

Coding Dojo: Mars Rover

Oct. 2014



Peter Kofler, 'Code Cop'

@codecopkofler

www.code-cop.org

Peter Kofler

- Ph.D. (Appl. Math.)
- Professional Software Developer for 15 years
- “fanatic about code quality”
- I help development teams



Coding Dojo?
Expectations?

Coding Dojo Mindset

- Safe place outside work
- We are here to learn
- Need to slow down
- Focus on doing it right
- Collaborative Game



Two Basic Rules

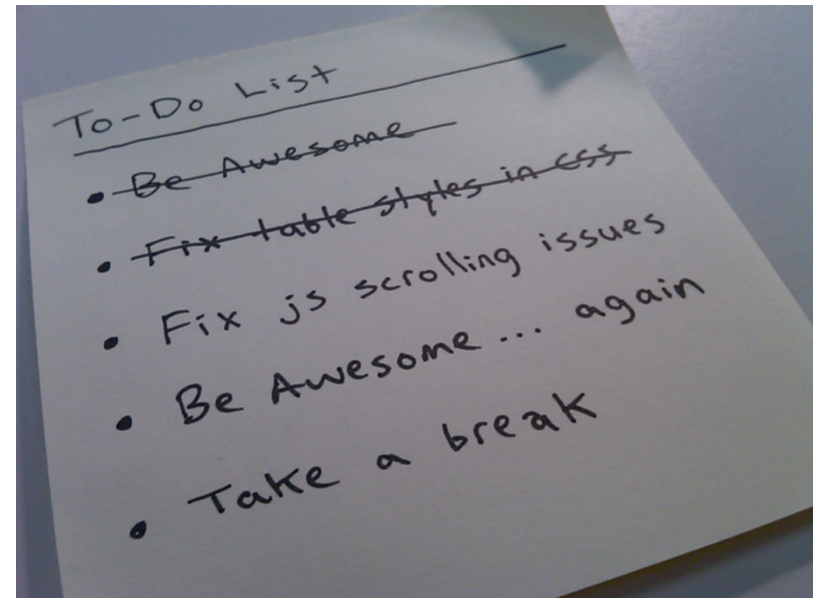
- Collaborative = Pair Programming
 - “Randori” (pairing on the projector)
 - or programming in pairs
- Test Driven Development
 - think about tests
 - write the test first

