

Avoid long functions. Avoid deep nesting.

4/18/2012



long python

nested alligator

q.e.d.

Stefan Kislinskiy
Dept. of Medical and Biological Informatics (E130)

A Bug Squashing Story

Bugzilla - Bug List

Home | New | Browse | Search | Search [2] | Reports | My Requests | Preferences | Help | Log out s.kislinskiy@dkfz-heidelberg.de

Wed Apr 18 2012 11:05:23 CEST
Fußball ist wie Schacht. Nur ohne Würfel. (Podolski)

Resolution: --- Component: Segmentation Product: MITK

22 bugs found.

ID	Product	Comp	Assignee	Status	Resolution	Summary	Changed
11318	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	CONF	---	2D interpolation preview still shifted	2012-03-28
11317	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	CONF	---	Preview of 3D region growing does not disappear after closing view	2012-03-23
11316	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	CONF	---	Update problem for defining contour when 3D interpolation is active	2012-04-11
11313	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	CONF	---	3D interpolation preview still shifted	03-23
11312	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	CONF	---	Preview of 3D region growing does not disappear after closing view	03-23
11311	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	CONF	---	Update problem for defining contour when 3D interpolation is active	03-23
11210	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	CONF	---	Change mouse cursor when a 2D segmentation tool is active	2012-03-23
11309	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	CONF	---	It should be able to activate the 3D interpolation afterwards	2012-03-23
11218	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	IN_P	---	Boolean View of segmentation should be able to work in images with different dimension	2012-04-10
11213	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	IN_P	---	Segmentation Bundle crashes when selecting the "Autocrop" Option	2012-04-12
11127	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	CONF	---	The position nodes of the 3D interpolation should contain a new data type	2012-03-05
10395	MITK	Segmenta	t.heimann@dkfz-heidelberg.de	CONF	---	Wrong interpolation after clipping a 3D image with a deformed plane	2012-01-16
10248	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	IN_P	---	Create a module for the segmentation	Mon 15:00
10333	MITK	Segmenta	t.schroeder@dkfz-heidelberg.de	IN_P	---	Implement an overwriteImageFilter based on the algorithm of vtkImageReslice	2012-04-11
9271	MITK	Segmenta	t.heimann@dkfz-heidelberg.de	CONF	---	Contouring tools don't work if e.g. measurement bundle is active	2012-01-11
9288	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	CONF	---	Reference image combo remains disabled	2011-11-10
9222	MITK	Segmenta	t.heimann@dkfz-heidelberg.de	UNCO	---	Segmentation is not possible in .dwi, .dti and .qbi images	2011-08-31
7994	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	CONF	---	Realize undo functionality for the segmentation in rotated planes	2011-07-27
6417	MITK	Segmenta	a.fetzer@dkfz-heidelberg.de	IN_P	---	Allow contouring for rotated slices	2011-10-12
5863	MITK	Segmenta	t.heimann@dkfz-heidelberg.de	CONF	---	no filewriter/reader for mitk::sacdsimage available	2011-10-12
4276	MITK	Segmenta	t.heimann@dkfz-heidelberg.de	CONF	---	Allow to choose the pixel value for the ROI and added a flag to erase only the pixels with a specified pixel value	2011-10-12

22 bugs found.

[Long Format](#) [CSV](#) | [Feed](#) | [Calendar](#) | [Change Columns](#) | [Change Several Bugs at Once](#) | [Send Mail to Bug Assignees](#) | [Edit Search](#) as

[XML](#)

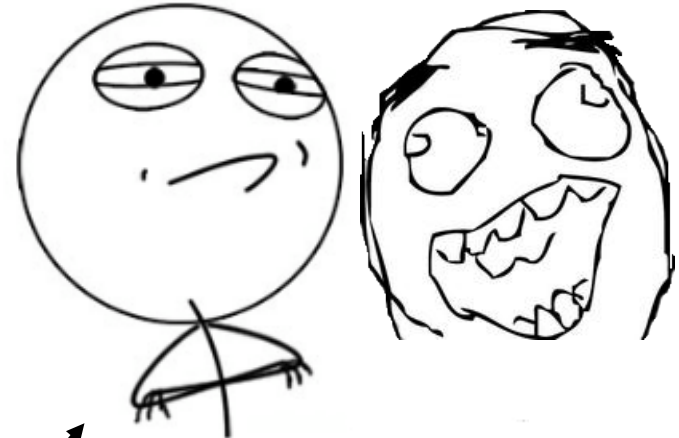
File a new bug in the "MITK" product

Home | New | Browse | Search | Search [2] | Reports | My Requests | Preferences | Help | Log out s.kislinskiy@dkfz-heidelberg.de

My Bugs
What did I do today?

