

2IP15: Programming Methods

Documentation for Assignment 3, part 3: FormulaLab and Undo/Redo

E.I.R. van Delden
student: 0618959

2008-07-03

Abstract

This document will give a, very, brief summary of my Undo/Redo Implementation for the FormulaLab program.

1 Introduction

I have added 2 new units, *UndoRedo* and *TreeEditorCommands*. These units are situated in

`/Components/Base/UndoRedo.pas`

and

`/Components/Base/TreeEditorCommands.pas`

respectively.

1.1 UndoRedo

The UndoRedo unit is exactly the same as seen in the Appendix of the UndoRedo note.

1.2 TreeEditorCommands

TreeEditorCommands is based on the unit *LineEditorCommands* from the extra UndoRedo note. The following assumption was made:

“Using the Undo and Redo commands, do not effect the clipboard.”

This is a commonly made assumption, seen in more applications, like Microsoft Word or the texteditor used for this document (Macromates' TextMate).

The following operations have been added to TreeEditorCommands, to support Undo-/Redoing:

- DoClear
- DoCut

- DoDelete
- DoExpand
- DoPaste

Because of the above made assumption, DoCopy was omitted. Every command has an *create*, *execute*, *reverse* and *reversible* method.

1.2.1 Create

For each command the variable *FTreeEditor* is given as an argument, so we can access the Tree form within *command.execute*. Expand and Paste have an extra argument, for replacing a part of the tree. All commands have at least 1 private field, *FDeletedTree*. *FDeletedTree* stores the old tree that's been deleted (the entire tree, not an subtree nor a pointer to a father).

1.2.2 Execute

The code for the old implementation is literally copied to the execute method and slightly adjusted. The assert's and all code for the clipboard was removed.

Preceding the copied code, there is an extra assignment

```
FDeletedTree := FTreeEditor;
```

So we don't lose any data when we want to undo the command

1.2.3 Reverse / Reversible

All given Commands change the tree in a way, so all commands are *Reversible* (result = true). Reverse has for all commands the following implementation:

```
FTreeEditor.Replace(FDeletedTree, FTreeEditor.Tree);
```

It restores the tree, before the command was executed.

2 Implementation

An extra private variable has been added, *FController*: *TController*, that is created on Initialization. For the 5 original commands, the asserts and all code with respect to the clipboard have been untouched. The real working code has been moved to corresponding execute command in the unit *TreeEditorCommands*.

The implementations of the 5 original commands have been replaced with:

```
Command := TCommand_DoClear.Create(FTreeEditor);
FController.DoCommand(Command);
```

The corresponding Command object is created and fed to the Controller, that executes the command and adjusts the 2 stacks accordingly.

2 new buttons have been added, *Undo* and *Redo*. These buttons can be used to Undo or Redo a certain command.

Note: The current implementation isn't bug free. 1 command can successfully be undone and redone, but troubles arise when trying to undo more commands. Part of it is, so I believe, because of the chosen implementation.