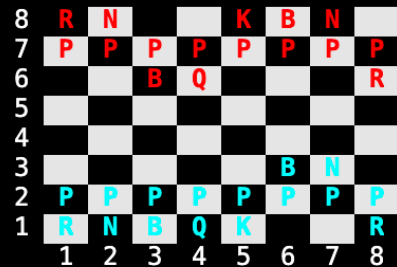
さらに、pawnの分身が残る特殊ルール.....

アンパッサン 2マス動いたpawnの後ろに自分を道連れにする分身が生じる

他にプロモーションがあるが今回は省略

特殊ルール・キャスリング (Castling)

初学者には難しいので、
熟練者向け



- 相手のターン - Bを動かしてきた

- 自分のターン - [座標を入力]

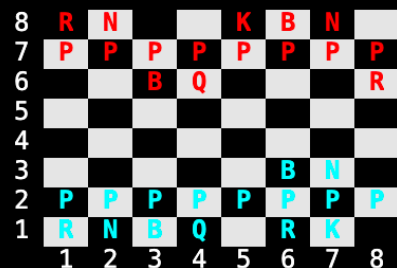
5 1

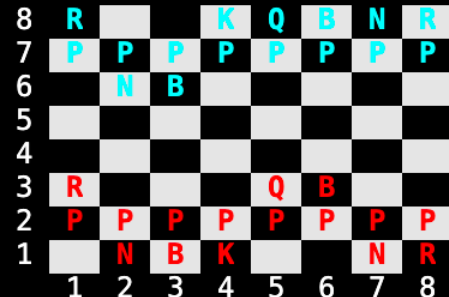
・ king 左右、上下、斜め方向に1マス移動できる。

- 自分のターン - K [移動座標を入力]

7 1

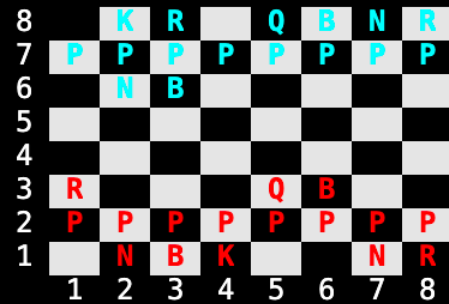
- 自分のターン - <- Castling ->





- 相手のターン - wait for answer

- 相手のターン - <- Castling ->



- 相手のターン - Kを動かしてきた

- 自分のターン - [座標を入力]

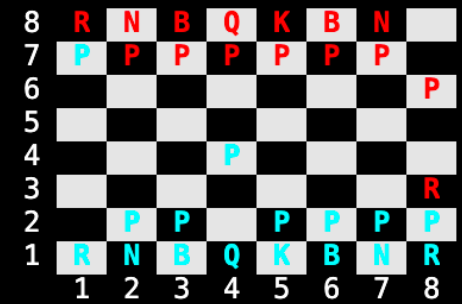
Serverがキャスリングした例



2つの駒が1手で移動

特殊ルール・アンパッサン (en passant)

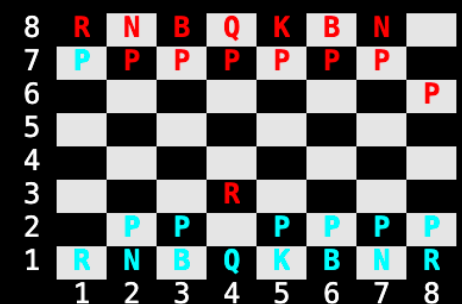
初学者には難しいので、
熟練者向け



8 R N B Q K B N
7 P P P P P P P
6
5
4
3
2 P P P P P P P
1 R N B Q K B N R
1 2 3 4 5 6 7 8

- 相手のターン - wait for answer

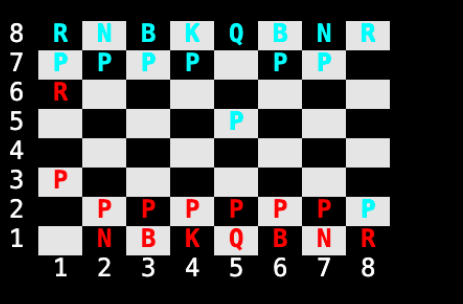
- 相手のターン - <- en passant ->



8 R N B Q K B N
7 P P P P P P P
6
5
4
3
2 P P P P P P P
1 R N B Q K B N R
1 2 3 4 5 6 7 8

- 相手のターン - Rを動かしてきた

- 自分のターン - [座標を入力]



8 R N B K Q B N R
7 P P P P P P P
6 R
5
4
3 P
2 P P P P P P P
1 N B K Q B N R
1 2 3 4 5 6 7 8

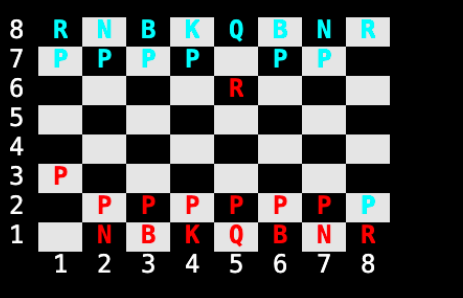
- 相手のターン - Pを動かしてきた

- 自分のターン - [座標を入力]
1 6

・ rook 左右、上下好きな距離移動できる。(将棋における飛車)

- 自分のターン - R [移動座標を入力]
5 6

- 自分のターン - <- en passant ->



8 R N B K Q B N R
7 P P P P P P P
6
5
4
3 P
2 P P P P P P P
1 N B K Q B N R
1 2 3 4 5 6 7 8

Clientがアンパッサンした例



2マス動いたPが、1ターン以内で
後ろの分身をとられ消滅

工夫した点 4

・チェスのルールを覚えたい！

(制作者はchess初心者)

