elm

Using Tests in Elm

Elm Singapore Meetup – hosted by ThoughtWorks

*** Why Testing? ***

How to Test?

Elm website quote

“

We’ve had zero run-time failures, the filesize is ridiculously small, and it runs faster than anything else in our code base.

”

– Jeff Schomay (Pivotal Tracker)

That's all Folks!

A step back: user motivations

1. **What** can I do with this code? (simple examples?)
2. **How** does this work? (good doc?)
3. Is it **reliable**? (are there build and tests status?)
4. Can I **contribute**? (without breaking everything?)

A step back: main builder motivations

1. Is this **useful**?
2. Or is this **funny**?
3. Is it **reliable**? (will others trust my code?)
4. Can I get others to **contribute**? (without breaking everything?)

A step back: to sum up

“ *Tests help building trust and community.* ”

– Me, just now
(and probably someone else before today)

Why Testing?

***** How to Test? *****

Testing even for a small lib?

Question on Elm Discuss:

me

"Best practices to create examples for an elm package?"

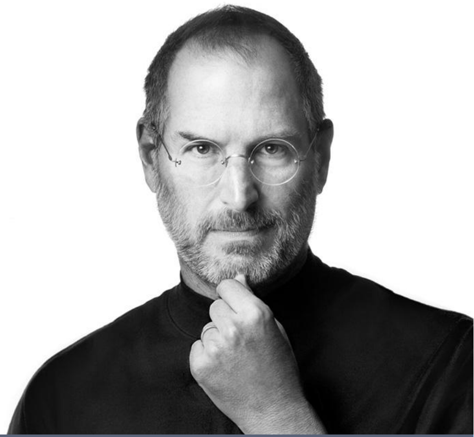
Noah
eeue56

- Use elm-doc-test for examples whereever possible, don't just write documentation examples. ...
- Have a test suite. ...
- If you have views, use elm-html-test. ...
- If you are exposing something which is browser-dependant, ..., You should have an integration test set up that runs your code in a real browser.

Testing even for a small lib?

Documentation examples:	elm-doc-test
Test suite:	elm-test
Views tests:	elm-html-test

“ *There’s an elm testing package for it* ™ ”



elm-doc-test

```
{-| Construct a line segment collinear with the given axis, with its endpoints
at the given distances from the axis' origin point.

    >>> import OpenSolid.Geometry.Types exposing (..)
    >>> import OpenSolid.Axis2d as Axis2d

    >>> 1 + 2
    3

    >>> along Axis2d.x 3 5
    LineSegment2d
        ( Point2d ( 3, 0 )
        , Point2d ( 5, 0 )
        )

-}
along : Axis2d -> Float -> Float -> LineSegment2d
along axis start end =
    LineSegment2d ( Point2d.along axis start, Point2d.along axis end )
```

