Eastern Finland Virtual University (ISVY)

Jeliot 3 - User Guide

Who, Where, and What of Jeliot – The Algorithm Theatre

Hello, User

This is the user guide for Jeliot 3 – the Algorithm Theatre. It answers questions like "*What does this button do?*" and "*How do I ask user for value in Jeliot?*", and acts as the most specific source of information on usage of Jeliot.

Professor: "As I often say, what you don't find elsewhere, you can find here. What you don't find here, you can go find somewhere else."

If you are looking for quick start installation and animation instructions, take a look at document titled *Jeliot 3 - Quick Tutorial.* In case you are looking for general guidelines and examples on how to use Jeliot , take a look at *Jeliot 3 - Animation Tutorial.* Everything else you may need should be in this document.

Here we could have some nice drawing, maybe even saying a comment if someone can think of a comment for this bad case of noncommentable place. Ugh.

Contents

1 Introduction

Jeliot is a program animation system intended for teaching introductory programming. Programs are animated *automatically*, requiring no modifications or annotations on the part of the user. While this limits the flexibility of the animation, Jeliot is extremely simple to use so that it is easily accepted by true novices, as well as by their teachers who do not have to invest on learning how to prepare animations.

comment: “It’s so simple even I can use it!”

comment: “use jeliot in the classroom to show things, no matter what language you use!”

Jeliot has been written in Java to gain maximum portability. It animates programs written in Java, but this does not restrict its use solely on courses that deal with Java. It has been successfully used on courses dealing with other programming languages, e.g. PASCAL. For more information on how to use Jeliot in class room, see [1].

2 History (?)

3 Release Notes (?)

4 Installation

Professor: “For quick start, see the quick start guide. However, ’Speed is nothing, understanding is everything”’ :P

Student: “Source code is also available. Don’t touch it! :)”

Here are the instructions for downloading and installing Jeliot 3. A quicker set of instructions, meant for users that do not need so detailed information can be found from *Jeliot 3 - Quick Start Guide*. Before you start, you should check that you have any of the Java environments of version 1.4 or newer installed. To verify whether you have it not, look for command `java.exe` or `java`. If it exists, you have Java (You still need to know the version. Try running `java -version` on commandline, it should tell you the version). Check `http://java.sun.com` for information on how to install Java.

The following installation instructions are split into two parts. First one deals with installation trough *Jeliot windows installer*, second with *Jeliot executable distribution*. Select the one you wish to use and proceed according to appropriate set of instructions. Windows users are suggested to select the *windows installer* version.

If you wish to actually use Jeliot, and not play around with the way it is created, you should pick either of the above mentioned distributions. However, for the ones interested, *Binary distribution* is also available. Documentation for source code is included in the package. You need to rely on that.

4.1 Installation with Windows Installer

These instructions describe the installation process of Jeliot with the *windows installer*. This is the best way for Windows users, as it is the easiest way to install, and it creates shortcuts to Start-menu and deskop for easy running of Jeliot.

4.1.1 Downloading

Go to Jeliot download-page at `http://www.cs.joensuu.fi/jeliot/downloads.php` and click on *Jeliot 3, windows installer*. Your browser should prompt you for running/saving the file. Here you should select "save", although it is possible to open the file directly from the web. However, we suggest you save it to disk, should the first installation fail.

The file you get is named in the following format: `Jeliot3-N.exe`, where N is the version number. E.g., version 2 preview 3 would be in a file called `Jeliot3-2P3.exe`.

4.1.2 Installation

Run the `Jeliot3-N.exe` you just downloaded by double clicking on it. A welcome window should appear (fig. 1(a)). Click Next. The next window prompts you for the directory you wish to install Jeliot in (fig. 1(b). Give a directory, or leave it at the default suggestion. Click Next. Next, you are prompted for the Start-menu folder to install Jeliot in (fig. 1(c)). Give a name for the folder, or leave it at the default suggestion. Click Next. From the next view, you can select whether you wish to have shortcuts created (fig. 1(d)). Select atleast the Start-menu)and click Next.

Window in figure 1(e) summarizes your selections. Click Install to start installation (fig. 1(f)). The final window just tells us that the installation is complete. Click Finish to exit installation. Now you are ready to start Jeliot!

