**SYSC5013 Assignment 1, Question 2. State-Based Agent.**

**Introduction and mathematical notations**

General definition of Agent is: RE -> Ac. Agents are not necessarily reactive as they don't only take the latest state of the environment, but they may use information from the past states that may have happened. Such behavior of an agent is called state-based, where information from the past can influence the agent's present behavior.

To fully define the behavior of state-based agent (Q X E -> Ac x Q), we need to enumerate over all possible pairs of Q (my current state) and E (Environment state), and then return the particular action Ac and the new state Q that you go into. A Finite State Machine diagram is useful in such scenarios (on next page).

**Environment and Actions**

The environment τ is also known as the state transformer function is denoted as: RAC ->E. For the agent, the scope of the environment (e1,e2,e3) is ball visibility, goal visibility and distance to the ball. The text file (textfile.txt) provided presents the environment states as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Text file representation: **Y** (Yes), **N**(No), **F**(In Front), **U**(Unknown) | | | | |
| Ball Visibility | [Y, | N, | F ] | ∈ | BallSeenByPlayer |
| Distance to the ball | [Y, | N, | U] | ∈ | BallCloseToPlayer |
| Goal visibility | [Y, | N, | F ] | ∈ | GoalVisibility |

For the agent, the actions (α1,α2,α3) are turn, kick or dash and can only be performed once/cycle.

[turn, kick, dash] ∈ Agent Actions.

**Reasoning of State-Based agent**

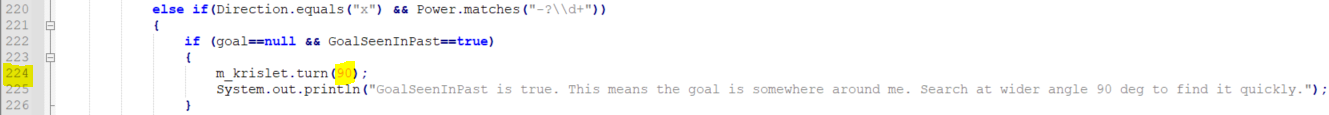
Agent's brain has two states. This functionality is implemented in Brain.java file using a Boolean variable (GoalSeenInPast) that will help to remember if the goal was seen in the last run N-1 of the while loop. The variable is set to false if TempEnvironmentStates.GoalVisibility in the current N cycle of the while loop is "N" (No) as seen in line 278 of Brain.java. That's how the agent becomes state based as it's able to remember something that happened in the previous cycle of while loop.

From the definition Q X E -> Ac x Q, the State-Based function becomes:

[GoalSeenInPast 1/0]x[BallSeenByPlayer, BallCloseToPlayer, GoalVisibility]->[Action]x[GoalSeenInPast 1/0]

To put the definition in one line, 1/0 simply means True/False for the Boolean variable GoalSeenInPast.

For state-based agent, the behavior is hard coded on line 224 of Brain.java file. The behavior of the state-based agent can be modified by providing any integer value as argument of the turn function.



I chose 90 assuming that agent may have seen goal just in the last cycle. Hence searching it with angle of 90 instead of default 40 will be quicker. I first tried with 180 degrees but that was resulting agent to flip multiple times before pointing to the goal. At line 224, it can be seen that agent checks it’s memory to determine if it saw goal in the past (i.e. previous cycle of while loop).

**Examples of editing the text file**

Text file and reactive agent mapping function is similar to the question 1 of this assignment. The functionality of the text file is still intact as in question 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Agent function in Text file** | BallSeenByPlayer | BallCloseToPlayer | GoalVisibility | **Action,Power/Angle,Direction** |
| F, Y, F -> Kick,100,goal | In **F**ront | **Y**es | In **F**ront | Kick,100,goal |
| N, U, F -> Turn,40,x | **N**o | **U**nknown | In **F**ront | Turn,40,x |

- Power/Angle means power OR angle and required when the Action is kick or dash. Otherwise, put x

- Direction is needed when Action is kick or turn. Otherwise, put x for direction.

**Running and testing the code**

Once theAssignment1Question2 folder is unzipped, run the provided batch file (Krislet/ TeamStart\_StateBasedAgent.bat). Text file parsing is performed at each cycle of the while loop in Brain.java, therefore, the behavioral changes of the agent can be seen on the fly while the game is in progress (in the while loop). To store the environment states, I created a new class EnvironmentStates (EnvironmentStates.java). Inside Brain.java while loop, the text file is parsed, and the function mappings are stored in a HashMap variable EnvStateToActionMapping. Keys of the HashMap are the environment states while values are the actions. To specifically test the state-based agent, follow the instructions under "Reasoning of State-Based agent" heading on previous page.

**State Machine Diagram and proof of state-based agent**

In state machine diagrams, event e is followed by actions α separated by / (e/α notation). The yellow highlights prove that the behavior is state-based. Had this been a reactive agent, for the N while loop cycle, the action would have been default Turn 40 degree instead of Turn 90 degree.

Diagram

Description automatically generated