



SOFTWARE COMPREHENSION AND MAINTENANCE

Project Report: Simredo 4: A java Unicode Editor



Alexandros Charos
Department of Management Science and Technology
Athens University of Economics and Business
Summer 2005

Project report

Simredo 4

Index:

1	Introduction.....	p 2
2	Simredo 4: Overview.....	p 2
3	Changing Simredo into a new version.....	p 6
3.1	Changing Cases.....	p 7
3.1.1	toLowerCase()/toUpperCase().....	p 8
3.1.2	toggleCase().....	p 9
3.1.3	Implementing the new methods.....	p 10
3.2	The Copy Bar.....	p 11
3.3	The Progress Bar.....	p 14
3.4	Simredo 4 in Greek (a.k.a. Lost in translation)	p 16
3.5	Overview of the changes.....	p 17
4.	Contributing to the open-source community: Conclusions.....	p 17
	Appendix: Communication with the development team.....	p 19

1 Introduction

The open source community was always something I admired. When I started to work on this course I had a lot of ideas. Open source projects that everyone uses (Firefox), open source projects that help information exchange (Azureus, DC++), open source projects that make developers' lives easier (Eclipse, JUnit) and many other popular examples were my first inspirations. Unfortunately, not only the people working on these projects are more than the KLOCs themselves, but I had difficulties even thinking of a feature that was not already implemented. So I decided to look for a less popular project that I could actually contribute and not just study its code.

Creating a text editor myself was something that I had been looking to do for a long time. With Java being the only language that not only have I been taught but also have studied by myself, I had no doubts on which language to use. Additionally, being a macintosh user, cross-platform programs were something I cherished.

Despite of what I thought, the project I liked the most was not on sourceforge.net. I was quite amazed when I discovered that the project that excited me the most was not registered on sourceforge. The editor I found was a unicode editor. It was written 100% on Java and I enjoyed reading its code. I edited it mainly on a PC (Windows XP sp2) using Eclipse and tested it on PC and Macintosh (Mac OS X v. 10.3.9).

2 Simredo 4: Overview

Quoting the Simredo website:

“Simredo is a free Java Unicode editor. Its features include:

- * Easy installation for GNU/Linux and Windows
- * Limited support for styled text documents (RTF-format)
- * The ability to convert over 100 encodings to and from Unicode
- * A flexible key map feature, for redefining the keyboard layout
- * Support for right-to-left languages such as Arabic, Persian, etc.

- * User interface in English, Esperanto, French, or Hindi
- * A spell checker for English and Esperanto
- * An encryption function”

It is obvious that Simredo is something more than a simple text editor.