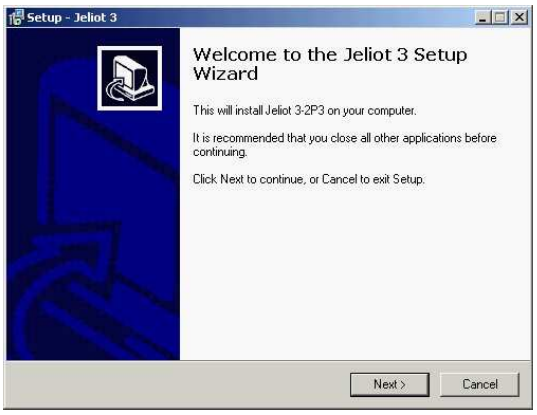
4.1.3 Running

If the installation with the Windows Installer finished correctly, and you selected the options for creating the Start-menu shortcuts, you should have Jeliot available in your Windows Start menu under folder with the name you entered during the installation. Depending on your selections, you might also have a shortcut on your desktop. Just select either one, and Jeliot should fire up. If the installer failed to insert the shortcuts, you can start Jeliot by doubleclicking on `Jeliot.bat` in the directory you installed Jeliot into. If you encounter problems, check the running instructions of *executable distribution* for help.

Now Jeliot should be running on your screen looking like the one in figure 2.

4.2 Installation of Executable Distribution

This set of instructions should guide you through the installation of Jeliot under Windows, Linux, *nix or any other operating system that has a working Java environment. In case of the more extreme OSs, we trust that


Setup - Jeliot 3

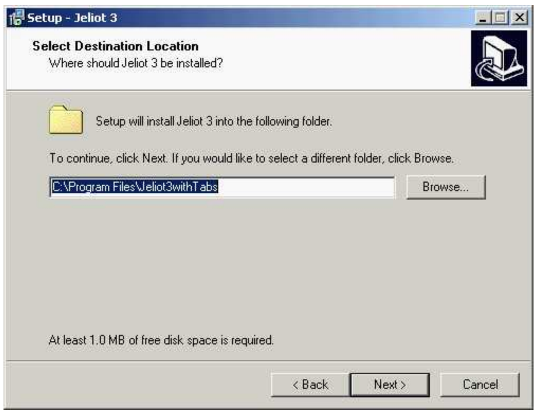
Welcome to the Jeliot 3 Setup Wizard

This will install Jeliot 3-2P3 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

(a) Window 1


Setup - Jeliot 3

Select Destination Location
Where should Jeliot 3 be installed?

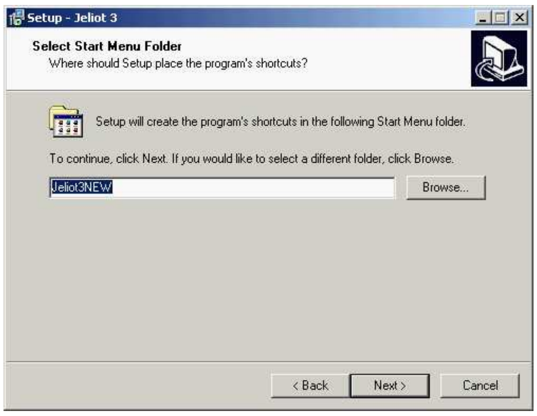
Setup will install Jeliot 3 into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Jeliot3withTabs

At least 1.0 MB of free disk space is required.

(b) Window 2


Setup - Jeliot 3

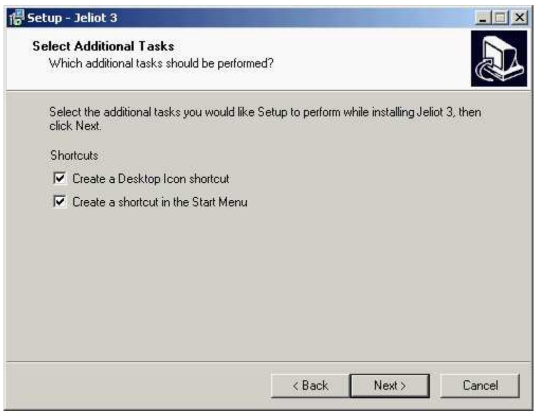
Select Start Menu Folder
Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Jeliot3NEW

