Developing for ImageJ and friends.

The Developer/Image analyst daily bread.

JYT image analysis work.

Work in a light-microscopy **facility.**
Service to users.
50% microscopy / 50% image analysis.

Bioimage Analyst

(Main) Tools of the trade:

- **Fiji**
  - Explore data
  - Use existing plugins
  - Script & Macros
  - Swiss army knife
- **Icy**
  - The same.
- **MATLAB**
  - Data analysis workflows
- **Python**
  - File processing
  - Text processing

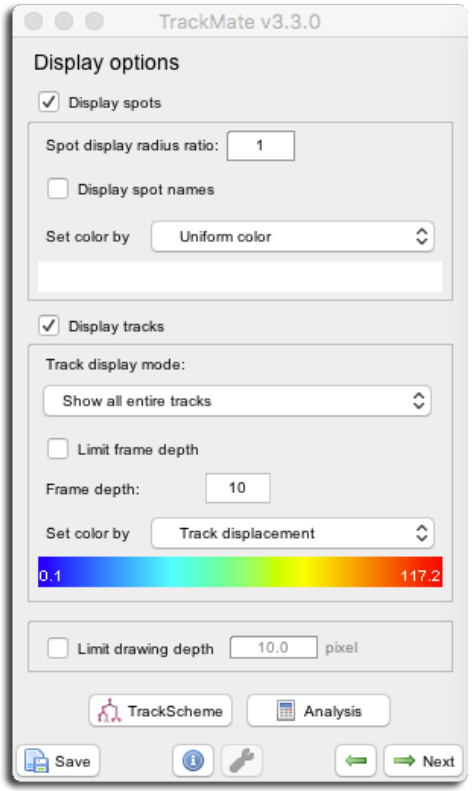
Developer

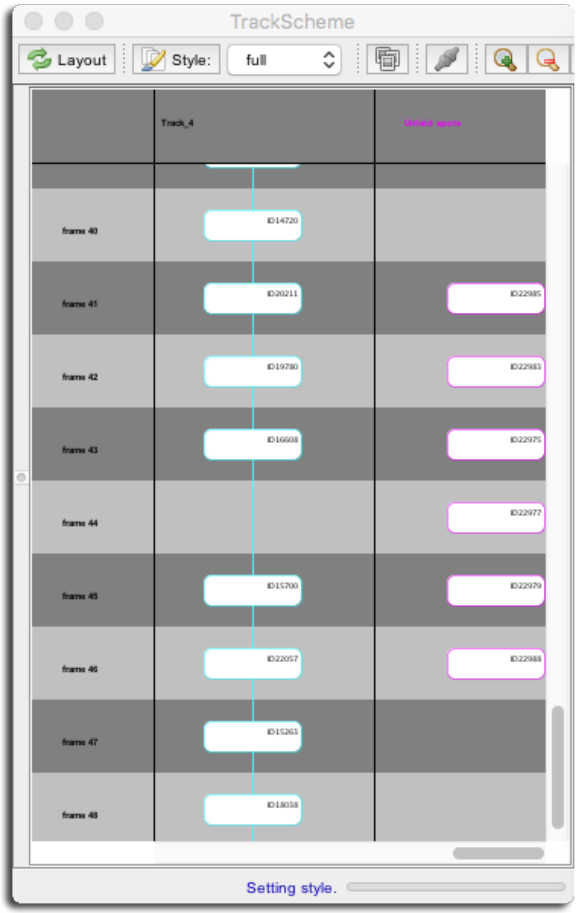
Java & co:

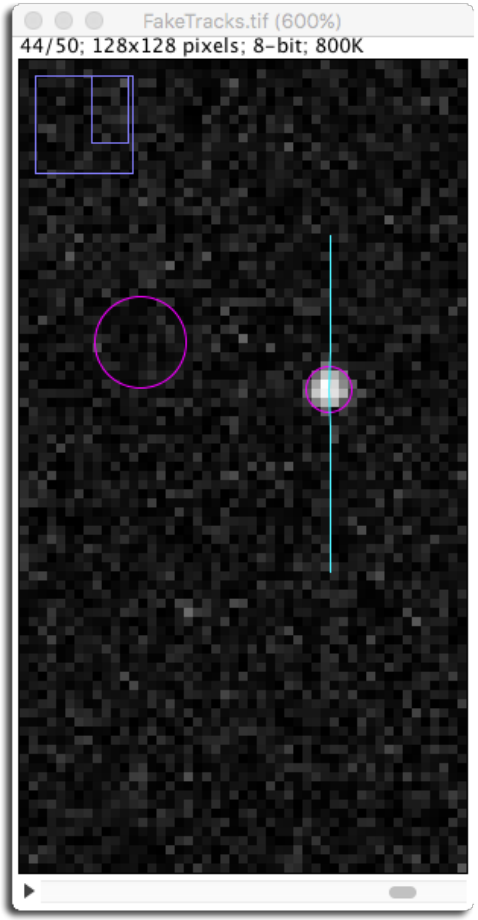
- GUI stuff.
- New plugins or algorithms.
- Pixel based stuff.
- Complex development or deployment

Why turning to development?

- GUI, visualization and user interaction:

