# Seattle.rb Workshop

Have laptop, will code!

March 2015

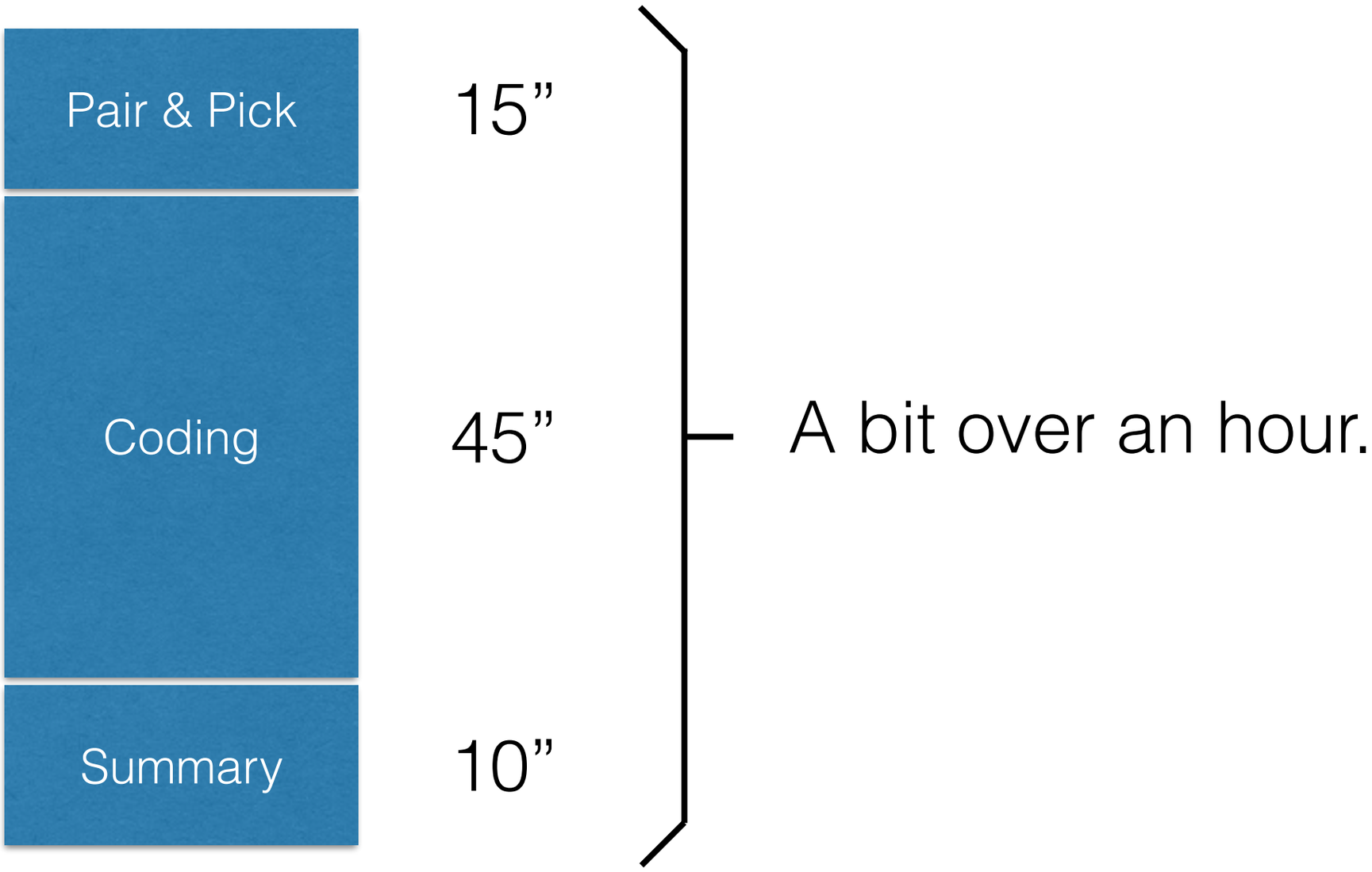
# What?