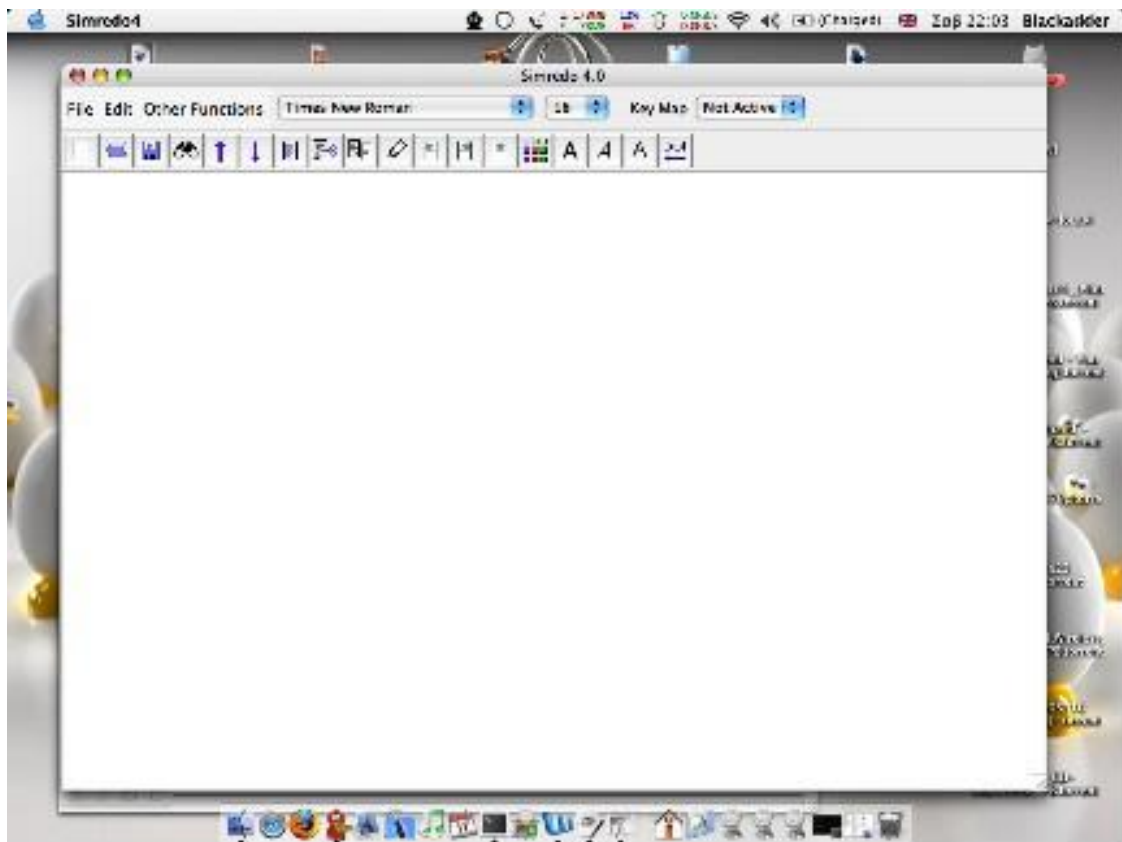


Figure 1 Simredo before changes

Simredo 4 class overview

Simredo 4 consisted of about 50 classes. The class hierarchy before my acts was as follows:

- class java.lang.Object
 - class javax.swing.AbstractAction (implements javax.swing.Action, java.lang.Cloneable, java.io.Serializable)
 - class [SearchDialog.SearchDown](#)
 - class [SearchDialog.SearchUp](#)
 - class [Simredo4.ChangeDirection](#)
 - class [Simredo4.ChangeLanguage](#)
 - class [Simredo4.Exit](#)
 - class [Simredo4.NewFile](#)
 - class [Simredo4.OpenFile](#)
 - class [Simredo4.PrintIt](#)
 - class [Simredo4.RedoAction](#)
 - class [Simredo4.ReverseText](#)
 - class [Simredo4.SaveFile](#)
 - class [Simredo4.SaveFileAs](#)
 - class [Simredo4.SearchDownx](#)
 - class [Simredo4.SearchUpX](#)
 - class [Simredo4.ShowChange](#)
 - class [Simredo4.ShowChars](#)
 - class [Simredo4.ShowColourSelectorDialog](#)
 - class [Simredo4.ShowConvertDialog](#)
 - class [Simredo4.ShowKeyboard](#)
 - class [Simredo4.ShowSearch](#)
 - class [Simredo4.ShowSpellDialog](#)
 - class [Simredo4.SimIconAction](#)
 - class [Simredo4.ToggleKeyMap](#)
 - class [Simredo4.UndoAction](#)
 - class java.awt.Component (implements java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable)
 - class java.awt.Container
 - class javax.swing.JComponent (implements java.io.Serializable)
 - class javax.swing.JPanel (implements javax.accessibility.Accessible)
 - class [Simredo4](#) (implements java.awt.event.ActionListener, javax.swing.event.DocumentListener, java.awt.event.KeyListener, java.awt.event.MouseListener, javax.swing.event.PopupMenuListener)
 - class javax.swing.text.JTextComponent (implements javax.accessibility.Accessible, javax.swing.Scrollable)
 - class javax.swing.JEditorPane
 - class javax.swing.JTextPane
 - class [SimTextPane](#) (implements java.awt.event.ActionListener, java.awt.print.Printable)
 - class javax.swing.JTextArea
 - class [SimTextArea](#)

- class javax.swing.JTextField (implements javax.swing.SwingConstants)
 - class [SimTextField](#)
- class java.awt.Window (implements javax.accessibility.Accessible)
 - class java.awt.Dialog
 - class javax.swing.JDialog (implements javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants)
 - class [ChangeDialog](#) (implements java.awt.event.ActionListener)
 - class [ConvertDialog](#) (implements java.awt.event.ActionListener)
 - class [ErrorDialog](#) (implements java.awt.event.ActionListener)
 - class [ExistsDialog](#) (implements java.awt.event.ActionListener)
 - class [FileChangedDialog](#) (implements java.awt.event.ActionListener)
 - class [LangDialog](#) (implements java.awt.event.ActionListener)
 - class [OpenConvertDialog](#) (implements java.awt.event.ActionListener)
 - class [OpenKeyDialog](#) (implements java.awt.event.ActionListener)
 - class [SaveConvertDialog](#) (implements java.awt.event.ActionListener)
 - class [SaveKeyDialog](#) (implements java.awt.event.ActionListener)
 - class [SearchDialog](#) (implements java.awt.event.ActionListener)
 - class [ShowCharsDialog](#) (implements java.awt.event.ActionListener)
 - class [SpellDialog](#) (implements java.awt.event.ActionListener)
 - class [TimedMessage](#) (implements java.awt.event.ActionListener)
 - class java.awt.Frame (implements java.awt.MenuContainer)
 - class javax.swing.JFrame (implements javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants)
 - class [ShowKeyboardDialog](#) (implements java.awt.event.ActionListener)
- class javax.swing.plaf.ComponentUI