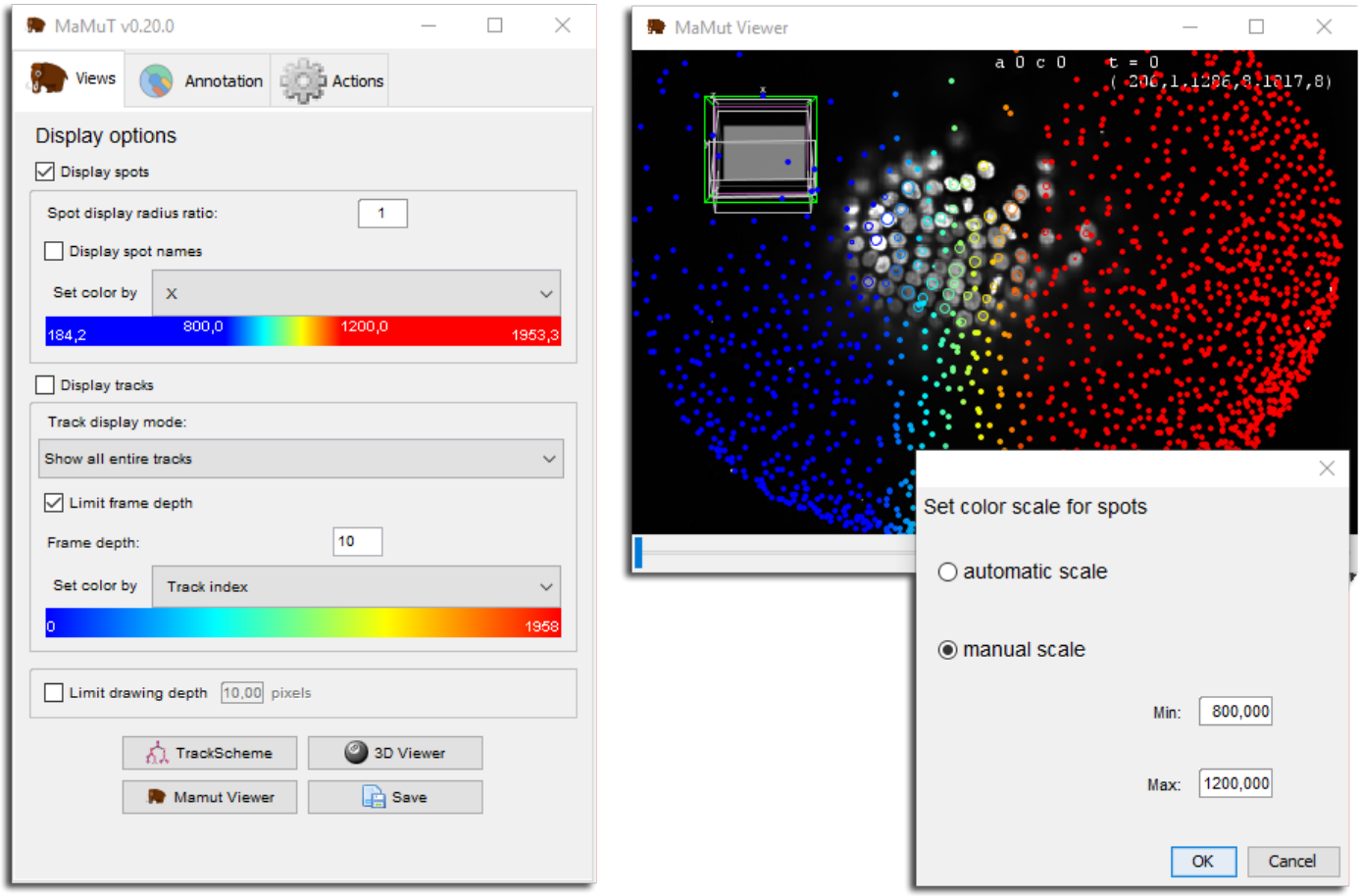






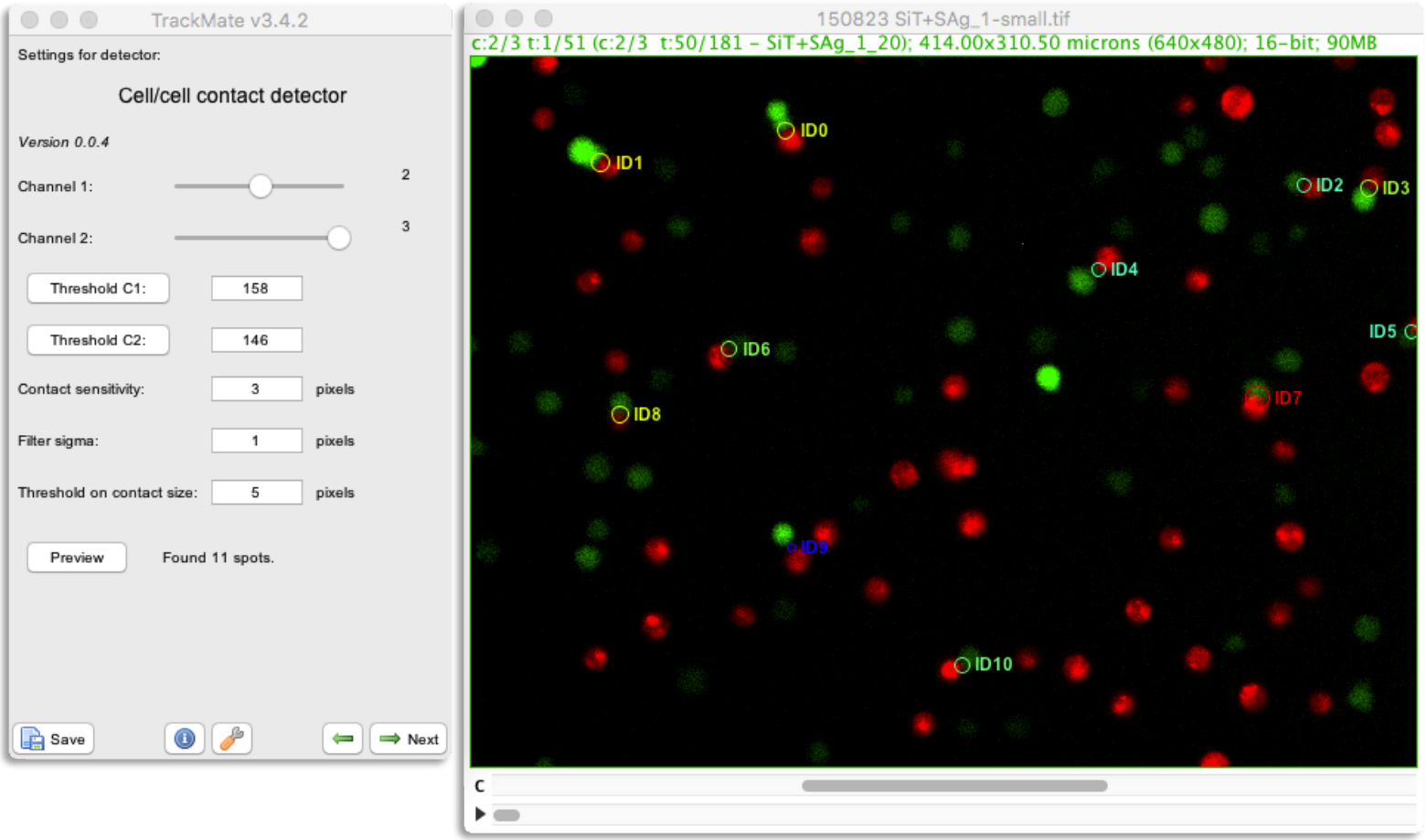
Why turning to development?

