Question 1) Path to download the code/binaries:

<https://drive.google.com/drive/folders/0B1IBCwXEnb-YYmZybmNLdmpBZEk?usp=sharing>

The agent function should be saved as a text file named “AgentFunction.txt” and placed in the binaries folder. A line that starts with a “;” is treated as a comment and therefore ignored by the parser. The agent function definition has 1 rule/mapping per line with the following format: [Environment State] -> [Action].

The environment states can be made of 1 or 2 object’s states whose formats are: [goal] and [ball,distance,direction]. “distance” can be “far” (>1.0) or “close” (otherwise) and “direction” can be “same” (0) or “different” (otherwise).

The actions format is as follows: [command, direction, power, wait]. “command” can be “dash”, “turn” or “kick”. “direction” can be any real number from -180.0 to 180.0. “power” can be any real number. “wait” can either be “wait” or ignored. If a field in the middle doesn’t need to be entered leave it empty e.g. [turn,,100]: “direction” is not needed and it is in the middle so it is left empty, “wait” is not needed but is at the end so it is just ignored, i.e. no comma is added.

* [ball,far,same] -> [dash,,10]: if the ball is far from the player and in the player has the correct direction, then the player dashes towards the ball at power ‘10’.
* [ball,far,different] -> [turn]: if the ball is far and the player player doesn’t have the correct direction, then the player turns in the correct direction.
* [ball,close,same]->[turn, 40,,wait]: if the ball is close and the player is in the correct direction but there is no info about the goal then the player turns by 40.0 and waits for an update from the environment.
* [ball,close,different]->[turn, 40,,wait]: if the ball is close and the player is not in the correct direction and there is no info about the goal then the player turns by 40.0 and waits for an update from the environment.
* [ball,close,same]|[goal] -> [kick,,100]: if the ball is close, the player has the correct direction and the player knows where the opponent’s goal is, then the player kicks the ball with power ‘100’ towards the opponent’s goal.
* [ball,close,different]|[goal] -> [kick,,100]: same as above except the player doesn’t have the right direction, nonetheless the player kicks the ball with power ‘100’ towards the opponent’s goal.
* [goal] -> [turn,40,,wait]: if we only have info about the goal, i.e. no info about the ball, the player turns by 40.0 and waits for an update from the environment.

The code was created by modifying mainly the “Brain” class of the “Krislet” program. The way it works is as follows: at the beginning of the “Run” function in the “Brain” class, the code parses the agent function definition in “AgentFunction.txt” and creates a table with all the rules, then while the game is on, i.e. between kickoff and the end of the game, every time an update is received by the agent it compares the current state with the states in each rule starting with the first one, and when a match is found the corresponding action is executed by the agent, otherwise the agent turns around by 40 and waits for the next environment update.

The provided code and agent function should give a behavior identical to the “Krislet” one.

To run the code, launch the server and the monitor, then go to the binaries subfolder in the path given above and execute the TeamStart.bat file just as in the provided “robocupDemo”.