- class javax.swing.plaf.TextUI
 - class javax.swing.plaf.basic.BasicTextUI (implements javax.swing.text.ViewFactory)
 - class javax.swing.plaf.basic.BasicEditorPaneUI
 - class javax.swing.plaf.basic.BasicTextPaneUI
 - class [SimTextPane.SimTextUI](#)
- class java.awt.event.WindowAdapter (implements java.awt.event.WindowFocusListener, java.awt.event.WindowListener, java.awt.event.WindowStateListener)
 - class [Simredo4.Closer](#)
- class [Detect](#)
- class [EngDictFlags](#)
 - class [EngDict](#)
- class [FileInformation](#)
- class [IsciiConverter](#)
- class [IsciiMappings](#)
- class [PageDivider](#) (implements java.awt.print.Pageable)
- class [SimCon](#)
- class [SimIran](#)
- class [UserKeyMap](#)
- class [Simredo4.KeyMapFilter](#) (implements java.io FilenameFilter)
- class [Simredo4.SimUndoableEditListener](#) (implements javax.swing.event.UndoableEditListener)

The main class is (obviously) `Simredo4`. This class is responsible for everything that happens inside `Simredo4` and also coordinates the rest classes. I understood the project's structure as soon as I studied this class.

It's also worth mentioning that for internationalization purposes `Simredo4` comes with a `Simredo4.properties` file that helps a developer interested in localizing `Simredo4`. The project has a total of about 15 KLOCs.

3 Changing Simredo into a new version

In this chapter I intend to show what my exact changes in `Simredo4` were and the reason I worked the way I did. It is important to point out that my initial thoughts of changes I intended to do were quite different than what I succeeded in doing.

A text editor is a kind of project that gives one a variety of possible changes. Everyone

has used text editors and everyone can think of a possible new feature that could make editing easier. So, the first idea I had was to make the editor identify an HTTP address (e.g. <http://foo.bar>) and make it a hyperlink. In spite of my impression that developing a class (or even a method) that would do such a thing could be easy, it turned out to be time – consuming and I was progressing really slow. So I had to drop this idea and implement other ones.

The changes I finally made were the following:

1. Changing cases: Gives the user the ability to change the case of the whole document to:
 - a. Upper Case
 - b. Lower Case
 - c. Toggle Case: meaning that the case of every line within the document toggles
2. The copy bar: a panel on the left of the document being edited that keeps the user's 10 last copies to the clipboard
3. Loading bar: a simple loading bar that shows the user the state of Simredo's initiation process
4. Localization: translation of the menu to Greek

My total changes to the project are the following:

4 new classes created
8 new methods added in existing classes
400 LOCs approximately added

A more detailed description follows description follows:

3.1 Changing cases

Adding an option that makes it easier for a user to switch case on his document is something underestimated in my opinion on the most popular document editors (Microsoft's Word, Apple's Xcode), so I decided to add it to Simredo. Here is how I did it:

3.1.1 toLowerCase()/ toUpperCase()

SimTextPane is the class that controls the text pane of the end user's document. I added two new methods within this class called toLowerCase() and toUpperCase(). What they actually do is to take the current document, transform it to lower or upper case, remove it from the pane and insert the new one.

Lets see how toLowerCase() works with more detail:

```
public void toLowerCase() {  
  
    Document doc = getDocument();  
    int length = doc.getLength();
```

By here we have stored the document's length in an integer variable called length.

```
    try {
```

I use "try" so that possible exceptions can be caught

```
        StringBuffer sb = new StringBuffer(doc.getText(0, length));  
        String stringToConvert = sb.toString();  
        stringToConvert = stringToConvert.toLowerCase();
```

I store the sting I want to convert inside a buffer, and read from the buffer.

```
        doc.remove(0, length);  
        doc.insertString(0, stringToConvert, null);
```

I remove everything from the current document and add the new string (stringToConvert)

```
    } catch (Exception e){  
        System.err.println("Exception : " + e.getMessage());  
    }  
}
```

Possible exceptions will have been caught by here.