(Maybe Some) Constraints

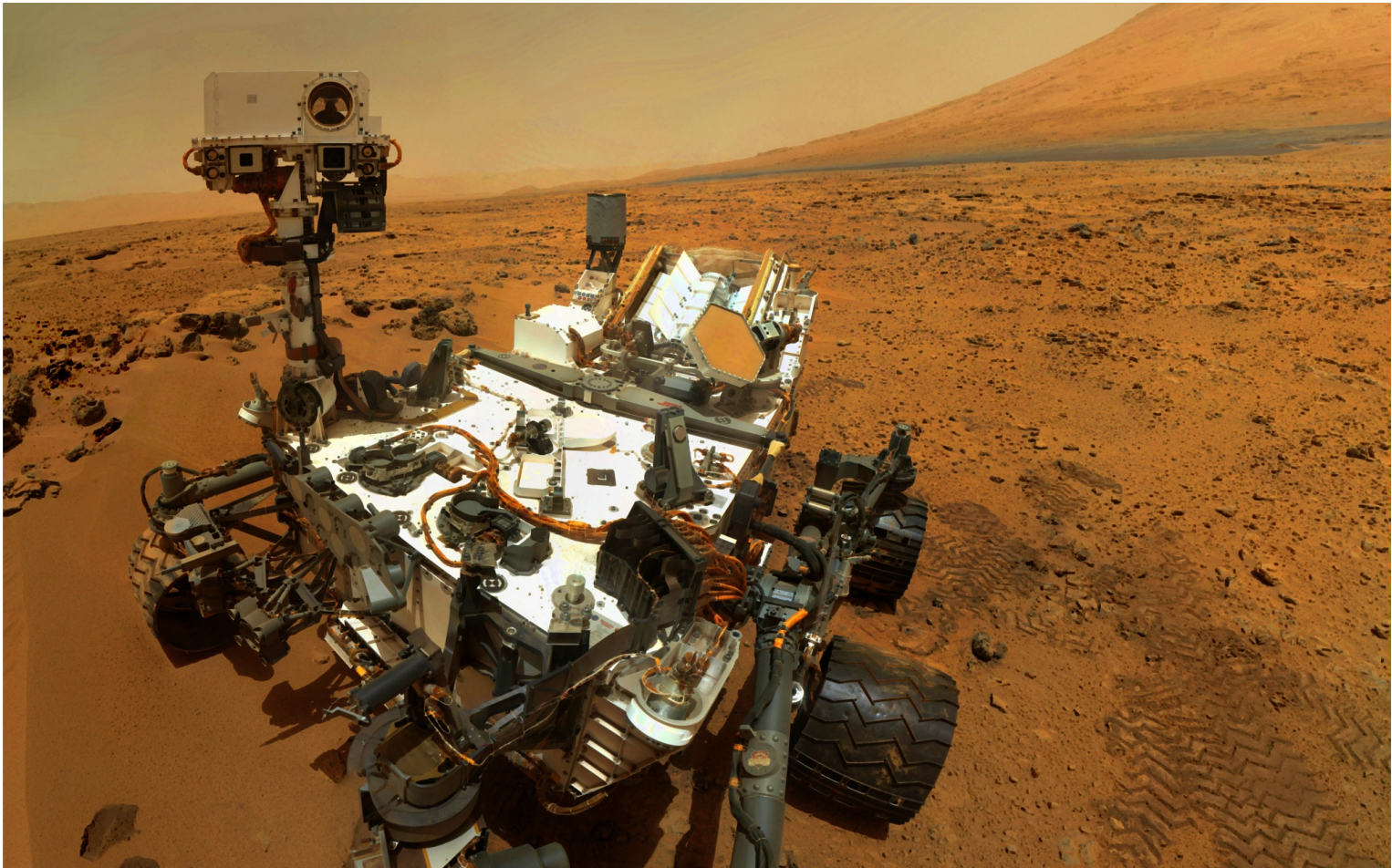
- Challenges during a dojo or coderetreat.
- Moving to the extreme is a way of learning
- Examples
 - Missing Tool (No Mouse, ...)
 - Missing Feature (No IFs, ...)

Dojo Structure

- Introduction 15'
- Coding 45'
- Interim (Retrospective) 15'
- Break 10'
- Coding 45'
- Retrospective 15'



