

2. Laboratorio

En `pacman.py` la clase `GameState`

El `GameState` contiene toda la información necesaria sobre:

- 1.- La comida que hay
2. - Las cápsulas
- 3.- Las configuraciones de los agentes (fantasmas y pacman)

En pacman.py la clase GameState

Métodos:

def getLegalActions(self, agentIndex=0):

Devuelve las acciones “posibles” a partir del estado en el que está. El parámetro agentIndex sirve para discriminar que agente es el actual (0 pacman, >0 fantasma)

def generateSuccessor(self, agentIndex, action):

Devuelve el estado sucesor una vez tomada la acción

def getGhostStates(self):

Devuelve los estados de los fantasmas. Por cada fantasma ghost al recorrerlo podré obtener su posición con el ghost.getPosition().

def getPacmanPosition(self):

Devuelve la posición del Pacman

def hasFood(self), x, y:

return self.data.food[x][y]

def hasWalls((self,x,y):

return self.data.layout.walls[x][y]

def getCapsules(self):

Devuelve la lista de posiciones (x,y) de las capsulas restantes

def getNumFood(self):

Devuelve un contador de las comidas que restan sin comer

def getWalls(self):

Devuelve una matriz de booleanos indicando si en cada x,y hay una pared

1 pregunta

```
successorGameState = currentGameState.generatePacmanSuccessor(action)
newPos = successorGameState.getPacmanPosition()
newFood = successorGameState.getFood()
newGhostStates = successorGameState.getGhostStates()
newScaredTimes = [ghostState.scaredTimer for ghostState in newGhostStates]
```