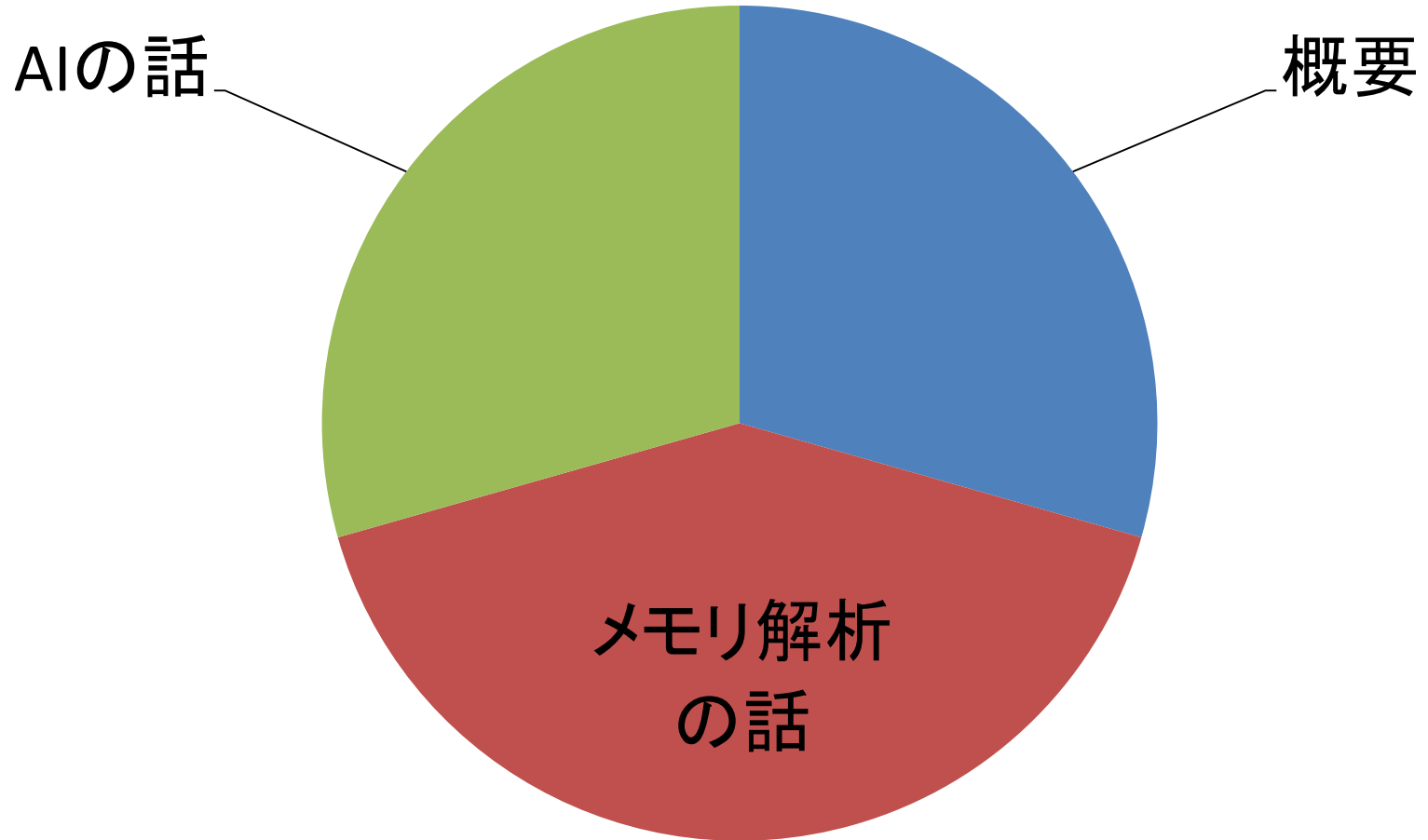


# 東方地霊殿の自動プレイ プログラムの作成

計算工学専攻 M1

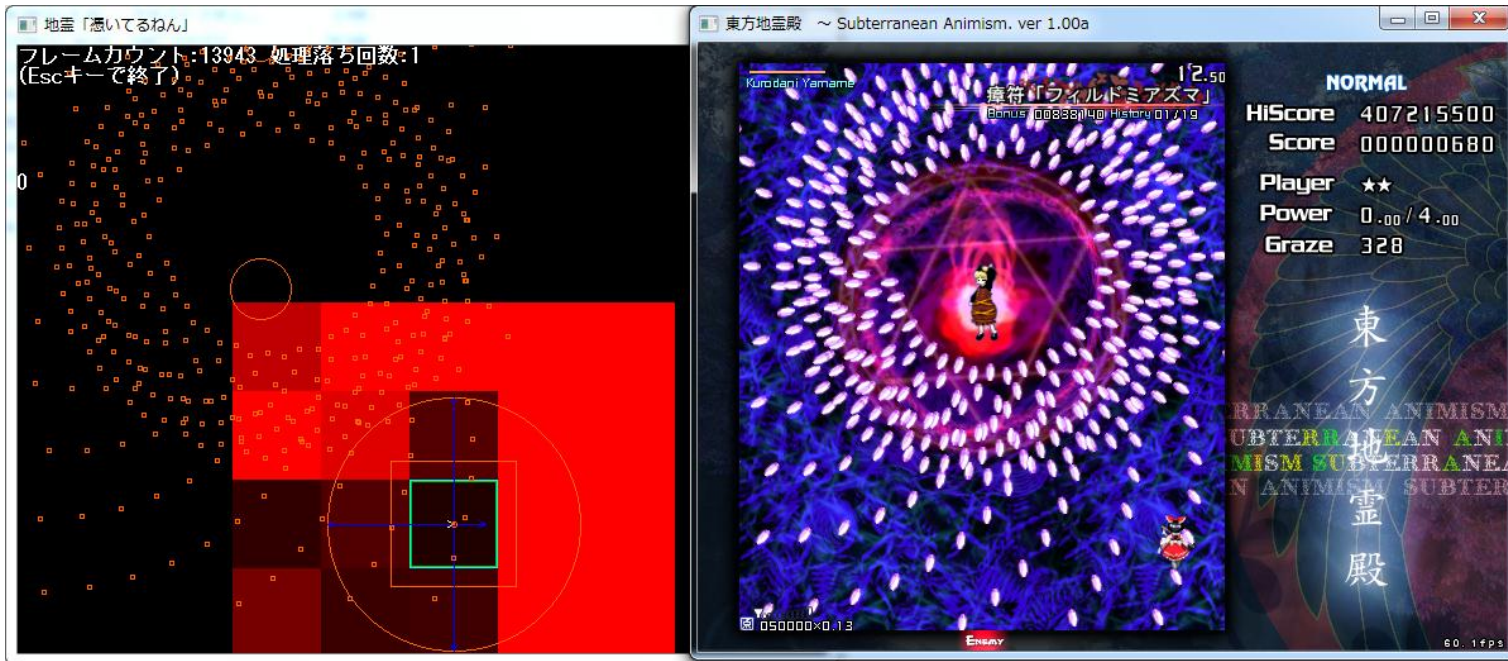
@ide\_an

# 話の流れ



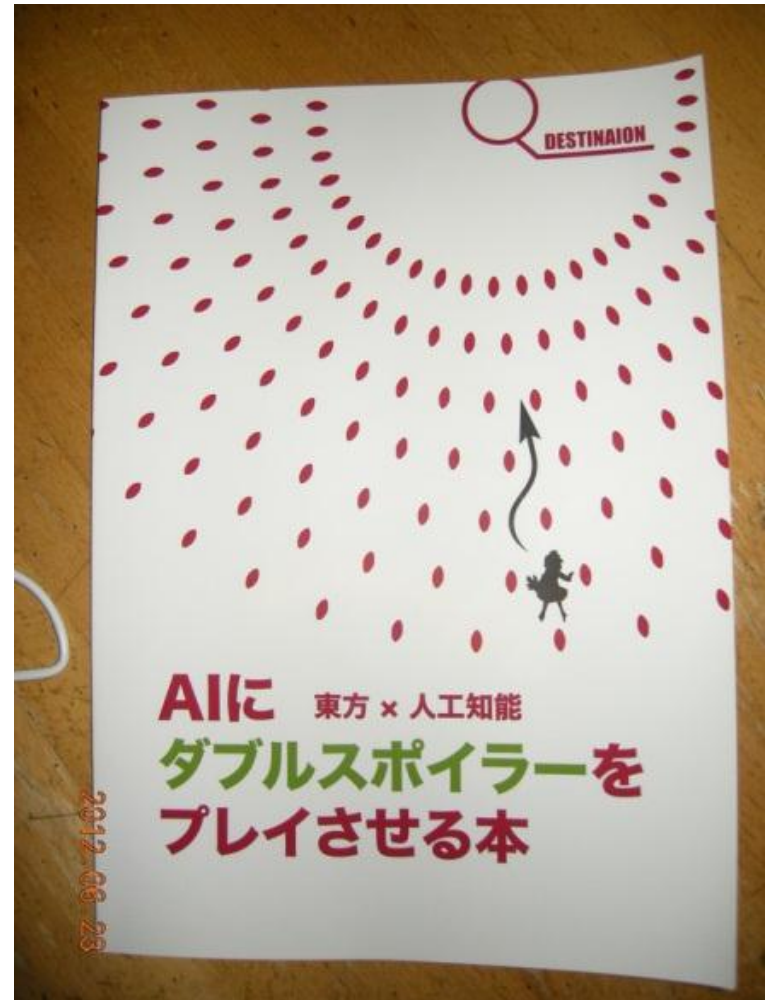