The code demonstrated above shows the way the method toLowerCase() worked. The same applies for toUpperCase()

3.1.2 toggleCase()

The way that the method toggleCase() works is a bit different than the other two methods related to “changing cases”. What I had to do here was to check every character one by one and substitute it with its opposite case. It is obvious that a more complicated algorithm is needed for the toggleCase() method:

```
public void toggleCase() {  
  
    Document doc = getDocument();  
    int length = doc.getLength();  
  
    try {  
  
        StringBuffer sb = new StringBuffer(doc.getText(0, length));  
        String stringToConvert = sb.toString();
```

The first part of the method apparently is the same as the other two.

```
String conversion = "";
```

I create a new String variable called “conversion”. Its role is to store there one by one the characters that their case is toggled.

```
for (int i=0; i<length; i++) {
```

with this loop I make sure that I have scanned every possible character of the document.

```
    if  
    ((int)stringToConvert.charAt(i)==(int)(stringToConvert.toLowerCase().charAt(i))) {  
what the above line does is : if a character is lower case...
```

```
        conversion+=stringToConvert.substring(i,i+1).toUpperCase();
```

```
    }
```

...change it to upper case and add it to our depository(conversion).

```
else if  
((int)stringToConvert.charAt(i)==(int)(stringToConvert.toUpperCase().charAt(i))) {
```

```
    conversion+=stringToConvert.substring(i,i+1).toLowerCase();
```

The same thing applies if the character is upper case

```
        else {  
                                conversion+=stringToConvert.substring(i,i+1);  
        }
```

If it is a different kind of character, just add it to our depository.

```
    }
```

```
    doc.remove(0, length);
```

```

        doc.insertString(0,conversion, null);
    }
    catch (Exception e){System.err.println("exception" + e.getMessage());
    }
}

```

The final lines are the same as the other two methods.

3.1.3 Implementing the new methods

Since the methods do what we want them to do, we have to place them in a class that will actually call them. The new classes are called ToggleCase, UpperCase and LowerCase and they all use the same template:

```

public class LowerCase extends AbstractAction {
    LowerCase(String default_label) {
        super(default_label);
    }
    public void actionPerformed(ActionEvent e) {
        text_pane.toLowerCase();
    }
}

```

With text_pane being the current SimTextPane running.

The next thing to do is to link my new classes with the main one. Simredo4.

```

action_hash.put("lower-case", new LowerCase("lower-case"));
action_hash.put("upper-case", new UpperCase("upper-case"));
action_hash.put("toggle-case", new ToggleCase("toggle-case"));

```

This way I add my new classes to the hash table that organizes all the actions. My final step is to define “lower-case”, “upper-case” and “toggle-case” on the Simredo4.properties file.

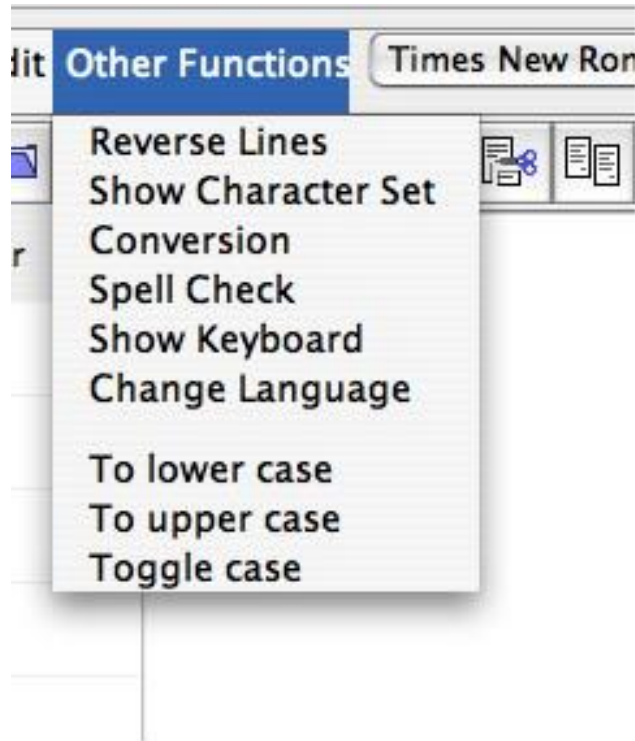


Figure 2 The case-changing menu

3.2 The Copy Bar

Having a place where one's previous copies are placed was the next idea I had about Simredo.

