

# E07 FF Planner(2)

---

18340013 Conghao Chen

October 19, 2020

## Contents

<b>1</b>	<b>Boxman Game</b>	<b>2</b>
<b>2</b>	<b>Notes</b>	<b>3</b>
<b>3</b>	<b>Codes</b>	<b>3</b>
<b>4</b>	<b>Results</b>	<b>7</b>

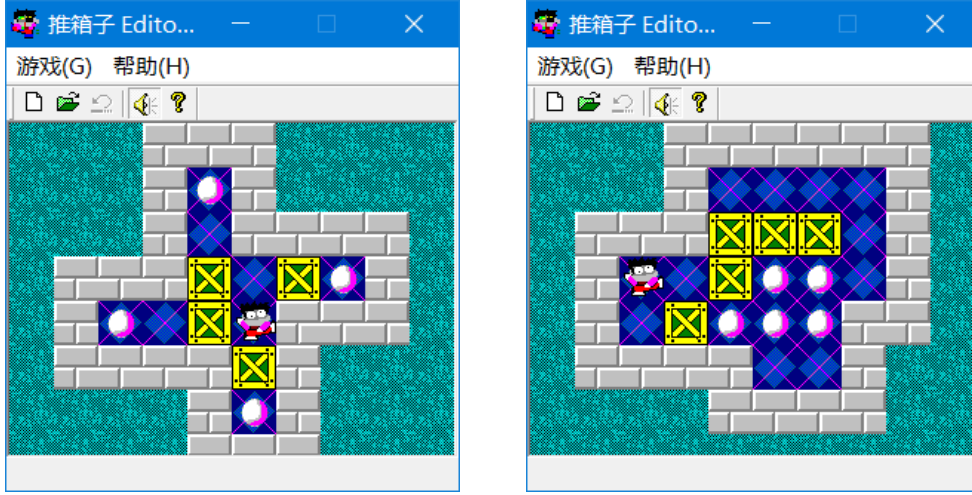


Figure 1: Boxman case1 (level 1) and case2 (level 10)

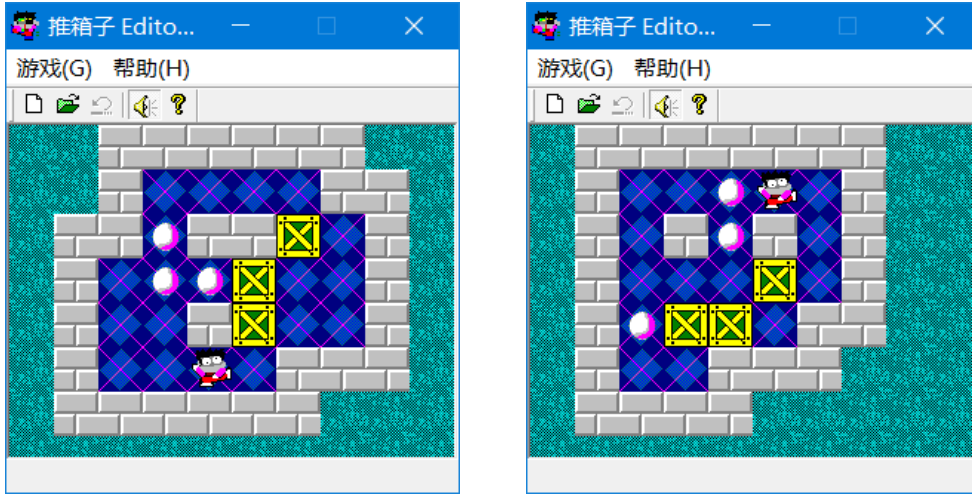


Figure 2: Boxman case3 (level 30) and case4 (level 40)

## 1 Boxman Game

If you don't know how to play the boxman game, you should open `BoxMan.zip` and click `BoxMan.exe` to have a try. You can also choose the level of the game to challenge yourselves. There are five cases choosed from level 1, 10, 30, 40, 50 in the following figures.

You can model the location information based on rectangular coordinates as mapped out in Figure 3. For example, we denote by P13 the position (1,3). The calculated action sequence can be like this: `MOVE P12 P13, PUSH BOX1 P14 P15...`, which means the guy runs from position (1,2) to position (1,3), and push the box1 from position (1,4) to position (1,5). However, this is only a very simple and intuitive approach to representing the actions and positions. If you have any other better methods, you can have a try.

