

サムライジョッキーゲームルール (暫定版)

情報処理学会プログラミングコンテスト委員会

2017/09/16

概要

Samurai Coding 2017-2018 に用いるゲームのルール案を述べる。このルール案は暫定版であり、詳細については今後の改訂の可能性がある。

1 ゲームの概要

AIの制御するふたりのプレイヤーがスタート位置から開始し、障害物が設置されたレースコース上でステップごとに位置を変えながら、ゴールに早く到達することを競うゲームである。

1 ゲームは、同じレースコースを用い両プレイヤーのスタート位置を交換した2レースからなる。両レースのゴールタイムの合計が短い方がゲームの勝者である。合計タイムが同一である場合は、そのゲームは引き分けである。

2 レースコース

レースコースは2次元格子状であり、ゲーム毎に大きさや障害物の位置が異なる。以下ではレースコースの幅を w 、コース長を l とし、各格子点の位置の座標を (x, y) ($0 \leq x < w, 0 \leq y$) とする。

コース上の格子点の一部は障害点である。障害点あるいは縦横斜めに隣接するふたつの障害点を結んだ線分を障害物と呼ぶ。障害物は固定されており、レース中に動くことはない。障害物を飛び越すことはできないので、障害物に囲まれた領域に入ることはできない。

プレイヤーを制御するAIプログラム(以下単に **AI** と呼ぶ)に与えられる障害物の情報は、プレイヤーの視界の範囲内のもの、すなわちプレイヤーの位置の前後一定範囲の y 座標にあるものについてのみである。加減速は限定された範囲でしかできないので、速度を上げ過ぎると障害物を避けられなくなる可能性がある。

図1にレースコースの例を示す。茶色の小円は障害点、線分や塗りつぶした領域は障害物およびそれらに囲まれた領域である。

レース中、両プレイヤーはいずれかの相異なる格子点にある。レース開始時の両プレイヤーの位置の x 座標は異なり、 y 座標は0である。両プレイヤーはステップごとに位置を変えて行き、コース長 l 以上の y 座標の地点に達したらゴールしたことになる。

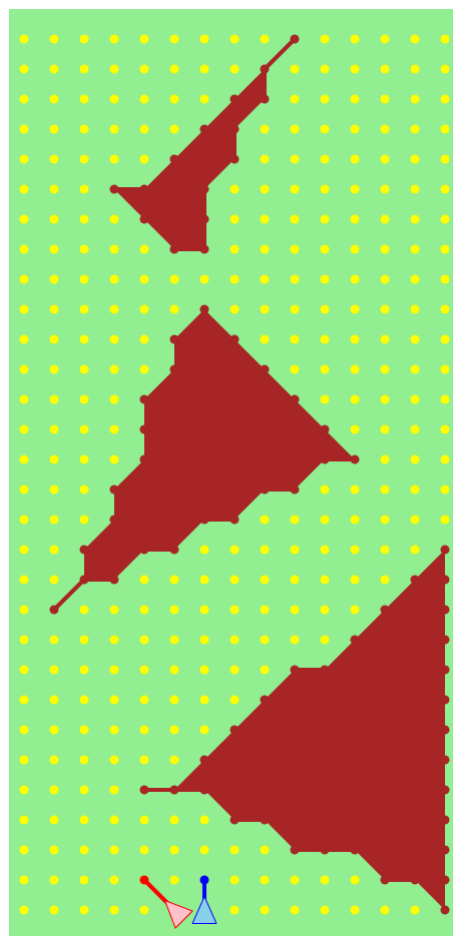


図 1: レースコースの例

コースには行き止まりがないものとする。コース内のどの格子点からも、十分小さい速度で移動すれば、一度も y 座標を減らすことなく障害物やコース端にぶつからずに y 座標がより大きい点に移動できる。

3 レースの進行

各レースでは、ステップごとに AI にレースの状況についての情報を与え、それに対する AI からの応答である加減速指示に従ってレースの状況を更新することを繰り返す。レース中に両プレイヤーが同時に同じ位置を占めることはない。

ステップは 0 から 1 ずつ進み、両プレイヤー共にゴールすればレースは終了である。ただし、制限ステップ数に達してもゴールしないプレイヤーがある場合、その時点でレースは終了し、ゴールに達していないプレイヤーのゴールタイムは制限ステップ数の 2 倍とする。

3.1 考慮時間

AI が 1 レースで使える考慮時間の合計には、制限を設ける。考慮時間はゲーム管理プログラムがプレイヤーに情報を送信し終わってから、プレイヤーからの応答を受信し終わるまでの実時間である。考慮時間の合計が制限値を超えたプレイヤーはそのレースを含むゲームを失う。¹