- Big data (images or annotations).



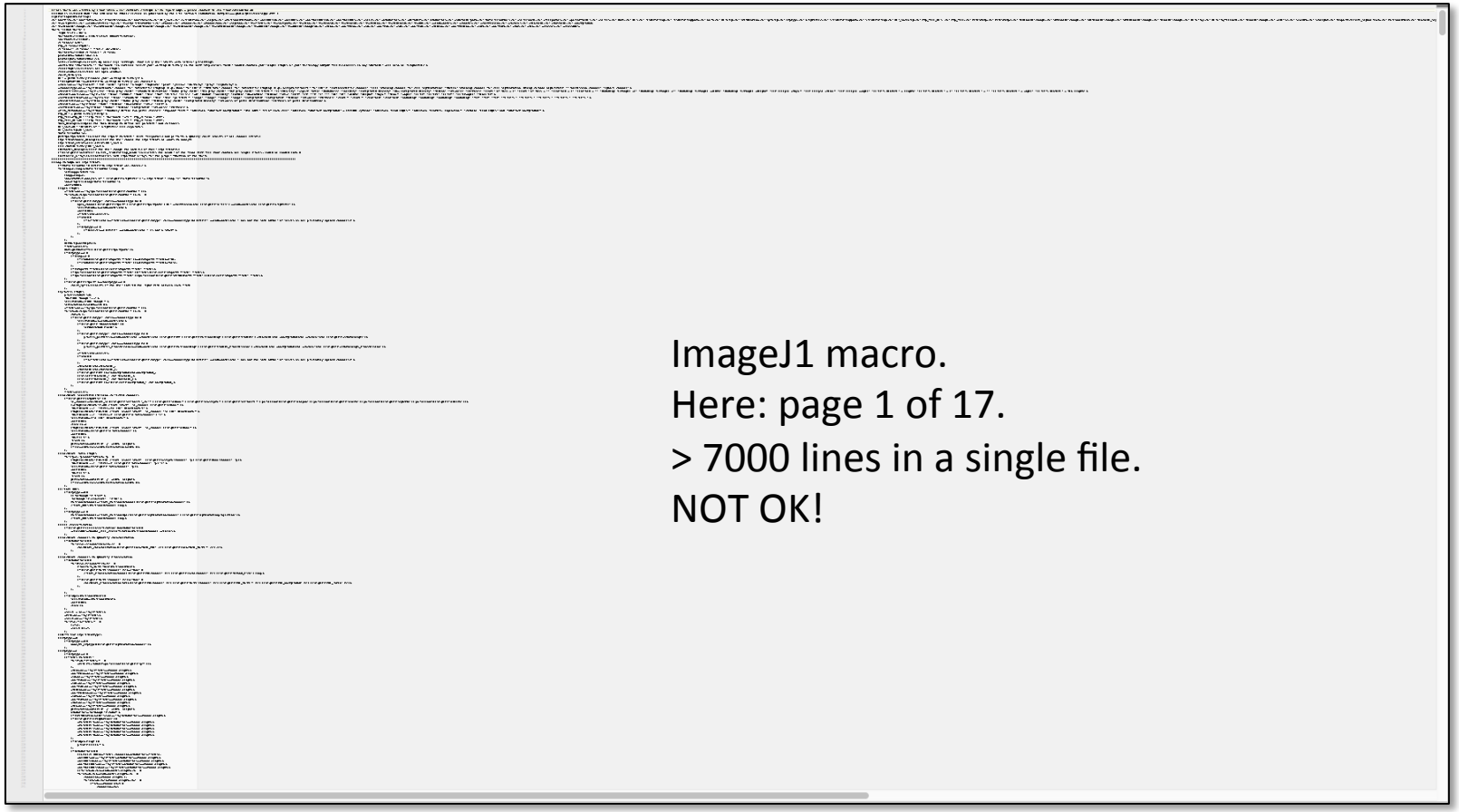
Why turning to development?

- Extending existing plugins requires programming:



Why turning to development?

- Projects is “large”.



ImageJ1 macro.
Here: page 1 of 17.
> 7000 lines in a single file.
NOT OK!

Why turning to development?

- Pixel-based operations, new algorithms.

```
final ArrayCursor< FloatType > c = kernel.cursor();
final long[] coords = new long[ nDims ];


// Work in image coordinates
while ( c.hasNext() )
{
        c.fwd();
        c.localize( coords );

        double sumx2 = 0.;
        double mantissa = 0.;
        for ( int d = 0; d < coords.length; d++ )
        {
                final double x = calibration[ d ] * ( coords[ d ] - middle[ d ] );
                sumx2 += ( x * x );
                mantissa += 1. / sigmaPixels[ d ] / sigmaPixels[ d ] * ( x * x / sigma / sigma - 1 );
        }
        final double exponent = -sumx2 / 2. / sigma / sigma;
```

Why turning to development?

