

プログラミング演習 最終課題について

どういう課題？

- 授業で学んだ知識を活かして，ゲームのAIを作ろう
- 規模の大きいプログラムの作成に慣れよう
- 他人が作ったプログラムと組み合わせて動く
プログラムの作成に慣れよう

ゲームについて

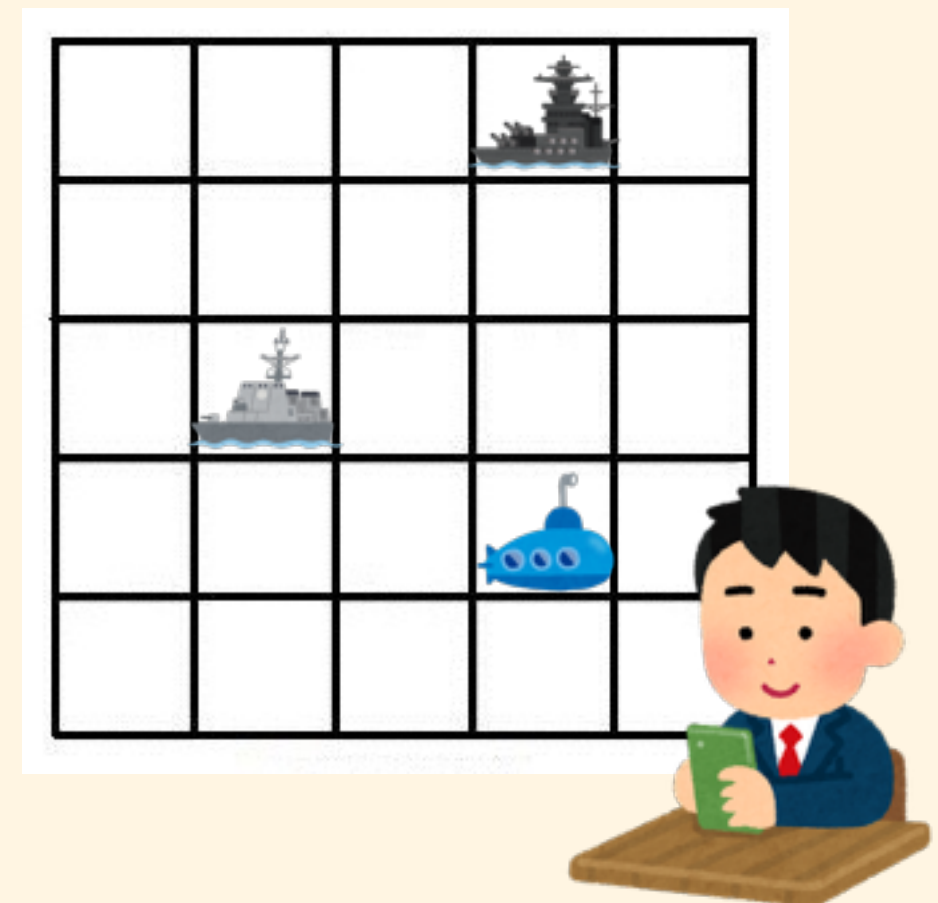
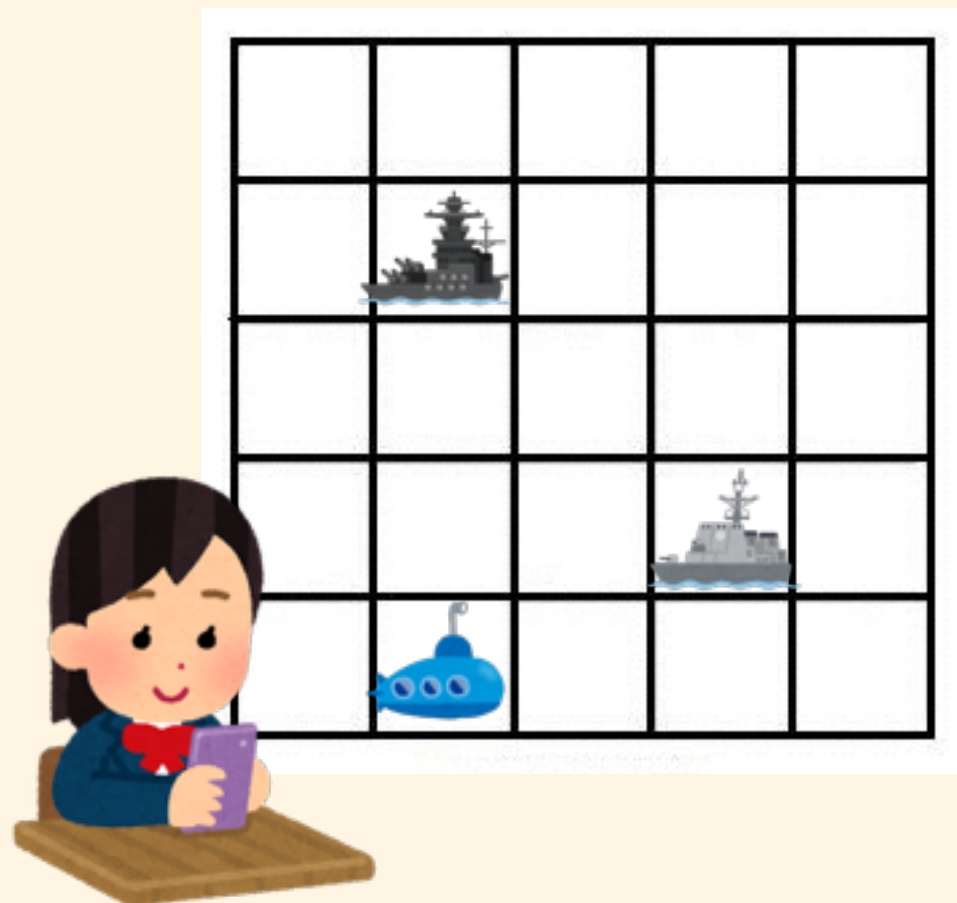
潜水艦ゲーム