3.2 プレイヤーの状態と加減速指示

プレイヤーは各ステップの開始時において位置 (x, y) と速度 (v_x, v_y) を持つ。

AI はステップごとに速度を変更する加速度 (a_x, a_y) を指定する。 a_x, a_y は各々 $-1, 0, 1$ のいずれかである。

後述するコースアウトや衝突によって動けない場合を除き、次ステップのプレイヤーの位置の座標は $(x + v_x + a_x, y + v_y + a_y)$ となる。この位置を以下では次ステップにおける予定位置と呼ぶ。また、次ステップにおけるプレイヤーの速度はコースアウトや衝突の有無に関わらず $(v_x + a_x, v_y + a_y)$ となる。

3.3 動線とコースアウト

プレイヤーの位置と次ステップにおける予定位置を両端とする線分を、このプレイヤーの当該ステップでの動線と呼ぶ。

以下の場合、当該プレイヤーはコースアウトを生じているという。

- 予定位置がコース外に出る、すなわち予定位置の座標 (x, y) が $0 \leq x < w$ かつ $0 \leq y$ を満たさない場合
- 動線が障害物と交差あるいは接触する場合

コースアウトしたプレイヤーの次ステップでの位置は元のままになる。速度についてはコースアウトしなかった場合と同じ加速度が適用される。

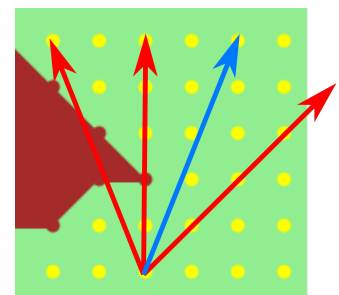


図 2: コースアウト
赤い動線はコースアウト

3.4 衝突と優先権

¹考慮時間の制限値は未定だが、数十秒程度となるだろう。

コースアウトがない場合に、両プレイヤーの動線が交差あるいは接触する場合、衝突が生じたという。

衝突が生じた場合は、優先権を持つプレイヤーの次ステップの位置は予定位置に、優先権のないプレイヤーの位置は元のままになる。いずれのプレイヤーについても、次ステップの速度は衝突しなかった場合と同じ $(v_x + a_x, v_y + a_y)$ となる。

優先権を持つのは、位置の y 座標がより小さいプレイヤーである。 y 座標が同一である場合、位置の x 座標がより小さいプレイヤーが優先権を持つ。ただし、動線が相手プレイヤーの動作前の位置に達するか通過する場合は、優先権が相手プレイヤーに移る。

両プレイヤーとも互いに相手の位置に達するか通過するような動線を持つ場合は、両者とも優先権を失い、次のステップでの位置は元のままとなる。

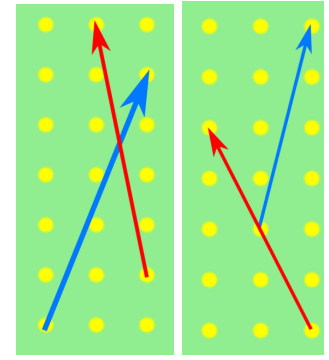


図 3: 衝突

いずれも、青い動線のプレイヤーが優先権を持つ

3.5 ゴールタイム

プレイヤーがステップ s で座標 (x, y) の位置にあり、次ステップで位置 (x', y') に達し、 $y' \geq l$ (l はコース長) であるとき、そのプレイヤーはゴールしたものとされる。このときゴールタイムは $s + (l - y)/(y' - y)$ で与えられる。

コースアウトしたプレイヤー、あるいは衝突を起こし優先権を持たないプレイヤーは、たとえ予定位置の y 座標がコース長以上であっても、次ステップの位置は元の位置のままとなるので、そのステップではゴールしない。

4 AIの動作

AI は初期化時にレース全体に関する情報を入力し、初期化終了を示す応答を出力する。ステップごとにはレース状況の情報を入力し、応答として加速度指示を出力する。

入力のひとつ以上の空白あるいは改行で区切った 10 進整数の並びである。負の数にはマイナス符号を前置する。初期化時の入力およびステップごとの入力の最後は必ず改行を置く。