PS: syntax is evolving very soon

elm-doc-test

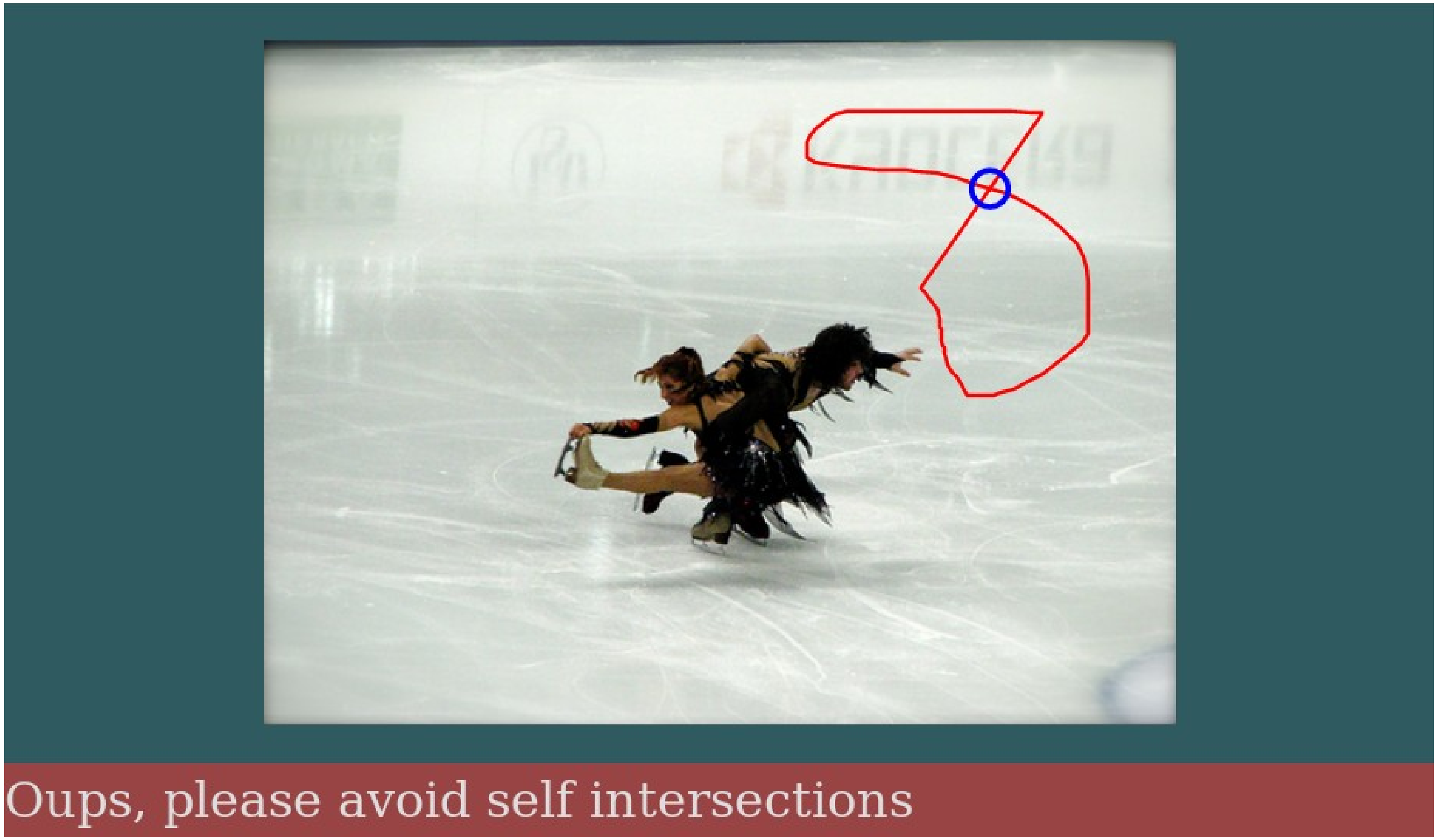
```
$ npm i elm-test -g
$ npm i elm-doc-test -g
$ elm-test init

$ nvim tests/elm-doc-test.json
```

```
{
    "root": "../src",
    "tests": [
        "OpenSolid.LineSegment2d"
    ]
}
```

