Creating SNePS/Greenfoot Agents and Worlds
SNeRG Technical Note 46

Jonathan P. Bona and Stuart C. Shapiro
Department of Computer Science and Engineering
201 Bell Hall
University at Buffalo, The State University of New York
Buffalo, NY 14260-2000
{jpbona|shapiro}@buffalo.edu

May 3, 2011

Contents

Abstract

This technical note explains how to use Greenfoot scenarios as worlds for SNePS-based agents.

1 Introduction

The Greenfoot framework is a pedagogical tool designed for teaching introductory object-oriented programming in Java to beginning students [Kölling, 2009]. Greenfoot runs on Java and is self-contained, even including its own integrated development environment. Normal use of Greenfoot involves creating scenarios consisting of a World (subclass of greenfoot.World) and Actors (subclasses of greenfoot.Actor), and modifying those Java classes to acheive the desired behavior for the scenario. The Greenfoot Tutorial [1] takes as an example a simple "Wombats" scenario – a 2D grid-based World with such Actors as Leaves and Wombats that move around the grid eating Leaves.

The subject of this document is an interface that allows code external to the Greenfoot scenario to manipulate the scenario. One example might be a Java program that starts a Greenfoot scenario and calls the public methods for that scenario's World or Actors. Because Greenfoot was not created for this type of use, it does not directly support such manipulation. We have created a Java class that is derived from the class responsible for booting Greenfoot in an exported scenario (`greenfoot.export.Main`). Our modified class `greenfoot.export.SNGFMain` (for SNePS/Greenfoot Main) includes a method that returns a reference to the World object for the scenario. Thus any Java program that has this class in its classpath can use it to start executing a scenario and to access the World and Actors in it.

We developed this interface to allow the use of Greenfoot scenarios as virtual worlds in which SNePS agents can operate. This is currently implemented by using Allegro Common Lisp's JLinker[2] to allow a SNePS agent running in Allegro to access Greenfoot scenarios. Though other configurations are possible, we have focused on the creation of SNePS agents that each manipulate a single Greenfoot Actor within a scenario, with the Actor implementing the agent's basic capabilities that allow it to perceive and act in the World.

2 SNePS/Greenfoot Scenarios

A SNePS/Greenfoot scenario consists of the following parts:

1. A Greenfoot scenario. The scenario should include a (public) subclass of `greenfoot.Actor` that implements the agent's body in the world. This class should expose as public methods the agent's basic capabilities. Greenfoot can "export" a scenario as an executable jar file that contains everything necessary to run the scenario.

2. A copy of `greenfoot.export.SNGFMain` in the classpath.

3. Lisp code that sets up Allegro's JLinker and initializes the Greenfoot scenario, and Lisp functions corrresponding to the Actor's public methods that the agent needs to access.

4. A SNePS implementation of the agent's mind, connected to the JLinker Lisp functions. We use agents that follow the GLAIR Cognitive Architecture [Shapiro and Bona, 2010], which divides the agent implementation into a gradation of layers, and makes the connection between the Lisp portion and the Java portion of the agent in the lower Perceptual Motor Layer.

[1] Greenfoot Tutorial: `http://www.greenfoot.org/doc/tutorial/tutorial.html`, last accessed 30 March 2011

[2] `http://www.franz.com/support/documentation/8.2/doc/jlinker.htm`

3 Tutorial Scenario

This section deals with a simple tutorial scenario that illustrates the basic structure of a Greenfoot scenario for use by a SNePS agent. The scenario ("SampleWorld") is a 2x3 grid with a simple Actor (to be controlled by SNePS) that can move forward, turn left or right, place a ball in its current cell, sense the presence of a ball in its current cell, and pick up a ball that is in its current cell. If the user clicks on an empty cell, a ball will be added in that cell. The scenario is shown in **Figure 1** below.

Figure 1: The SampleWorld scenario

3.1 Location

The directory `/projects/robot/Greenfoot/Tutorial/` on the UB CSE servers contains all files for the SampleWorld scenario. In the rest of this section, we refer to this directory as `$TUT_ROOT`.

3.1.1 Greenfoot Scenario

- `$TUT_ROOT/scenario-source/` contains the source (and compiled) version of the scenario, which can be opened, viewed, modified, and recompiled within Greenfoot.
- `$TUT_ROOT/SampleWorld/SampleWorld.jar` is the executable exported version of the scenario.