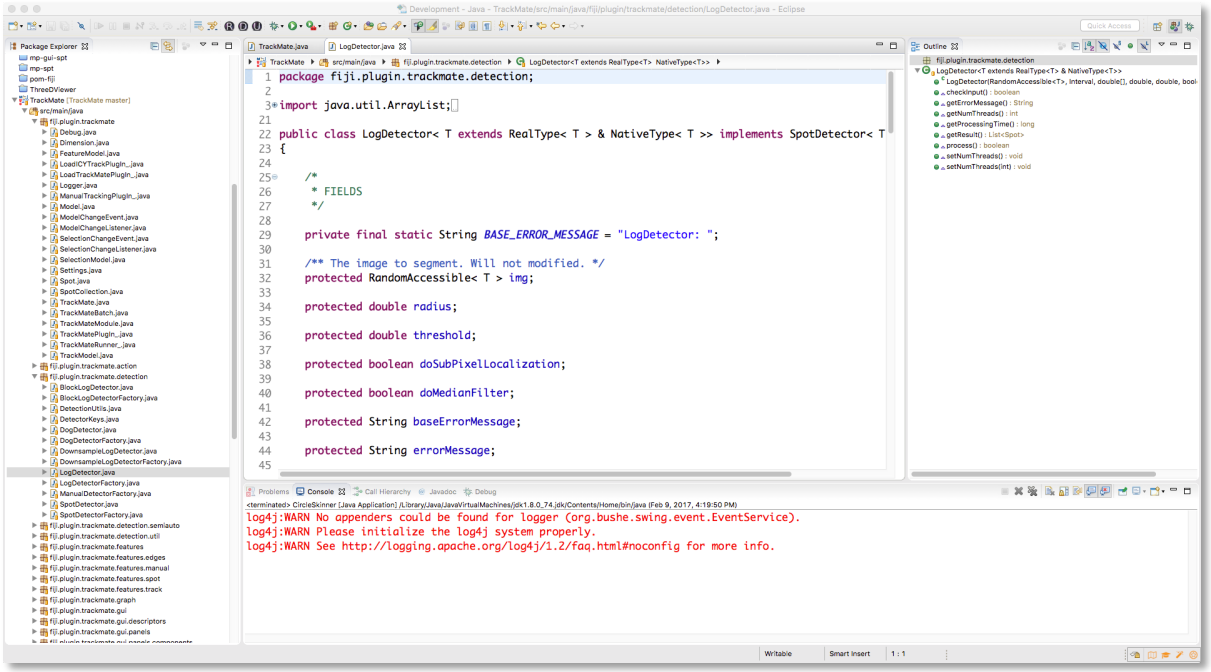
- It’s fun and fulfilling!

TOOLS FOR DEVELOPMENT.

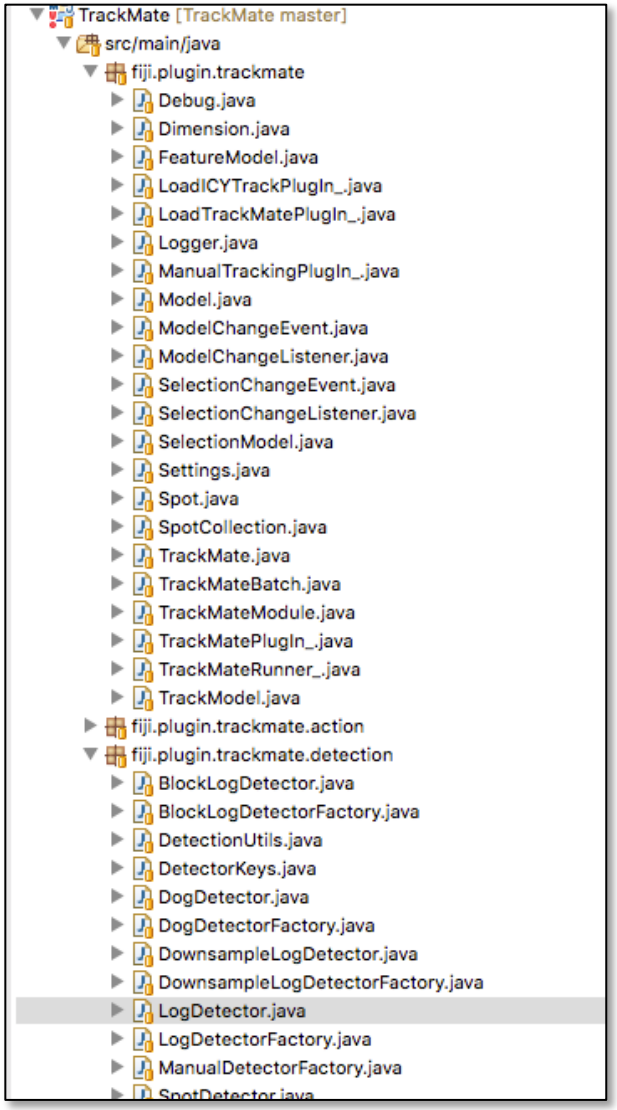
INTEGRATED DEVELOPMENT ENVIRONMENT (IDE).

Integrated Development Environment (IDE).

A software to write code, conveniently.