The first thing to do was to create a method that get the Clipboard contents.

```
public String getClipboardContents() {
    String result = "";
    Clipboard clipboard = Toolkit.getDefaultToolkit().getSystemClipboard();
    Transferable contents = clipboard.getContents(null);
    boolean hasTransferableText = (contents != null) &&
    contents.isDataFlavorSupported(DataFlavor.stringFlavor);
    if ( hasTransferableText ) {
        try {
            result =
            (String)contents.getTransferData(DataFlavor.stringFlavor);
        }
        catch (UnsupportedFlavorException ex){
            System.err.println("Exception : "+ex.getMessage());
        }
    }
}
```

```

        catch (IOException ex) {
            System.err.println("Exception : "+ex.getMessage());
        }
    }
    return result;
}

```

My getClipboardContents() method returns the String contained within the system clipboard

```

private JPanel copy_panel = new JPanel(new BorderLayout());
public Thread copyThread;

public JLabel copyTitle = new JLabel();

public JTextArea copyArea[] = new JTextArea[10];{
    for (int i = 0;i<10;i++){
        copyArea[i] = new JTextArea(4,8);
        copyArea[i].setEditable(false);
        copyArea[i].setLineWrap(true);
    }
}

```

I initiate the copy bar by creating a panel that will hold it and the text areas that will keep the last copies. I also create a thread that will be used to control the checks on the clipboard.

I enter the following within the Simredo4 class:

```

copyTitle.setText(copybar_text);
copy_panel.add(copyTitle);

GridLayout gly = new GridLayout(11,1, 1,1);
copy_panel.setLayout(gly);
copy_panel.add(copyArea[0],BorderLayout.WEST);
copy_panel.add(copyArea[1],BorderLayout.WEST);
copy_panel.add(copyArea[2],BorderLayout.WEST);
copy_panel.add(copyArea[3],BorderLayout.WEST);
copy_panel.add(copyArea[4],BorderLayout.WEST);
copy_panel.add(copyArea[5],BorderLayout.WEST);
copy_panel.add(copyArea[6],BorderLayout.WEST);
copy_panel.add(copyArea[7],BorderLayout.WEST);
copy_panel.add(copyArea[8],BorderLayout.WEST);
copy_panel.add(copyArea[9],BorderLayout.WEST);
Thread copyThread = new Thread(this);
copyThread.start();
add("West", copy_panel);

```

The control for the clipboard's contents are control from the thread's run() method. The thread runs as long as Simredo runs and it sleeps for 1 second. The clipboardCounter variable is increased by one every time the clipboard content is different than the last text copied to the copy bar and is being reset as soon as it reaches 10. The full algorithm is shown below:

```

public void run() {
    int clipboardCounter = 0;
    String lastText = "";
    while (true) {

        try {
            if (!lastText.equals(getClipboardContents())) {
                lastText = getClipboardContents();
                copyArea[clipboardCounter].setText(lastText);
                clipboardCounter++;
            }
        } catch (IllegalStateException ex)
        {
            System.out.println("IllegalStateException. View Bug ID: 4475489.");
        }
        if (clipboardCounter==10) {
            clipboardCounter=0;
        }
        try {Thread.sleep(1000);}

        catch(InterruptedException e)
        {
            System.out.println("interrupted");
        }

    }
}

```

At this point it is interesting to explain the reason I entered a “catch” for `IllegalStateException`.

Older versions of java, and coincidentally the version of the computer that I was testing the project (1.3.1) throw this kind of exception as a result of the bug 4475489 (more info on http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=4475489). This bug is supposedly in newer releases of JDK.



Figure 3 The copy bar

3.3 The progress bar

A progress bar indicating the state of Simredo when it is loading was something missing. It was the first thing I noticed when I ran Simredo for the first time. Adding it was quite simple since all I needed (from my point of view) was a new thread and a `JProgressBar`. The class creating the instance is named `ProgressShow` and its role is twofold: firstly it shows a frame on the user's screen thus assuring him/her that Simredo is starting up. Secondly it updates the loading bar whenever a large portion of the loading period is done. The code is:

```

import javax.swing.*;

public class ProgressShow extends JFrame {
    static JProgressBar current;
    JLabel out;
    Thread runner;
    private static int num = 0;
    static JPanel pane = new JPanel();

    public ProgressShow(int numb) {

        super("Simredo 4");

        setSize(200,80);
        out = new JLabel("Simredo4 is Loading...");
        current = new JProgressBar(0,numb);
        current.setValue(num);
        current.setStringPainted(true);
        pane.add(current);
        pane.add(out);
        setContentPane(pane);
        setBounds(400,400, 200,80);
        setVisible(true);
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
    }

    public void stop() {
        setVisible(false);
    }

    public void iterate(int numm) {
        current.setValue(numm);
        try {
            Thread.sleep(1000);
        }catch (Exception e ) {
        }
    }
}

```

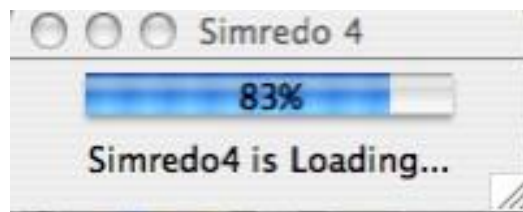


Figure 4 The progress bar

The iterate() method is the one being called by Simredo4 class after each stage of the loading process.