3.1.2 SNGFMain Class

- `$TUT_ROOT/SampleWorld/sneps-gf.jar` contains `greenfoot.export.SNGFMain`, which allows control of Greenfoot scenarios from external code.

3.1.3 SNePS SampleAgent

`$TUT_ROOT/SampleWorld/sneps` (henceforth `$SNEPS`) contains the SampleWorld SNePS agent's SNePSLOG knowledge layer implementation and Lisp PML implementation, as well as files required by the JLinker:

- `$SNEPS/SampleAgent.snepslog` contains the SNePSLOG implementation of the agent's knowledge layer. It contains a single ActPlan, `operate`, which causes the agent to turn randomly left or right, move forward and sense for the presence of a ball. If a ball is present in the current cell the agent removes it. Otherwise the agent places a ball in the cell. Finally it performs operate again.
- `$SNEPS/pmla.cl` contains the Lisp implementation of the agent's efferent primitive actions, e.g. `turn`, `goForward`, etc.
- `$SNEPS/pmls.cl` contains the Lisp implementation of the agent's sensory act `see`.
- `$SNEPS/pmlb.cl` contains Lisp functions that are called by the PMLa and make the connection to the Java/Greenfoot world. This file is also responsible for setting up JLinker's classpath and calling `jl-config.cl`.

- `$SNEPS/jl-config.cl` configures the JLinker.
- `$SNEPS/jlinker.jar` contains classes that make up the JLinker library.

3.2 Running SampleWorld Agent and Scenario

To run the SampleWorld agent and scenario on timberlake:

- Change to the directory `/projects/robot/Greenfoot/Tutorial/SampleWorld/sneps`:
  `cd /projects/robot/Greenfoot/Tutorial/SampleWorld/sneps`
- Start SNePS in Allegro CL. The preferred way to run SNePS in Allegro involves the use of emacs, as described here: `http://www.cse.buffalo.edu/sneps/instructions.html`.
  Another way is to run Allegro (`/util/bin/composer`) and load SNePS within Allegro:
  `cl-user(1): (load "/projects/snwiz/bin/sneps")`
- Start SNePSLOG:
  `cl-user(2): (snepslog)`
- Run the agent:
  `: demo SampleAgent.snepslog`
- After the agent has loaded, the scenario should appear in a new window. Click the `Run` button.
- Tell the agent to begin operation:
  `: perform operate().`
  The robot should begin moving around the world, placing and removing balls in accordance with the agent's `operate` ActPlan.
- To end the agent's operation, close the Greenfoot scenario window using the small `x` in the upper right corner. This may produce an error on the Lisp side, as the agent will continue to try accessing the Greenfoot scenario. This error may be ignored. Return to the Lisp top level using the appropriate command in your environment.

3.3 Loading SampleWorld in Greenfoot 2.0.1

To load the SampleWorld scenario in Greenfoot on `timberlake.cse.buffalo.edu` or another CSE server:

1. Launch Greenfoot: the latest version of Greenfoot is in `/util/bin/greenfoot`. Assuming `/util/bin` is in your path, you may run Greenfoot simply by typing `greenfoot &` at your shell prompt. For more information on running Greenfoot, see `https://wiki.cse.buffalo.edu/services/content/greenfoot`.
2. Load the scenario: click on `Scenario` in the menu at the top of the Greenfoot window, and select the menu option `Open...` [3]
3. Navigate to the the directory `$TUT_ROOT/scenario-source/` in the file chooser dialog and select `SampleWorld`. Click the `Open` button. The scenario should appear as shown in **Figure 2** below.

   Note that running the scenario at this point will have no effect, as the Bot in the scenario requires the SNePS SampleAgent to be connected in order to operate. **Section 3.4.3** describes how to modify the scenario in a way that will cause it to do some things independently of the agent.
4. Close the scenario - by clicking the `x` in the upper right corner of the scenario, or by clicking `Scenario` in the menu bar and selecting `Close`) - before moving on to the next section.

[3] The first time a user loads Greenfoot on a system, it may present a welcome menu with the option to "Choose a scenario." Selecting this option is one alternative way of completing this step.