Code a kata together to practice and learn.

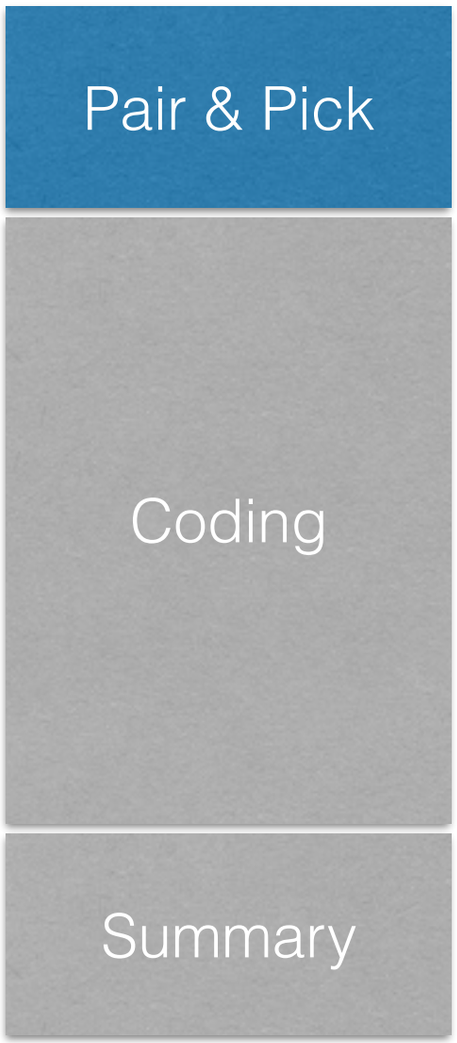
You choose the level of difficulty.

It is about how we code, the process,
not the end product.

# How long?



# How will it work?

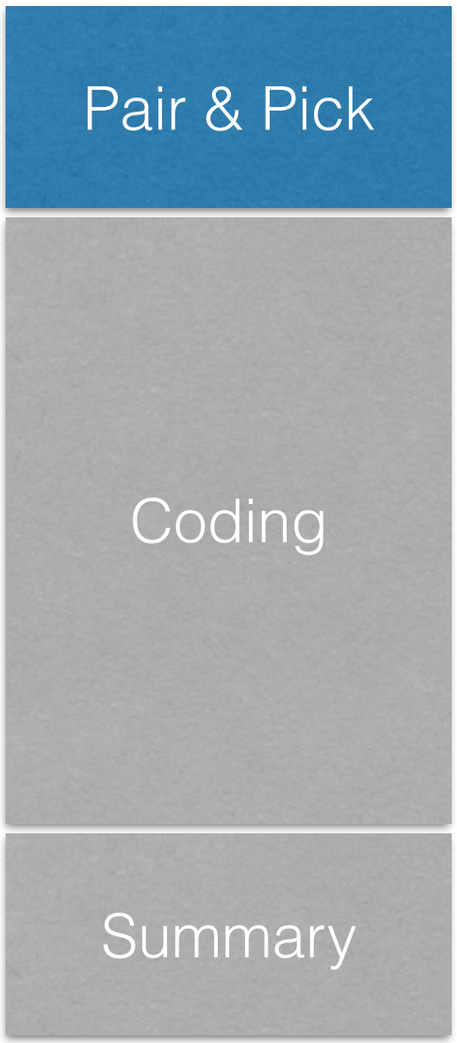



**Step 1: Pair Up!**

By experience level.

Yet totally flexible.