3.4 Simredo 4 in Greek (a.k.a. Lost in Translation)

Localizing Simredo was the first thing that the main developer advised me when on our first contact but it never was my first priority. Despite that, I translated Simredo in Greek. The only thing worth mentioning here is that I had to enter the Unicode version of each Greek letter in the Simredo4.properties, something that made my work less exciting and more time-consuming. The result was the following:

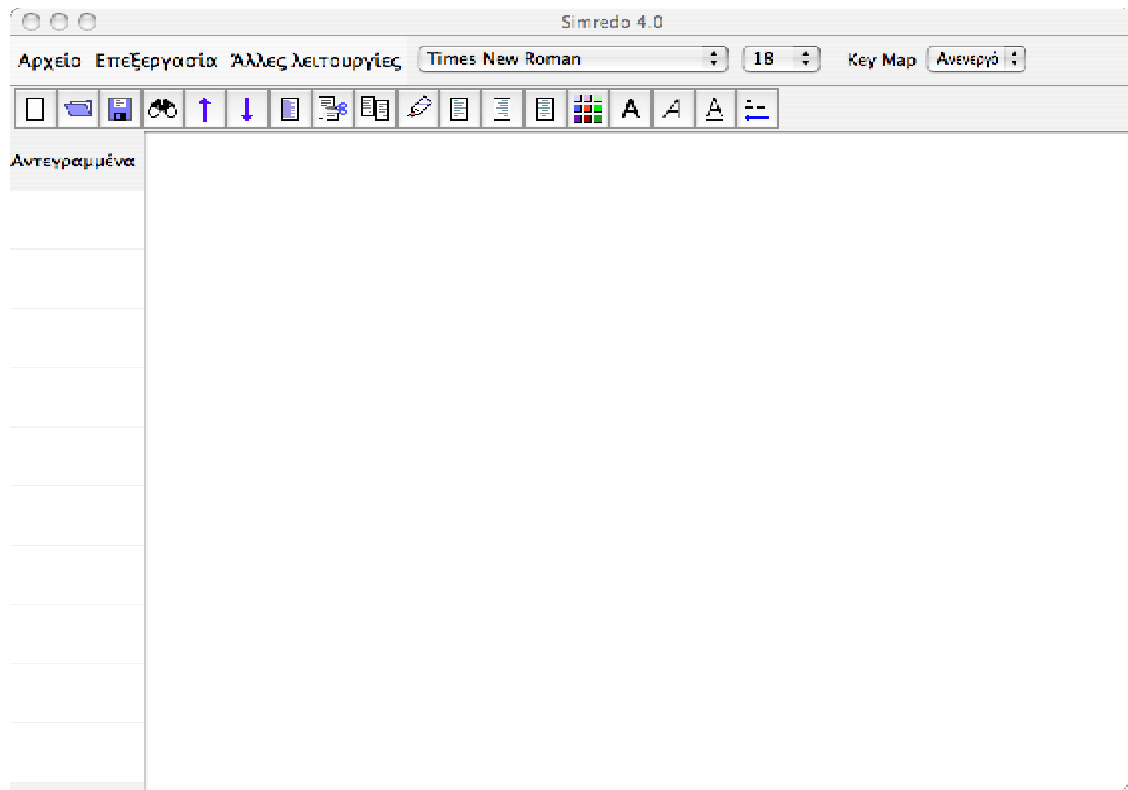


Figure 5 The Greek version of Simredo4

3.5 Overview of the changes

Simredo's new features make Simredo, in my opinion, user-friendlier and even more, an environment that thinks more like the end user does. Having to remember "where that last text that I copied before is now?" should not be the user's responsibility. The change-case menu may not be a feature that will be used on an every day basis by the final user, but still, implementing it can't harm anyone. Finally, the progress bar I added is definitely a must-have on programs that need some time to initiate.

To sum up, the changes implemented in my new version of Simredo, somehow change this project from a Unicode text editor to a document editor. RTF support, an already existing feature, also shows that. I do not hesitate to say that my changes can easily be considered as changes that take Simredo to a "beta" stage. Processing time, for example, is something that has to be looked up, but nonetheless, I consider my changes to take Simredo a step further than backwards.

4 Contributing to the open-source community: Conclusions

When I decided on the project I wanted to work on, I thought that there would be slight chances that the project administrator would even bother replying to e-mails I might sent him. To be honest, I was quite amazed when the project administrator not only replying within one day to all of my e-mails, but was also eager to read my code and suggesting new ideas. As said before, I always admired the open-source community but I had the impression that each developer minds his own work and is reluctant to help a newbie in his community. The truth is the exact opposite. The way the project administrator always responded to my e-mails helped me a lot when some ideas of mine weren't progressing.