AI の応答出力も空白あるいは改行で区切った 10 進整数の並びで、負の数にはマイナス符号を前置する。出力の最後には必ず改行を置く。

4.1 初期化時の入力

初期化時の AI の入力は以下の項目がこの順に並ぶものである。

考慮時間 t_{given} このレースで使える考慮時間をマイクロ秒単位の整数値として。

制限ステップ数 s_{max} ゴールまでのステップ数の上限を整数値として。

コースサイズ コースの幅 w および長さ l を、ふたつの整数値として。

視界 d 視界に入る y 座標の範囲を整数値 d として。位置 (x, y) にあるプレイヤーを制御する AI をに与えられる相手プレイヤーや障害点の情報は、 y 座標が $y - d$ から $y + d$ の範囲 (両端を含む) にあるもののみが与えられる。

4.2 入力形式

t_{given}
 s_{max}
 w_l
 d

4.3 初期化時の出力

AI は初期化が終了したことを整数 0 ひとつで伝える.

0

4.4 ステップごとの入力情報

ステップごとの AI の入力は以下の項目がこの順に並ぶものである.

ステップ s 当該のステップの値を整数ひとつで与える.

残り考慮時間 t_{left} このレースで使える考慮時間の残りをマイクロ秒単位の整数値として.

自プレイヤーの状態 当該の AI が制御するプレイヤーの位置の x 座標と y 座標, 速度の x 成分 v_x と y 成分 v_y の 4 整数.

相手プレイヤーの状態 相手プレイヤーの位置の x 座標 x_e と y 座標 y_e , 速度の x 成分 v_{x_e} と y 成分 v_{y_e} の 4 整数. ただし, 相手プレイヤーの位置が視界の外であれば, y 座標は -1 , 他は 0 になる.

障害点 視界内の各格子点が障害点であるか否かの情報で, $(2 \times d + 1) \times w$ 個の整数からなる. 視界内の $y - d$ から $y + d$ までの各 y 座標に対して w 個の整数 $o_{0,y}, o_{1,y}, \dots, o_{w-1,y}$ が, y 座標の順に与えられる. $o_{x,y}$ が 1 ならば格子点 (x, y) は障害点であり, 0 ならばそうではない. ただし, $y < 0$ であるコース外の格子点については 1 が与えられる.

4.5 入力形式

s
 t_{left}
 $x_y_v_x_v_y$
 $x_e_x_e_v_{x_e}_v_{y_e}$
 $o_{0,y-d}_o_{1,y-d}_ \dots _o_{w-1,y-d}$
 $o_{0,y-d+1}_o_{1,y-d+1}_ \dots _o_{w-1,y-d+1}$
 \dots
 $o_{0,y+d}_o_{1,y+d}_ \dots _o_{w-1,y+d}$

4.6 ステップごとの出力

AI は当該ステップの加速度 (a_x, a_y) をふたつの整数で指定する.

$a_x_a_y$

4.7 入力例

右に実際の入力例を示す。これは初期化から2ステップまでの入力となっている。

初期化時の入力について、考慮時間、制限ステップ数、コースの幅、長さと視界がそれぞれ 20000, 100, 15, 100, 8 となっていることが分かる。

この入力を受け取ったらプレイヤーは0と改行をを出力しなければならない。

以降はステップごとの入力情報になるが、まず最初のステップについて現在のステップと残り考慮時間がそれぞれ 0, 19981 となっていることが確認できる。このAIは初期化時に 19 [μs] だけかかり、残り 19981 [μs] 残されているようだ。

次の2行は自プレイヤーと相手プレイヤーの状態が示されており、このAIの現在の位置は $x = 5, y = 0$ で速度は x 軸方向 y 軸方向ともに0となっている。

次に続く17行（視界: $8 \times 2 + 1$ ）は視界内の格子点が障害点か否かを示している。 $y < 0$ におけるすべての格子点が障害点になることが分かる。前方には中央に障害物が集中しているようだ。

この入力を受けたら、加速度 (a_x, a_y) を出力する必要がある。すると、また次のステップの入力が与えられる。先ほどと同様の入力形式となっているが、どうやらこのAIは前ターンの加速度として $(-1, 1)$ を出力したようだ。

```
20000
100
15 100
8
0
19981
5 0 0 0
9 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
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 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0
1
19955
4 1 -1 1
8 1 -1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
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 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0
```

4.8 情報の記憶

ゲーム管理システムは、各ステップについて出力を終了したAIの動作を一時停止し、次ステップの情報を与える際に動作を再開させる。このため、ステップの間に計算を進めることはできないが、変数値などの実行コンテキストは1レースの間保持し続けることができる。

AIはファイル出力やネットワークアクセスができない環境で動作し、ゲーム管理システムはレース毎にAIを初期状態から立ち上げ直す。このため、AIは複数のレース間で情報を受け渡すことができない。

以上