# How will it work?

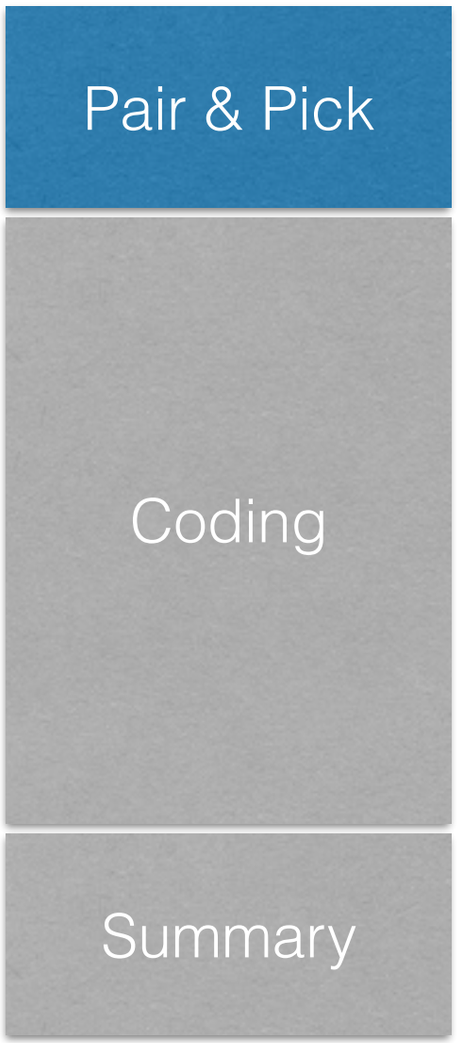



**Step 2: Pick your poison!**

The exercise is to recode…

…a kata you know…

# How will it work?



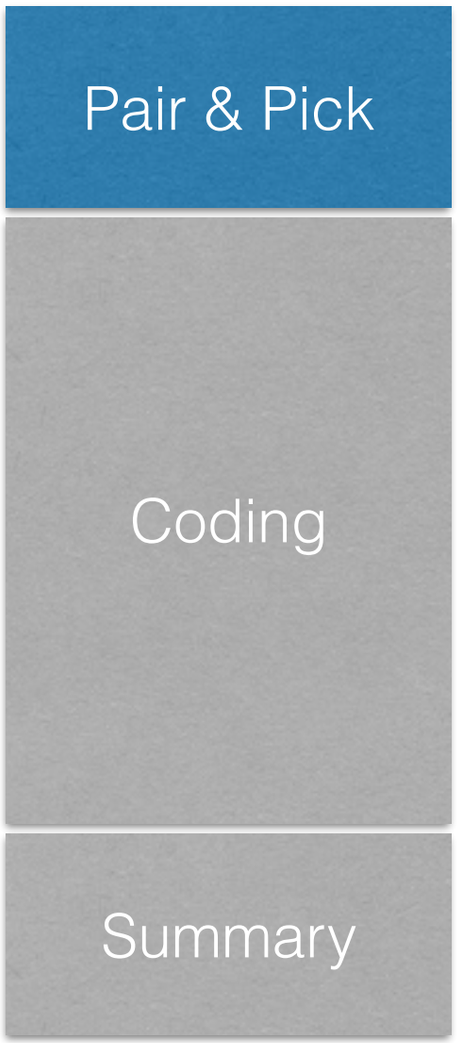

**Step 2: Pick your poison!**

The exercise is to recode…

…a kata you know…

…with added restriction/s

# How will it work?




**Step 2: Pick your poison!**

The exercise is to recode…

…a kata you know…

…with added restriction/s

