Name: Donghao Li
SU email: dli106@syr.edu

Question1:

1. linearrandom: This option causes the agent to execute the rules in sequential but randomized order. The rules are evaluated linearly, but the starting point within the list is chosen at random each time.
   all other options:
   linear: rules follow the order of linear list
   random: rules evaluated by completely random order
   priority: rules evaluated by pre-defined priority
   first: only consider the first applicable rule in the list
   last: only consider the last applicable rule in the list
2. At any time, the there is only one rule in the order list is being executed.
   Because of the linear random order, one cycle will have the first rule executable, and then the following rules will not be executed, for the worst case in linear random order, the last rule is executable, the whole list will evaluated but only the last one will be executed. normal case would be the first executable rule in the linear random list that would be executed.

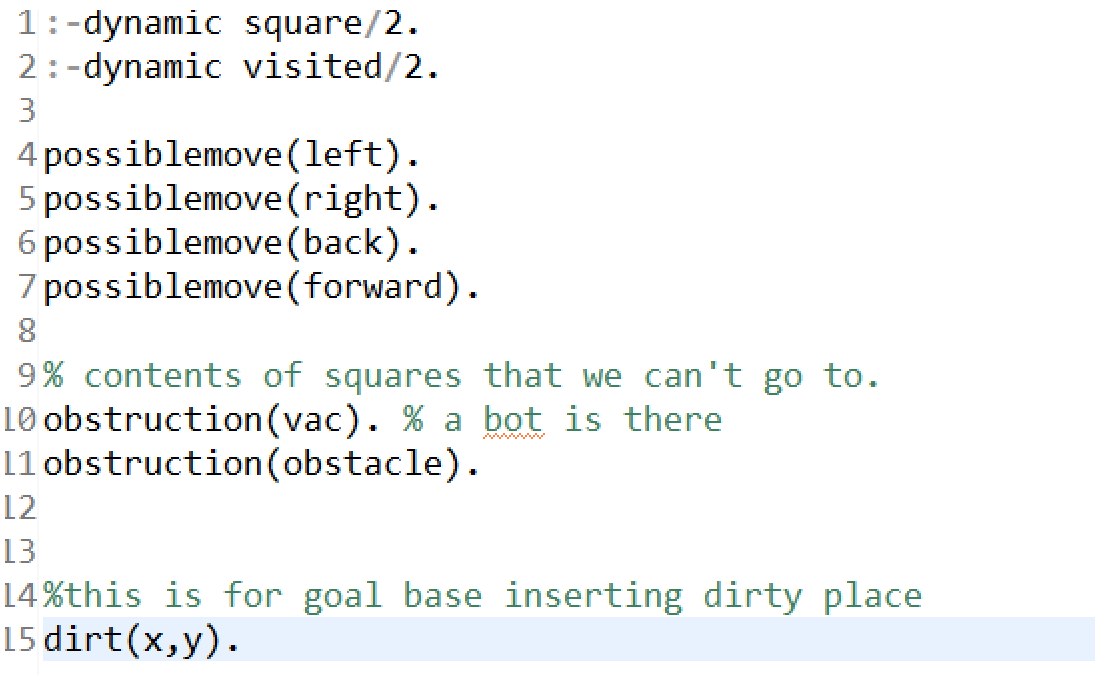
Question2:

1. Since the next line is "**forall percept** (square(Name,Contents)) **do insert**(square(Name,Contents))." The idea of this line of code is to insert the observation of the sensor into the model as the belief base if in a new cycle, the model do not clean up the belief base used by the previous cycle, there will exist a conflict between the new belief base and the old one. So we need to delete the old belief base at the beginning of a new cycle.
2. Continue with the idea of the last question, when we delete the old belief base, we need to insert new belief base for our agent to predict its action in this cycle that's the idea of this insert code mainly doing.