# 何をやったの？

- 東方地霊殿をプレイするAIを作った！
  - 新歓で展示
  - 避け方がキモいと評判に



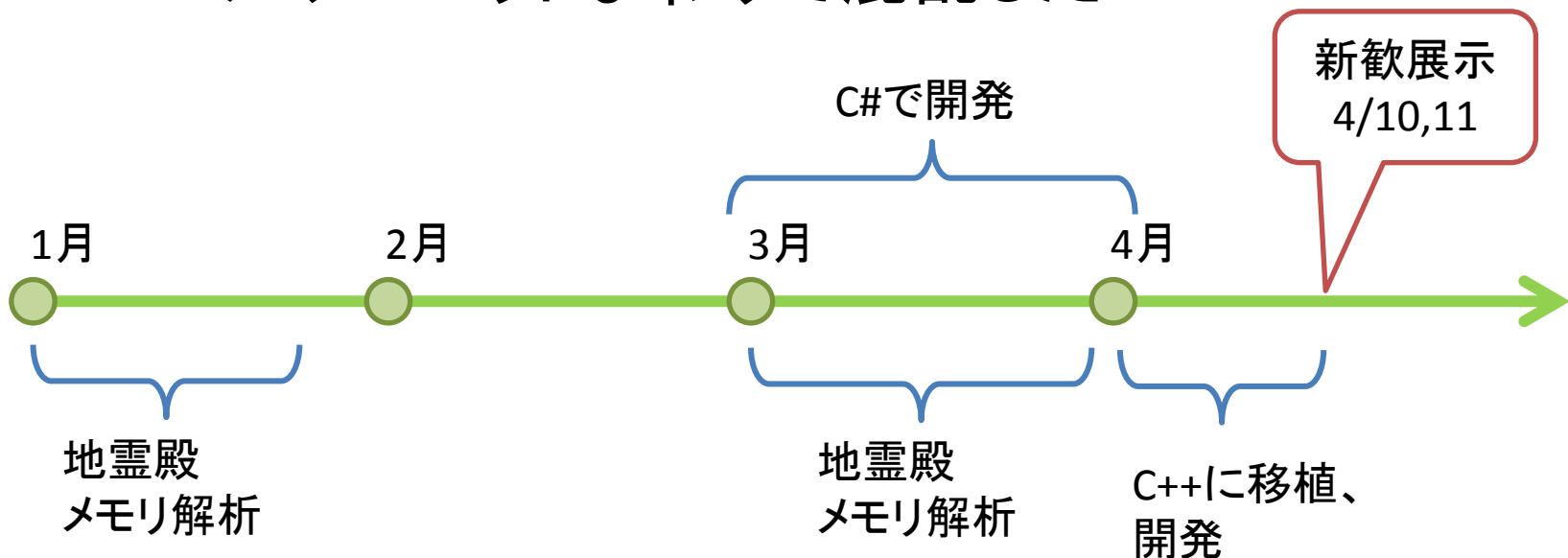
# 動機

- 冬コミで買った薄い本の影響を受けた
  - 画像処理で弾の位置を推測するアプローチ
  - じゃあ俺はメモリ解析でやろう、ということで



# スケジュール

- メモリ解析：コーディング ≒ 1:1
- C#で書いてたけど後にC++に移行
  - パフォーマンスが足りない
  - マルチスレッドまわりで混乱した



# プログラムの流れ

1. 地霊殿プロセスのメモリから自機や弾などの情報を読み取る
2. 読み取った情報を元に次に自機がすべき操作を決める
3. 操作をキー入力に変換して地霊殿プロセスに送りつける