- プレイヤーは2人の対戦ゲーム



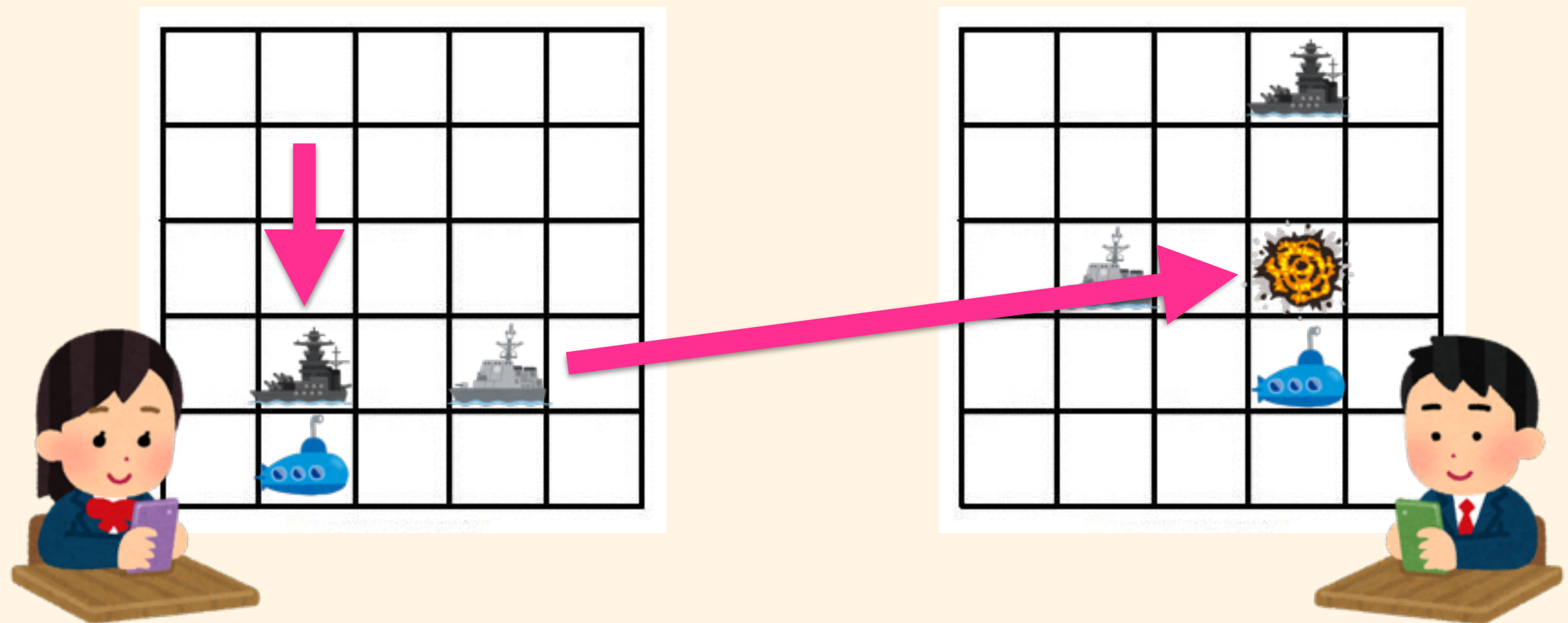
潜水艦ゲーム

- 5x5のマップに, 戦艦, 駆逐艦, 潜水艦の3種類の軍艦を1隻ずつ配置する.



