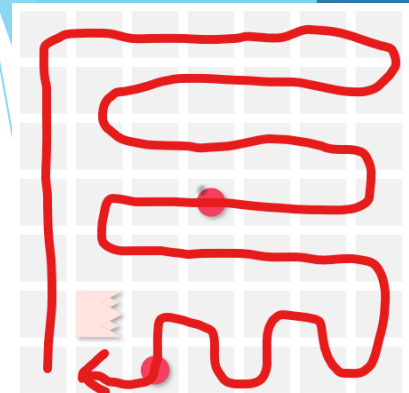


# 基礎工学PBL Solo発表 第3班

浅海雄士  
川口晴大朗  
佐藤慶季  
畑大輝  
山久保孝亮

# プロダクト

## 蛇の戦略



### 最初に実装を行った戦略

盤面全体を通るルートを予め定めておき、その経路を常に通過し続ける。

- ・ 自分自身にぶつかる処理を記述する必要がなく、容易に実装できた。
- ・ ルール改定後、常に空白のマスのうちの3マスにfoodがあらわれることから、ほとんどすべてのマスを通る経路を通ればすぐに蛇の長さが最大に達するため、この戦略は不利になった。

# プロダクト

## 蛇の基本戦略

今回の基本戦略は以下の3つである。

1. 次のターンで安全な方向を調べ、消去法で次のターンの行動を決定する。
2. 基本的には外周を回るようにする。
3. Healthをできるだけ使い果たしてから、foodを食べるようにする。

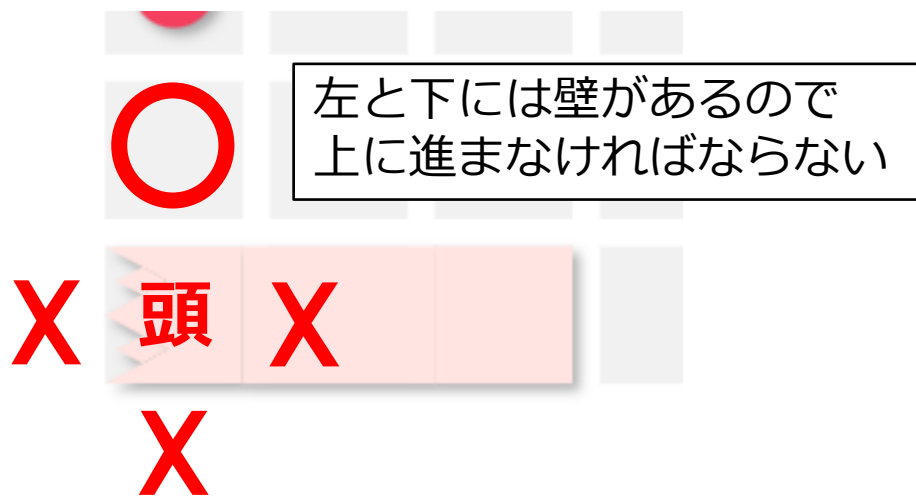
# プロダクト

## 蛇の基本戦略

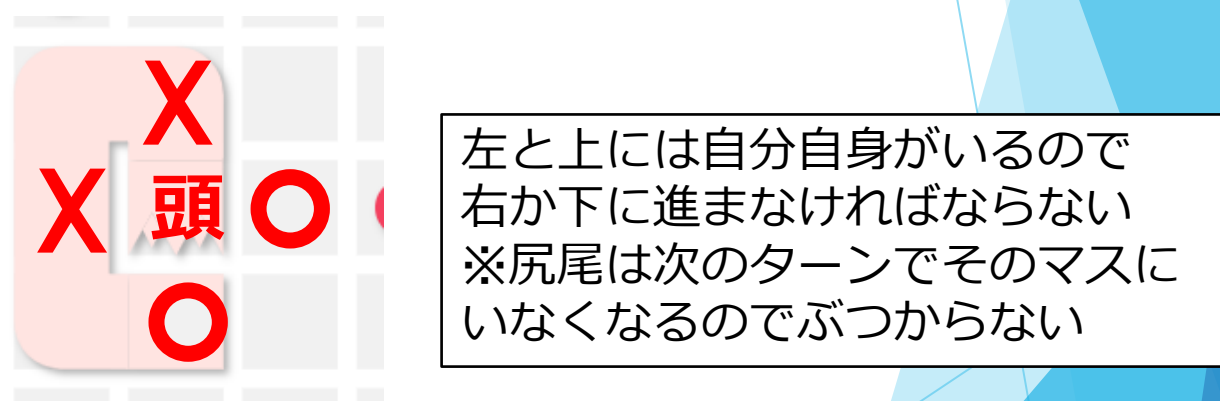
基本戦略 1. 次のターンで安全な方向を調べ、消去法で次のターンの行動を決定する。

