

Assignment 7

SE1 Proseminar, winter 2007/2008

Stefan Resmerita

stefan.resmerita@cs.uni-salzburg.at

In this assignment, you are requested to achieve implementation of a simple scripting language and execution of script programs. The Command design pattern should be employed in your solution.

A sentence in the scripting language has one of the following forms:

1. `<time_step>: <variable> = "<command>"`
2. `<time_step>: <command>`

where:

- `<time_step>` is a natural number representing the simulation time when the statement after the semicolon must be executed.
- `<variable>` is an alphanumeric string without spaces.
- `<command>` is any of the commands currently supported in the system, with the corresponding parameters

A statement of type 1 means that the command starts execution at the specified time step, then the variable is assigned the string value corresponding to the output of the command *at the time step when the command finishes execution*. A statement of type 2 means that the command starts execution at the specified time step and the output is displayed at the console *when the command finishes execution*.

A parameter of a command may be of the form `$(variable)`. In this case, if the variable has been previously assigned a value, then `$(variable)` is replaced with that value before the command is processed. Otherwise, an error message should be displayed at the console.

A script (or a program) consists of a sequence of statements of the form 1 or 2, where each statement begins at a new line. Statements are interpreted in the order of increasing time steps.

To run scripts, implement the following command:

ExecuteScript `<filename>`

The parameter `<filename>` represents the absolute path of a file containing a program for the system.

Script example:

```
1: id_robot = "PowerOn TransportRobot"
1: id_team = "CreateTeam PaintingRobot=1"
2: ShowContents $id_team
3: AddToTeam $id_team $id_robot
5: ShowContents $id_team
8: TransportItem (3,9) (9,9) 4 horizontal
12: TransportItem (6,3) (10,11) 2 vertical
```

When executing the above script, the output should look like the following (assume that `id_robot` is 100 and `id_team` is 1000).

```
Current time: 2
Command: ShowContents 1000
Start time: 2
Output:
    Begin Team 1000
        PaintingRobot 101
    End Team 1000
```

```
Current time: 5
Command: ShowContents 1000
Start time: 5
Output:
    Begin Team 1000
        Painting Robot 101
        TransportRobot 100
    End Team 1000
```

```
Current time: 54
Command: TransportItem (6,3) (10,11) 2 vertical
Start time: 12
Output:
    Robot: TR1
    Initial coordinates: (0,0)
    Duration to the starting point: 18s
    Duration of the transportation: 24s
```

```
Current time: 96
Command: TransportItem (3,9) (9,9) 4 horizontal
Start time: 8
Output:
    Robot: TR2
    Initial coordinates: (0,0)
    Duration to the starting point: 48s
    Duration of the transportation: 40s
```

If there are statements with the same time step, they are executed in the order in which they appear in the input file. You may assume that the statements are syntactically correct and that the time step of a statement is not greater than the time step of any succeeding statement in the file. Note that the `TransportItem` command is the only command which takes time to execute. A `TransportItem` command may power on a transportation robot, if needed to execute the command.

The checking interface for this assignment should be the same as for the previous assignment.