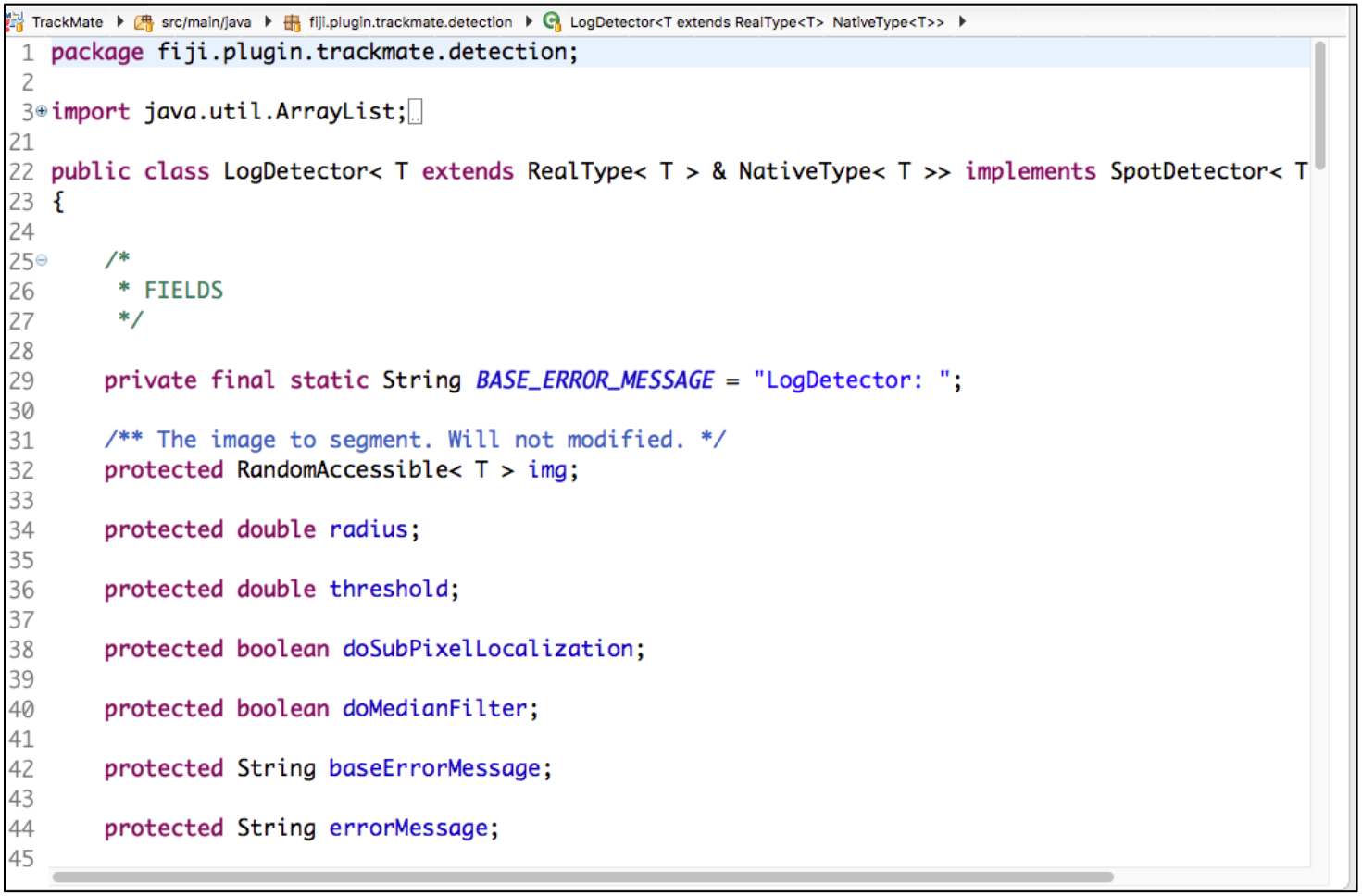


Benefits of an IDE.



Project & code organization.

Benefits of an IDE.




```
1  package fiji.plugin.trackmate.detection;
2
3  import java.util.ArrayList;
21
22 public class LogDetector< T extends RealType< T > & NativeType< T >> implements SpotDetector< T
23 {
24
25     /*
26      * FIELDS
27      */
28
29     private final static String BASE_ERROR_MESSAGE = "LogDetector: ";
30
31     /** The image to segment. Will not modified. */
32     protected RandomAccessible< T > img;
33
34     protected double radius;
35
36     protected double threshold;
37
38     protected boolean doSubPixelLocalization;
39
40     protected boolean doMedianFilter;
41
42     protected String baseErrorMessage;
43
44     protected String errorMessage;
45
```

Syntax highlighting -> easier to orient yourself.

Benefits of an IDE.

```
1  package fiji.plugin.trackmate.detection.util;
2
3  import java.util.Iterator;
4
5  import net.imglib2.IterableInterval;
6  import net.imglib2.Localizable;
7  import net.imglib2.Positionable;
8  import net.imglib2.RandomAccess;
9
10 import net.imglib2.RandomAccessibleInterval;
11 import net.imglib2.RealPositionable;
12 import java.awt.Window;
13
14 import net.imglib2.outofbounds.OutOfBoundsFactory;
15 import net.imglib2.view.ExtendedRandomAccessibleInterval;
16 import net.imglib2.view.Views;
17 import net.imglib2.Cursor;
18 import net.imglib2.RandomAccessibleInterval;
19
20 public class SquareNeighborhood3x3 <T> implements Positionable, Ite
21
22     private RandomAccessibleInterval<T> source;
23     private final long[] center;
24     private final ExtendedRandomAccessibleInterval<T, RandomAccessi
25
26     /*
27      * CONSTRUCTOR
28      */
```

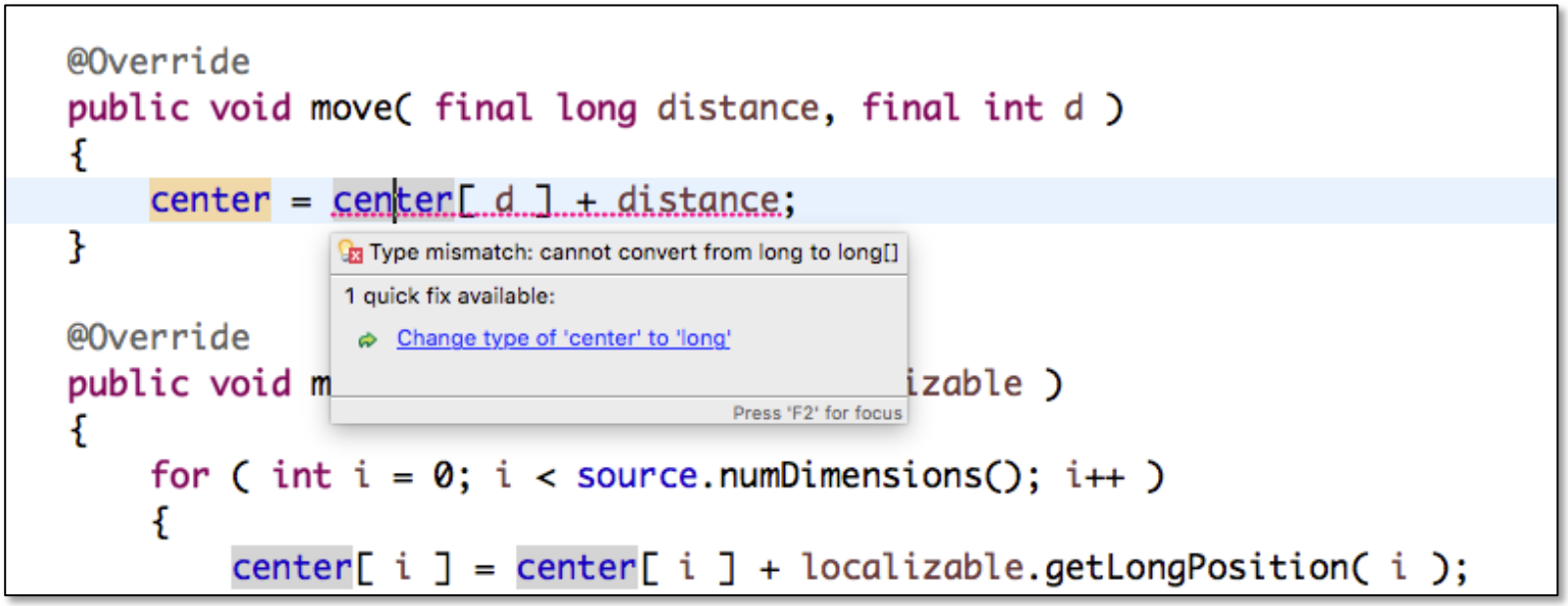
Automatic code style.

```
1  package fiji.plugin.trackmate.detection.util;
2
3  import java.util.Iterator;
4
5  import net.imglib2.IterableInterval;
6  import net.imglib2.Localizable;
7  import net.imglib2.Positionable;
8  import net.imglib2.RandomAccess;
9  import net.imglib2.RandomAccessibleInterval;
10 import net.imglib2.RealPositionable;
11 import net.imglib2.outofbounds.OutOfBoundsFactory;
12 import net.imglib2.view.ExtendedRandomAccessibleInterval;
13 import net.imglib2.view.Views;
14
15 public class SquareNeighborhood3x3< T > implements Positionable, IterableInterval< T >
16 {
17
18     private RandomAccessibleInterval< T > source;
19
20     private final long[] center;
21
22     private final ExtendedRandomAccessibleInterval< T, RandomAccessibleInterval< T > > extended
23
24     /*
25      * CONSTRUCTOR
26      */
27
28     public SquareNeighborhood3x3( final RandomAccessibleInterval< T > source, final OutOfBounds
```

