

Layer0-Stage03 左に進めるなら左を向く

code

```
-- ===== 追加部分(ここから) =====
if map:CanEnter(-1, 0) then
    if his.prev_action ~= "TurnLeft" then
        return "TurnLeft"
    end
end
-- ===== 追加部分(ここまで) =====

if map:CanEnter(0, 1) == false then
    return "TurnRight"
end

return "MoveForward"
```

または

```
-- ===== 追加部分(ここから) =====
if map:CanEnter(-1, 0) and his.prev_action ~= "TurnLeft" then
    return "TurnLeft"
end
-- ===== 追加部分(ここまで) =====

if map:CanEnter(0, 1) == false then
    return "TurnRight"
end

return "MoveForward"
```

意識

上の例の場合

- もし、左側が通れる、なら
 - もし、前回の行動が「左を向く」-ではない、なら
 - 行動決定「左を向く」
- もし、前方が通れ-ない、なら
 - 行動決定「右を向く」
- 行動決定「前に進む」

下の例の場合

- もし、左側が通れる、なおかつ、前回の行動が「左を向く」-ではない、なら
 - 行動決定「左を向く」
- もし、前方が通れ-ない、なら
 - 行動決定「右を向く」
- 行動決定「前に進む」

解説

- 前回の行動をチェックする(左を向いているなら条件を成立させない)ことで、連続で左を向かないようにしている

※ 以降は下の例のプログラムを元に進めていきます

検索キーワード

- 「Lua if文」 ... if文と論理演算子（and,or等）の使い方を調べる