(c) Window 3

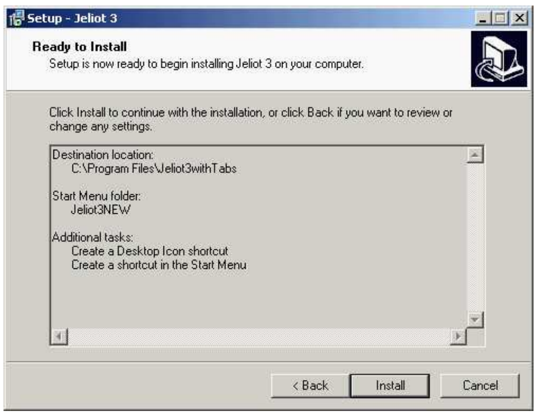

Setup - Jeliot 3

Select Additional Tasks
Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Jeliot 3, then click Next.

Shortcuts
- [✓] Create a Desktop Icon shortcut
- [✓] Create a shortcut in the Start Menu

(d) Window 4


Setup - Jeliot 3

Ready to Install
Setup is now ready to begin installing Jeliot 3 on your computer.

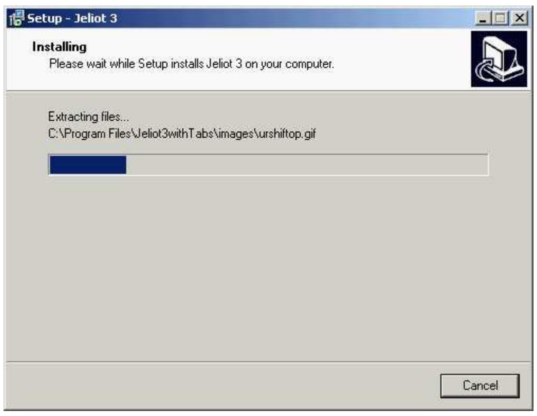
Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Destination location:
    C:\Program Files\Jeliot3withTabs

Start Menu folder:
    Jeliot3NEW

Additional tasks:
    Create a Desktop Icon shortcut
    Create a shortcut in the Start Menu
```

(e) Window 5

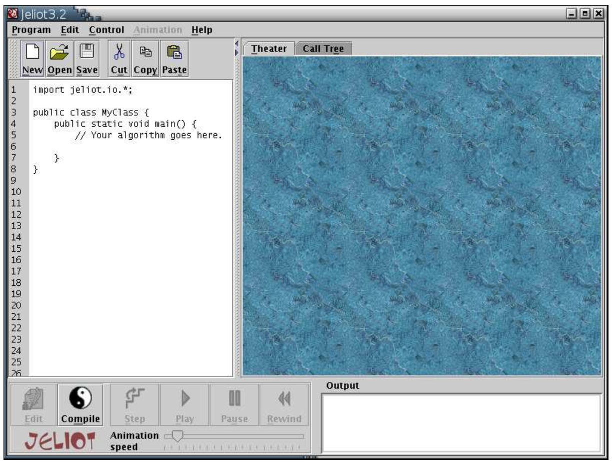

Setup - Jeliot 3

Installing
Please wait while Setup installs Jeliot 3 on your computer.

Extracting files...
C:\Program Files\Jeliot3withTabs\images\urshiftop.gif

(f) Window 6

Figure 1: Screenshots of the installation process



```
import jeliot.io.*;

public class MyClass {
    public static void main() {
        // Your algorithm goes here.

