In [1]:
```
from agents import *
```

In [2]:
```
class SondaMarciana(Agent):
    def absorveu(self, thing):
        print("Sonda : Absorveu uma bateria em {}.".format(self.location))
    def registrou(self, thing):
        print("Sonda : Registrou um marciano em {}.".format(self.location))
```

In [3]:
```
Spirit = SondaMarciana()
```

```
Can't find a valid program for SondaMarciana, falling back to default.
```

In [4]:
```
class Baterias(Thing):
    pass
class Marciano(Thing):
    pass
```

In [51]:
```
class Marte(Environment):
    def percept(self,agent):
        #retornar uma lista de coisas que estao na proximidade da sonda
        coisas = self.list_things_at(agent.location)
        return coisas
    def execute_action(self, agent, action):
        #altera o estado do ambiente baseado no que faz o agente"
        if action == "ir em frente":
            print('{} resolveu {} na posicao {}'.format( str(agent)[1:-1],action,
            agent.vaiemfrente()
        elif action == "absorver":
            items = self.list_things_at(agent.location, tclass= Baterias)
            if len(items)!=0:
                if agent.absorveu(items[0]): # Sonta absorveu a bateria
                    print('{} resolveu {} na posicao {}'.format( str(agent)[1:-1]
                    self.delete_thing(items[0])# remove a bateria absorvida
        elif action == "registrar":
            items = self.list_things_at(agent.location, tclass= Marciano)
            if len(items)!=0:
                if agent.registrou(items[0]): # Sonta registrou marciano
                    print('{} resolveu {} na posicao {}'.format( str(agent)[1:-1]
                    self.delete_thing(items[0])# ignora marciano registrado

    def is_done(self):
        no_edibles = not any( isinstance(thing, Baterias) or isinstance(thing, Ma
        dead_agents = not any( agent.is_alive() for agent in self.agents)
        return dead_agents or no_edibles
```

In [58]:
```
class SondaMarciana(Agent):
    location = 1
    def vaiemfrente(self):
        self.location +=1

    def absorveu(self, thing):
        if isinstance(thing, Baterias):
            return True
        return False
    def registrou(self, thing):
```

```
        if isinstance(thing, Marciano):
            return True
        return False
```

In [59]:
```
def program( percepts):
    for p in percepts:
        if isinstance(p, Baterias):
            return 'absorver'
        elif isinstance(p,Marciano):
            return 'registrar'
    return 'ir em frente'
```

In [60]:
```
planicieMarciana = Marte()
```

In [61]:
```
Spirit = SondaMarciana(program)
```

In [62]:
```
bateria1 = Baterias()
bateria2 = Baterias()
marciano1 = Marciano()
marciano2 = Marciano()

planicieMarciana.add_thing(Spirit,1)
planicieMarciana.add_thing(bateria1,5)
planicieMarciana.add_thing(bateria2,9)
planicieMarciana.add_thing(marciano1,12)
planicieMarciana.add_thing(marciano2,14)
```

In [63]:
```
planicieMarciana.run(20)
```

```
SondaMarciana resolveu ir em frente na posicao 1
SondaMarciana resolveu ir em frente na posicao 2
SondaMarciana resolveu ir em frente na posicao 3
SondaMarciana resolveu ir em frente na posicao 4
SondaMarciana resolveu absorver na posicao 5
SondaMarciana resolveu ir em frente na posicao 5
SondaMarciana resolveu ir em frente na posicao 6
SondaMarciana resolveu ir em frente na posicao 7
SondaMarciana resolveu ir em frente na posicao 8
SondaMarciana resolveu absorver na posicao 9
SondaMarciana resolveu ir em frente na posicao 9
SondaMarciana resolveu ir em frente na posicao 10
SondaMarciana resolveu ir em frente na posicao 11
SondaMarciana resolveu registrar na posicao 12
SondaMarciana resolveu ir em frente na posicao 12
SondaMarciana resolveu ir em frente na posicao 13
SondaMarciana resolveu registrar na posicao 14
```

In [ ]:

In [ ]: