**Running the Reactive Agent:** The “ReactiveAgent” folder should replace the demo Krislet folder. The agent is executed identically to Krislet (i.e., “java Krislet”). An agent’s behavior can be changed without recompiling the code by editing AgentSpec.txt.

**Environment Model:** For the agent, the ball is either “visible”, “directly in front”, or “not visible” and is either “far”, “close”, or “unknown” in terms of proximity to the agent. Likewise, the goal is either “visible”, “directly in front”, or “not visible” to the agent.

**Agent Specification Syntax:** An agent behavior is specified in the following form:

(<ball visibility>, <ball distance>, <goal visibility>) -> <action>:<power>:<direction>

Where <ball visibility> {Visible, NotVisible, DirectlyInFront}, <ball distance> {Close, Far, Unknown}, <goal visibility> {Visible, NotVisible, DirectlyInFront}, and <action> {Kick, Dash, Turn}. Along with the action output, a value(s) must also be specified that further describes the action performed. These are shown in the following table:

|  |  |
| --- | --- |
| **action** | **value** |
| Kick | The power of the kick (integer) and direction of the action (integer or one of {ball, goal}) |
| Dash | The speed of the dash (integer) |
| Turn | The direction the player turns (integer or one of {ball, goal}) |

If a behavior is not specified for a given environment the agent finds itself in, no action will be performed.

**Example Agent Behaviors:**

|  |  |
| --- | --- |
| **Behavior Rule** | **Description** |
| (DirectlyInView, Far, NotVisible) -> Dash:50 | When the ball is in the center of the agents view and far away, dash forward at 50 speed |
| (Visible, Far, Visible) -> Kick:100:goal | When the ball and goal are visible and the ball is far away, kick with 100 power in the direction of the goal. |

**Expected Behavior of the Provided Agent:** The agent specification provided in the submission is intended to behave similarly to Krislet. The agent will turn until it is in line with the ball, then dash towards the ball. When the agent is close enough, the agent will the kick the ball towards the net.

**Agent Design:** The reactive agent mapping is achieved with a HashMap. On startup, the AgentSepc.txt file is read and converted into a mapping of discrete environment states to actions. On each tick, the environment is discretized into an object that is used as a key into a HashMap to retrieve the necessary action.