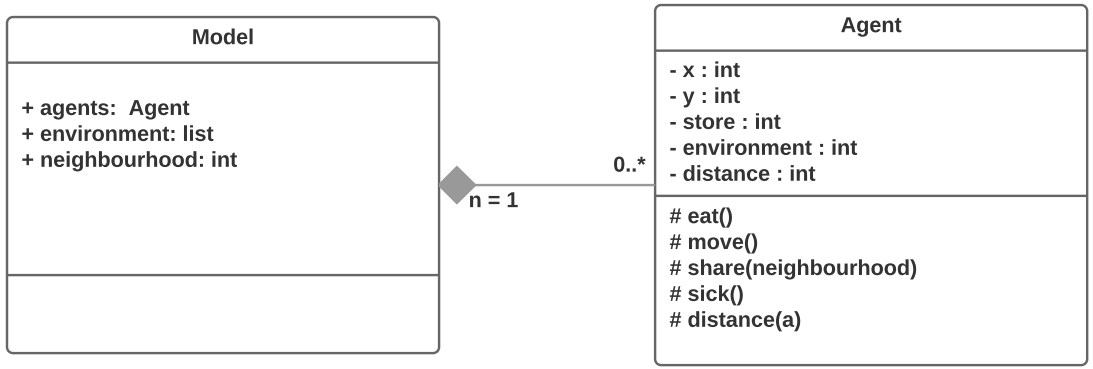
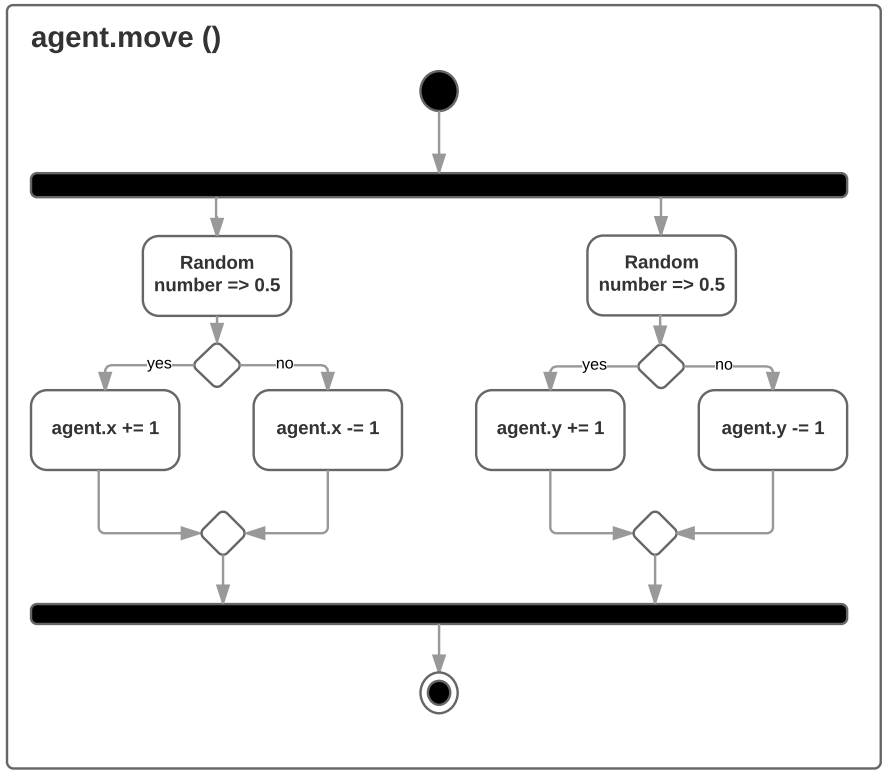
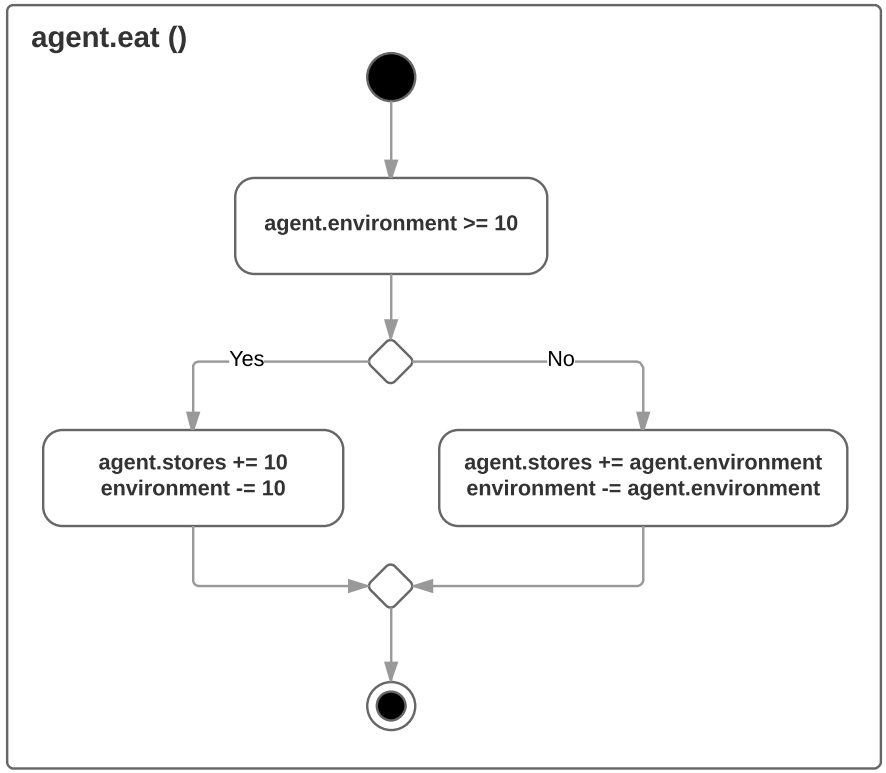