Aesthetic satisfaction.
Common grounds when multiple developers work on the same code.
Enforce project standards.

Benefits of an IDE.

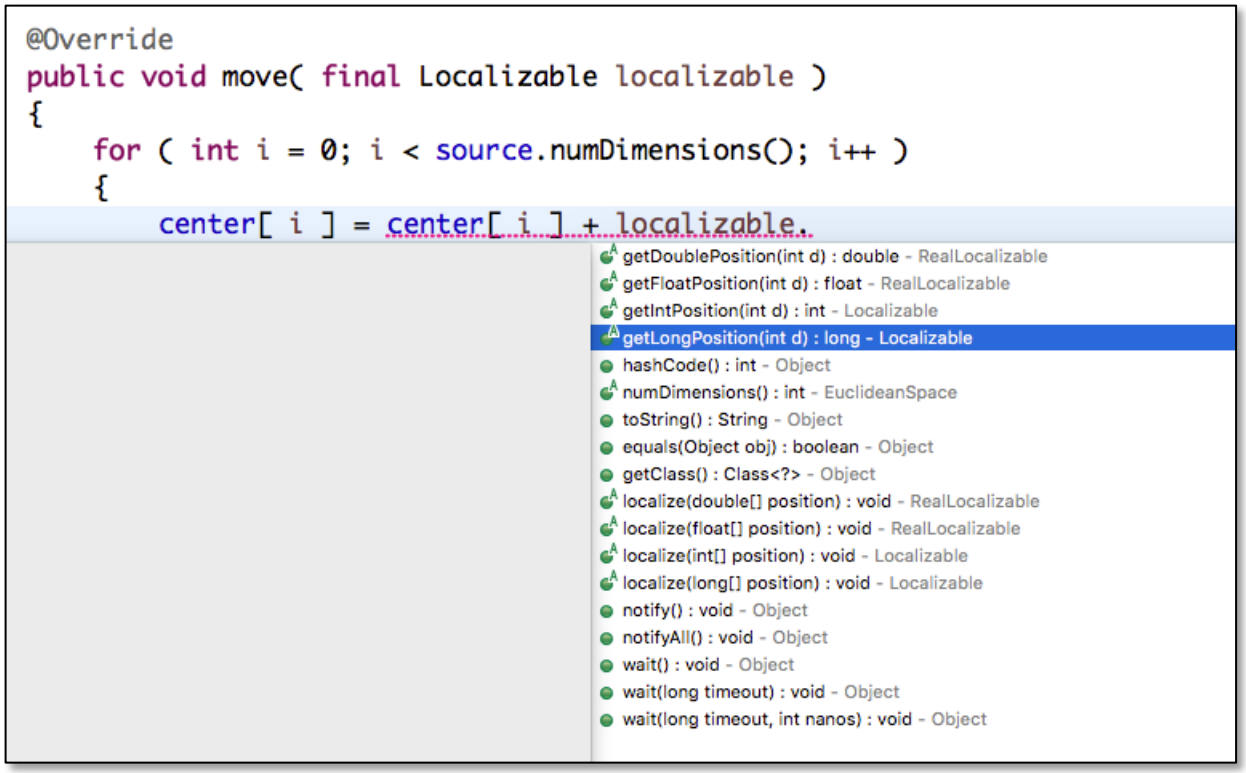
Errors and warnings reporting.



Benefits of an IDE.

Code completion.

Save live 500% of time (conservative estimate).



CODE VERSION CONTROL SYSTEM (CVS).

Check code status.

