

SM2-TES: Functional Programming and Property-Based Testing, Day 7

Jan Midtgaard

MMMI, SDU

Other QuickCheck Frameworks

John Hughes: Certifying Your Car with Erlang

Project Ideas

Project/exam formalities

Other QuickCheck Frameworks

[Your presentations]

John Hughes: Certifying Your Car with Erlang

[Video talk]

Project Ideas

Project ideas (1/3)

Here is first a list of generic proposals:

- Test the last program you wrote
- Test the last library you used

Project ideas (1/3)

Here is first a list of generic proposals:

- Test the last program you wrote
- Test the last library you used
- Is there a program/library/module at work which could benefit from testing?

Project ideas (1/3)

Here is first a list of generic proposals:

- Test the last program you wrote
- Test the last library you used
- Is there a program/library/module at work which could benefit from testing?
- Test software from the last course you took

Project ideas (1/3)

Here is first a list of generic proposals:

- Test the last program you wrote
- Test the last library you used
- Is there a program/library/module at work which could benefit from testing?
- Test software from the last course you took
- Test the implementation of the last non-trivial algorithm you wrote/reused

Project ideas (2/3)

- **Test properties of XML or JSON tools** (both require a generator of random document data, . . . which properties?)

Project ideas (2/3)

- **Test properties of XML or JSON tools** (both require a generator of random document data, . . . which properties?)
- **Test properties of Java's class loading library** (a generator of classes? which properties?)

Project ideas (2/3)

- **Test properties of XML or JSON tools** (both require a generator of random document data, . . . which properties?)
- **Test properties of Java's class loading library** (a generator of classes? which properties?)
- **Test graphical properties of a 2D/3D engine** (a generator of random vectors, matrices, . . . which properties?)

Project ideas (2/3)

- **Test properties of XML or JSON tools** (both require a generator of random document data, . . . which properties?)
- **Test properties of Java's class loading library** (a generator of classes? which properties?)
- **Test graphical properties of a 2D/3D engine** (a generator of random vectors, matrices, . . . which properties?)
- **Test properties of a sound library** (a generator of sound clips? which properties?)

Project ideas (3/3)

- **Model-based GUI testing**: a language of 'graphical actions' (push button, select item in dropdown menu, ...) and an state machine generator to produce them

Project ideas (3/3)

- **Model-based GUI testing**: a language of 'graphical actions' (push button, select item in dropdown menu, ...) and an state machine generator to produce them
- **Test (parts of) a standard library** (Sets, maps, hashtables, queues, ...)

Project ideas (3/3)

- **Model-based GUI testing**: a language of 'graphical actions' (push button, select item in dropdown menu, ...) and an state machine generator to produce them
- **Test (parts of) a standard library** (Sets, maps, hashtables, queues, ...)
- **Test a graph library** (which properties?)

Project ideas (3/3)

- **Model-based GUI testing**: a language of 'graphical actions' (push button, select item in dropdown menu, ...) and an state machine generator to produce them
- **Test (parts of) a standard library** (Sets, maps, hashtables, queues, ...)
- **Test a graph library** (which properties?)
- **Test a REST API**
(which properties? general or domain-specific?)
 - Concrete suggestion: Orbit's REST API

Project ideas (3/3)

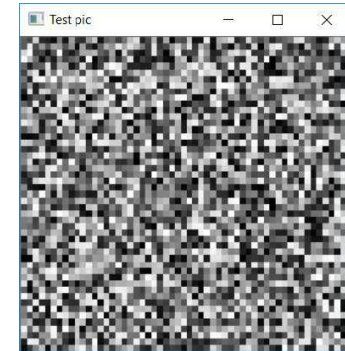
- **Model-based GUI testing**: a language of 'graphical actions' (push button, select item in dropdown menu, ...) and an state machine generator to produce them
- **Test (parts of) a standard library** (Sets, maps, hashtables, queues, ...)
- **Test a graph library** (which properties?)
- **Test a REST API**
(which properties? general or domain-specific?)
 - Concrete suggestion: Orbit's REST API
- **Test a protocol** (HTTP, WebSockets, ... ?)
(which properties – model based?)

Previous project: Testing OpenCV

OpenCV is a big C++ library for image manipulation and analysis. In 2018: a project on testing parts of it...

A first generator of images:

100*100 px, each w/256 shades of gray.

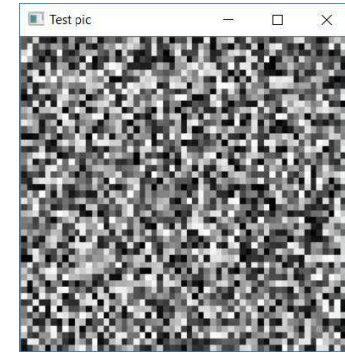


Previous project: Testing OpenCV

OpenCV is a big C++ library for image manipulation and analysis. In 2018: a project on testing parts of it...

A first generator of images:

100*100 px, each w/256 shades of gray.



Q: Chance of generating uniform black/white/gray image?

Previous project: Testing OpenCV

OpenCV is a big C++ library for image manipulation and analysis. In 2018: a project on testing parts of it...

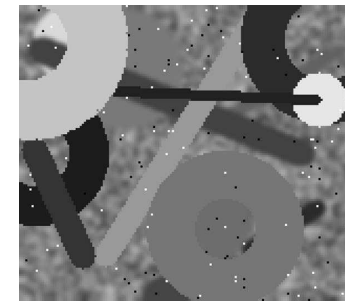
A first generator of images:

100*100 px, each w/256 shades of gray.



Q: Chance of generating uniform black/white/gray image?

A revised generator based on symbolic commands (after discussion with Jan):



Properties:

Initially: differential testing w/prev. reimplementation

Later: domain-specific (color inversion, cut-off, ...)

Project/exam formalities

Project/exam formalities

- **Mar 24** (at the latest)
 - Hand in a project description (talk to Jan first)
 - After **a green light** you can start project work
 - You can do a group project (1-4 members)
- **May 11**, you give a preliminary project presentation (topic, prelim. results and ideas, ...)
- **May 31**, hand in project report (and code)
- **June ??**, oral exam: project presentation + answer questions
 - You receive a combined grade for the project report + presentation + answers
 - Measure of success: ability to apply QuickCheck (to a project of your choice)

The project report

Explain

- your project domain
- how QuickCheck works in the port you've chosen to work with
- how your generator works
- which properties you test for
- your experiments and results

The project report

Explain

- your project domain
- how QuickCheck works in the port you've chosen to work with
- how your generator works
- which properties you test for
- your experiments and results

For length, I expect more from, e.g., a 4-person group than 1 person. Something like:

$5 + (n - 1) * [2 - 3]$ where n is the number of group members

e.g., ~ 5 pages for 1 person and $\sim 11-14$ pages for 4 people.

Take these numbers as estimates: $\pm 10-15\%$ can be reasonable, but 50% or 250% should be reconsidered.

Summary and conclusion

- We've seen **a number of frameworks** expressing the core ideas of QuickCheck in other languages
- We've seen an **extensive application** of the techniques to Volvo/AUTOSAR
- We discussed various project ideas
- Perhaps you got **an idea for a project** from one of the above?
- We've discussed the project/exam formalities