    }
}
```

Figure 2: First view of Jeliot.

you know what you're doing. In case there is difference between what you are supposed to do in OSs, Windows users should check option (1) and others option (2).

4.2.1 Downloading

Go to Jeliot download-page at `http://www.cs.joensuu.fi/jeliot/downloads.php` and click on *Jeliot 3 executable distribution (zipped file)*. Your browser should prompt you for opening/saving the file. Here you should select "save", although it is possible to open the file directly from the web if your system is properly configured. For the sake of generosity, we suggest you save the file to disk.

The file you get is named in the following format: `Jeliot3-N.zip`, where N is the version number. E.g., version 2 preview 3 would be in a file `Jeliot3-2P3.zip`.

4.2.2 Uncompressing

Uncompressing the file you have just fetched from the web will result in few files and some subdirectories. First, you should create a directory for Jeliot, for example `c:\jeliot\` (Windows) or `/home/user/jeliot/` (Linux/*nix) depending on your operating system and access to it. Any folder or directory is ok, as long as you know where it is. Your unzipping program might also be able to create a directory for the files.

Next, decompress the `Jeliot3-N.zip` to the previously created directory using

1. WinZip, or some similar tool in Windows (doubleclick on the `Jeliot3-N.zip` file and select "extract" from the program. Select the directory that you just created as the target.
2. command `unzip Jeliot3-N.zip` in Linux/*nix environment. Note that this command will uncompress the files to the directory you are in, and will need the file `Jeliot3-N.zip` in the same directory. Specify the full path to uncompress the packet from elsewhere, and use -d option to specify the target directory.

Now you should have Jeliot uncompressed on your disk. To verify, you can check that files `jeliot.jar`, `jeliot.bat`, `jeliot.ico`, `license.txt` and subdirectories `images`, `examples` and `docs` exist in the directory you created.

4.2.3 Running

To run Jeliot,