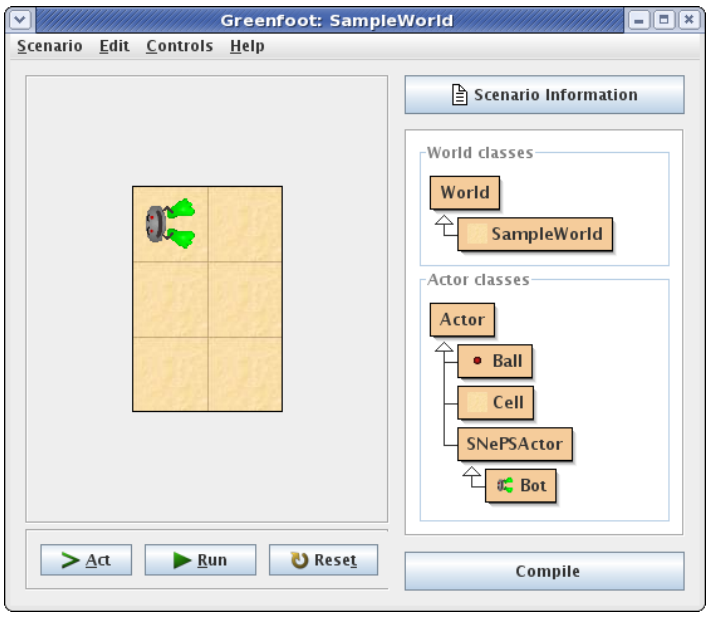

Figure 2: The SampleWorld scenario in Greenfoot

3.4 Modifying SampleWorld Agent and Scenario

This section describes how to create a new directory that contains all the pieces necessary to modify and run the SampleWorld Greenfoot scenario and SNePS agent.

3.4.1 Obtaining Local Copy

1. Create a new directory for the SampleWorld scenario and agent files in your home directory, or in another location to which you have write access. In the rest of this section, we refer to this directory as `$HOME_TUT`

2. Copy the directory `$TUT_ROOT/scenario-source/` into `$HOME_TUT`

3. Copy the directory `$TUT_ROOT/SampleWorld` into `$HOME_TUT`. This will copy `sneps-gf.jar` and the `sneps` subdirectory, which are required to run the scenario and the agent, respectively. It will also copy the subdirectory `javadocs` and the exported scenario `SampleWorld.jar`.

4. Load the scenario following the instructions in **Section 3.3** but using `$HOME_TUT/scenario-source/` rather than the the one in `$TUT_ROOT`.

   Note that Greenfoot keeps track of which scenario was last opened and may automatically open that scenario when Greenfoot starts. Since this section deals with an identical copy of the SampleWorld, this behavior may be especially confusing.

3.4.2 Running Local Version

To run your local version of the SampleWorld agent and scenario, first complete the instructions in **Section 3.4.1** above.

The SNePS agent should run in this new location without modification to any of its files, including those files that set up the classpath for JLinker, because the classpath in `pmlb.cl` uses relative paths, and following

the above instructions maintains the scenario and interface jar files in the correct locations relative to the other files.

The steps to run your local version of the SNePS agent and scenario are identical with the steps to run the version in `/projects/robot/Greenfoot/Tutorial`, given in **Section 3.2**, *except for the location of the files.*

- Change to the directory `$HOME_TUT/SampleWorld/sneps`
- Start SNePS in Allegro CL
- Start SNePSLOG
- Run the agent: `: demo SampleAgent.snepslog`
- After the agent has loaded, the scenario should appear in a new window. Click the `Run` button.
- Tell the agent to begin operation: `: perform operate().` The robot should begin moving around the world, placing and removing balls in accordance with the agent's `operate` ActPlan.
- To end the agent's operation, close the Greenfoot scenario window using the small `x` in the upper right corner.

3.4.3 Modifying the Scenario

The scenario can be modified by altering source code or changing other properties within greenfoot itself. This section describes how to make a simple change to the scenario: we add a new type of ball and code that makes it appear at random and disappear after a set number of time steps.

- Start Greenfoot and load your local copy of the scenario.
- On the right edge of the Greenfoot window is a class browser that shows the classes in the scenario. Right-click on the `Ball` class and select the menu option `New subclass...`
- Enter "DBall" (for disappearing ball) as the new class name and click the `OK` button. The `DBall` class will appear in the class hierarchy under `Ball`
- Double-click the `Cell` class. Its source will appear. Locate the `act` method:

```
public void act()
{
    // if the mouse was clicked on this cell, add a ball here
    if(Greenfoot.mouseClicked(this)){
        getWorld().addObject(new Ball(), getX(),getY());
    }
}
```

  Every `greenfoot.Actor` has an `act` method that is called once for every time step in Greenfoot. The `act` method in `Cell` adds a `Ball` to the cell if the user clicked on that cell since the last time step.