Deciding to participate in Simredo's development was easy. Making it happen was the tough part. Since I had never participated in a project as big as Simredo (15 KLOCs approximately), my view on software developing was limited on writing the code, or even better, concluding to a suitable algorithm. Studying the project's code enriched that view. Software developing is much more than just that. So many aspects are included that I had never thought about before.

Despite of what one may thinks, the area where this project helped me the most was time management. Even though I didn't have a specific time frame to complete this project, I understood the importance of managing your time correctly inside a large project. Features that I intended to add and may have been more useful than the loading bar, for example, would have taken up a large portion of my time and at the bottom line, I had not guarantee that they would work properly. So, time is what I had to manage better than developing skills, code-reading skills and communication skills.

On a final conclusion on my "Software Comprehension and Maintenance" project, I found it quite interesting and exciting. My first experience with the open-source community helped me understood things I hadn't before and clearly improved my developing skills in general. I am already planning to implement ideas that I couldn't before on Simredo but also taking part in other projects as well.

APPENDIX : Communication with the development team

The main developer of Simredo 4 is Klivo Lendon. My correspondence with him is as follows:

Date:
Sat, 30 Jul 2005 02:51:49 -0700 (PDT)
From:
"Al-x- Haros" <realtowz@yahoo.com>
Subject:
simredo 4.0 project
To:
indriko@yahoo.com

Mr Klivo,

I am writing this e-mail due to your java unicodeeditor, Simredo 4.0 that I downloaded from your website.

I am an undergraduate student from the department of Management and Technology (<http://www.dmst.aueb.gr/>) of the University of Economics and Business of Athens of Greece.

A course that I have chosen is called "Software Comprehension and Maintenance" (<http://www.dmst.aueb.gr/dds/ismr>). The assignment I have to do is to change or add a new feature to an open-source project. I have chosen Simredo and I intend to make the following changes:

1. I will add an UPPER CASE, a lower case and a tOGgLE cAsE feature.
2. I will make the editor highlight and make an HTTP address clickable (something like a link)
3. I will create a panel where the 10 last clipboard copies will be held and the user may use them.
4. I am thinking of creating an HTML editor interface.

It would be really helpful if you checked my changes and made any suggestions for adding other features to your program,

Thank you in advance,

Alexandros Charos

Date:

Sun, 31 Jul 2005 04:18:57 -0700 (PDT)

From:

"Klivo Lendon" <indriko@yahoo.com>

Subject:

Re: simredo 4.0 project

To:

"Al-x- Haros" <realtowz@yahoo.com>

>Mr Klivo,

Hello Mr. Charos,

'Klivo' is actually the Esperanto form of my given name, 'Cleve'. If you wish to be formal, you can call me Mr. Lendon, but Klivo or Cleve, without the Mr., is quite acceptable to me.

> I am writing this e-mail due to your java
>unicodeeditor, Simredo 4.0 that I downloaded from your
>website.

>I am an undergraduate student from the department of
>Management and Technology(<http://www.dmst.aueb.gr/>)
>of the University of Economics and Business of Athens
>of Greece.

> A course that I have chosen is called "Software
>Comprehension and Maintenance"
>(<http://www.dmst.aueb.gr/dds/ismr>). The assignment I
>have to do is to change or add a new feature to an
>open-source project. I have chosen Simredo and I
>intend to make the following changes:
>1. I will add an UPPER CASE, a lower case and a
>toggle
>case feature.
>2. I will make the editor highlight and make an HTTP
>address clickable (something like a link)
>3. I will create a panel where the 10 last clipboard
>copies will be held and the user may use them.
>4, I am thinking of creating an HTML editor
>interface.

They sound like interesting projects, but you might want to reconsider that last one. When I started working on Simredo 4, I wanted to support HTML. I was able to load and save HTML-files, but there were so many problems with Java's HTML editor kit, that I had to give up. I found out that the support for RTF, although not complete, was at least reliable.

>It would be really helpful if you checked my changes
>and made any suggestions for adding other features to
>your program,

I would be glad to.

I assume you are Greek. Have you considered translating the interface to Greek? I would be very willing to include such a translation in a future release. Of course, translating isn't a lot of fun. It's just a thought.

>Thank you in advance,

>Alexandros Charos

Regards,

Cleve
(or Klivo)

Date:

Sun, 31 Jul 2005 12:57:41 -0700 (PDT)

From:

"Al-x- Haros" <realtowz@yahoo.com>

Subject:

Re: simredo 4.0 project

To:

"Klivo Lendon" <indriko@yahoo.com>

Dear Cleve,

It seems like I will implement all the suggestions on my last email and I will leave the HTML editor thought last. As you said, translating Simredo to greek is just simple work so I think I will do it. My professor also suggested to add a greek keymap and I am considering implementing this idea.