1. in Windows, you can use `jeliot.bat`. Double click on it the start Jeliot. It might be a good idea to create a shortcut for `jeliot.bat` with `jeliot.ico` icon.
2. in Linux/*nix, you should give the command `java -jar jeliot.jar` in the directory where Jeliot was extracted. Actually, this is exactly what `jeliot.bat` does.

Now you should have Jeliot running on your screen. Figure 2 shows the startup state of Jeliot.

5 Graphical User Interface

Jeliot 3 is always used through its graphical user interface (GUI). A screenshot of the Jeliot GUI is shown below in figure 3. In this chapter we go through the different parts of the GUI, one by one, describing their meaning and behavior.

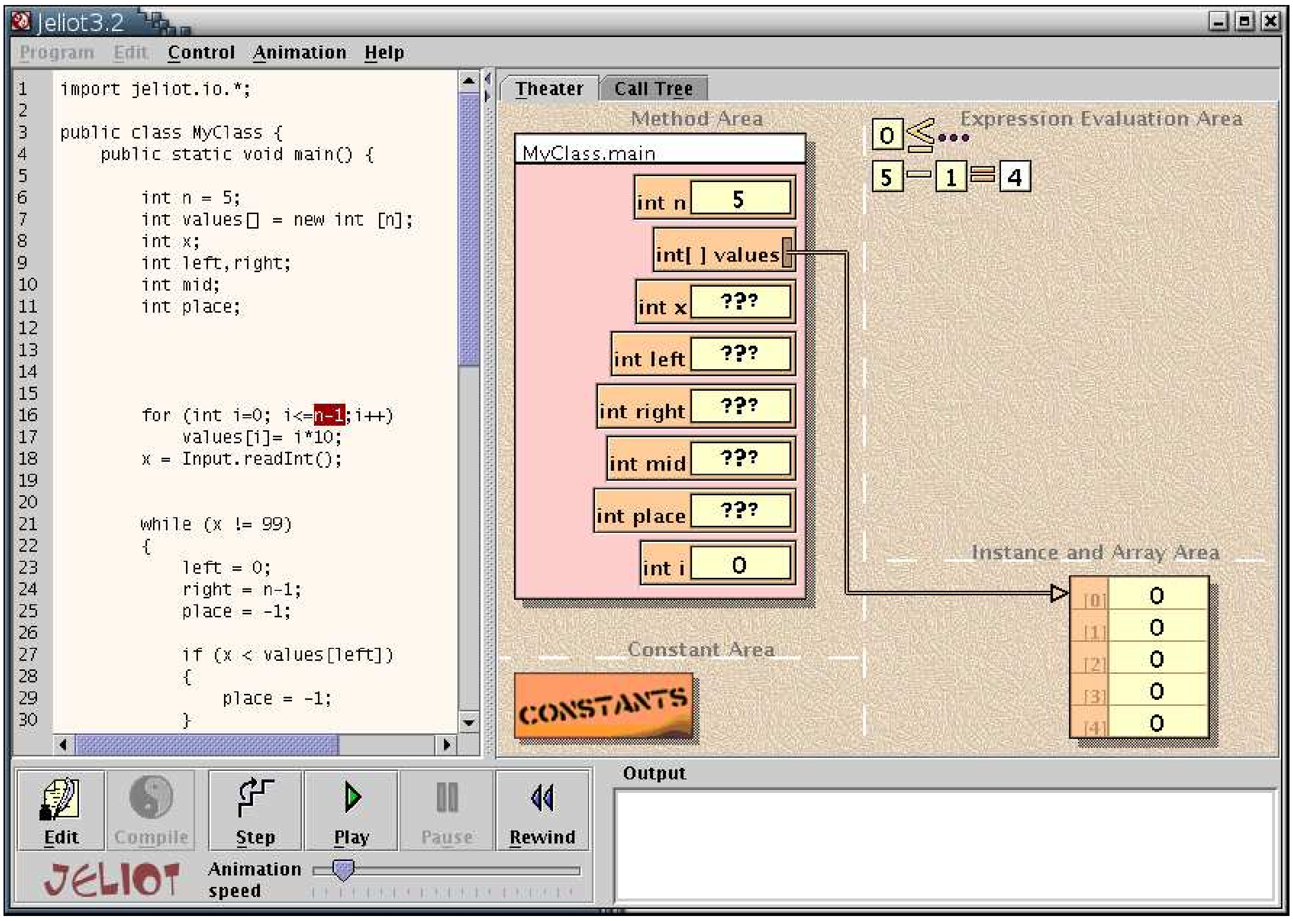
5.1 Source Frame

The frame on the left hand side of the screen, consisting of linenumbered text-area, is called the source frame. When you open or create a new file, it will be opened to the source frame for editing. Clicking on Compile will remove the editing toolbar on the top, switch the source frame to uneditable mode and slide open the curtains on animation frame. After animation, clicking on Edit will return the source frame to editing state.

During animation, the code that is currently being animated is highlited in the source frame. For example, while in figure 3 the animation frame shows `n-1` being calculated, the same operation is highlited in the source frame. Another example: While entering a loop, animation frame gives a message about it, and source frame will highlite the whole area of loop. You will get the hang of it when you see it happening.

5.2 Animation Frame

Animation frame is the main area of Jeliot. This is where all the interesting animation takes place. While in editing state, it is covered with a blue curtain. When you move to animation state, the curtain slides open and reveals a light brown background. When you start the animation, the frame is divided into four separate areas with dashed white lines. The areas in left-right, top-bottom order are *Method Area*, *Expression Evaluation Area*, *Constant Area*, and *Instance and Array Area*.



