Debugging : one of the joys of Smalltalk!

To encourage fearless programming, the 'debugger' shouldn't be called a 'debugger'. It's not that you're cleaning up mistakes, but most of the time you're just exploring how something works. A name as for example 'flowInspector', 'liveInspector' more reflects and encourages the actual process. When I started programming in Smalltalk I didn't use the power of the debugger, after reading the first lines of the debug message I went back to my code and tried to fix the bug. I almost never pushed the debug button. Now, 'debugging' my code is actually more than half the part of 'making the solution'. I'm not fearing the debugger anymore. I just program/prototype my unclear ideas, I let come alive in the 'liveInspector', and refine them to a better solution.

Mathieu van Echtelt

Standard guidelines to minimize bugs' proliferation

Do not assume that a method answers what you expect it to answer

Use parentheses when sending several keyword messages in one expression

Avoid modifying a collection while iterating over it

Avoid altering the behaviour of well-known messages

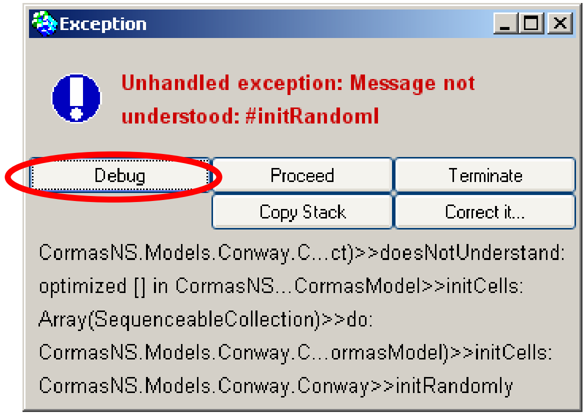
Standard error messages

« message not understood `self` »
usually means that a period (to separate two consecutive expressions) is missing

« does not understand `messageName` »
usually means the receiver is not of the right kind

- it may be undefined (`nil`)
- « does not understand `whileTrue:` » means the receiver is not a block (enclosed by `[]`)

The « **Debuger** » window

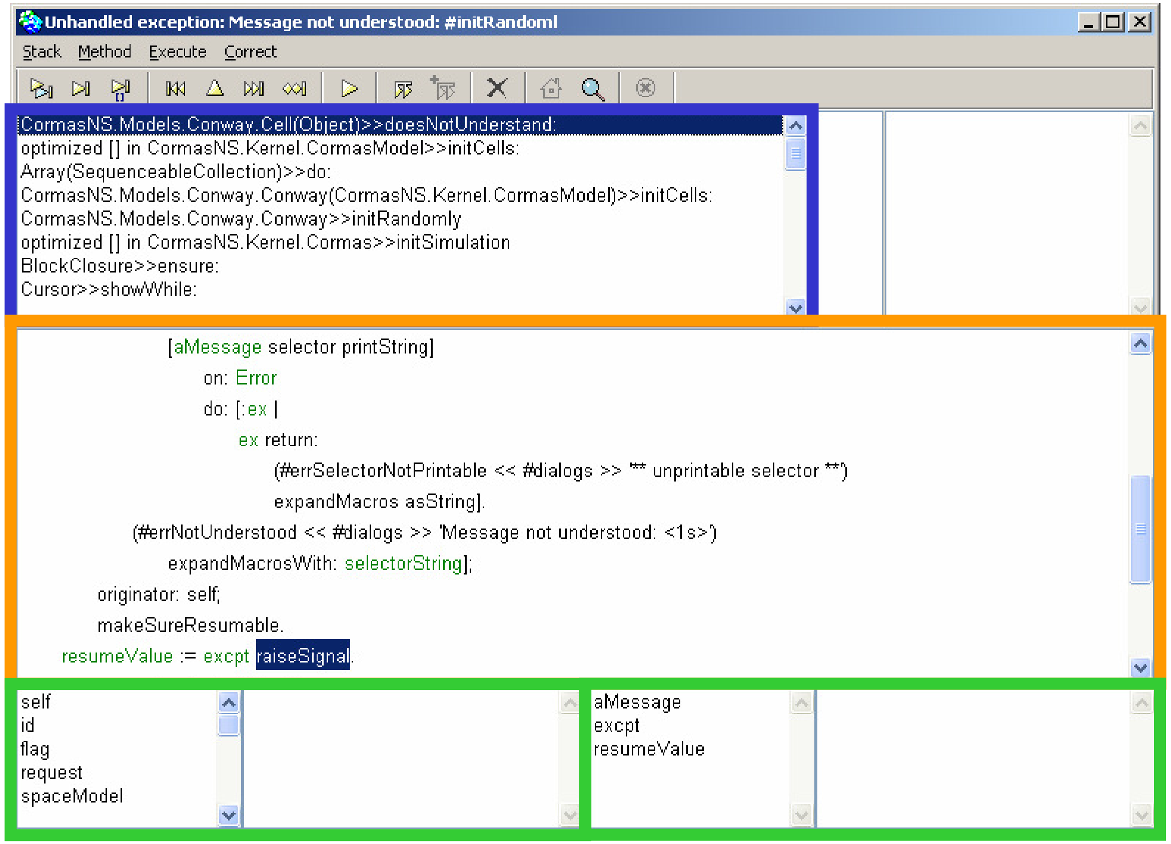


Unhandled exception: Message not understood: #initRandoml

```
CormasNS.Models.Conway.C...ct)>>doesNotUnderstand:
optimized [] in CormasNS...CormasModel>>initCells:
Array(SequenceableCollection)>>do:
CormasNS.Models.Conway.C...ormasModel)>>initCells:
CormasNS.Models.Conway.Conway>>initRandomly
```

« Stack » of messages sending

Code editor



```
CormasNS.Models.Conway.Cell(Object)>>doesNotUnderstand:
optimized [] in CormasNS.Kernel.CormasModel>>initCells:
Array(SequenceableCollection)>>do:
CormasNS.Models.Conway.Conway(CormasNS.Kernel.CormasModel)>>initCells:
CormasNS.Models.Conway.Conway>>initRandomly
optimized [] in CormasNS.Kernel.Cormas>>initSimulation
BlockClosure>>ensure:
Cursor>>showWhile:
```

```
                    [aMessage selector printString]
                        on: Error
                        do: [:ex |
                            ex return:
                                (#errSelectorNotPrintable << #dialogs >> '** unprintable selector **')
                                expandMacros asString].
                (#errNotUnderstood << #dialogs >> 'Message not understood: <1s>')
                    expandMacrosWith: selectorString];
        originator: self;
        makeSureResumable.
    resumeValue := excpt raiseSignal.
```

```
self
id
flag
request
spaceModel
```

```
aMessage
excpt
resumeValue
```

Inspector for instance variables

Inspector for temporary variables