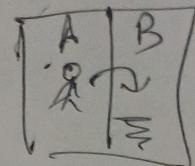


1/16/12

Labs

AI vacuum cleaner



agent(state, location) {

 if (isdirty){

 vacuum

}

 agent(false, left) //

 agent(false, right)

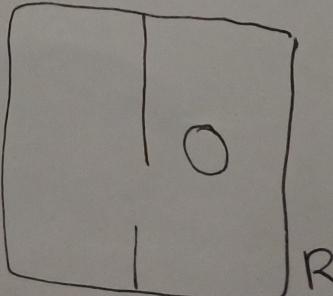
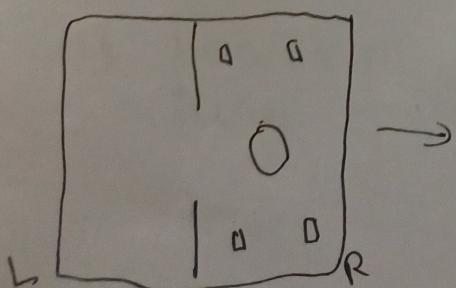
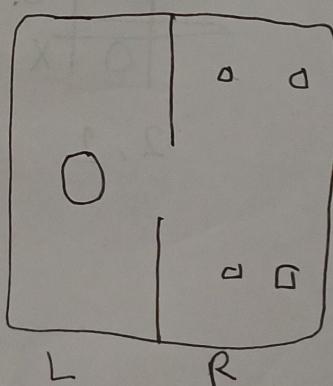
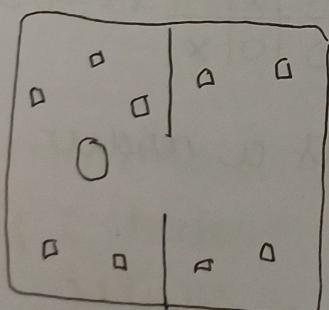
 return // exit condition

}

vacuum

 agent(false, left)

 agent(left, state)



(A, Right) (B)

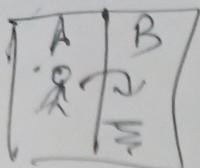
Roam Cle.

(dirty, A)

(dirty, B)

(clean, A)

(clean, B)



- ① (A, Right, Clean) \rightarrow Move right
 (B, Clean) \rightarrow Move left

(A, Right) (B, Clean) (B, Left) (A, Clean)

Roam Clean Mount

(Dirty, A) \rightarrow (Clean, A) $\xrightarrow[\text{right}]{\text{move}}$

(Dirty, B) \rightarrow (Clean, B) $\xrightarrow[\text{left}]{\text{move}}$

(Clean, A) $\xrightarrow[\text{right}]{\text{move}}$

(Clean, B) $\xrightarrow{\text{stop}}$

Algorithm

class vacuum:

def __init__(self):

self.rooms = {'A': 'dirty', 'B': 'dirty',
'C': 'dirty', 'D': 'dirty'}

self.position = 'A'

self.room_order = ['A', 'B', 'C', 'D']

self.current_index = 0

def status(self):

if self.rooms[self.position] == 'dirty':

self.rooms[self.position] = 'clean'

else

print(f"Room {self.position})")

def move_clockwise(self):

self.current_index = (self.current_index + 1) % 4

self.position = self.room_order

def move_counter(self):

self.current_index = (self.current_index - 1) % 4

self.position = self.room_order

[self.current_index]

def run

def run(self):

while 'dirty' in self.rooms_value():
self.status()
if self.rooms[self.position] == 'dirty':
self.suck()

else

if self.position == 'b':

self.moveCounterclockwise()

else

self.moveClockwise()

if -name- == "main":

vacuum = VacuumClean()

vacuum.run()

8/1/10

- 127.4

tex]