Utrecht, 22-03-'10

Environment Interface Standard Documentation
MSAgentEnvironment

Author: Hagenaars, Ramòn

Introduction

This document describes the MSAgentEnvironment as released on the 22th of march 2010, conform the Environment Interface Documentation Policy by T. M. Behrens. For this document, knowledge of the Environment Interface Standard is required since terms like *Percepts* and *Entities* will not be explained here.

The MSAgentEnvironment has a folder called *"dll"* which contains a dll-file that is necessary to run the Microsoft agents. In order to make the environment find this library, it must remain in the same directory as the *msagentenvironment.jar*. Do not rename or move the jar-file or the dll-file.

Documentation

Environment description: The MSAgentEnvironment is a simple Environment which can be used to control Microsoft Agent characters.
Please note that in order to run this environment, you must be running a Windows platform and have installed the required drivers for the agents. If the drivers are not installed yet, please read the user guide which describes where to find these drivers.

Jar-file: *msagentenvironment.jar*.
It should contain the following folders: *META-INF, eis, msagent, msagentenvironment.*

Entities: There are 4 possible entities: $E=\{e_1, \ldots\}$ where e_x is unique; it is not possible to create more than one of any of these entities.

The following entities are supported:

- **Merlin :** a funny looking wizard, probably the most well-known MS agent.
- **Genie :** a ghost with an Arabic look.
- **Peedy :** a green parrot.
- **Robby:** a robot with a robot voice.

Each entity is of the *MSAgent* class and has one *Character* object attached with it. Each Character c has a predefined set of animations A .

Actions: We define the following actions and their arguments:

show(*identifier*)

Makes the MSAgent appear on the screen.
The identifier is to determine whether to execute the action animated or just plain. It should be either *“yes”* or *“no”*.

hide(*identifier*)

Makes the MSAgent disappear.
Again, the identifier determines whether to animate or not while hiding and should be “yes” or “no”.

speak(*identifier*)

Lets the MSAgent say a text.
The identifier is the text to speak.

think(*identifier*)

Lets the MSAgent think a text.
The identifier is the text to think.

gestureAt(*number* , *number*)

Lets the MSAgent gesture at a location on the screen.
The numbers represent x and y coordinates with respect to the screen.

moveTo(*number* , *number* , *number*)

Moves the MSAgent to a specific location on the screen. If the MSAgent is visible while executing this action, it will perform this action animated.
The first two numbers represent x and y coordinates with respect to the screen. The third argument determines how long it should take to move to the position.

perform(*identifier*)

Performs a specific animation, where the identifier is the name of the animation. Please see the Javadoc for the possible animations performed per Character.

All actions as well as their arguments are case insensitive.

Percepts: For each action you will receive a percept after the action was executed successfully. You will receive: $msaae(c)$, where $msaae$ stands for *MSAgentActionEvent* which could be one of the following:

- createavatar;
- detach;
- gestureat;
- hide;
- initial;
- moveto;
- playanimation;
- setlanguageid;
- show;
- speakaudio;
- speaktext;
- think.

Environment-management: The environment can be initialized by specifying the MSAgents to load. So for example, if only Genie and Robby should be loaded, the set of arguments would be something like $P = genie, robby$.

Note that if no arguments are set, the environment will load all four of the MSAgents.