$\longrightarrow$ elm-test

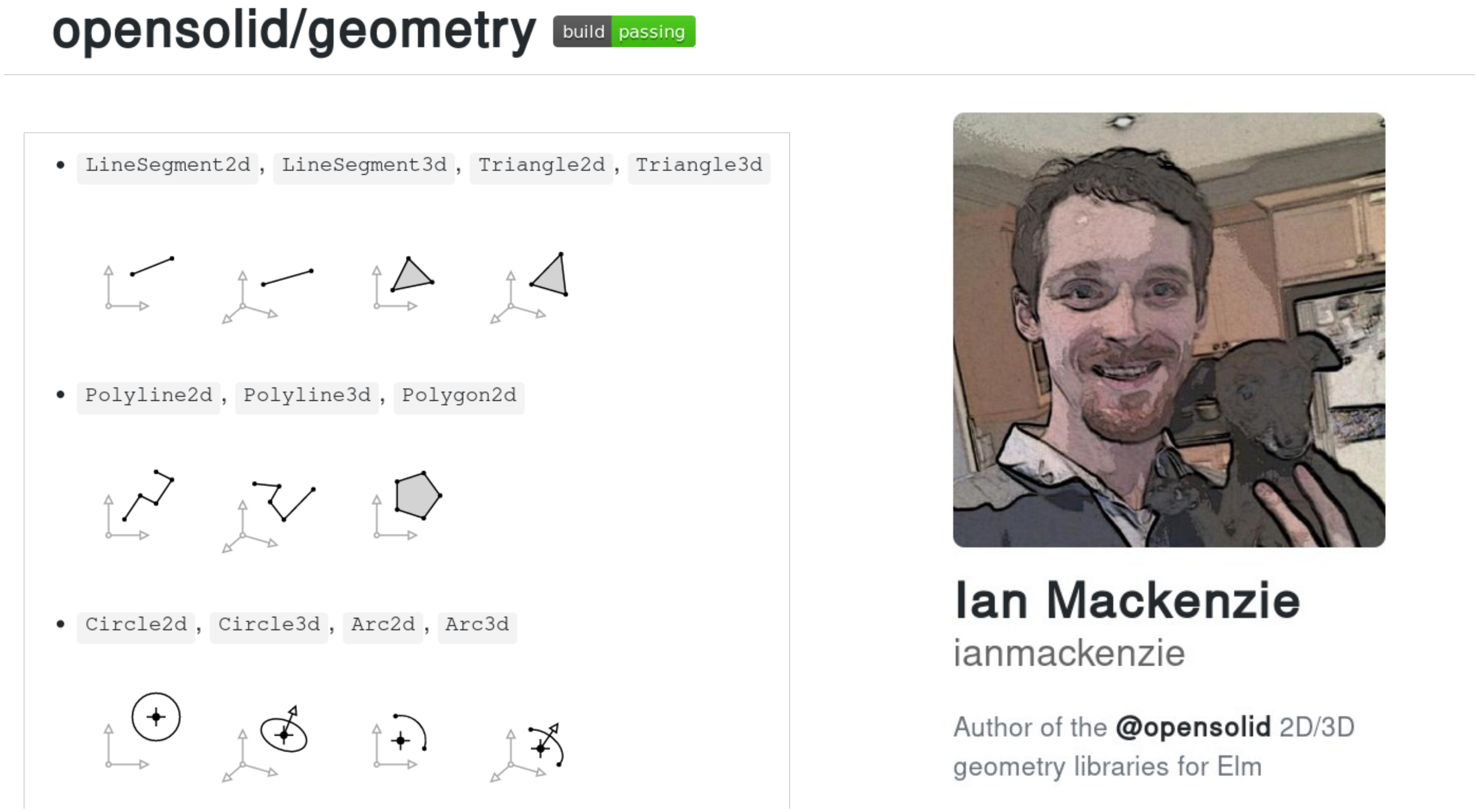
```
$ elm-doc-test && elm-test tests/Doc/OpenSolid/LineSegment2dSpec.elm
```