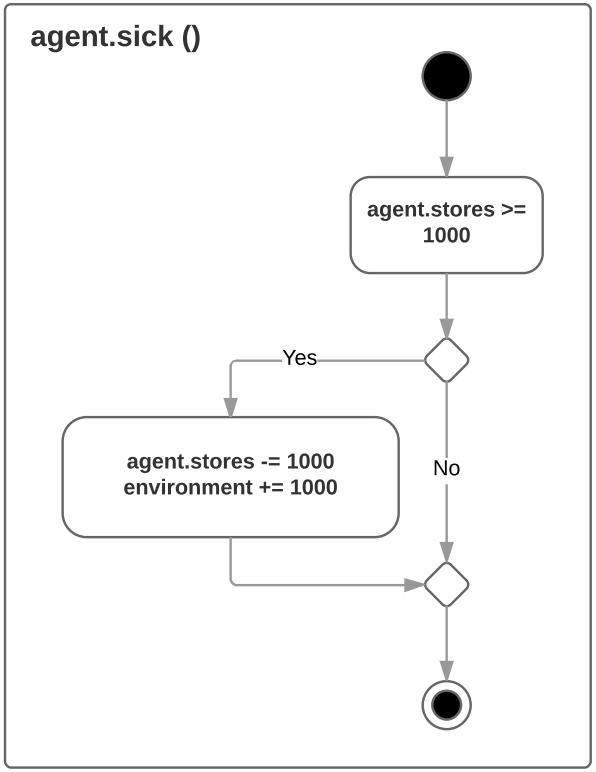
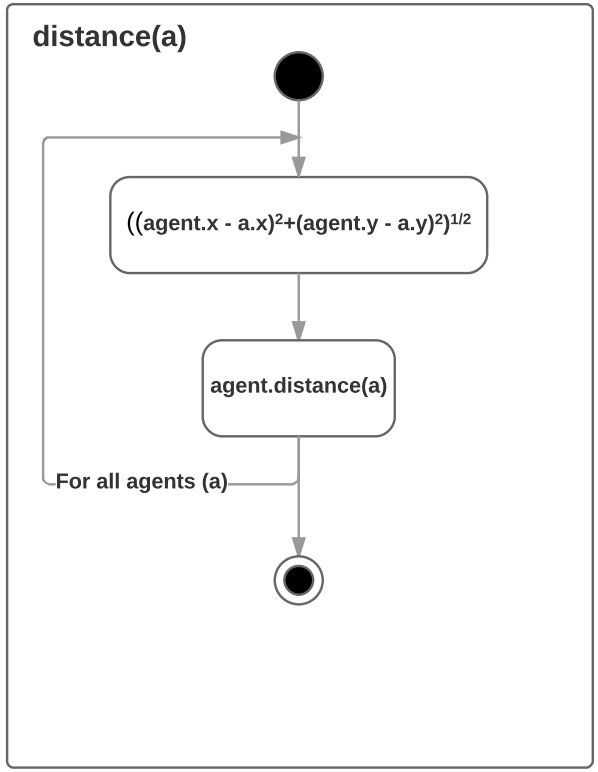
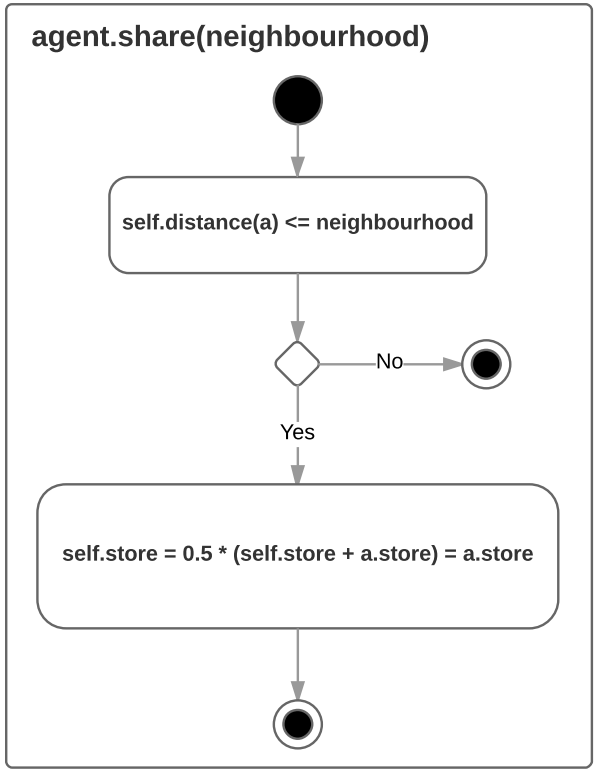