以下の2パターンは容易に回避することができる。

1. 壁に衝突する。



2. 自分自身にぶつかる。



他のどの戦略やアルゴリズムよりもこの結果を**最優先**する。  
安全な選択肢が複数あるときは他の戦略やアルゴリズムによって絞り込む

# プロダクト

## 蛇の基本戦略

基本戦略 1. 次のターンで安全な方向を調べ、消去法で次のターンの行動を決定する。

### メリット

- ・一ターン後に力尽きることを防ぐことができる。
- ・安全な方向が一つならそれ以上処理を行わなくて良い。

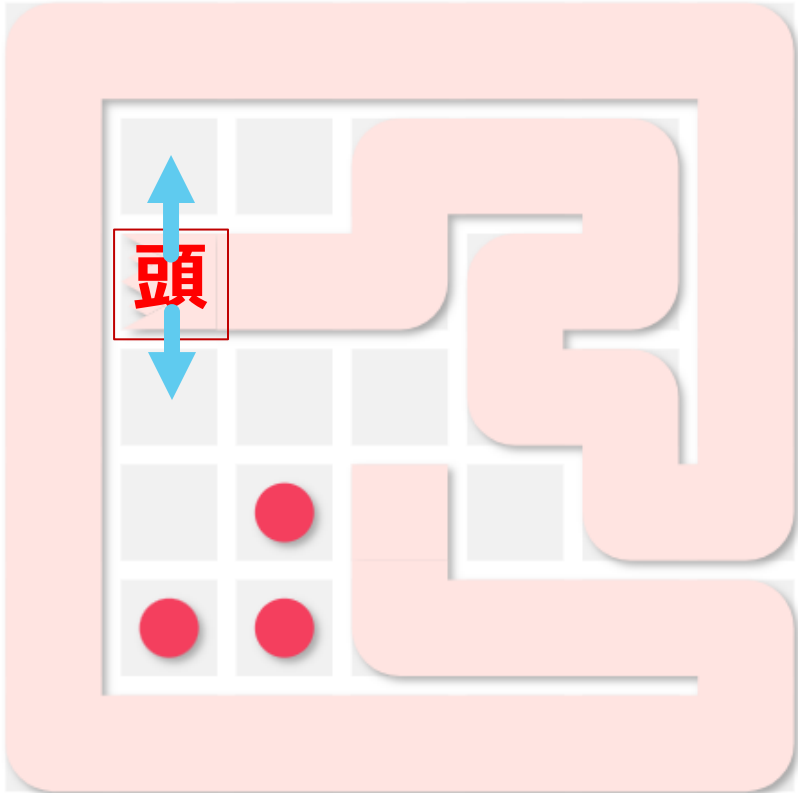
### デメリット

- ・数ターン先まで予測することができない。

プロダクト

## 発生した問題

- ・ 数ターン先まで予測することができない。



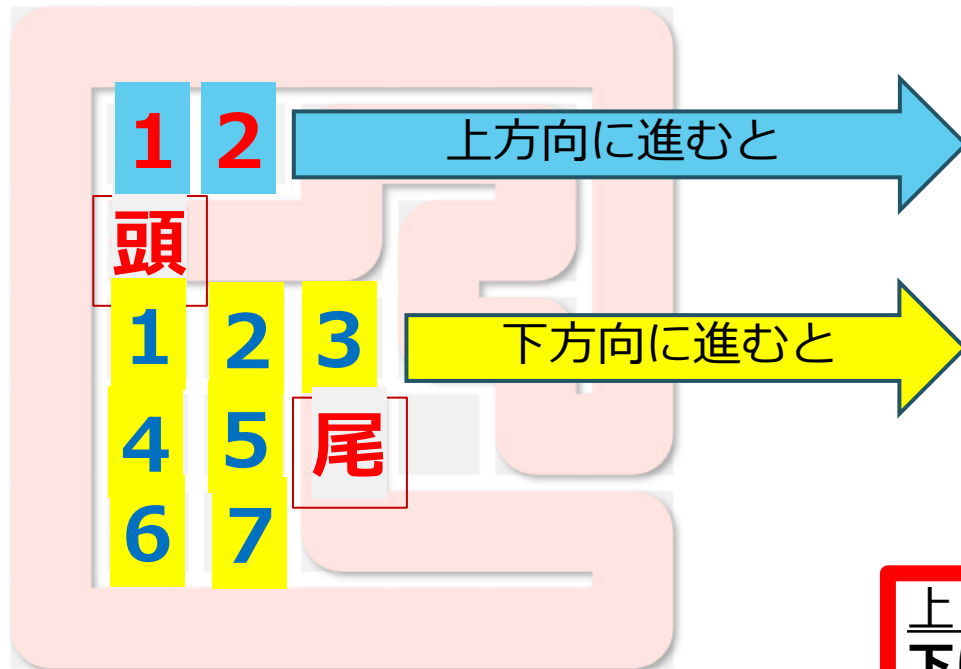
- ・ 1ターン後に安全な方向  
上と下であることは予測できる

しかし、上に進んだ場合、  
2ターン後に周りが自分自身で囲  
まれてしまい、進めるマスがなく  
なってしまう。

# プロダクト

## 解決方法

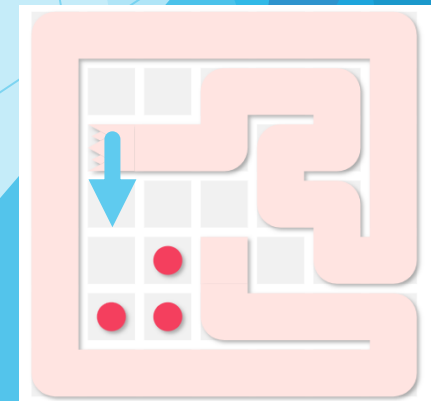
### ・ Flood fill 法を応用した順序付け



- ・ 現在 2 マスの空白のマス
  - ・ 現段階では尻尾にアクセスできない
- 囲まれてしまう可能性が高い

- ・ 現在 7 マスの空白のマス
  - ・ 尻尾にアクセスできる
- 囲まれる可能性が低い

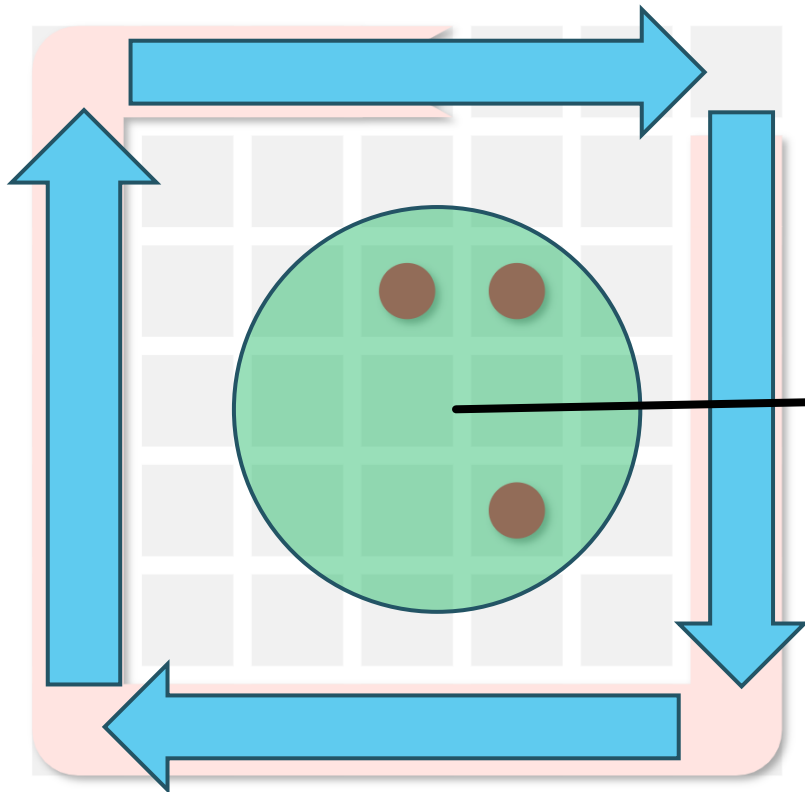
上と下なら  
**下に進む方が安全**であると考えられる



# プロダクト

## 蛇の基本戦略

### 基本戦略2. 基本的には外周を回るようにする。



- ・ 蛇がいるマス目にFoodが出現することはない

外周を回することで、  
Foodが中心に集まる確率が高くなる

Foodが中心に集まっているほど  
壁や自分自身に囲まれてしまうリスク  
が減るのではないか。



# プロダクト

## 蛇の基本戦略

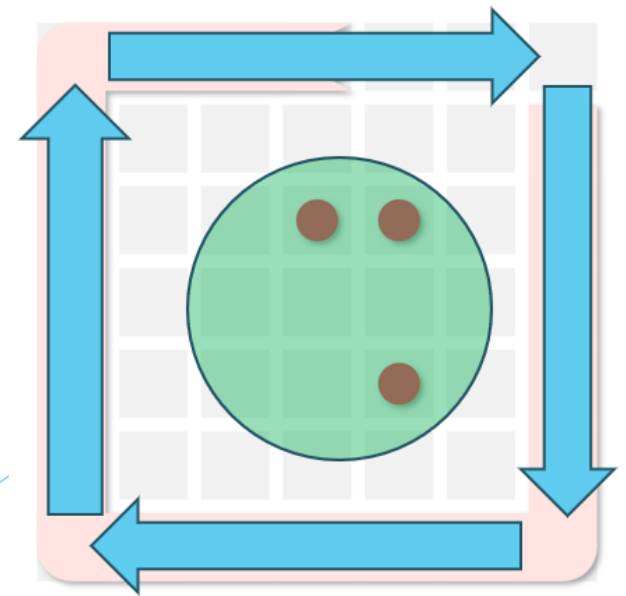
### 基本戦略 2. 基本的には外周を回るようにする。

#### メリット

- Foodが中心に集まることでFoodを取りに行きやすくなる。
- 進む向きが予め決まっており、処理時間が短く済む

#### デメリット

- 蛇の長さが長くなってくると外周を回ることができず、決められたルートから外れざるを得なくなる。

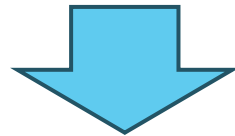


# プロダクト

## 蛇の基本戦略

基本戦略3. Healthをできるだけ使い果たしてから、Foodを食べるようにする。

- foodを食べると蛇の長さが長くなる。
- boardの大きさは7×7マスと有限であり、蛇は長いほど、自分に衝突しやすくなる。