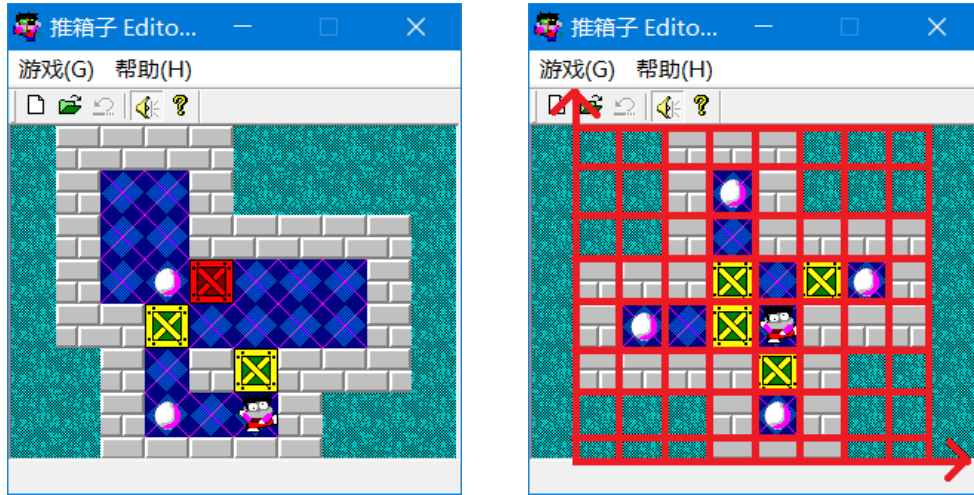


Figure 3: Boxman case5 (level 50) and modelling

Please solve the boxman game by using FF planner. You should hand in 2 files, including a domain file (boxman\_domain.pddl) and data file (boxman5.pddl).

## 2 Notes

Please send **E07\_YourNumber.zip** which should contain the codes(ai\_2020@foxmail.com).

## 3 Codes

boxman\_domain.pddl

```

1 (define (domain boxman)
2   (:requirements :strips :typing :equality)
3   (:types physob location)
4   (:constants PER - physob) ;表示推箱子的人
5
6   (:predicates
7     (at ?x - physob ?loc - location) ;x在位置 loc
8     (clear ?loc - location) ;位置 loc上没有箱子，但可能有人
9     (left ?loc1 ?loc2 - location) ;位置 loc1在 loc2的左边且相邻
10    (down ?loc1 ?loc2 - location) ;位置 loc1在 loc2的下面且相邻
11  )
12

```

```

13 (:action move ; 推箱子的人从from走到to
14 :parameters (?from ?to – location)
15 :precondition (and
16 (at PER ?from)
17 (clear ?to)
18 (or
19 (left ?from ?to) (left ?to ?from)
20 (down ?from ?to) (down ?to ?from)
21 )
22 )
23 :effect (and
24 (at PER ?to)
25 (not (at PER ?from))
26 )
27 )
28
29 (:action push ; 推箱子的人目前在pxy, 他把在from处的箱子推到位置to
30 :parameters (?box – physob ?pxy ?from ?to – location)
31 :precondition (and
32 (not (= ?box PER))
33 (at ?box ?from)
34 (at PER ?pxy)
35 (clear ?to)
36 (or ;
37 (and (left ?pxy ?from) (left ?from ?to)) ; 向右推
38 (and (left ?to ?from) (left ?from ?pxy)) ; 向左推
39 (and (down ?pxy ?from) (down ?from ?to)) ; 向上推
40 (and (down ?to ?from) (down ?from ?pxy)) ; 向下推
41 )
42 )
43 :effect (and
44 (not (at PER ?pxy))
45 (not (at ?box ?from))

```

```

46 (clear ?from)
47 (at PER ?from)
48 (at ?box ?to)
49 (not (clear ?to))
50 )
51 )
52 )