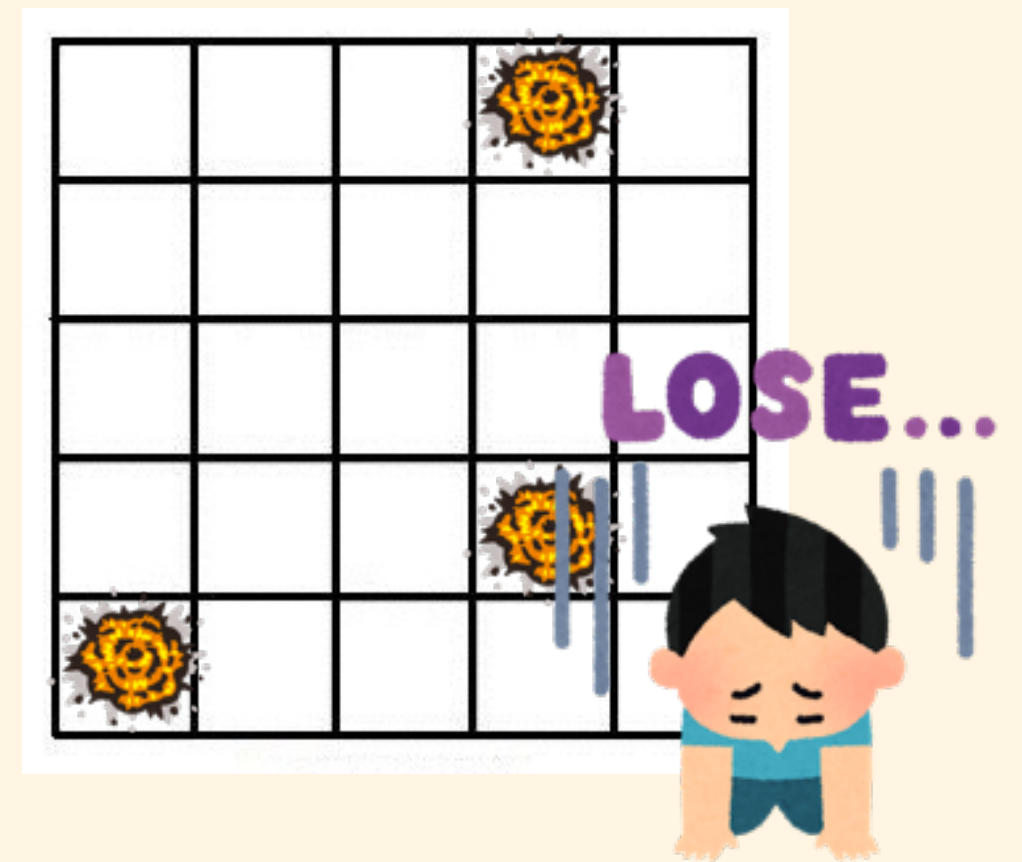
潜水艦ゲーム

- プレイヤーが交互に行動 { 魚雷攻撃, 艦の移動 } を選択



潜水艦ゲーム

- 敵艦を全て撃沈したプレイヤーが勝利



ルール

- 攻撃
 - 艦を中心とする周囲9マス

攻撃可能な範囲の例
(赤色が攻撃可能な範囲)

	A	B	C	D	E
1					
2		C	W		
3					
4					
5					S

ルール

- 移動
 - 東西南北の方向に1マス以上
 - 自分の艦と位置がかぶらない

移動可能な範囲の例
(青色が戦艦(W)が移動可能な範囲)

	A	B	C	D	E
1					
2		C	W		
3					
4					
5					S

ルール

- お互いに、相手の艦の配置はわからない。
- 相手の位置を知るヒントとなる情報
 - 魚雷を食らった or 近くに当たった場合に、**攻撃マス**の近くにいる**艦種**が公開される（「水しぶき」）
 - 例) [3,4]を攻撃 → (戦艦に命中, 潜水艦が近くにいる)
- 艦を移動させたときは、**艦種**、及び、**移動した距離と方角**が公開される

