Agent needs to keep track of:

* Intersection Agent is located at
* How Agent got to where it is
* Exit door and how Agent got there and steps it took away from it
* How many keys Agent has

System needs to keep track of:

* All Intersections
* Where keys are
* Where exit is

Intersection: Any place where there is more than two paths out of the square

Dead-end: Any place where there is only one path attached the square

(coordinate block0 0 0)

(coordinate block1 0 1)

.

.

.

(coordinate block63 7 7)

(agentAt block)

(win False)

(keyCount 0)

(isWalk block)

(hasKey block)

(isExit block)

(isStart block)

(isIntersection block)

(isDeadend block)

((connected ?blockA ?blockB) ⇐

(isWalk ?blockA)

(isWalk ?blockB)

(coordinate ?blockA ?xA ?yA)

(coordinate ?blockB ?xB ?yA)

(equals (abs (subtraction ?xA ?xB)) 1))

//differenceFn

//(evaluate ?x (DifferenceFn ?xA ?xB))

((connected ?blockA ?blockB) ⇐

(isWalk ?blockA)

(isWalk ?blockB)

(coordinate ?blockA ?xA ?yA)

(coordinate ?blockB ?xA ?yB)

(equals (abs (subtraction ?yA ?yB)) 1))

((agentCanWalk ?blockStart ?blockEnd) ⇐

(agentAt ?blockStart)

(connected ?blockStart ?blockEnd))

((isIntersection ?block) ⇐

(connected ?block ?blockA)

(connected ?block ?blockB)

(connected ?block ?blockC)

(different ?blockA ?blockB)

(different ?blockB ?blockC)

(different ?blockA ?blockC))

((isIntersection ?block) ⇐

(connected ?block ?blockA)

(connected ?block ?blockB)

(connected ?block ?blockC)

(connected ?block ?blockD)

(different ?blockA ?blockB)

(different ?blockB ?blockC)

(different ?blockA ?blockC)

(different ?blockA ?blockD)

(different ?blockB ?blockD)

(different ?blockC ?blockD))

((isDeadend ?blockA) ⇐

(eql (numberOfBindingsFn (connected ?blockA ?blockB) 1))

//closedRetrievalSetOf

((win True) ⇐

(agentAt ?block)

(isEnd ?block)

(keyCount 3))

ACTIONS:

openDoor: agent is in the square with the exit and agent keycount is 3; game is ended

moveToBlock: move to any adjacent square,agent is in a different block, blocks connect

claimKey: agent is in a square and key is in a square; keycount++, space no longer has key

startMaze: keycount is 0, agent is at entrance