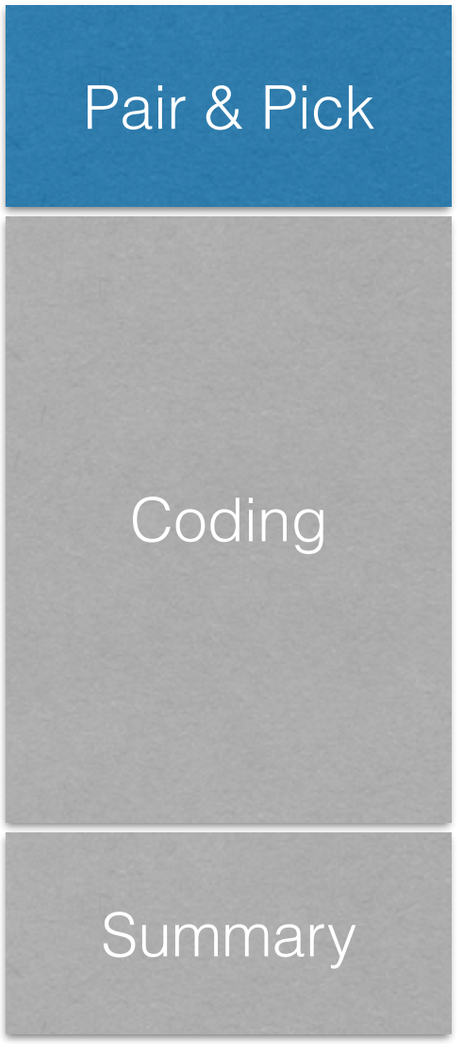
…randomized!!

# Wheel of Misfortune


Example Restrictions:

- mute ping pong
- no conditionals
- no primitives as I/O
- methods <= 3 lines
- no return values
- 1
- 2

# How will it work?

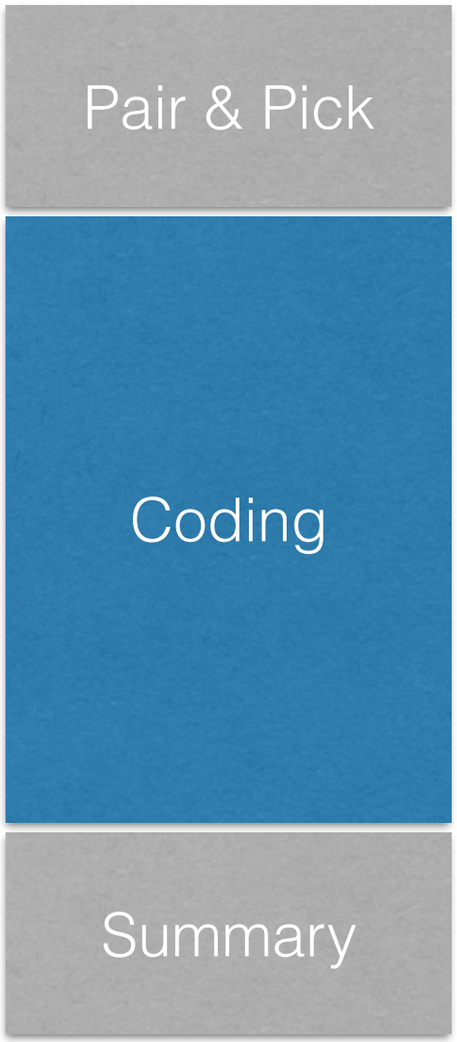



**Fear not!**

- You can pick your restriction,
- …or let fate decide (i.e. wheel!),
- …or come up with your own,
- …or even choose to code in Scheme!