```
import jeliot.io.*;

public class MyClass {
    public static void main() {

        int n = 5;
        int values[] = new int [n];
        int x;
        int left,right;
        int mid;
        int place;



        for (int i=0; i<=n-1;i++)
            values[i]= i*10;
        x = Input.readInt();


        while (x != 99)
        {
            left = 0;
            right = n-1;
            place = -1;

            if (x < values[left])
            {
                place = -1;
            }
```

Figure 3: Screenshot of Jeliot GUI.

Method Area holds activation frames for all the methods that are currently being processed. When there is nothing left in *Method Area*, there is nothing left to animate. Activation frames are displayed as boxes, that hold variables inside, drawn as subboxes. Return values are animated with a larger box holding the value inside. For variables of primitive or `String` type, the value is displayed adjacent to the name, others are shown as links to *Instance and Array Area*, or electrical ground symbol in case they are `null`.

Expression Evaluation Area is just what the name says, area for evaluationg expressions. Whatever instruction needs to be executed, it is always displayed here. Information on the results of evaluting the expressions are also shown here, as well as the dialog boxes for user input.

Whenever any literals are needed by the code, they are brought to the animation from the *Constants box* from *Constant Area*. This applies for all literals, no matter what type they are.

Finally, the *Instance and Array Area* holds dynamically reserved objects, such as instances of classes and arrays. These are connected to activation frames in *Method Area* by links.

Objects in the Animation Frame follow the following color coding:

Table 1: Color legend for animation frame.

Object	Color	Object	Color
Methods	Pink box with white header	Long	Light Green
Floats	Purple	Strings	Red
Integers	Brown	Doubles	**What is this color?**
Chars	Green	Class Objects	Yellow

5.3 Call Tree

Call tree displays the executed function calls and their return values as a tree, `className.main()` being the natural root node. You can view the call tree by clicking on Call Tree tab at the top of animation frame. The call tree is then displayed at the place of animation. Switching back to animation happens by clicking on Theater tab. You can switch between these displays whenever you wish. However, dialog boxes will only appear on theater display. You should remember this, if you wish to follow execution from call tree view. Example of Call Tree is shown in figure 4.

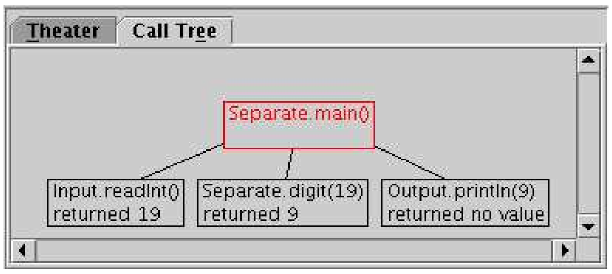


Figure 4: Example view on call tree.

5.4 Menus

Menus in Jeliot are quite self-explanatory. Program menu holds commands for file operations new, open, save, exit and Edit menu holds the clipboard functions cut, copy, save and select all. Quick summary of what these commands do is shown in table 2. You can find most of the commands also in the Toolbar on top of the source frame.

Control menu holds Edit and Compile commands. Animation menu holds command for controlling the flow of animation. Again, most of the commands in these two menus can be found from the toolbar at the bottom of the screen (figure 6) . Check the meanings of common commands from sections 5.6 and 5.7.

In addition to the commands that are also located in the toolbar, the animation menu holds two important commands: Pause on message and Run until.... The first is a checkbox, that will stop animation on any message (such as *"continuing for loop"*) if it is set. The second will ask for a linenumber, and set a breakpoint there. When execution comes to this line, it is paused.

Note that Animation menu is disabled while you are in editing state. In the same way Program and Edit menus will be disabled in animation state.

Table 2: File and Edit operations.

Command	Description
New	Inserts an empty template to source frame.
Open	Opens a dialog for selecting a file to open to source frame.
Save	Saves the code in source frame to given file.
Exit	Exits from Jeliot.
Cut	"Cuts" the selected text away from source frame, and places it on the clipboard.
Copy	Copies the selected text from source frame to clipboard.
Paste	Puts the content of clipboard to the source frame, starting from the row where the cursor is.
Select All	Selects everything in source frame.

Last menu is Help. There you can find information on Jeliot (About) and the Help document. Help document is opened on its own window when you select it from the menu, or press F1 on your keyboard.

5.5 Toolbar Buttons