遊んでみる

プログラムについて

プログラムについて

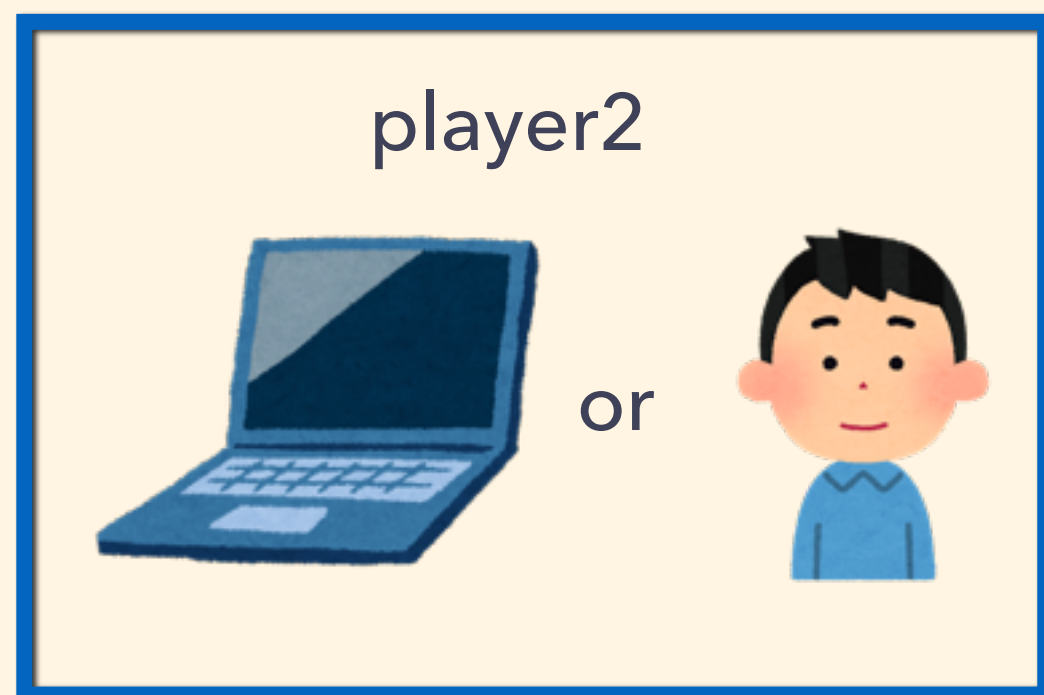
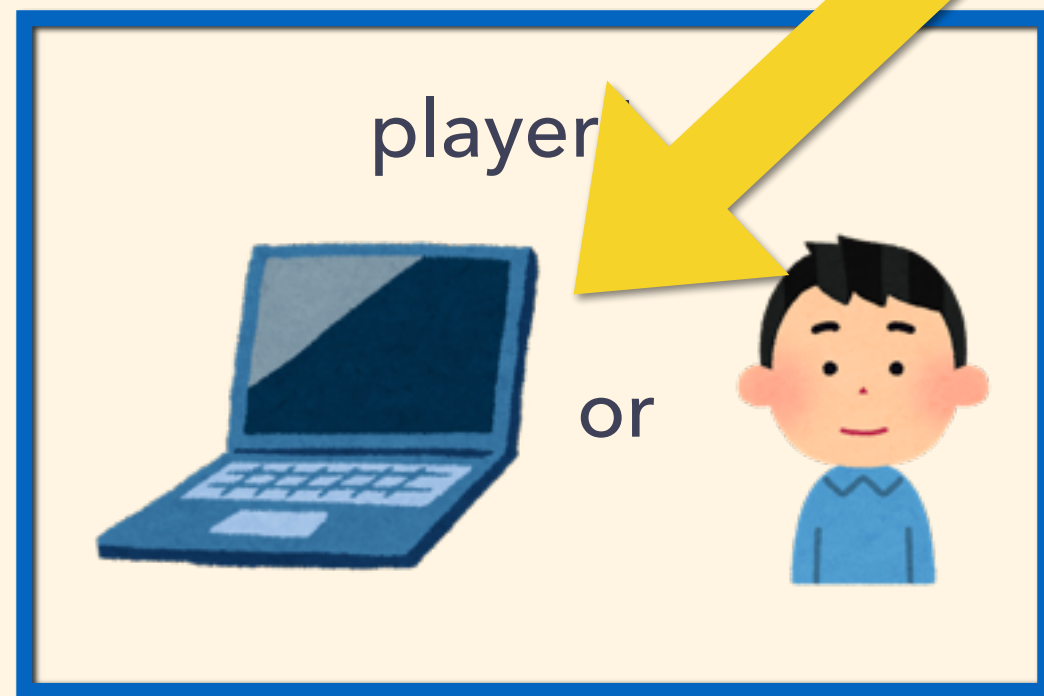
- 一昨年のプロ演の課題で現M1の3人チームで作成された
 - サーバー担当1人, AI担当2人
- 元の実装はRubyで, 今年度からAI作成部をPython化