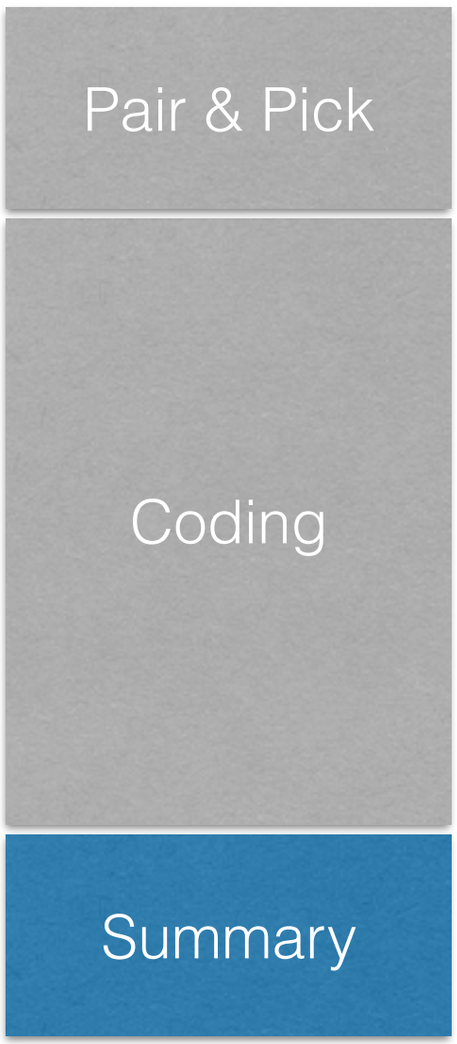
**You and your partner decide.**

# How will it work?




**Coding** for 45 minutes.

# How will it work?




**Summary**

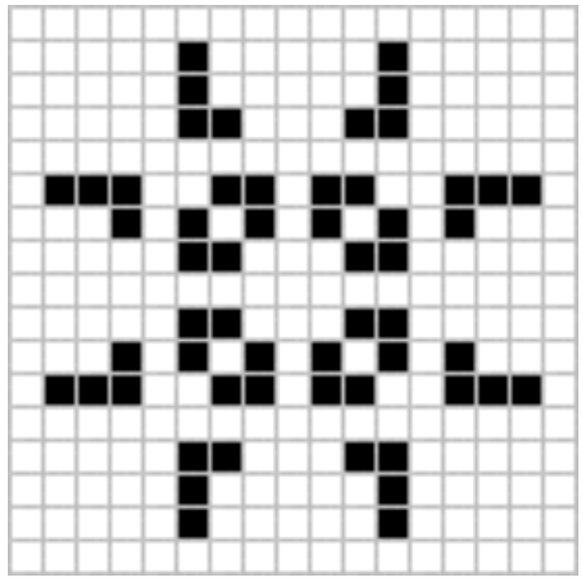
Volunteer basis.

Step up and share conclusions.

Show off cool code.

# Which kata?

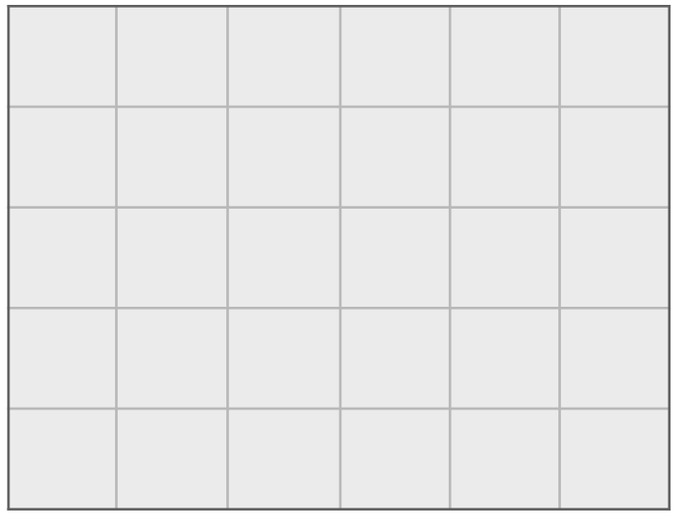
**Conway's Game of Life (GoL):**



- Easy to code, yet full of subtleties.
- Set up an initial pattern in the board.
- Run program and system evolves through generations.