Super easy tiny nano party bug

CHALLENGE ACCEPTED



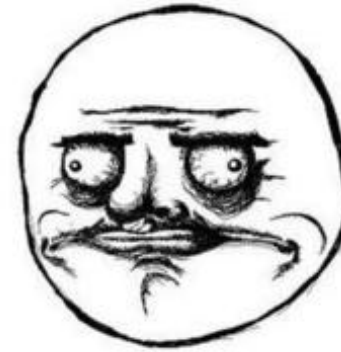
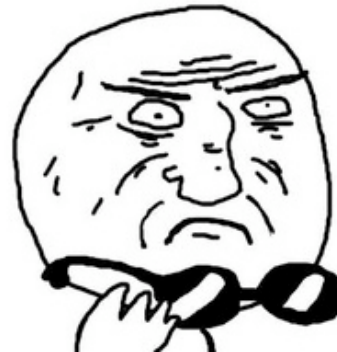
bug squashing bros

```
610 {
611     m_Controls->refImageSelector->show();
612     m_Controls->lblReferenceImageSelectionWarning->hide();
613
614     bool isBinary(false);
615     selectedNode->GetBoolProperty("binary", isBinary);
616     if ( isBinary )
617     {
618         FireNodeSelected(selectedNode);
619         selectedNode->SetVisibility(true);
620     }
621     else if (node != m_Controls->m_ManualToolSelectionBox->GetToolManager()->GetReferenceData(0))
622     {
623         if (m_Controls->m_ManualToolSelectionBox->GetToolManager()->GetReferenceData(0))
624             m_Controls->m_ManualToolSelectionBox->GetToolManager()->GetReferenceData(0)->SetVisibility(false);
625         if (m_Controls->m_ManualToolSelectionBox->GetToolManager()->GetWorkingData(0))
626         {
627             m_Controls->m_ManualToolSelectionBox->GetToolManager()->GetWorkingData(0)->SetVisibility(false);
628         }
629         FireNodeSelected(selectedNode);
630         selectedNode->SetVisibility(true);
631         SetToolManagerSelection(selectedNode, NULL);
632     }
633 }
634 else
635 {
636     m_Controls->refImageSelector->hide();
637     m_Controls->lblReferenceImageSelectionWarning->show();
638 }
639 }
640
641
642 void QmitkSegmentationView::OnShowMarkerNodes (bool state)
643 {
644     mitk::SegTool2D::Pointer manualSegmentationTool;
645
646     unsigned int numberOfExistingTools = m_Controls->m_ManualToolSelectionBox->GetToolManager()->GetTools().size();
647
648     for(unsigned int i = 0; i < numberOfExistingTools; i++)
649     {
```

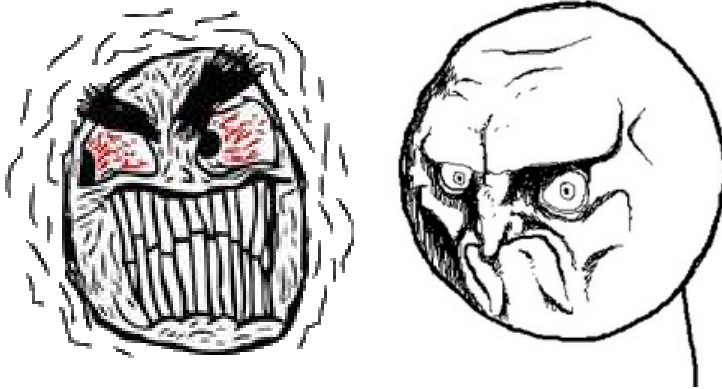
*million LOC's in functions,
trillion LOCs in file...*