You will have news from me as soon as I have a version of simredo that I judge as good.

Thank you,

Alex Charos

PS. you can call me Alex

Date:
Wed, 17 Aug 2005 16:52:14 +0100 (BST)
From:
"Al-x- Haros" <realtowz@yahoo.com>
Subject:
Re: simredo 4.0 project
To:
"Klivo Lendon" <indriko@yahoo.com>

Dear Klivo,

I have just finished the following:

1. I have added under the "other functions" menu the options "upper case, lower case, toggle case" so that the whole document can be converted to the appropriate case
2. I have added on the right of the Simredo4 program a panel that "remembers" the user's last 10 copies.
3. I have translated Simredo4 to greek

I think I will give up trying to make an http address, i am thinking of adding a "loading" bar at the beggining of Simredo4. maybe even adding a greek dictionary..

Are the above enough to be implemented on simredo's next version?

thanks in advance

Date:
Thu, 18 Aug 2005 07:41:19 -0700 (PDT)
From:
"Klivo Lendon" <indriko@yahoo.com>
Subject:
Re: simredo 4.0 project
To:
"Al-x- Haros" realtowz@yahoo.com

Hello Alex,

You work fast!

I certainly would like to include your Greek translation in the next release (4.1). Feature number 2 sounds very useful, but I'd like to try it out before deciding whether to include it. I'm a bit hesitant about the case change feature.

A loading bar would be a very good feature. Also, I'd love to include a Greek spelling dictionary, if you have time for that.

I should say a couple things about the "license". Simredo has an "I don't care" license. :-) That means as far as I'm concerned, people can use the program and the source anyway they like.

If this is acceptable to you, I will gladly include some of your features into the next release of Simredo.

If you don't like the idea of giving hard work away, you are quite free to distribute (or even sell) your own version of Simredo. In the past, a couple people have done this.

I'm planning to release 4.1 next January. Please send me your enhancements, if you are willing to have them included in the next release. I will, of course, mention your contribution on the 'Thanks' page.

Cheers,

Klivo

Date:
Thu, 18 Aug 2005 16:08:09 +0100 (BST)
From:
"Al-x- Haros" <realtowz@yahoo.com>
Subject:
Re: simredo 4.0 project
To:
"Klivo Lendon" <indriko@yahoo.com>

Dear Klivo,

by now i have added the loading bar but stopped trying to make a greek dictionary since i had some encoding problems.

I would surely like to incude my contibutions to the next simredo release. Actually, my course aims to have our implementations included to the next release. Do you have a number of how many downloads per month simredo has?

that's all for now, i will have sent you the code by tomorrow

best regards,
alex

Date:

Sat, 20 Aug 2005 08:18:22 -0700 (PDT)

From:

"Klivo Lendon" <indriko@yahoo.com>

Subject:

Re: simredo 4.0 project

To:

realtowz@yahoo.com

--- Al-x- Haros <realtowz@yahoo.com> wrote:

> Dear Klivo,

>

> by now i have added the loading bar but stopped
> trying to make a greek dictionary since i had some
> encoding problems.

>

> I would surely like to incude my contibutions to the
> next simredo release.

Wonderful.

> Actually, my course aims to have
> our implementations included to the next release. Do
> you have a number of how many downloads per month
> simredo has?

Actually, I do record downloads. My Access database is at work, so I can't give exact numbers right now, but Simredo is downloaded about seven times a day, over two hundred times a month.

You can look at the log, if you like. This only shows downloads since July 1st. The file was growing large, so I archive the first six months.

<http://www4.vc-net.ne.jp/~klivo/cgi->

[bin/download/log/SimInstall.jar.log](#)

```
> that's all for now, i will have sent you the code by  
> tomorrow  
>  
> best regards,  
>  
> alex
```

Regards,

Klivo

Date:

Sat, 20 Aug 2005 15:23:14 +0100 (BST)

From:

"Al-x- Haros" <realtowz@yahoo.com>

Subject:

Re: simredo 4.0 project

To:

"Klivo Lendon" <indriko@yahoo.com>

Hello klivo!

As I informed you two days ago I am sending you my
-probably final version of simredo.

I have made changes on Simredo4.java,
simredo4.properties and i have added a new class
called progressshow.java...

that's all for now.

Date:
Sun, 21 Aug 2005 06:07:53 -0700 (PDT)
From:
"Klivo Lendon" <indriko@yahoo.com>
Subject:
Re: simredo 4.0 project
To:
"Al-x- Haros" <realtowz@yahoo.com>

Hello Alex,

Thank you for the files. I'm looking forward to seeing Simredo in Greek, a language which I've always admired.

Efkharisto.

Klivo