よくあるChessゲームなどでは駒はルールで決められた範囲にしか移動できない。(範囲外のマス選択ができない)

→プレイヤーはルールを知らなくても適当に指せる。

本ゲーム

ここで、現実と同じく、ルールを無視すれば盤面のどこにでも駒を移動できるゲームを制作。

→プレイヤーはルールを知ろうとしないといけない。

プレイヤーが駒を選択すると動きの説明が表示される仕様に

工夫した点 4(example)

プレイヤーが駒を選択すると動きの説明が表示される仕様に

- 自分のターン - [座標を入力]

1 1

・ rook 左右、上下好きな距離移動できる。(将棋における飛車)

- 自分のターン - [座標を入力]

3 1

・ bishop 斜めで好きな距離移動できる。(将棋における角)

- 自分のターン - [座標を入力]

5 2

・ pawn 基本的に前へ1マス進む。敵を倒すのは斜め前方への移動のみ。
※それぞれのPは、初期位置からの移動のみ2マス動ける。
また2マス移動の直後1ターンは1マス後方に残像を残し、
この残像が攻撃されると本体も消えてしまう。
※Pが最奥へ到達するとR,N,B,Qのうち好きなコマへ昇格できる。

(制作者は将棋はある程度知っているなので、将棋で喩えている)

このゲームの強みまとめ

初心者向け！爽快chessゲーム

- ①本当のchess初心者でも何も調べたりせずに、理解しながらゲームを進められる！
- ②盤面がターミナルに残っているので、以前の盤面を簡単に見直すことが可能！
- ③プレイ途中で飽きた場合、ルールを無視して敵のkingを取りゲームを終了できる！ You win！

実装解説

1. 盤面配列
2. エスケープシーケンス
3. 盤面のカラフル表示
4. アンパッサンの実装

実装解説 1.盤面配列

Char Chess[64][2] で盤面を保存

chess[i][0]='R'のように駒の頭文字を格納している。

chess[i][1]='e'のように敵の駒は 'e',自分の駒は 'm'として格納。

空マスは{'0','0'}のようにしている。

```
static char[][] chess = {
    {'R','e'},{'N','e'},{'B','e'},{'Q','e'},{'K','e'},{'B','e'},{'N','e'},{'R','e'},
    {'P','e'},{'P','e'},{'P','e'},{'P','e'},{'P','e'},{'P','e'},{'P','e'},{'P','e'},
    {'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},
    {'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},
    {'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},
    {'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},{'0','0'},
    {'P','m'},{'P','m'},{'P','m'},{'P','m'},{'P','m'},{'P','m'},{'P','m'},{'P','m'},
    {'R','m'},{'N','m'},{'B','m'},{'Q','m'},{'K','m'},{'B','m'},{'N','m'},{'R','m'}
};
```

実装解説 2.エスケープシーケンス

ESCの準備(何度も使うので宣言しておいた方が楽)

```
12 static String bluebackwhite = "\u001b[00;46;37m";  
  
20 static String end = "\u001b[00m";  
21 static String endback = "\u001b[49m";
```

00で現在のesc初期化
46で背景を青に
37で文字を白に

実際に標準出力

```
System.out.println("\n"+bluebackwhite+"          - Original Chess Game -          / 1 ▶ start "+end+endback);
```

出力結果

Endだけでも良い

```
          - Original Chess Game -          / 1 ▶ start  
1
```

※ESCで出力様式を変更させた後、初期化を忘れずに

実装解説 3.盤面のカラーフル表示

コマ判別

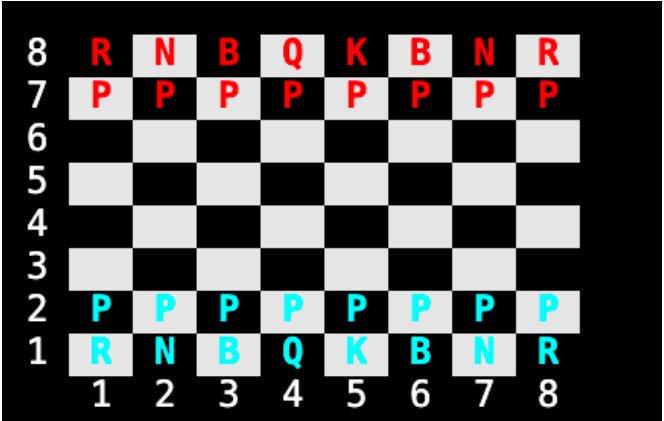
```
if(chess[i][1]=='m')System.out.print(blue);//敵のコマ  
else if (chess[i][1]=='e')System.out.print(red);//自分のコマ
```

敵のコマの時は文字色を青に
自分のコマの時は文字色を赤に

白黒判別

Int iなどを取っておきiを1マスずつ反転させていき、iに対応する背景をESCで表示

```
if(j==1)System.out.print(whiteback);  
else System.out.print(blackback);  
j=-1*j;
```



実装 4. アンパッサンの実装

基本のマス配列は $\{P, m\}$

自分のPの2マス移動時のみ、 $\{0, 1\}$ をPの後ろのマスに格納
(盤面表示は '0'を確認したら空マスで出力するのでOK)

・1ターンのみ生存

While
 自分のターン ←ここで $\{0, 1\}$ を盤面で発見したら $\{0, 0\}$ に戻す。
 相手のターン

・消滅

敵のコマが $\{0, 1\}$ 移動されたらPを見つけ消す