elm-test



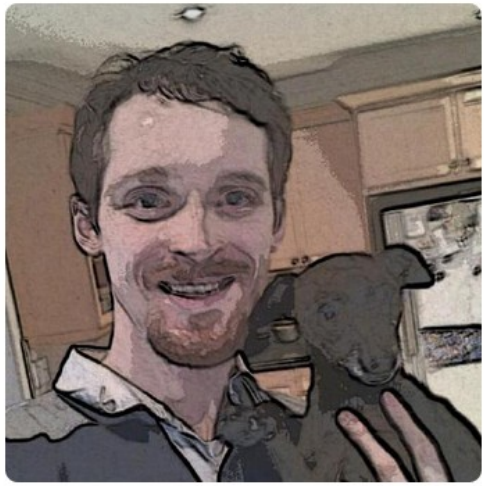
Oups, please avoid self intersections

elm-test



opensolid/geometry build passing

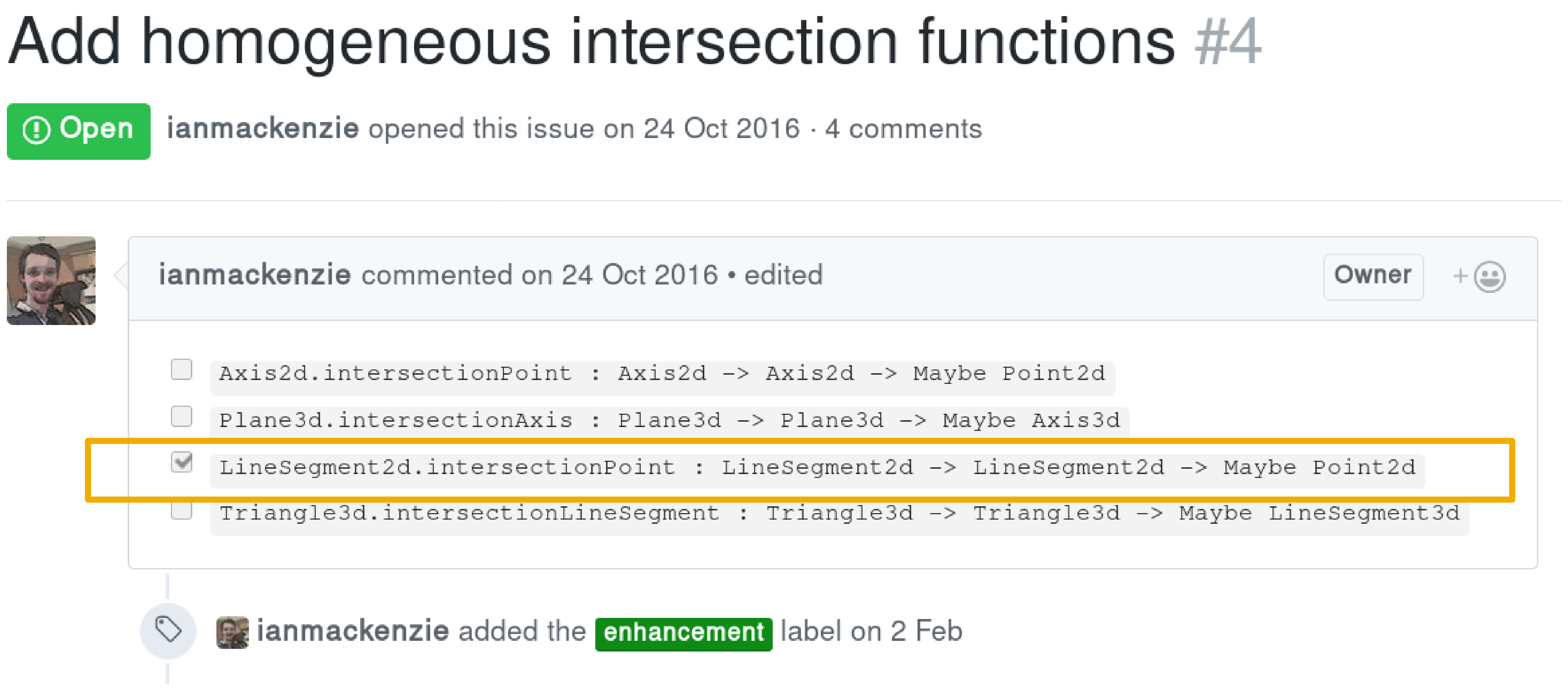
- `LineSegment2d`, `LineSegment3d`, `Triangle2d`, `Triangle3d`
- `Polyline2d`, `Polyline3d`, `Polygon2d`
- `Circle2d`, `Circle3d`, `Arc2d`, `Arc3d`



Ian Mackenzie
ianmackenzie

Author of the **@opensolid** 2D/3D geometry libraries for Elm

elm-test



Add homogeneous intersection functions #4

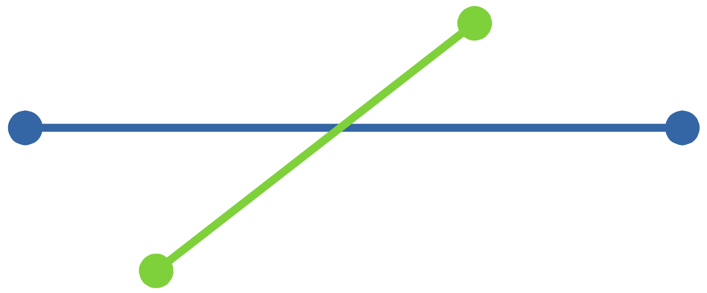
Open ianmackenzie opened this issue on 24 Oct 2016 · 4 comments