TrackMate — -bash — 105×41

```
[tinevez@lilium:~/Development/TrackMate$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   src/main/java/fiji/plugin/trackmate/detection/util/SquareNeighborhood3x3.java

no changes added to commit (use "git add" and/or "git commit -a")
tinevez@lilium:~/Development/TrackMate$
```

TrackMate — less • git log — 83×35

```
    More seriously, the color bar in the configure view panel does not
    fit below this size.
:...skipping...
commit 5532cd4544e73cf6345b1a41eab12f4ae48c8ccf
Author: Jean-Yves Tinevez <jean-yves.tinevez@pasteur.fr>
Date:   Wed Aug 31 17:47:43 2016 +0200

    Make the textfields of the Downsample LoG detector config panel resizable.

commit 2d10a625ba796ab6d51a3d53b9850f2d9b386aee
Author: Jean-Yves Tinevez <jean-yves.tinevez@pasteur.fr>
Date:   Wed Aug 31 17:45:13 2016 +0200

    Make the textfields in the Blobk LoG detector config panel resizable.

commit e801c826989df07f0eedeab16e42719cc3c2e50e
Author: Jean-Yves Tinevez <jean-yves.tinevez@pasteur.fr>
Date:   Wed Aug 31 17:37:46 2016 +0200

    Make the text fields in the LoG detector config panel resizable.

commit 2c082031563efd52fb11142c05f9d9e483d733a5
Author: Jean-Yves Tinevez <jean-yves.tinevez@pasteur.fr>
Date:   Wed Aug 31 17:13:07 2016 +0200

    Remove debug code.

commit b8101e248c84ee034e04d0bef2b6ce961cc0f9f8
Author: Jean-Yves Tinevez <jean-yves.tinevez@pasteur.fr>
Date:   Wed Aug 31 14:48:06 2016 +0200

    Make TrackMate GUI trousers bigger by 10 pixels.

    More seriously, the color bar in the configure view panel does not
```

Code history.

TrackMate — less • git diff — 105×41

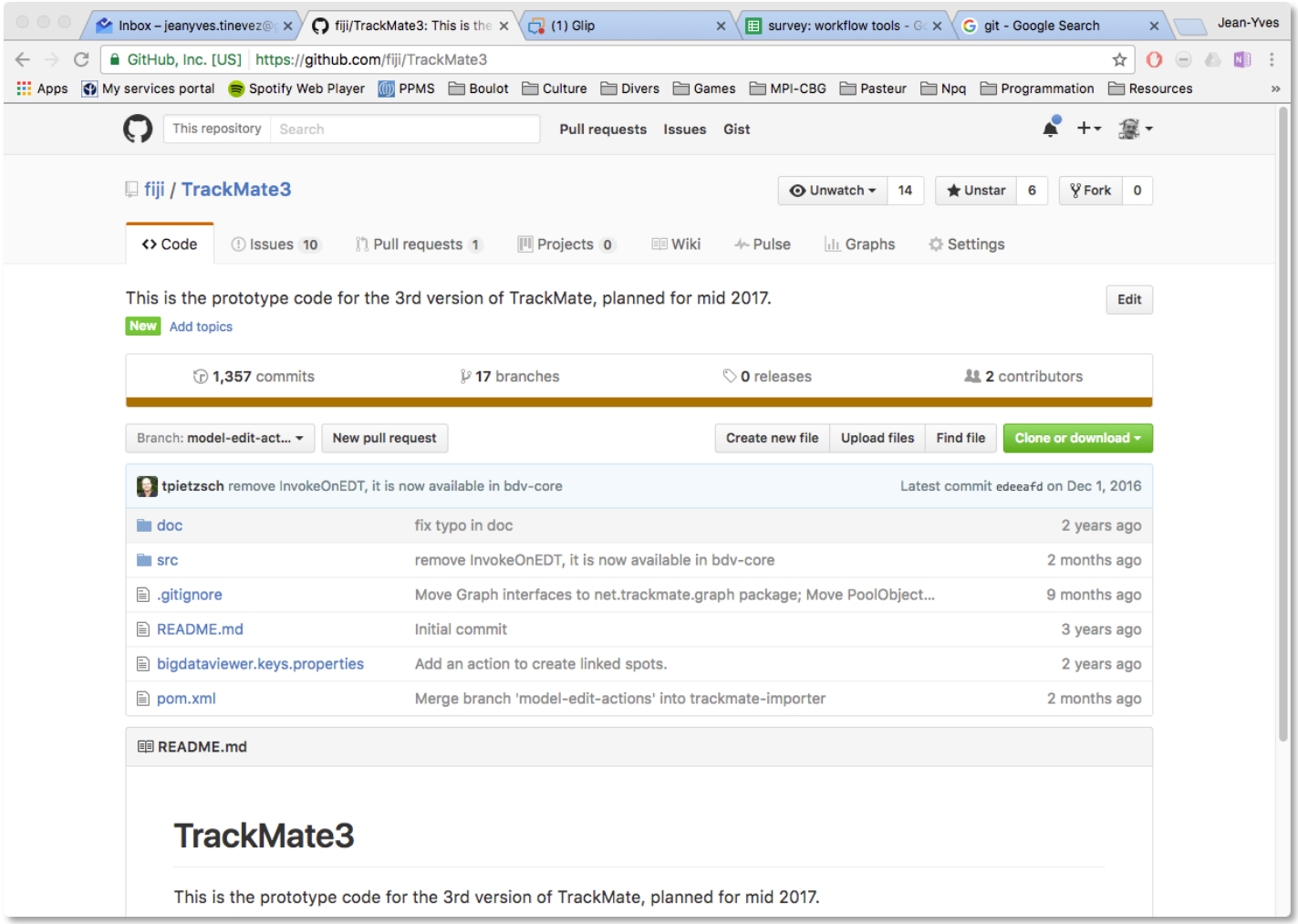
```
--- a/src/main/java/fiji/plugin/trackmate/detection/util/SquareNeighborhood3x3.java
+++ b/src/main/java/fiji/plugin/trackmate/detection/util/SquareNeighborhood3x3.java
@@ -12,227 +12,273 @@ import net.imglib2.outofbounds.OutOfBoundsFactory;
 import net.imglib2.view.ExtendedRandomAccessibleInterval;
 import net.imglib2.view.Views;

-public class SquareNeighborhood3x3 <T> implements Positionable, IterableInterval<T> {
+public class SquareNeighborhood3x3< T > implements Positionable, IterableInterval< T >
+{
+
+ private RandomAccessibleInterval< T > source;

-   private RandomAccessibleInterval<T> source;
        private final long[] center;
-   private final ExtendedRandomAccessibleInterval<T, RandomAccessibleInterval<T>> extendedSource;
-
+
+ private final ExtendedRandomAccessibleInterval< T, RandomAccessibleInterval< T > > extendedSource;
+
        /*
         * CONSTRUCTOR
         */
-
-   public SquareNeighborhood3x3(RandomAccessibleInterval<T> source, OutOfBoundsFactory<T, RandomAccessib
leInterval<T>> outOfBounds) {
+ public SquareNeighborhood3x3( final RandomAccessibleInterval< T > source, final OutOfBoundsFactory< T,
RandomAccessibleInterval< T > > outOfBounds )
+ {
                this.source = source;
-          this.center = new long[source.numDimensions()];
-          this.extendedSource = Views.extend(source, outOfBounds);
+         this.center = new long[ source.numDimensions() ];
+         this.extendedSource = Views.extend( source, outOfBounds );
        }
-
-
+
        /*
         * METHODS
:
```

Changes details.