Assignment

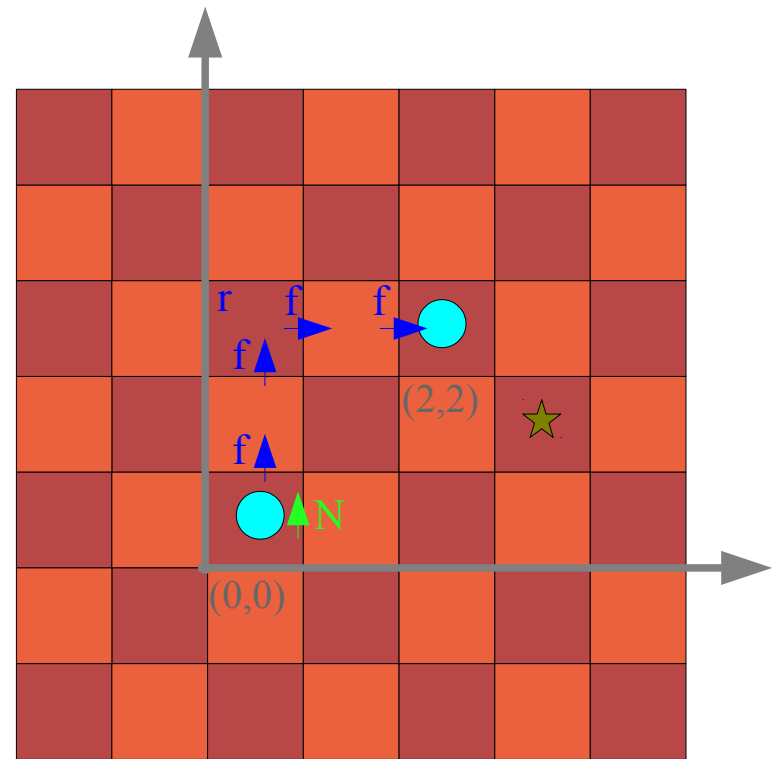


Mars Rover

- Develop an API that moves a rover around on a grid.
- You are given the initial starting point (x,y) of a rover and the direction ('N','S','E','W') it is facing.
- The rover receives a character array of commands.
- Move the rover forward/backward ('f','b').
- Turn the rover left/right ('l','r').
- Wrap the grid. (Planets are spheres after all.)
- Detect obstacles before each move to a new square. If a sequence of commands encounters an obstacle, the rover moves up to the last possible point and reports obstacle.

Mars Rover Example

- The **rover** is on a **100x100** grid
- at location **(0, 0)**,
- facing **"N"ORTH**.
- The rover is given commands **"ffrff"**
- and should end up at **(2, 2)**.



Remember the Rules!



Pair Programming & TDD

- regular Pair Programming
 - do not talk for too long
 - do not interrupt the other
 - no “keyboard hugging”
- use TDD (or at least “sort of” TDD)
 - write a test before you write code
 - refactor mercilessly
 - no debugger

With a Constraint



“Yes, and ...”

- When working in pairs none of the pairs is allowed to delete the code of the other person.
- We are looking for a common solution, that both of the pairs will agree.
- Learn to build on top of the ideas already presented.

“Yes, and ...” Rules

- improve existing code of the other
- can not delete code of the other
 - whenever one tries to delete code, say:
“You should improve the existing one.
Please go back and say Yes, and... I will do...”
 - Can not be angry on request of not deleting code.
 - Only unused code can be deleted.
 - Both must agree that the code can be deleted.

Prepare

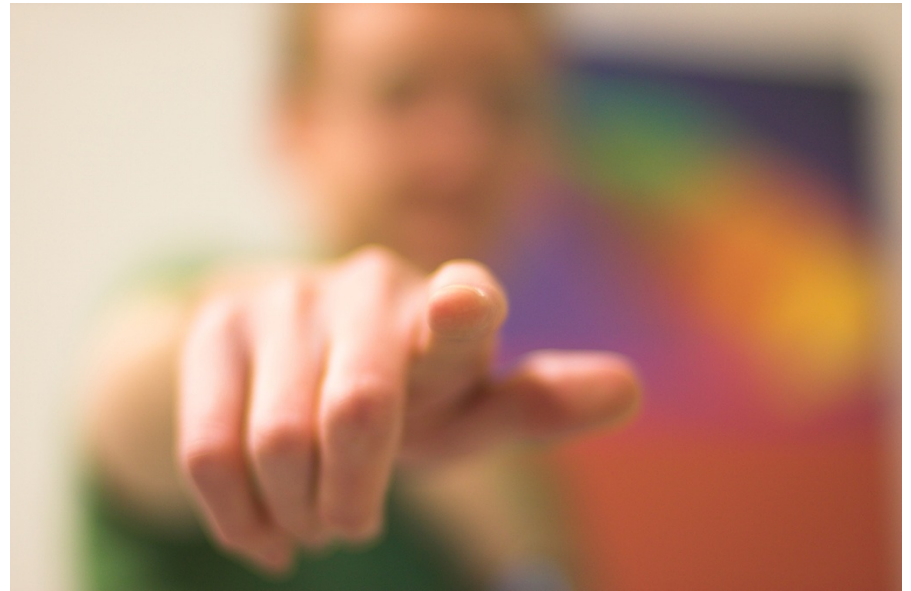
- Find a pair.
- Choose a programming language.
- Set up the environment.
 - Create new project.
 - Add testing framework.
- Check requirements.
- Implement Mars Rover.