ianmackenzie commented on 24 Oct 2016 • edited Owner

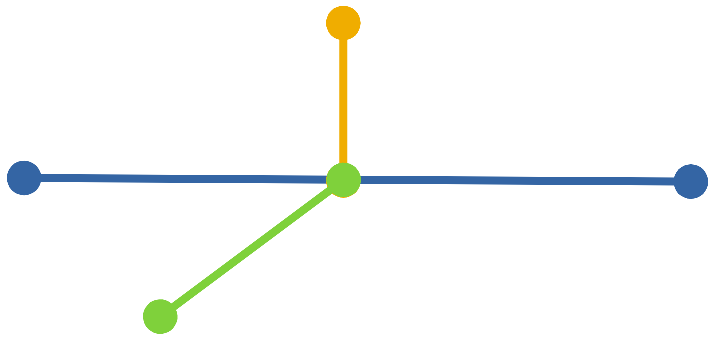
- [ ] `Axis2d.intersectionPoint : Axis2d -> Axis2d -> Maybe Point2d`
- [ ] `Plane3d.intersectionAxis : Plane3d -> Plane3d -> Maybe Axis3d`
- [x] `LineSegment2d.intersectionPoint : LineSegment2d -> LineSegment2d -> Maybe Point2d`
- [ ] `Triangle3d.intersectionLineSegment : Triangle3d -> Triangle3d -> Maybe LineSegment3d`

ianmackenzie added the enhancement label on 2 Feb

elm-test



Relatively easy



???

elm-test

```elm
module LineSegment2d
    exposing
        ( intersectionFindsCoincidentEndpoints
        , intersectionFindsCollinearCoincidentEndpoints
        , intersectionIsSymmetric
        , intersectionOfEqualLineSegmentsIsNothing
        , intersectionOfEqualPointSegmentIsPoint
        , intersectionOfReversedEqualLineSegmentsIsNothing
        , intersectionWorksProperly
        , jsonRoundTrips
        , sharedEndpointOnThirdSegmentInducesAnIntersection
        )
```

elm-test

Tree hierarchy:

```
your_projectd/
  - src/
  - tests/
    - elm-package.json
    - LineSegment2d.elm
    - Doc/...
```

$\longrightarrow$

```elm
import Expect
import Fuzz
import OpenSolid.Geometry...
...
import Test exposing (Test)
...
```

elm-test

```
Test.test : String -> (() -> Expectation) -> Test
Test.fuzz : Fuzzer a -> String -> (a -> Expectation) -> Test
```

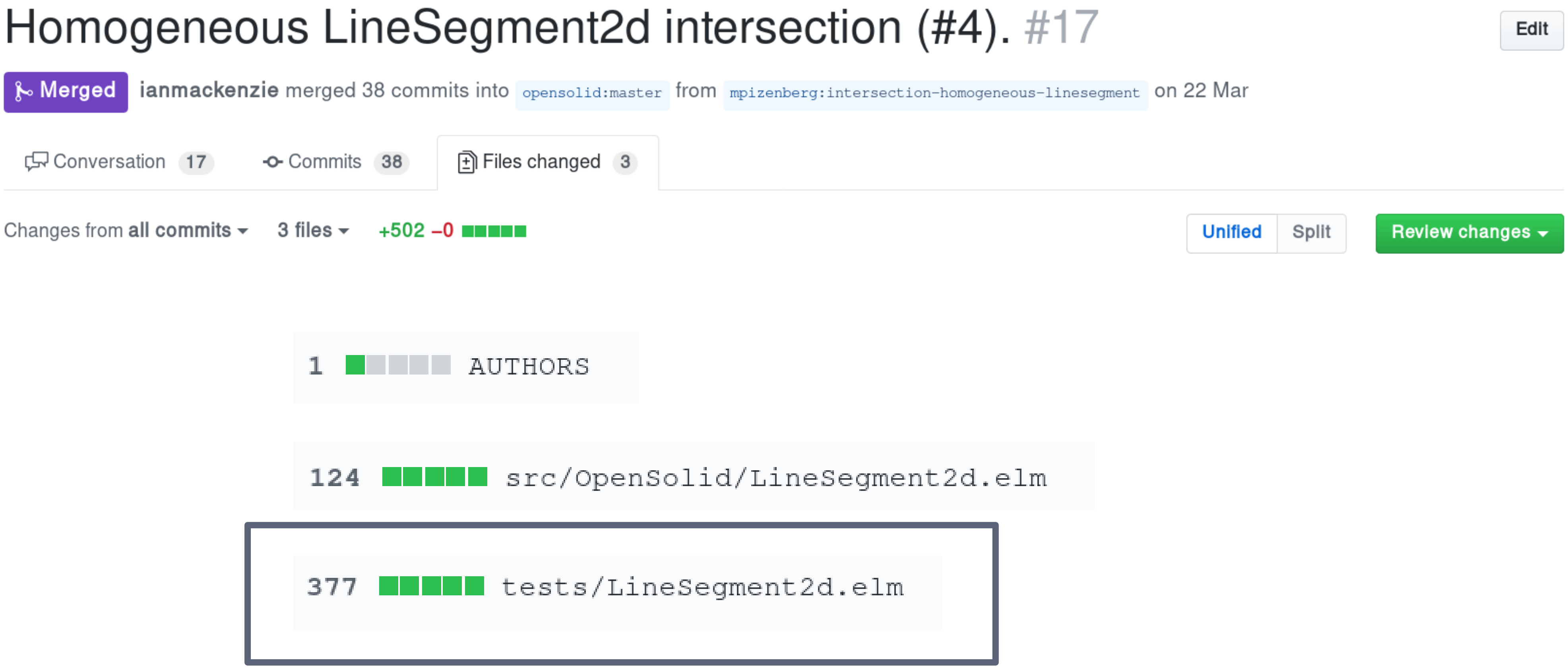
```
intersectionIsSymmetric : Test
intersectionIsSymmetric =
    Test.fuzz2
        Fuzz.lineSegment2d
        Fuzz.lineSegment2d
        "Intersection should be symmetric"
        (\lineSegment1 lineSegment2 ->
            Expect.equal
                (LineSegment2d.intersectionPoint lineSegment1 lineSegment2)
                (LineSegment2d.intersectionPoint lineSegment2 lineSegment1)
        )
```