プログラムについて

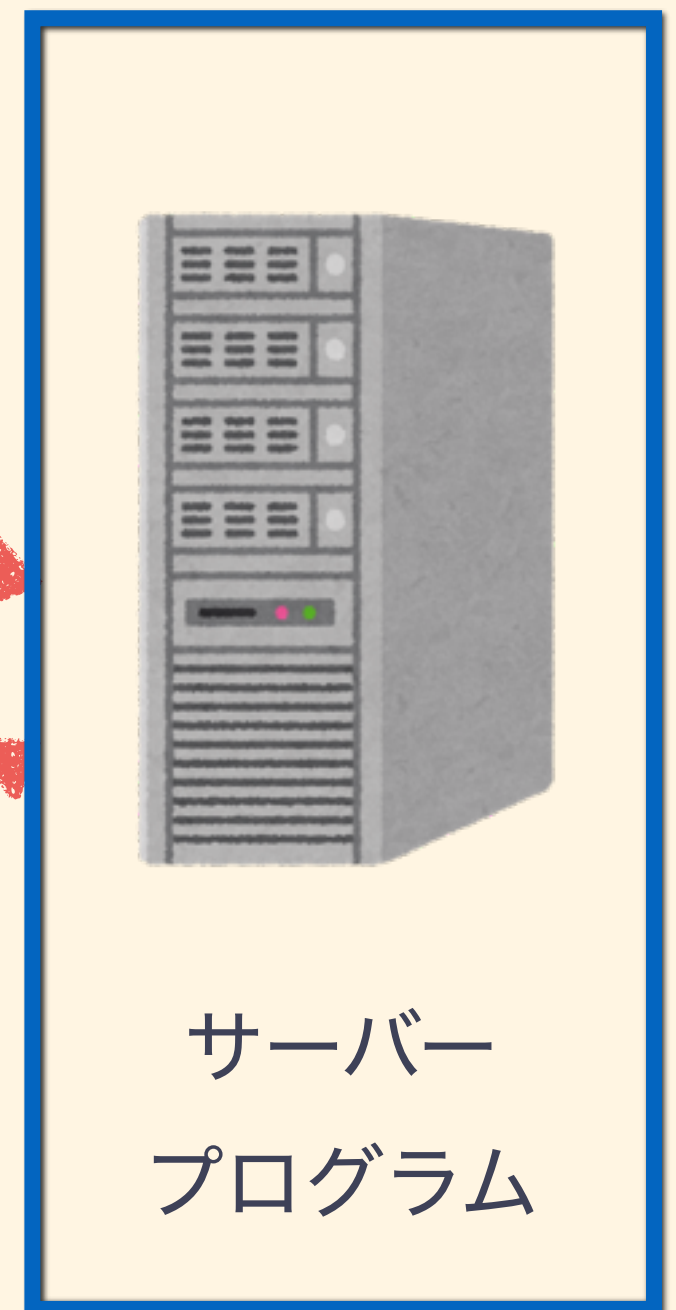
- 課題に取り組むにあたっては
 - 特に8章～10章あたりを思い出す
 - (optional) 付録Aで紹介されているgitの使い方を理解する
- をしておけばOK

実装

ここを作る



jsonデータを
やりとり



クライアント側

- 自分で実装する必要があるのは,
「自分の行動を選択する関数を記述する」 こと
- サンプルコードを改造すると良いです.
- 通信周りのめんどくさいことはライブラリが吸収してくれます
- がんばって強くしよう

クライアント側

目標

1. 合法手を返す
 - サンプルコードはちゃんと合法手を返すようになっています
2. 自分で考えた戦略に基づいて手を返す
3. 強くする

アドバイス

- 多少の工夫でランダムプレイヤーよりは強くなれます
- AIを強くする
 - 強くしたAIと自分(人間)で対戦してみる
 - 何が弱点になっているかを考えてAIを強くする
 - 強くしたAIと自分(人間)で対戦してみる
- ...
- 友達のAIプレイヤーと戦わせてみると楽しい
- プログラムの規模が大きくなってくると,
 - (この課題程度でも) どこがバグの原因になっているかわからなくなってしまうので, こまめにunittestを書く

サーバー側

- 見た目がCUIで寂しいので, 見た目をかっこよくする
- 人間プレイがターミナルを通じてしかできないので, GUI
の人間用クライアントを作る
- ルールを独自拡張してみる
- マップを面白い形状にしてみる
 - クライアント側の対応も必要なことに注意

アドバイス

- jsonでデータをやり取りできればなんでもいいので, 自分の得意な言語/フレームワークなどを使えます