# Game of Life

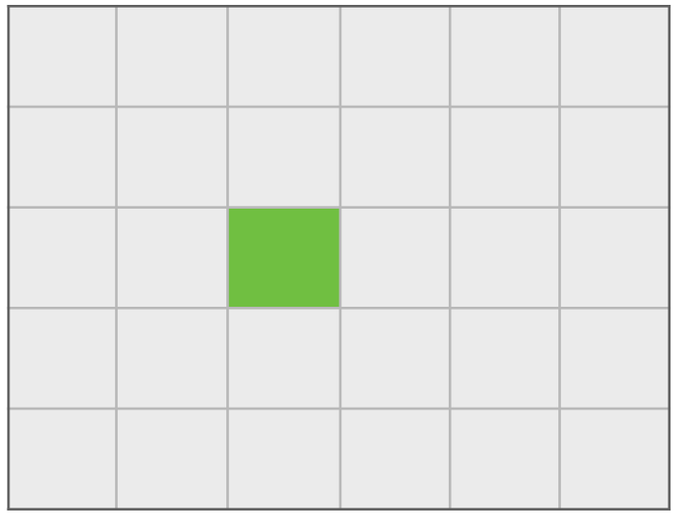
An infinite two-dimensional grid of square cells.



# Game of Life

An infinite two-dimensional grid of square cells.

Each cell is in one of two possible states, dead or alive.

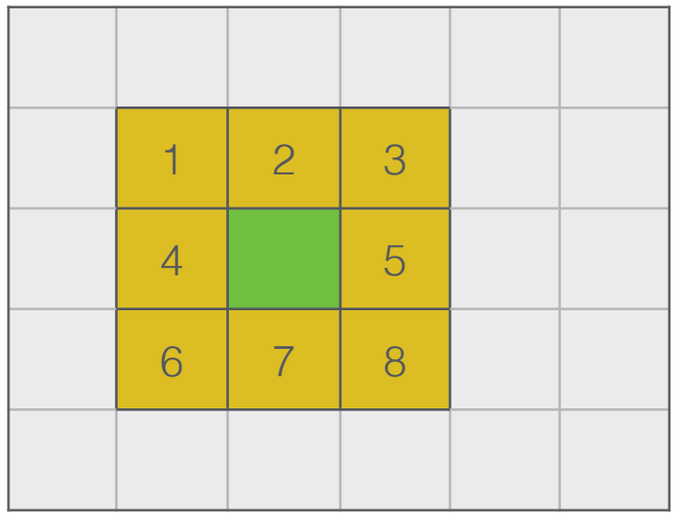


# Game of Life

An infinite two-dimensional grid of square cells.

Each cell is in one of two possible states, dead or alive.

Every cell interacts with its eight neighbors.



# Game of Life

An infinite two-dimensional grid of square cells.

Each cell is in one of two possible states, dead or alive.

Every cell interacts with its eight neighbors.

At each step in time, 4 rules decide which cells live and die.

# Game of Life

An infinite two-dimensional grid of square cells.

Each cell is in one of two possible states, dead or alive.

Every cell interacts with its eight neighbors.

At each step in time, 4 rules decide which cells live and die.

Births / deaths happen simultaneously in a tick of the clock.