elm-test

.travis.yml

```
language: node_js
node_js: node

os:
  - linux

env: ELM_VERSION=0.18.0

before_install:
  - echo -e "Host github.com\n\tStrictHostKeyChecking no\n" >> ~/.ssh/config

install:
  - node --version
  - npm --version
  - npm install -g elm@$ELM_VERSION elm-test
  - git clone https://github.com/NoRedInk/elm-ops-tooling
  - elm-ops-tooling/with_retry.rb elm package install --yes

script:
  - elm-test tests
```

elm-test



Homogeneous LineSegment2d intersection (#4). #17

Merged ianmackenzie merged 38 commits into `opensolid:master` from `mpizenberg:intersection-homogeneous-linesegment` on 22 Mar

1	🟩⬜⬜⬜⬜	AUTHORS
124	🟩🟩🟩🟩🟩	src/OpenSolid/LineSegment2d.elm
377	🟩🟩🟩🟩🟩	tests/LineSegment2d.elm

elm-html-test

```elm
import Html
import Html.Attributes exposing (class)
import Test exposing (test)
import Test.Html.Query as Query
import Test.Html.Selector exposing (text, tag)


test "Button has the expected text" <|
    \() ->
        Html.div [ class "container" ]
            [ Html.button [] [ Html.text "I'm a button!" ] ]
            |> Query.fromHtml
            |> Query.find [ tag "button" ]
            |> Query.has [ text "I'm a button!" ]
```

elm-html-test

```
someCasualHtmlView
    |> (1) Query.fromHtml              : Html msg -> Single msg
    |> (2) transform query             : Single msg -> Multiple msg
    |> (3) verify expectation          : Multiple msg -> Expectation
```

Questions ?

References

1. Best practices to create examples for an elm package?, Elm Discuss, https://groups.google.com/d/topic/elm-discuss/rddeM28_C5A/discussion

2. Elm doc tests, Christoph Hermann, https://github.com/stoeffel/elm-doc-test

3. Elm tests, Alex Neslusan (deadfoxygrandpa), Max Goldstein (mgold), Richard Feldman (rtfeldman), …, https://github.com/elm-community/elm-test

4. Line segment intersection example, https://github.com/opensolid/geometry/pull/17

5. Elm html test, Noah Hall (eeue56), https://github.com/eeue56/elm-html-test