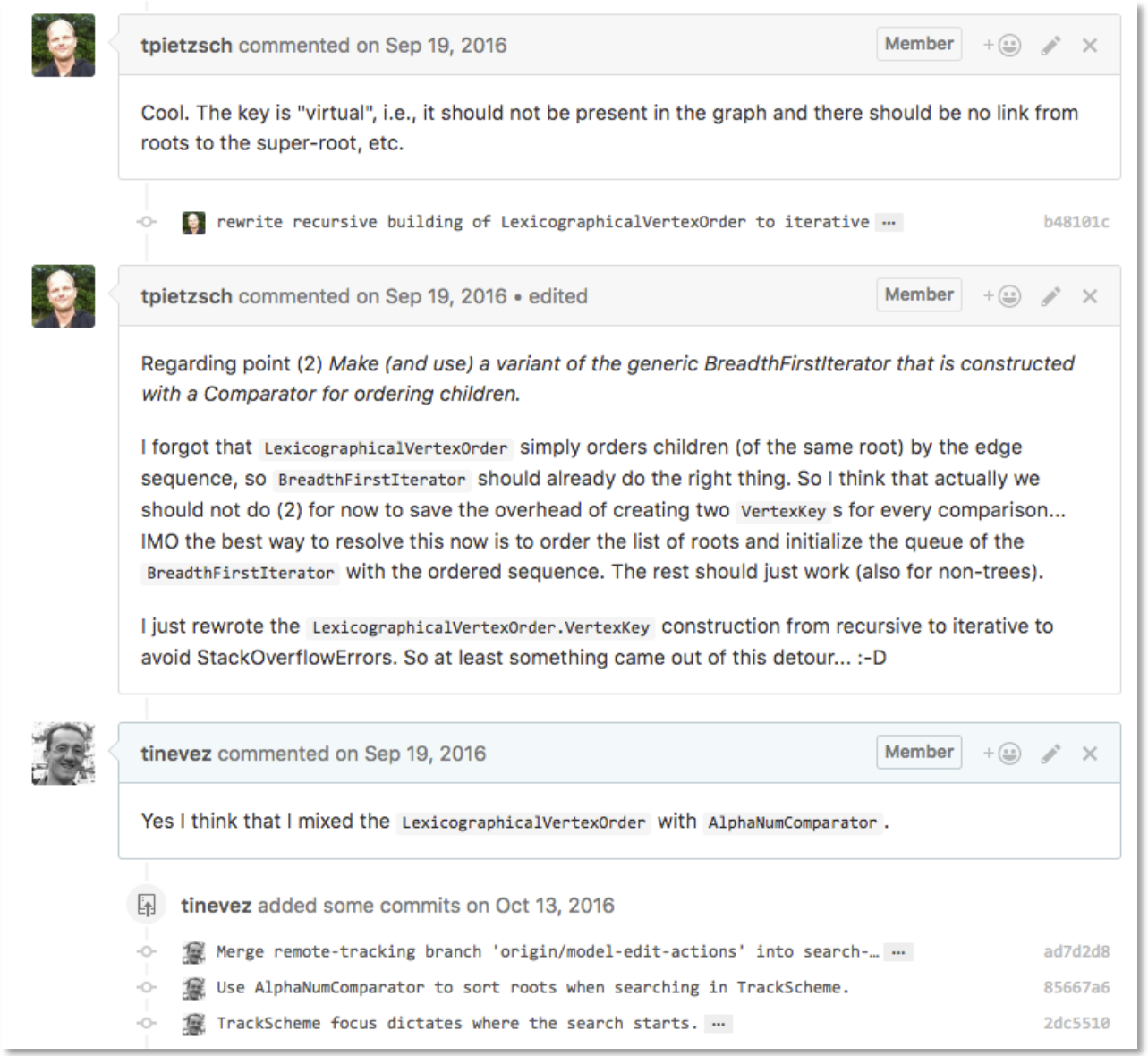
github
SOCIAL CODING



Online storage and sharing for your projects.

github
SOCIAL CODING

Work as a community.



tpietzsch commented on Sep 19, 2016 — Member

Cool. The key is "virtual", i.e., it should not be present in the graph and there should be no link from roots to the super-root, etc.

`rewrite recursive building of LexicographicalVertexOrder to iterative ···` `b48101c`

tpietzsch commented on Sep 19, 2016 • edited — Member

Regarding point (2) *Make (and use) a variant of the generic BreadthFirstIterator that is constructed with a Comparator for ordering children.*

I forgot that `LexicographicalVertexOrder` simply orders children (of the same root) by the edge sequence, so `BreadthFirstIterator` should already do the right thing. So I think that actually we should not do (2) for now to save the overhead of creating two `VertexKey` s for every comparison... IMO the best way to resolve this now is to order the list of roots and initialize the queue of the `BreadthFirstIterator` with the ordered sequence. The rest should just work (also for non-trees).

I just rewrote the `LexicographicalVertexOrder.VertexKey` construction from recursive to iterative to avoid StackOverflowErrors. So at least something came out of this detour... :-D

tinevez commented on Sep 19, 2016 — Member

Yes I think that I mixed the `LexicographicalVertexOrder` with `AlphaNumComparator` .

tinevez added some commits on Oct 13, 2016

`Merge remote-tracking branch 'origin/model-edit-actions' into search-… ···` `ad7d2d8`

`Use AlphaNumComparator to sort roots when searching in TrackScheme.` `85667a6`

`TrackScheme focus dictates where the search starts. ···` `2dc5510`