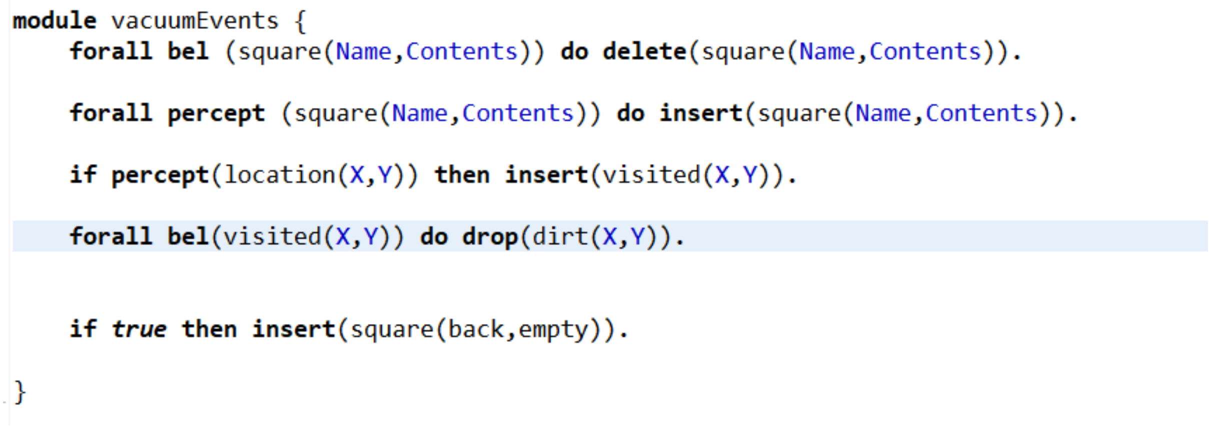
Question3:

I use adopt function in insert dust locations into goal base in the vacuumInit.mod2g.
in the vacuumEvent.mod2g i insert every current location of the robot into the belief base then drop te dirt location with same X and Y values. so that when the bot clean up a dust, the dirty location in the goal base will be dropped,
Finally when the goal base is empty, i use exit=nogoals to exit the program.

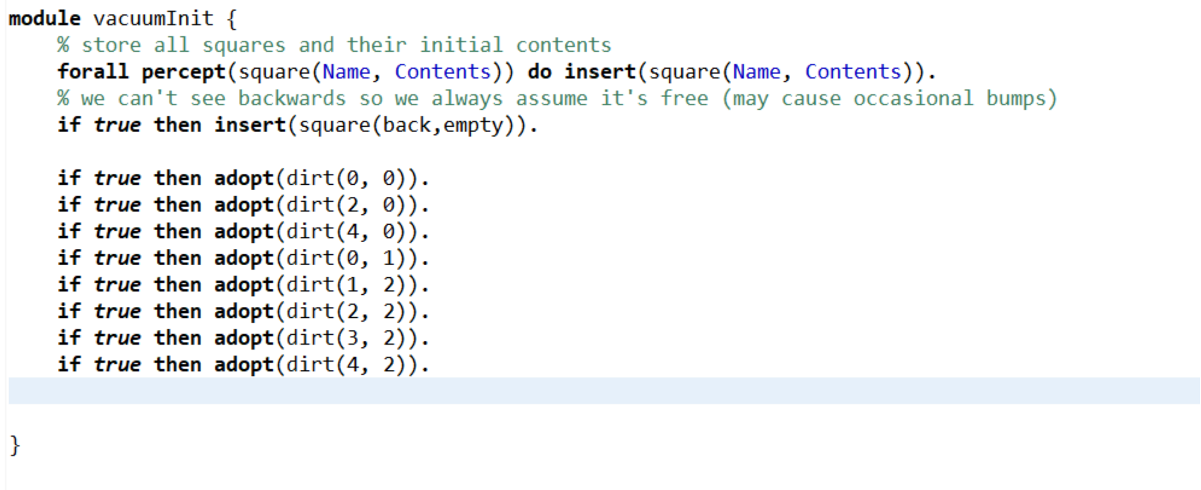
The goal-based adopting and dropping process is clearly shown in the video. Since the goal base model in the terminal does not show anything.



```
:-dynamic square/2.
:-dynamic visited/2.

possiblemove(left).
possiblemove(right).
possiblemove(back).
possiblemove(forward).

% contents of squares that we can't go to.
obstruction(vac). % a bot is there
obstruction(obstacle).


%this is for goal base inserting dirty place
dirt(x,y).
```



```
module vacuumEvents {
    forall bel (square(Name,Contents)) do delete(square(Name,Contents)).

    forall percept (square(Name,Contents)) do insert(square(Name,Contents)).

    if percept(location(X,Y)) then insert(visited(X,Y)).

    forall bel(visited(X,Y)) do drop(dirt(X,Y)).


    if true then insert(square(back,empty)).

}
```



```
module vacuumInit {
    % store all squares and their initial contents
    forall percept(square(Name, Contents)) do insert(square(Name, Contents)).
    % we can't see backwards so we always assume it's free (may cause occasional bumps)
    if true then insert(square(back,empty)).

    if true then adopt(dirt(0, 0)).
    if true then adopt(dirt(2, 0)).
    if true then adopt(dirt(4, 0)).
    if true then adopt(dirt(0, 1)).
    if true then adopt(dirt(1, 2)).
    if true then adopt(dirt(2, 2)).
    if true then adopt(dirt(3, 2)).
    if true then adopt(dirt(4, 2)).


}
```