While in edit state, there is a detachable[1] toolbar with file and clipboard commands on the top of the source frame (figure 5). These work in the same way as the commands in the Program and Edit menu do. When you compile your code, the toolbar is hidden beneath the source frame, or just disabled in case you have detached it and left it in its own window.

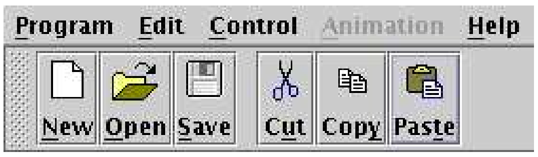


Figure 5: Program menus and File and Clipboard operations toolbar.

5.6 Edit and Compile

The toolbar shown in figure 6 has buttons called Edit and Compile. It is located in the lower left corner of the window. With these buttons you control whether Jeliot is in animation or editing state. From the editing state, you can start animating by clicking on Compile. You can return to edit the code at any time from animation state by selecting Edit.

5.7 Animation Controls

The animation can be controlled by the VCR-like buttons (figure 6) on the lower right corner of the Jeliot window. Have a look at the information on Animation menu in section 5.4 for additional control features.

[1] *detachable* meaning a toolbar that can be drawn out of its place and relocated elsewhere by dragging it with a mouse.

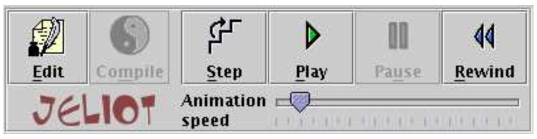


Figure 6: Edit, Compile and Animation Control toolbar.

Table 3 describes the animation controls on the toolbar.

Table 3: Animation toolbar commands.

Button	Function
Step	Proceeds the animation with one step with the set speed.
Play	Proceeds the animation continuously with the set speed untill Pause is clicked or the program ends.
Pause	Pauses running animation to situation when you click the button.
Rewind	Takes you back to the beginning of program.
Animation Speed	Tune the speed of animation. Left is slower, right is faster. You can also control the speed in steps from the Animation menu.

5.8 Output Frame

Output frame is located in the lower right corner of the screen. As you can see in figure 7, it is just a white textbox, that shows the data that is outputted from the animated program. All the values that are outputted are grabbed from the animation frame by a hand coming out of the box.

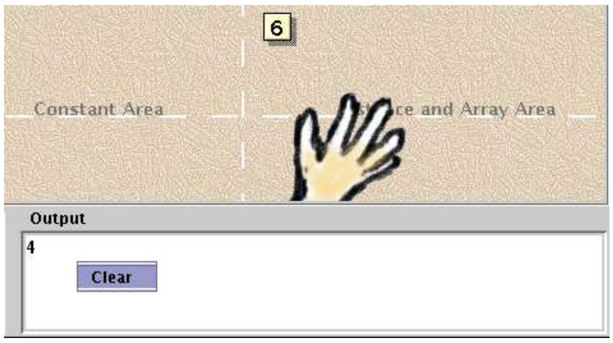


Figure 7: Output box, clear option in menu and value-grabbing hand.

Clicking on output frame gives you a menu with single selection for clearing the screen. You can use this option at any time to erase everything in the output box.

5.9 Error Display

In case there is an error in the code, in most cases Jeliot will tell you about it when you click on Play or Step for the first time. If you are trying to use an uninitialized variable, the animation will proceed normally until the point of error, where it is stopped with en error message.

A notification on error will be displayed on animation frame. Clicking OK will take you back to animation screen, but you cannot proceed with the animation untill you correct the faulty code. That is still your job, Jeliot will not (yet) do that for you.

6 Java Issues

There are two incompatibilities between Jeliot and Java.

1. All classes must be in a single source file.
2. For I/O, import the package `jeliot.io.*;` which provides the methods `void Output.println()`,`int Input.readInt()`,`double Input.readDouble()`, `char Input.readChar()`, `String Input.readString()`. Standard output is also already supported.

Jeliot uses DynamicJava (`http://koala.ilog.fr/djava/`) as a front-end and thus accepts almost all Java features that you would want to use for introductory programming, however, the implementation of the animation might not animate all features. Currently, the implementation includes:

- Values of type `String`, all primitive types and one-dimensional arrays.
- Static variables.
- Expressions including all unary and binary operations except `instanceof`.
- All the control statements (`if`, `while`, etc.).
- Conditional expressions (`exp?exp1:exp2`).
- Method invocation, including recursive invocation.
- Constructors, allocation of objects and invocation of methods on objects.

Not implemented:

- Super field accesses.
- Arrays with components of reference type (except `String`)
- Two or more dimensional arrays.
- Array initializers.
- Java 2 SDK API classes' methods cannot return object (except `String` type) or array types (e.g. `object.getClass()` that returns a `Class` instance).
- The used classes hashCode() -method has to return always a unique value.

References

[1] Ronit Ben-Bassat Levy, Mordechai Ben-Ari, and Pekka A. Uronen. The Jeliot 2000 program animation system. *Computers & Education*, 40(1), 1-15, 2003.

Document License (?)