αβ(y軸方向への速度を押さえつける機構の修正編)

　田中 健人

ここでは、サンプルαβのソースのy軸方向への速度を押さえつける機構を、visionの1/2でなく、visionと同じ値、押さえつける機構を完全に取り外すとき、スコアがどう変化するか測定する。

サンプルのαβだけでなく、すでに作成した3種類の評価関数修正したAIでも同様に修正する。

サンプルαβに対して、visionと同じ値まで考慮するソースが以下(pab\_vis1)。

<https://drive.google.com/file/d/1R6E_PJbcERqhPdLcHyxL-LVJsRSUfK8i/view?usp=sharing>

全く考慮しないソースを以下。(pab\_vis2)

<https://drive.google.com/file/d/1R6E_PJbcERqhPdLcHyxL-LVJsRSUfK8i/view?usp=sharing>

同様に、pseudo\_alphabeta\_cal1に対して、visionと同じ値まで考慮するソースが以下。(pab\_cal1\_vis1)

<https://drive.google.com/file/d/1WhG7oUcu2MQEBcFrn5jjkxTGMeMbxd5W/view?usp=sharing>

全く考慮しないソースを以下。(pab\_cal1\_vis2)

<https://drive.google.com/file/d/1YQjFAxX_0KmpN32VRbEuamJYTRZA35Gz/view?usp=sharing>

pseudo\_alphabeta\_cal2に対して、visionと同じ値まで考慮するソースが以下。(pab\_cal2\_vis1)

<https://drive.google.com/file/d/1lO2at23C48hGrDT6FSGDfZ5Tar5c-IVN/view?usp=sharing>

全く考慮しないソースを以下。(pab\_cal2\_vis2)

<https://drive.google.com/file/d/1mncCmLmcOmDQ732PhuVeEOLpCHyxEjwj/view?usp=sharing>

pseudo\_alphabeta\_cal3に対して、visionと同じ値まで考慮するソースが以下。(pab\_cal3\_vis1)

<https://drive.google.com/file/d/1OZeqCr3FYyAVjlCqLhW1bnJyswb2YENF/view?usp=sharing>

全く考慮しないソースを以下。(pab\_cal3\_vis2)

<https://drive.google.com/file/d/1lRH0vdVNeq3FF9MZbqNpkulXS1vvaPPK/view?usp=sharing>

これらのAIで、

敵がいない場合のスコアをサンプルステージ11種類に対してplayer1として、測定する。

対象サンプルAIはランダム以外の4種類。

スコアは小数点第二以降は四捨五入した。

<player1>

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | sample-course | course01 | course02 | course03 | course04 | course05 | course06 | course07 | course08 | course09 | course10 |
| pseudo\_alphabeta | 51 | 22.5 | 41.5 | × | 69.4 | 50 | 41 | 32.3 | 36.8 | 20 | 26.4 |
| pab\_vis1 | 60.4 | 22.5 | 41.5 | × | 72.4 | 21 | 39.5 | 37.3 | 46.4 | 25 | 30.7 |
| pab\_vis2 | 60.4 | 22.5 | 41.5 | × | 72.4 | × | 41.6 | 41.1 | 46.4 | 25 | 30.7 |
| pseudo\_alphabeta\_cal1 | 49 | 24.3 | 39.5 | 77 | 68 | 50 | 41 | 32.3 | 36.8 | 19 | 26 |
| pab\_cal1\_vis1 | 61.8 | 24.1 | 39.5 | 77 | ×  ※1 | 21 | 39.5 | 39.1 | 48.4 | 18 | 30.7 |
| pab\_cal1\_vis2 | 68 | 24.1 | 39.5 | 77 | ×  ※1 | ×  ※2 | 41.6 | ×  ※2 | 48.4 | 18 | 30.7 |
| pseudo\_alphabeta\_cal2 | 49 | 23.8 | 41.5 | 77 | 68.4 | 50 | 41 | 32.3 | 36.8 | 20 | 26 |
| pab\_cal2\_vis1 |  |  |  |  |  |  |  |  |  |  |  |
| pab\_cal2\_vis2 |  |  |  |  |  |  |  |  |  |  |  |
| pseudo\_alphabeta\_cal3 | 49 | 24.3 | 41.5 | 77 | 68 | 50 | 41 | 32.3 | 36.8 | 19 | 25.8 |
| pab\_cal3\_vis1 | 61.8 | 24.1 | 41.5 | 39.1 | ×  ※1 | 21 | 39.5 | 39.1 | 47.4 | 18 | 30.7 |
| pab\_cal3\_vis2 |  |  |  |  |  |  |  |  |  |  |  |

※1.考慮時間使いすぎで、失格

※2.ステップオーバー

いや、y軸の速度は視野の1/2以上出すのは危険やな。

全てを計測せずとも、そのことが分かるので、残りは計測しない。

冷静に考えたらわかることやったわ。

偶然、スコアがよくなることはあっても、見えない障害物にぎりぎり対応できる限界が、視野の1/2までだわ。

もし、y軸の速度が視野の1/2だったら、新しく出てくる障害物は最高で視野の1/2先。

今、y軸の速度が視野の1/2なので、加速度-1で調整することが可能。

もし、y軸の速度が視野の1/2以上だったら、次のターンで見えるようになった障害物にぶつかる可能性がある。

ステージを作っている人の意図を考えて、見えない障害物を予測するのであるば、速度の1/2以上出しても大丈夫だが、

(例)スタートから、しばらく障害物が存在していなかった時、つぎの場所に障害物は存在しにくいなど。

しかし、ステージが完全ランダムで作成するなら、y軸の速度が視野の1/2以上出さないことが、理論上一番スコアを挙げれる。

また、視野の1/2以上の選択肢を削ることが、考慮時間的にも有効な省略のしかた。

しなかった場合、考慮時間が足りなくなるケースが、サンプルコースの中のみで考えても出てくる。