**SYSC5013 Assignment 1, Question 1. Reactive Agent.**

**Introduction and mathematical notations**

To design an agent, we need to look at the characteristics of the environment. When something changes in the environment, the agent should do something about it. This means that the agent needs to be responsive or "reactive" to those changes in the environment. Mathematically, an agent function is denoted as: E-> AC where e1,e2,e3… ∈ E; and α1,α2,α3… ∈ AC.

**Environment and Actions**

The environment τ is also known as the state transformer function is denoted as: RAC ->E. For the reactive 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 reactive agent, the actions α1,α2,α3 are turn, kick or dash and can only be performed once/cycle.

[turn, kick, dash] ∈ Agent Actions.

**Text file and agent function mapping**

As explained in the beginning of this document, a reactive agent is E-> AC. The agent function is stored in an editable text file (textfile.txt) separate from the code. One should be able to modify the behavior of the agent by simply editing/saving the text file and without recompiling the code. The general from of agent function, i.e. the mapping between the environment state and the actions as defined in the text file is:

BallSeenByPlayer,BallCloseToPlayer,GoalVisibility->Action,Power/Angle,Direction

**Examples of editing the text file:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |
| F, Y, Y -> Turn,x,goal | In **F**ront | **Y**es | **Y**es | Turn,x,goal |
| N, U, F -> Turn,40,x | **N**o | **U**nknown | In **F**ront | Turn,40,x |

- All comments in the text file start with //.

- 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 theAssignment1Question1 folder is unzipped, run the provided batch file (Krislet/ TeamStart\_ReactiveAgent.bat). The user can save the changes in the text file prior to running the batch file or can make changes while the game is in progress. 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, values are the actions.

**Examples**: When the agent function is Y, Y, Y -> Kick,0,goal instead of Y, Y, Y -> Kick,100,goal in the text file, agent won't be able to kick the ball near goal as the power is 0. Similarly, F, N, Y -> Dash,5,x instead of F, N, Y -> Dash,100,x in the text file will slow down the agent.