以上のことを繰り返す

# プログラムの流れ(cont.)

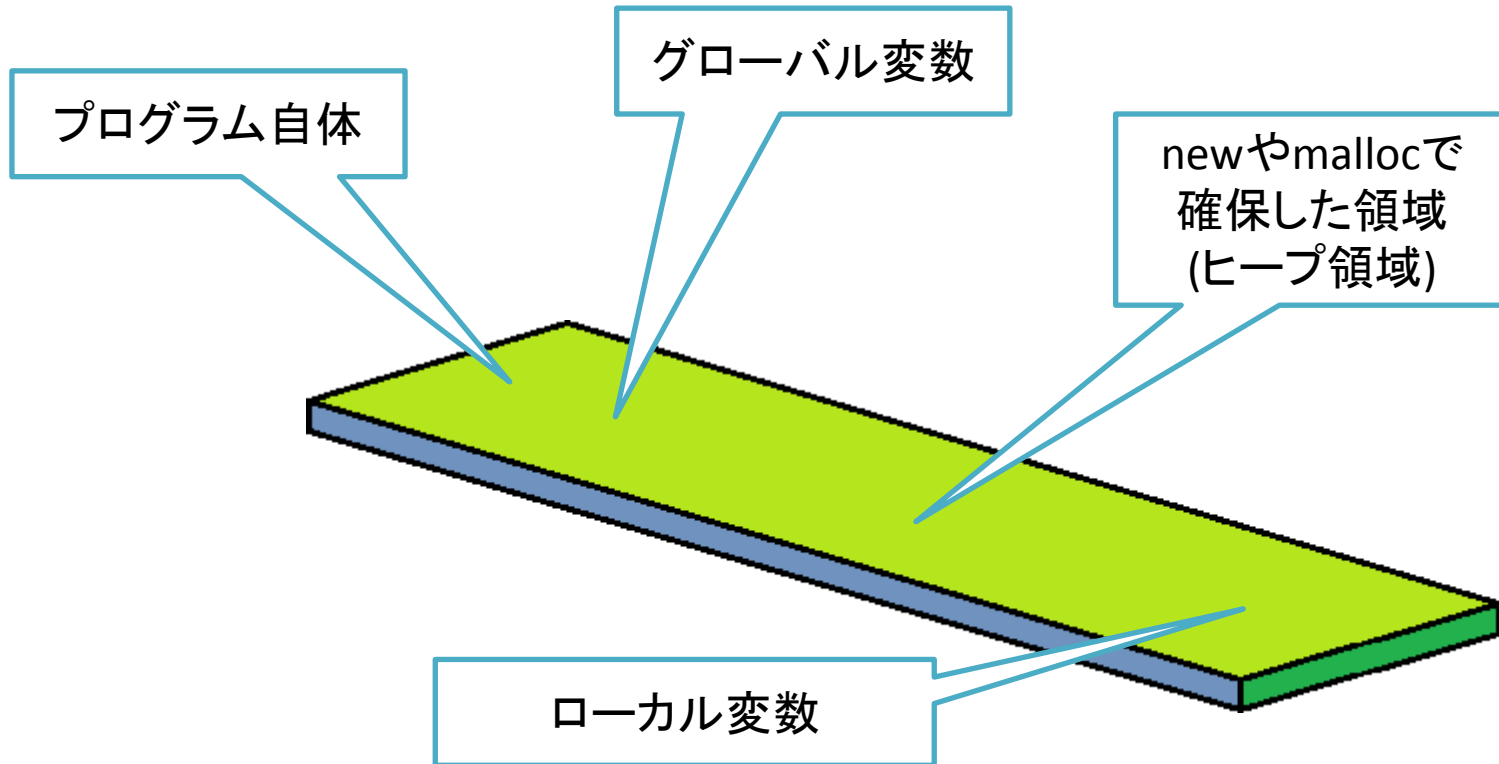
- メモリからどうやって情報を読み取るの？
  - あらかじめメモリ上での表現を(人力で)調べとく
  - メモリを読むこと自体はWin32APIで可能
    - ReadProcessMemory()を使う
- 操作をどうやって決めるの？
  - いい感じに動きそうなモデルを考える
    - 自分のプレイとか他人のプレイを参考に
  - 人間っぽい動きにするのは難しい・・・

