

S5 course_generatorの仕様確認

【S5タスクの目的】

今後利用できるように、course_generator.pyが、ルール上可能性のあるすべてのコースを作成するように記述してあるか確認する。その他、確認すべき仕様をソースから確認する。

【まとめ】

■わかったこと

- ・ルールにはコース幅、コース長の制限が記述されていないが、course_generator.pyでは制限があった。コース幅は10~30の範囲、コース長は40~200の範囲。
- ・視界はコースごとに指定されるとルールに記述してあるが、どんな範囲から指定されるかについては記述がない。course_generator.pyでは視界は3~20の範囲からランダムに視界が選ばれている。

■必要であれば今後検証すること

- ・ルールには”行き止まりはないと記述されているが、course_generator.pyも”行き止まり”を作成しないようなソースになっているかどうか。
- ・course_generator.pyが作成するコースが、ゴールできるようになっているかどうか。

※10/23現在のルール。今後改訂される可能性あり。

以下はタスクで行ったことの詳細。

言いたいことは↑の【まとめ】に書いたので、詳細が気になる部分があれば読んでいただくと。

【コースに関するルールについて】

rule-jp.htmlからポイントだけ抜粋

コースの座標(x, y),コース幅W, コース長さlとすると、以下の条件が成立。

- ・ $0 \leq x < W$
- ・ $y < 0$ の範囲にも動ける
- ・ $y = l$ の時ゴールとする
- ・ レース開始時,playerの座標は $y=0$, x は異なる値に位置する
- ・ 障害や水溜りの座標は $0 \leq x < w$, $0 \leq y < l$ の範囲に限られる。つまり, コース幅の外, スタート位置より前, ゴールより後には存在しない
- ・ コースには行き止まりがない
- ・ レース開始時の両プレイヤーの位置から到達可能なコース内のどの柵目からも, 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

- ・ 視界はコースごとに定める

※ルールを見て気になったのは以下の二つ。これらについてルールと `course_generator.py` の整合性がとれているか調べた。

- ・ ”行き止まり”
- ・ コース幅とコース長、視界

【ルールと `course_generator.py` の整合性について】

■行き止まり

ルールに記載してある”行き止まり”を「ゴールができるかどうか」と解釈していたが、多分間違い。

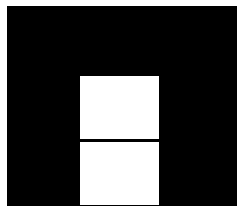
“行き止まり”の正しい解釈は、AIがy座標を負に移動しなければ進めない障害の並び方のこと。

例) 以下のような障害の並び方は行き止まりになる。1 が障害で0が障害でも池でもないことを表す。

[111]

[101]

[101]



ルールには”行き止まり”は存在しないと記述してある。`course_generator.py`が”行き止まり”をコースに含む可能性があるように記述されているか確認する必要がある。必要であれば後ほどやります。

■作成されたコースが、ゴールできるようになっているか否か

最初”行き止まり”のことを「ゴールができないこと」と間違って解釈していて、その解釈のまま調べていたので、やったこととして念のため記述する。

ゴールできないコースは、以下の二つの場合があると考えた。

① コース 1 行が全て障害(障害は数字の1で表す)の場合

コース幅が6だとすると、

[111111]

1	1	1	1	1	1
---	---	---	---	---	---

②障害に接しないと進めない場合(接するだけで衝突と判定されるので進めない)

コース幅が6だとすると、

[010100]

[101011]

	1		1		
1		1		1	1

これらの順列を含むリストがcorse_generator.pyによって作成されるか確認をした。
理論上100%だと断定はできないが、作成されたコースに行き止まりがあるかどうかを1000回ほど確認することでその代わりとした。

①について

大きさがコース幅である、要素が全て1のリスト [1,1,1,...]が、コースのそれぞれの行に含まれているかどうかを確認することで検証した。

コース作成を1000回行ってみて上記の検証を試したところ、①のような障害が一行に全て並ぶコースは生成されなかった。①のような場合は起こらないと考えてもいいと思う。

※ test_course_generator.pyの「validation1」の部分(204~204行あたり)

②について

・ **方法1** : (あんまり調べた意味ないかも)

「コースの各列全てに対して、(障害の数)<(行の総数-3)が成り立つ」ことは、ゴールできるコースであることの必要条件だと考えてとりあえず試してみた。

コース作成を1000回行ってみて上記の検証を試したところ、上記の必要条件は満たしていた。

※ソースの「validation2」の部分(216~235行あたり)

・ **方法2** : 深さ優先探索・幅優先探索

冷静に考えれば探索してゴールまでたどり着けば、ゴールできると言えることに気づいた。
これは簡単なAIを作ることと同じな気がする。必要であればやります。

■コース幅、コース長、視界

・コース幅、コース長：

ルールにはコース幅・コース長の制限が今のところ記述されていないが、ソース内には指定があり、この指定された制限を満たすコースが作成されていた。

・視界：

course_generator.pyでは3~20の範囲からランダムに視界が指定されるようになっている。ルールには、視界はコースごとに定められると記述されているが、どんな範囲で指定されるかについては記述がない。

```
wmin = 10 # コース幅の最小を10にする
wmax = 30 # コース幅の最大を30にする
lmin = 40 # コース長の最小を40にする
lmax = 200 # コース長の最大を200にする
vmin = 3 # 速度の最小を3にする
vmax = 20 # 速度の最大を20にする
```

※制限がないと参加者もAI作成の方針を立てにくいと思うので、今後ルールにコース幅、コース長、視界の制限が追記されるかもしれない。運営にコース幅・コース長の制限について直接聞くのもあり。

【参考：corse_generator.pyについて】

自分の確認用でメモしておきたかったので、ここからは読まなくていいです。

corse_generator.pyの目的はコースに必要な情報を作成すること。
作成された情報は、.crsファイルとして以下のように辞書型で記述される。

```
{"filetype":"race course 2018", "width":12, "length":182, "vision":13,
"squares":[(0, 1, 2 のいずれかからなる大きさwidth*lengthのリスト)],
"thinkTime":120, "stepLimit":182, "x0":4, "x1":8}
```

それぞれのwordの意味は以下の通り。

filetype : (略)
width : コース幅
length : コース長

vision : 作成されたコースに指定される視界

squares : コースを表すリスト。大きさは、(コース幅)*(コースの長さ)。各インデックスには0, 1, 2のいずれかが格納されている。0,1,2の意味は以下の通り。

0: 障害でも水溜りでもない

1: 障害

2: 水溜り

thinkTime : 考慮時間

stepLimit : 制限ステップ数

x0 : プレイヤーのスタート時のx座標

x1 : もう一方のプレイヤーのスタート時のx座標

以下のようにコース情報である.crsファイルを指定することで、指定したコース上でレースを実行できる。実行したレースの結果は.racelogファイルに保存され、このファイルをviewerで開くことで視覚的にレースの様子を確認できる。

```
official/official samples/sample.crs player/greedy Mary player/greedy  
John >/tmp/test.racelog
```