




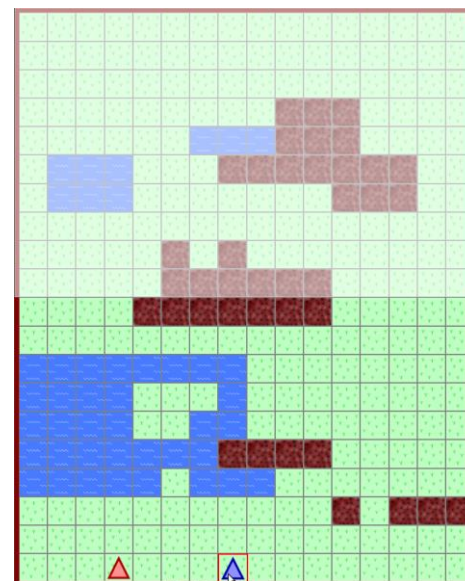
# 副テーマ進捗報告： SamurAI Coding 2018-2019



M1 山内 拓磨

# SamurAI Coding

- 2人のプレイヤーが障害物のあるコース上をステップごとに加速・減速しながらゴールに到達するステップ数を競う
- プレイヤーの制御はAIが行い、ステップごとに加速度を決定する
- プレイヤーには視界内の障害物の情報が与えられ、ステップ毎に更新される



# Samurai Coding

- ・コースごとに視界距離、考慮時間が与えられる
- ・与えられた考慮時間を使い切る、または制限ステップ数に到達すると失格となる
- ・コースには行き止まりはない
- ・加速度( $a_x, a_y$ )はそれぞれの軸に対して  $\{-1, 0, 1\}$  のいずれかを選択する
- ・相手の進路を妨害することも可能
- ・障害にぶつかるような場合、移動は行われない

# これまでの作業

- ・ 環境作成、進め方を決定
- ・ 昨年度のサンプルAIを調査
- ・ 昨年度のルール上でAIを開発
- ・ 今年度のルールに合わせて移植

# 昨年度のサンプルAIを調査

- ・ サンプルAI、昨年の決勝8位のAIを調査
- ・ 調査したAI同士を総当たりさせた
  - alphabetaを用いたAIは枝刈りがうまくいけばいい結果が出せそう
- ・ 工夫できそうな点
  - 探索手法、評価関数、視界外の予測

# 昨年度のルール上でAIを開発

- ・ 考慮時間切れが発生する
  - - ・ 探索深さを調整
    - ・ 枝刈りのために探索順序を変更
- ・ 評価関数を改良
  - コストマップを作成し、それを基に評価

# 今年度のルールに移植

- ・ ルールの主な変更点
  - 池の追加 (池で停止すると速度がリセット)
  - 接触の判定が変更
- ・ 接触判定は今年度のルールに適応
- ・ コストマップの作成が未対応
  - 池をどうコストマップに反映するべきか

# 現状の課題など

- ・ コストマップに池をどのように反映させるか
- ・ 探索深さの最適化
- ・ alphabetaよりも効果的な手法はあるのか
- ・ より効率よく枝刈りを行うには
- ・ 相手の動きをより正確に予測するには
- ・ 使用可能な考慮時間などの目安が現状ではない