Model
+ agents: Agent
+ environment: list
+ neighbourhood: int
n = 1
0..*
Agent
- x : int
- y : int
- store : int
- environment : int
- distance : int
eat()
move()
share(neighbourhood)
sick()
distance(a)


agent.move ()
Random number => 0.5
Random number => 0.5
yes
no
yes
no
agent.x += 1
agent.x -= 1
agent.y += 1
agent.y -= 1


agent.eat ()
agent.environment >= 10
Yes
No
agent.stores += 10
environment -= 10
agent.stores += agent.environment
environment -= agent.environment







Notes to the model

Agents are created within the model from the agent framework.

The model reads in an environment file as a list and parses into a two dimensional array.

Agents move at random usnig the agent.move() function. The model calls this with the agent.eat() function which reduces the environment score at any given (x,y) location of the agent, and increases the variable agent.store() by the same value (here 10), with a lower bound of zero.

Calling the agent.sick() function assesses whether agent.store() has reached a threshold (here 1000) above which, agent.store() is reduced, and the environment file at the agent's location is increased by the same value.

Agent.share() uses the function distance(a). Distance(a) calculates the pythagorean dsitance between agent() and all other agents(a) and stores the distance as a list agent.distance(). Where agent.distance(a) is smaller than the neighbourhood threshold defined within the model, agent.store() and a.store() are summed, and divided equally between the two variables.