# メモリ解析



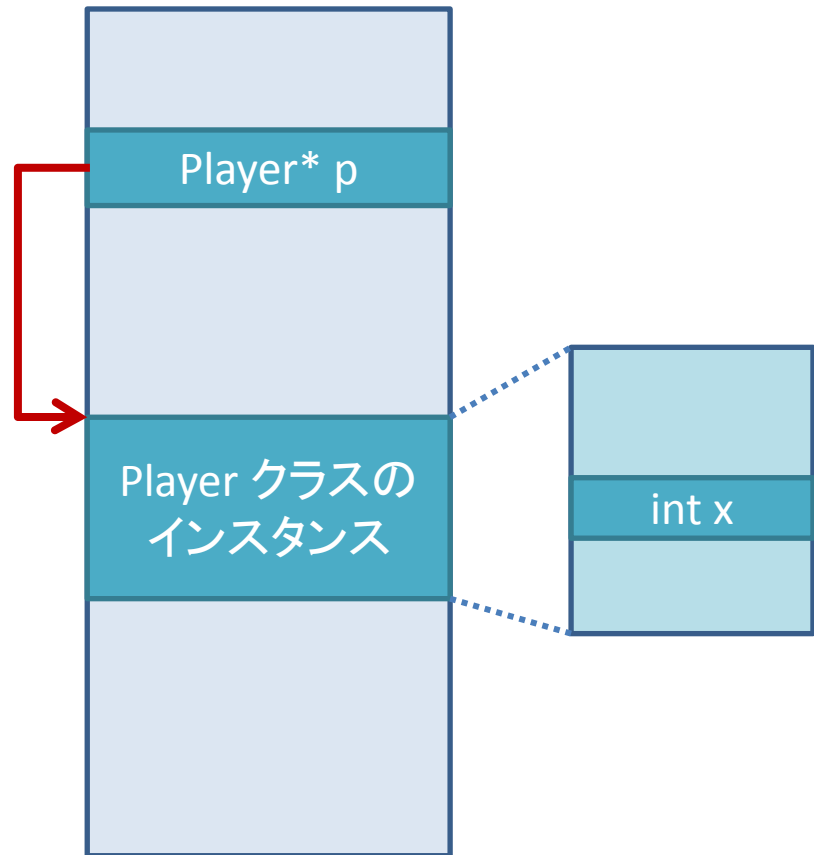
# メモリとプログラム

- プログラムは色々なデータをメモリに置く

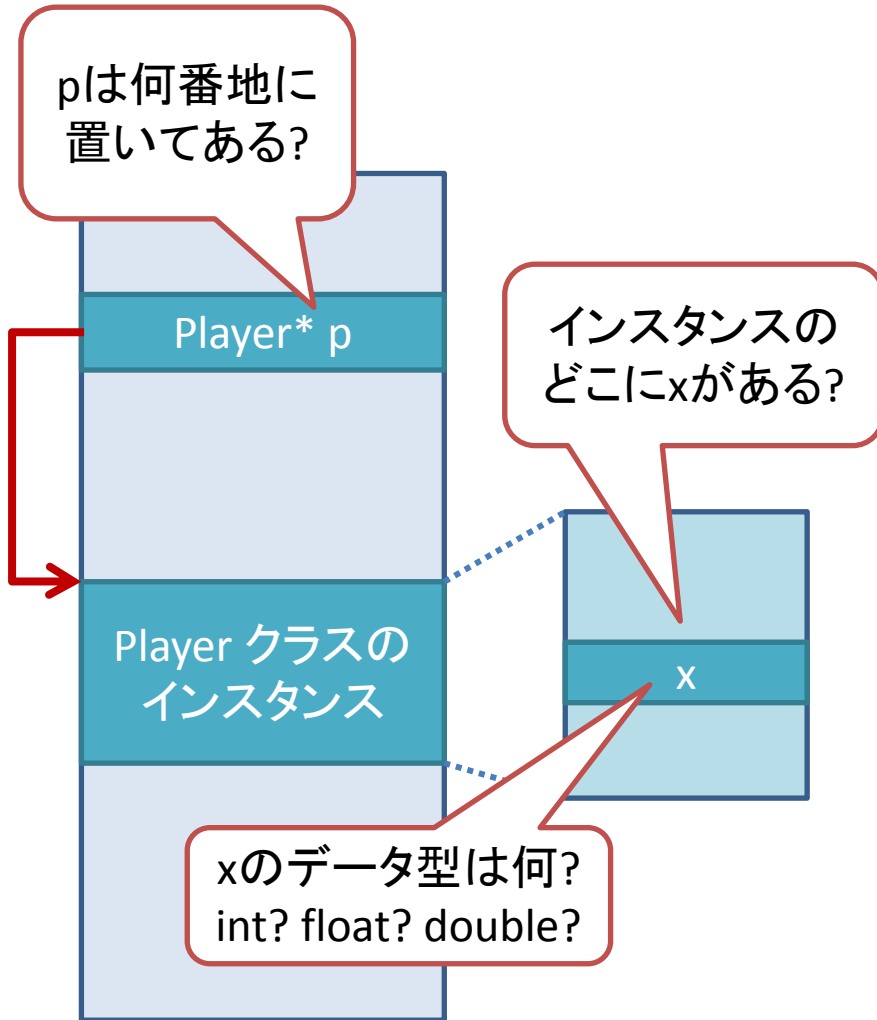


# メモリとプログラム(cont.)

```
class Player{  
    int x;  
    // ...  
};  
Player* p;  
int main(void){  
    p = new Player();  
    // ...  
}
```



# メモリ解析



メモリ解析=

- どこに
  - 何が
  - どんな型で
- 置いてあるかを調べること

# どうやるの？

- メモリの値の変化の特徴を手がかりにする
  - ex. 自機のX座標は横移動に合わせて増減
- メモリブレークポイントを仕込む
  - データをいじるコードが見つかる
- 逆アセンブルしたコードを読む
  - 読んでも分からんところはデバッガで

# メモリ解析デモ

- 自機のX座標を探そう
  - 自機はヒープ領域にあるはず
    - 普通は自機のクラスを作ってnewするので
  - 型はfloatと推定
    - Direct Xだと座標をfloatで扱うので

# 地霊殿のメモリ解析

## 1月中に解析したもの

- 自機の情報配置
- 当たり判定処理コードの場所と処理内容

## 3月に解析した(1月の解析で挫折した)もの

- 敵の情報配置
- 弾の情報配置
- 特殊な当たり判定処理の処理内容