Healthが十分あるときはFoodをできるだけ避けるようにした。

# プロダクト

## 蛇の基本戦略

基本戦略3. Healthをできるだけ使い果たしてから、foodを食べるようにする。

### メリット

- 蛇の長さがすぐには長くないので、自分自身にぶつかりにくくなる。

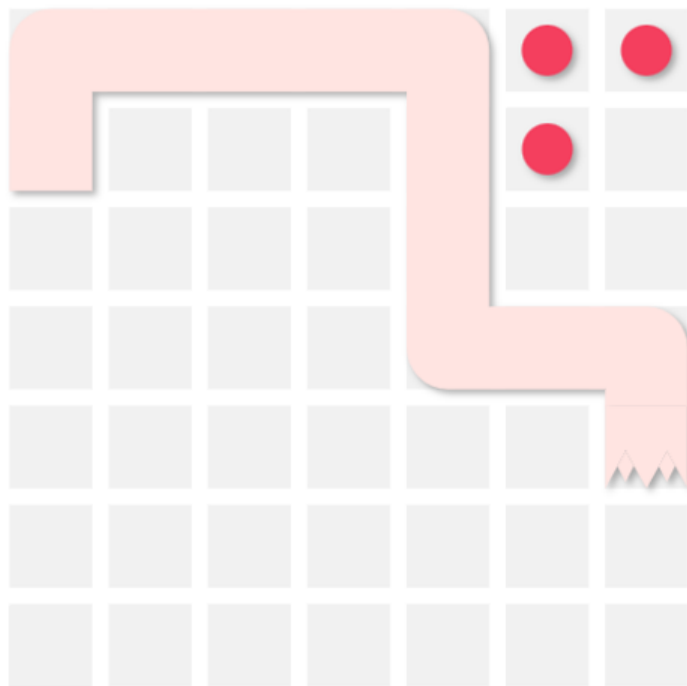
### デメリット

- foodが隅に集まってしまうと、取りに行くことが困難となる
- FoodをHealthの値がなくなる直前まで避け続けると、Health不足で力尽きやすくなる。

# プロダクト

## 発生した問題

foodが隅に集まってしまうと、取りに行くことが困難



改善案：

Healthが十分あるときは、外周を回るのではなく、  
盤面の中央あたりで自分の尻尾を追いかけるようにする。 (未実装)

# プロセス

## 作業分担——役割決め

作業	メンバー
コード作成	佐藤、畑
アルゴリズムの相談	全員
デバッグ	浅海、川口、山久保
レポート作成	川口、山久保
プレゼン作成	浅海、山久保

# プロセス

## 使用したツール

- ・ GitHub

プログラムの共有に使用

- ・ battelesnakeのローカル環境の構築

各自でプログラム作成や、デバッグを行えるようにした

- ・ CC

他のグループの蛇の動きを参考にした

- ・ PowerPoint ・ Overleaf

班員全員でのスライド・レポート作成に使用