- Add the following to the `act`:

```
// Add a DBall 1% of the time
 if(Math.random() < .01){
     getWorld().addObject(new DBall(), getX(),getY());
 }
```

This will cause each cell to add a disappearing ball about 1% of the time.

- Return to the main window for the scenario (optionally, close the `Cell` class window first) and double-click on the `DBall` class in the main window.
- Add a variable to `DBall` that will store the number of time steps before it disappears:

```
private int life = 50; // the ball remains for 50 time steps
```

- Modify `act`, which should be empty initially, to decrement the life counter and remove the `DBall` from the world once the counter reaches 0:

```
public void act()
{
    if(life-- < 1){
        getWorld().removeObject(this);
    }
}
```

- Return to the main window for the scenario and click the `Compile` button to compile the scenario. Click the `Run` button to test the addition of randomly-appearing disappearing balls. Note that the Bot will not move since its `act` method is empty. Balls should appear and disappear. Clicking the `Pause` button at any time will stop the scenario.
- Follow the instructions in **Section 3.4.4** below and **Section 3.4.2** above to export the modified scenario and run the agent in it.

3.4.4 Exporting Scenario

The simplest way to connect SNePS to a Greenfoot scenario through the JLinker is to export the scenario to a jar file that is then place in the JLinker classpath. This section describes that process.

Note that the scenario must be re-exported whenever it has been modified and those modifications need to be part of the version that the SNePS SampleWorld agent uses.

1. With the scenario open and compiled, click on `Scenario` in the menu at the top of the Greenfoot window, and select the menu option `Export...`
2. Click on the `Application` tab to export the scenario as an application.
3. Click on the `Browse` button to select a location for the exported jar. Select the directory `$HOME_TUT/SampleWorld` and name the file `SampleWorld.jar`. Click `Choose` to confirm the file name and location, and to return to the `Greenfoot: Export` window.
4. Click `Export` in the `Greenfoot: Export` window to write the exported scenario jar file. When the export has completed, the message "Export complete" will be displayed. Click `Close` to leave the `Greenfoot: Export` window.

3.5 Scenario Class Documentation

`/projects/robot/Greenfoot/Tutorial/SampleWorld/javadocs/index.html` contains javadoc-generated documentation for classes in the SampleWorld scenario.

3.5.1 Regenerating Javadocs

To produce updated javadocs after changing the scenario:

1. Make sure you have exported the changed scenario jar file to the location listed in **Section 3.4.4** above.
2. Change to the directory `$HOME_TUT/SampleWorld/javadocs`
3. Execute the following command:

```
javadoc -link http://download.oracle.com/javase/1.5.0/docs/api/
        -link http://www.greenfoot.org/doc/javadoc/
        -classpath ..:../SampleWorld.jar
        ../../scenario-source/SampleWorld/*.java
```

3.6 Creating a New SNePS/Greenfoot Scenario

Though the scenario discussed above does not have elements that are necessarily specific to environments for use by SNePS agents, it does have some features that are used by all currently-existing SNePS/Greenfoot agents. For instance, the `SampleWorld` class is a 2D grid-based world with fixed-size cells and private static members that determine the size of the grid, and it fills each location with an `Actor` (`Cell`, in this case). The `SNePSActor` class, which is a superclass of `Bot` in this scenario, contains standard method useful for navigating in a 2D grid world: `moveForward`, `turnLeft`, etc. Any scenario built for similar agents may and, probably should, initially make use of these classes.

The Greenfoot website contains many tutorials and other documentation that are helpful when building a new scenario. One recommended basic introduction is at `http://www.greenfoot.org/doc/intro1.html` The Greenfoot tutorial (`http://www.greenfoot.org/doc/tutorial/tutorial.html`) is more comprehensive.

Greenfoot comes packaged with some sample scenarios, including the wombats scenarios used in the basic introduction. Sample scenarios for Greenfoot 2.0.1 are located on CSE systems in the directory `/util/greenfoot-2.0.1/scenarios`.

References

[Kölling, 2009] Kölling, M. (2009). *Introduction to programming with Greenfoot*. Prentice Hall.

[Shapiro and Bona, 2010] Shapiro, S. C. and Bona, J. P. (2010). The GLAIR cognitive architecture. *International Journal of Machine Consciousness*, 2(2):307–332.