  - レーザーとかオニオンリングとか

**弾避けAI**

# 弾避けAIのお仕事

- 次にすべき操作をいい感じに決めること
    - 「被弾しない」が目標
    - 今の弾の位置や速度は分かっている
      - ごりごりメモリ解析したのはこの辺を知るため
    - できる操作は
      - 上下左右の高速移動(斜め移動はなし)
      - 停止
- のいずれか



# 作成したAI

- 2つのポリシーに基づいて避ける
  - 脊髄反射的
    - 弾の動きを予測して、当たりそうなら避ける
    - 当たりそうになれば動かない
    - 細かく避ける
  - 積極的
    - 弾幕が薄いところに逃げる
    - 逃げるときのルートはA\*アルゴリズムで決める
    - 大雑把に避ける

AI デモ

# 戦果

- 地霊殿Easyすらクリアできないorz
  - 大局的な回避ができずに詰む
    - 霊鳥路空の巨大弾で角に追い込まれる><
  - 初見殺しを学習せずに詰む
    - 三歩必殺の泡弾に埋もれる><
  - 予測がアレで死ぬ
    - レーザーの太さが変わるとか予測できない><
- とはいえそこそこ避けられる
  - が、人間的じゃなくてキモい><

# まとめ

- 弾避けAIむずい
  - だからこそ面白い！
- メモリ解析つらい
  - 背徳的な面白さはあるよ！
    - 東方の当たり判定がアレだとか円対円の当たり判定をミスってるとか結構グローバル変数(static変数かもしれない)が多いとか色々知ることができた
- つらいけどその分だけの面白さはあった