- ・ Slack

班員内の連絡・デバッグの報告・ファイルの共有等に使用

# プロセス

## 作業分担——具体的な分担方法

- ▶ プログラミング知識に偏りがあったため、コード班の2人とプレゼン・レポート班の3人に分かれて作業を行った。
- ▶ 蛇の死亡する原因を実際に動かして探る際、時間がかかりかかることから、プレゼン・レポート班に分かれて数十回動かしてデータを収集した。
- ▶ レポートのプロダクトの節については、コード班が主に編集することにした。プレゼンの発表についても、質疑応答の質を鑑みた結果コード班が行うことにした。

# プロセス

## 作業分担——反省点と改善方法

- ▶ コード班は毎週活動のためにコードを書く必要があったが、レポート・プレゼン班はある程度コードが出来ないと作業に取り掛かりにくく、結果として二班の作業量に差が出来てしまった。
  - コード班でないメンバーもある程度コードを書く。
- ▶ コード班がコードを修正する際、デバッグがまだ終わっていないということがあった。
  - コード班とデバッグ担当で都合を合わせる。またデバッグ担当を増やす。



# プロセス

## 議論の工夫

- ▶ プログラミング知識の偏りを考慮して、授業内では主に戦略を中心として議論を行った。
- ▶ 全体でアルゴリズムについて考える際、ホワイトボードやPCで内容を洗い出し、全員が視覚的に内容を理解できるようにした。また議長が、班員全員の意見を汲み取ることで、議論をより洗練されたものにした。