

1. Among the main predicates we have the following:

- `room(X,Y)` %The localisation of the room
- `adjacentTo(room(X,Y),room(XT,YT))` %detection of adjacent rooms and their content
- `location(room(X,Y))` %the position of the hunter
- `stench(room(X,Y))` %the scent of the wumpis if in a 1 block radius
- `breeze(room(X,Y))` %the wind, meaning there is a put in a 1 block radius
- `glitter(room(X,Y))` %meaning there is a gold in the current position

And many more

2.

*%Can You grab Gold*

```
grabGold(location) :-  
    location(room(A,B)),  
    gold(room(X,Y)),  
    (X==A,Y==B).
```

*%Current Location of agent*

```
location(room(2,3)).
```



*grabGold(room(2,3))*

true

```
shootWumpus(room(X,Y)) :-
```

```
    room(X,Y),  
    room(W,Z),  
    wumpus(room(W,Z)),  
    adjacentTo(room(W,Z),room(X,Y)).
```

*%There is a wumpus in a location*

```
wumpus(room(_1,_2)) :-  
    room(_3,_4),  
    room(_1,_2),  
    stench(room(_3,_4)),  
    adjacentTo(room(_3,_4), room(_1,_2)).
```

*%stench Locations*

```
stench(room(1,2)).  
stench(room(1,4)).  
stench(room(2,3)).
```



*shootWumpus(room(2,3))*

true



*shootWumpus(room(2,2))*

false

3. As a limitation, we can't code for sure where there is a pit and where there are none because in a corner due to the size of the map as in a normal game, the agent wouldn't know that the position (4,2) is not a pit knowing that (4,1), (3,2), and (4,3) are breezes.



The Safe function also has limitations as it doesn't exactly give a precise answer when looking for safe positions.