Don't Focus on
Getting it Done.
Focus on Doing
It Perfectly.

→ Practice

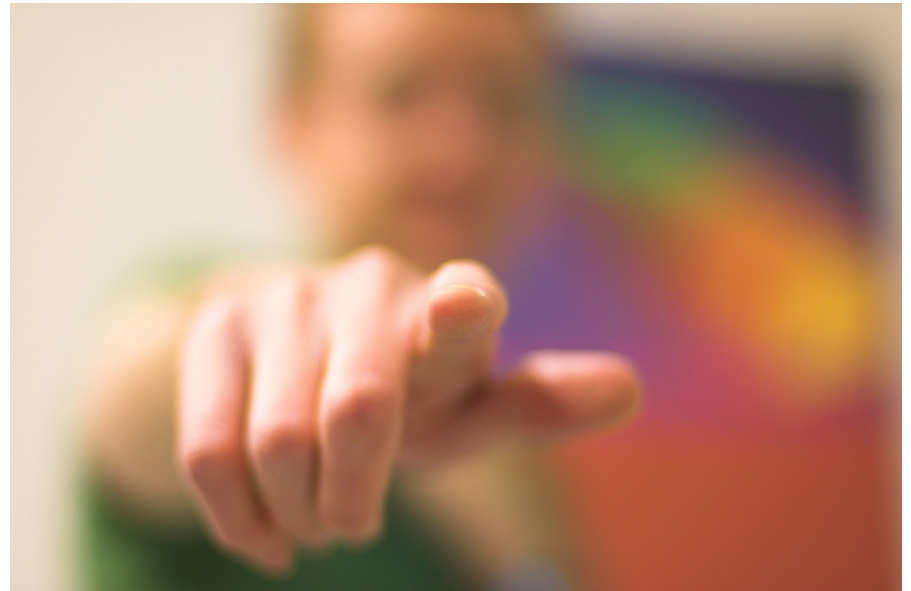
Short Closing Circle

- How did it work out?
- How do you feel?
- What did you learn and want to share with the group?



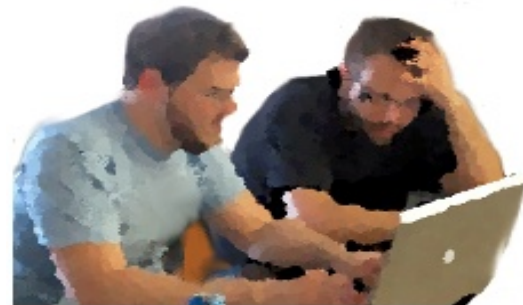
Full Closing Circle

- What did you learn today?
- What surprised you today?
- What will you do differently in the future?



Global Day of Code Retreat

- A world-wide event celebrating passion and software craftsmanship.
- GDCR 2014:
15th November
- Follow #GDCR14



Coderetreat

- honing the craft together

<http://globalday.coderetreat.org/>



Peter Kofler

@codecopkofler

www.code-cop.org

Game by

Adrian Bolboaca

@adibolb

<http://blog.adrianbolboaca.ro/2013/12/pair-programming-game-yes-and/>

CC Images

- Rover Selfie <http://www.flickr.com/photos/gsfcr/7971341456>
- Dojo
<http://www.flickr.com/photos/49715404@N00/3267627038/>
- Todos
<http://www.flickr.com/photos/kylesteeddesign/3724074594/>
- Curiosity <http://www.flickr.com/photos/mendhak/8162305237>
- Rule <http://www.flickr.com/photos/phunk/4188827473>
- Model <http://www.flickr.com/photos/eldave/8430367749>
- Wants you <http://www.flickr.com/photos/shutter/105497713/>