*... stuffed with über-loops
and mo-mo-mo-monster
conditional branches*

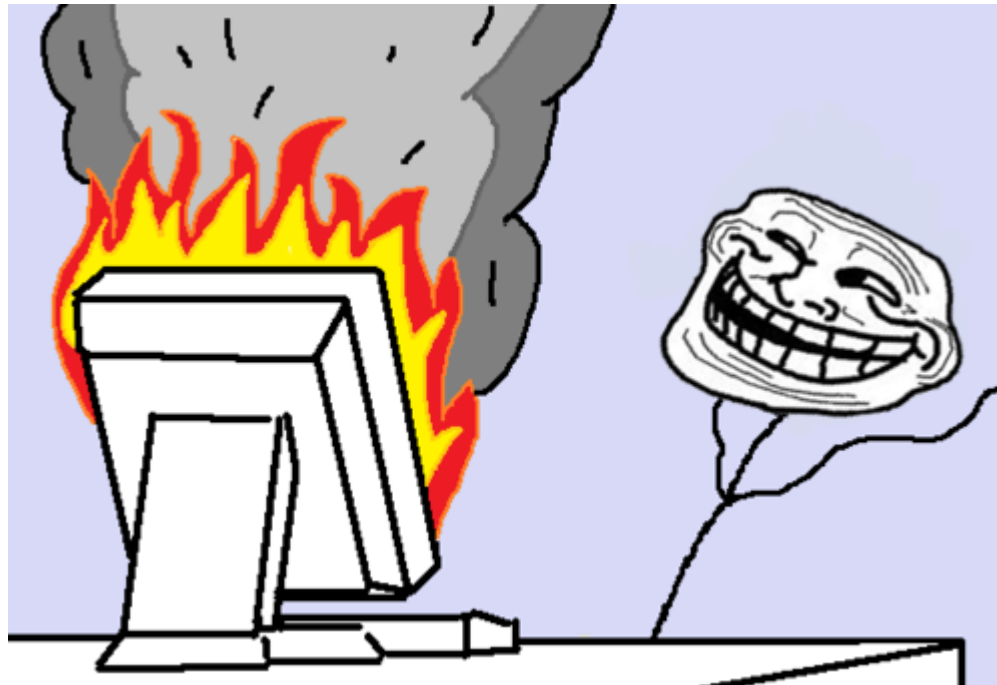
MOTHER OF GOD...



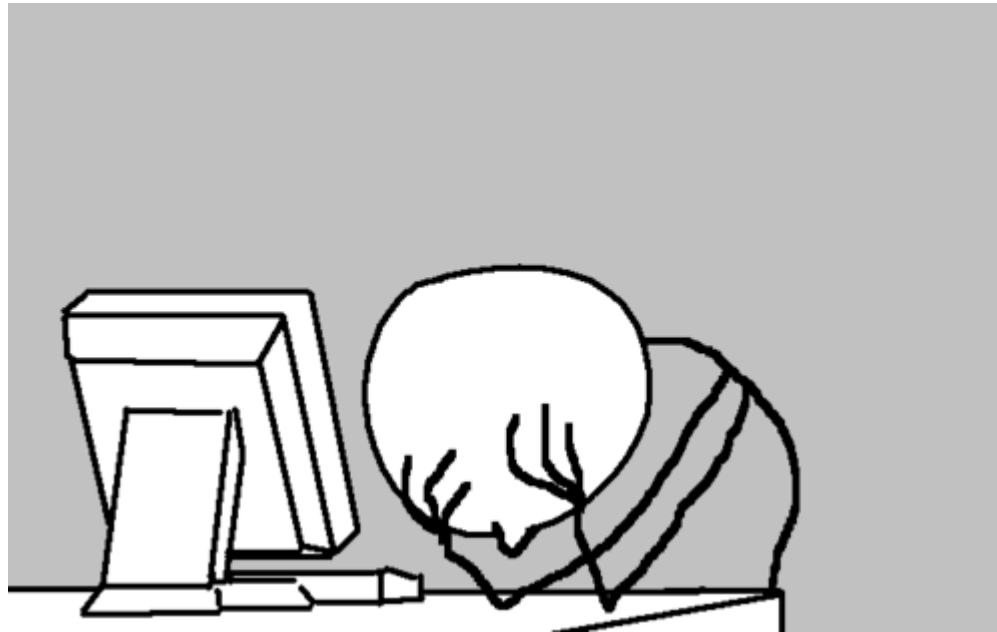
a few hours later...



... later that day ...



Finally found truly super easy tiny nano ultra well hidden party bug...



... but somehow there is no satisfaction in the air. - THE END -

- Avoid long functions
 - Every function should be a coherent unit of work
 - One function – one responsibility
- Avoid deep nesting
 - Rhetorical question: Have you ever found a closing brace in someone's code and wondered which of the many **fors**, **whiles**, or **ifs** it matched?

- Prefer cohesion: Give one function one responsibility. No, really!
- Don't repeat yourself: Prefer named functions over C&P'ed code snippets.
- Prefer algorithms: Flatter than loops, and your/my code usually sucks compared to them. :-)

- Prefer writing nonmember nonfriend functions
 - Improve encapsulation
 - Break apart monolithic classes
 - Reduce dependencies / coupling
- Functions have pre- and post-conditions – respect them!
 - Fail gracefully. If your function works only for 3D-Images, make sure input IS a 3D-image instead of crashing hard.



- How many hours did you spend searching for bugs and how many hours did you spend to actually squash those nasty bitc... I mean bugs?
- Would that bug be even present when there would be shorter and flatter functions with single responsibilities instead of that monolithic “piece of ... code”?



We should endorse **code reading** and **(preemptive) refactoring** during bug squashing parties.

Thank you! :)

