サムライジョッキーデータファイルの形式

近山 隆

2017/09/25

概 要

サムライジョッキーゲームシステムが用いるデータファイルの記述形式を述べる.

1 データの種類

早駆けゲームシステムが用いるデータファイルには以下のものがある. いずれも JSON 形式である.

コース記述 ゲームで用いるコースを記述する形式. コースのサイズや障害物位置, プレイヤのスタート位置を記述する. 通常 editor で設計したコースを書き出したものを official で読み込んで用いる.

編集用コース記述 ゲームで用いるコースを編集するための形式. 障害物位置を多角形で指定する. 通常 editor でしか用いない.

レースログ 1 レースの経緯と結果を記録するもの. 通常 official が書き出したものを viewer で読み込んで表示するのに用いる.

2 コース記述

コースデータの記述ファイルは以下の形式である.

```
{
  "filetype":  "race course",
  "width":  w,
  "length":  l,
  "x0":  x0,
  "x1":  x1,
  "d":  d,
  "obstacles":  [
    [b1,1, b2,1, ..., bw,1],
    [b1,2, b2,2, ..., bw,2],
    ...,
    [b1,l, b2,l, ...bw,l]
  ]
}
```

ここで w はコースの幅, l はコースの長さ, x_0, x_1 はそれぞれプレイヤ 0 と 1 の初期 x 座標, d は視界に入る縦方向の距離, $b_{x,y}$ は座標が (x, y) である格子点の記述であり, 当該格子点が障害点であれば 1, さもなければ 0 である。

3　編集用コース記述

編集用コース記述は以下の形式である.

{
  "filetype": "race course for editing",
  "width": w,
  "length": l,
  "x0": x_0,
  "x1": x_1,
  "d": d,
  "obstacles": [$o_1, \ldots, o_n$]
}

ここで w から d までは上述のコース記述と同様である.

$o_1, \ldots o_n$ は障害物指定の記述である. 障害物指定は指定に用いる頂点座標

{"x": x, "y": y}

のリストである. 一点だけからなる障害物指定は頂点ひとつ, 線分による障害物指定は頂点ふたつ, 多角形による指定は 3 頂点以上で指定する.

4　レースログ

レースログファイルは以下の形式である.

{
  "filetype": "race log",
  "course": $course$,
  "name0": $name_0$,
  "name1": $name_1$,
  "time0": $time_0$
  "time1": $time_1$
  "log0": log_0,
  "log1": log_1
}

各項目は以下の通り.

$course$: レースが行われたコースの記述で, 上述のコース記述 (2) と同じ形式である.

$name_0, name_1$: レースを競った両プレイヤの名称の文字列である.

$time_0, time_1$: 両プレイヤのゴールタイムである. ゴールタイムは十進数で, 小数部を持つ場合もある. 制限ステップ以内にゴールできなかった場合は制限ステップ数の 2 倍になる.

log_0, log_1 各プレイヤのプレイ情報のリストで, ステップごとのプレイ内容とそれによって生じた結果のリストである.

プレイ情報リストの長さは当該プレイヤがゴールするまでのステップ数 (ゴールタイムを整数値に切り上げたもの) +1 であり, したがって両プレイヤで異なることもある. 当該プレイヤが制限ステップ数以内にゴールできなかった場合, 長さはその制限ステップ数 +1 となる.

プレイ情報の形式は以下のとおりである.

{
"step": $step$,
"before": $before$,
"velocity": $velo$,
"vision": "min": y_{min}, "max": y_{max} , "acceleration": $accel$,
"result": $result$,
"after": $after$
}

ここで各要素は以下を意味する.

$step$: このプレイのステップ番号である.

$before$: 動作前の位置.

$velo$: 動作前の速度.

y_{min}, y_{max}: 視界範囲の y 座標の最小値と最大値.

$accel$: 指示した加速度.

$result$: 動作結果を表す整数値で, 取りうる値とその意味は以下の通り.

-1: タイムアウト. 当該プレイヤの応答が制限時間以内に受け取れなかった.

0: 正常. 指示通りの動きができた. ゴールした場合も含む. 相手プレイヤと接触したが, 優先権を持っていたため, 指示通りの動きができた場合も含む.

1: 停止. コースの外に出る, コース上の障害物と接触, 相手プレイヤとの接触のいずれかにより, 元の位置に留まった.

$after$: 動作後の位置.

位置, 速度, 加速度はいずれも 2 次元のベクトルで, 以下の形式である.

{ "x": x, "y": y }

$r = 0$ の場合, 次ステップの位置は $(x + s_x + a_x, y + s_y + a_y)$ になる. それ以外の場合は, 次ステップの位置はこのステップと同じ (x, y) である. ただしゴールしたステップの $after$ は $x = y = -1$ で表す.

次ステップの速度はいずれの場合も $(s_x + a_x, s_y + a_y)$ となる.