```

#### boxman5.pddl

```

1 (define (problem prob)
2 (:domain boxman)
3
4
5 (:objects
6  n25 n26 n27 n32 n33 n34 n35 n36 n37 n42 n44 n45 n52 n53 n54 n55 n64 n65
   n74 n75 — location
7  box1 box2 box3 — physob
8 )
9
10 (:init
11 (at box1 n34)
12 (at box2 n45)
13 (at box3 n53)
14 (at PER n52)
15
16 (clear n25)
17 (clear n26)
18 (clear n27)
19 (clear n32)
20 (clear n33)
21 (clear n35)
22 (clear n36)
23 (clear n37)

```

24	(clear n42)
25	(clear n44)
26	(clear n52)
27	(clear n54)
28	(clear n55)
29	(clear n64)
30	(clear n65)
31	(clear n74)
32	(clear n75)
33	
34	
35	(left n32 n42)
36	(left n42 n52)
37	(left n34 n44)
38	(left n44 n54)
39	(left n54 n64)
40	(left n64 n74)
41	(left n25 n35)
42	(left n35 n45)
43	(left n45 n55)
44	(left n55 n65)
45	(left n65 n75)
46	(left n26 n36)
47	(left n27 n37)
48	(down n25 n26)
49	(down n26 n27)
50	(down n32 n33)
51	(down n33 n34)
52	(down n34 n35)
53	(down n35 n36)
54	(down n36 n37)
55	(down n44 n45)
56	(down n52 n53)

```
57 (down n53 n54)
58 (down n54 n55)
59 (down n64 n65)
60 (down n74 n75)
61 )
62
63 (:goal (and
64 (not (clear n32))
65 (not (clear n35))
66 (not (clear n45))
67 )
68 )
69 )
```

## 4 Results

最终成功过关, 所用的步数是 50 步, 我根据 **Planner** 输出的结果自己推演了一遍, 最终的确能够让三个箱子到达目标位置。在定义位置的时候只定义了图中的蓝色区域; 灰色区域和绿色区域并未定义 (即直接按不存在处理)。还有就是注意 **clear** 函数, 这个函数为真当且仅当箱子不在该位置, 其他情况均为假。所以最后的 **goal** 就是 **not clear**, 也就是当且仅当箱子在目标位置。

## Found Plan (output)

(push box3 n52 n53 n54)

(move n53 n52)

(move n52 n42)

(move n42 n32)

(move n32 n33)

(push box1 n33 n34 n35)

(move n34 n44)

(push box3 n44 n54 n64)

(move n54 n55)

(move n55 n65)

(move n65 n75)

(move n75 n74)

(push box3 n74 n64 n54)

(push box3 n64 n54 n44)

(move n54 n53)

(move n53 n52)

(move n52 n42)

(move n42 n32)

(move n32 n33)

(move n33 n34)

(push box1 n34 n35 n36)

(push box2 n35 n45 n55)

(push box2 n45 n55 n65)

(move n55 n54)

(move n54 n64)

(move n64 n74)

(move n74 n75)

(push box2 n75 n65 n55)

(move n65 n64)

```
(:action push
:parameters (box3 n52 n53 n54)
:precondition
  (and
    (not
      (= box3 per)
    )
    (at box3 n53)
    (at per n52)
    (clear n54)
    (or
      (and
        (left n52 n53)
        (left n53 n54)
      )
      (and
        (left n54 n53)
        (left n53 n52)
      )
    )
    (and
      (down n52 n53)
      (down n53 n54)
    )
    (and
      (down n54 n53)
      (down n53 n52)
    )
  )
)
:effect
  (and
    (not
      (at per n52)
    )
    (not
      (at box3 n53)
    )
    (clear n53)
    (at per n53)
    (at box3 n54)
    (not
      (clear n54)
    )
  )
)
```



(move n64 n54)	
(push box3 n54 n44 n34)	
(move n44 n45)	
(move n45 n35)	
(push box3 n35 n34 n33)	
(push box3 n34 n33 n32)	
(move n33 n34)	
(move n34 n44)	
(move n44 n54)	
(move n54 n64)	
(move n64 n65)	
(push box2 n65 n55 n45)	
(move n55 n54)	
(move n54 n44)	
(move n44 n34)	
(move n34 n35)	
(move n35 n25)	
(move n25 n26)	
(move n26 n27)	
(move n27 n37)	
(push box1 n37 n36 n35)	