# It’s Alive!

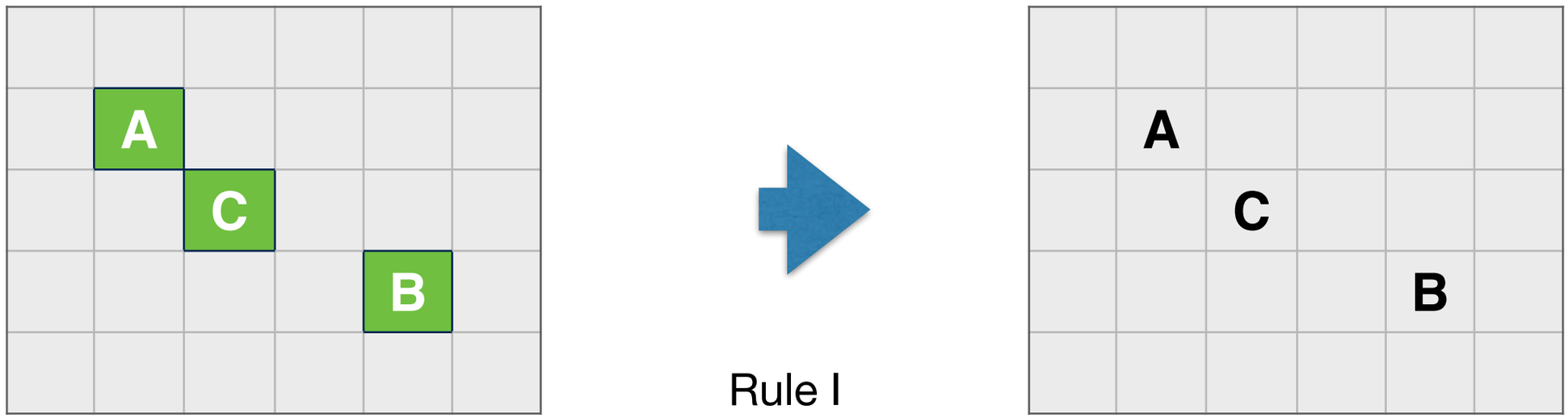
Visual example of how it looks

# It’s Alive!

Spaceship

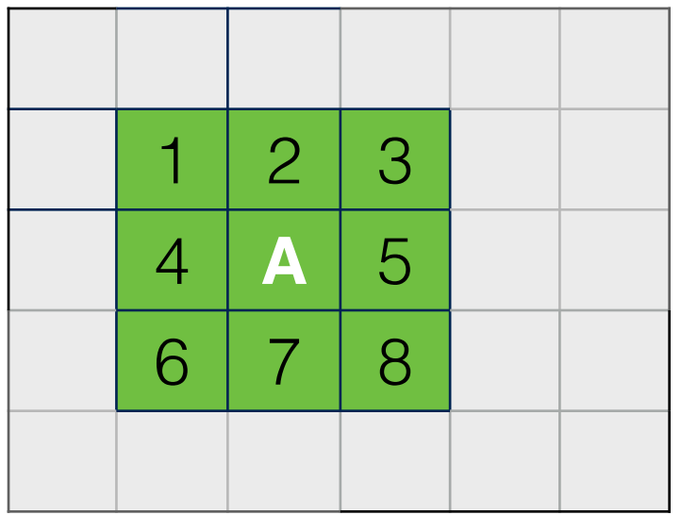
# Game of Life Rules

I. A live cell with less than 2 live neighbors dies: underpopulation.

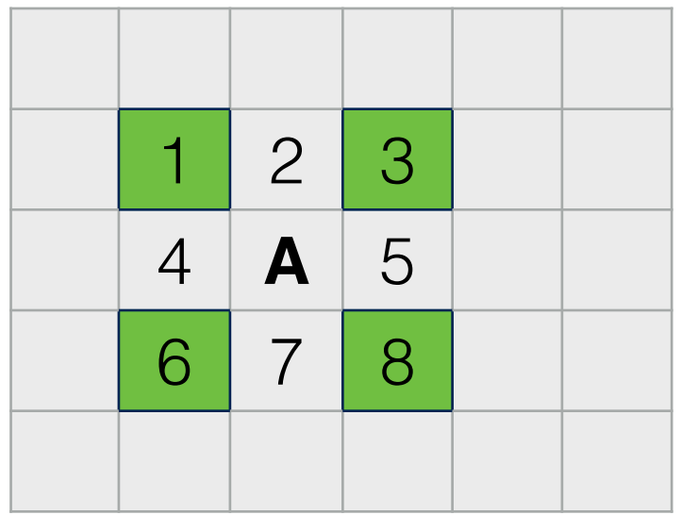


# Game of Life Rules

II. A live cell with more than 3 live neighbors dies: overcrowding.

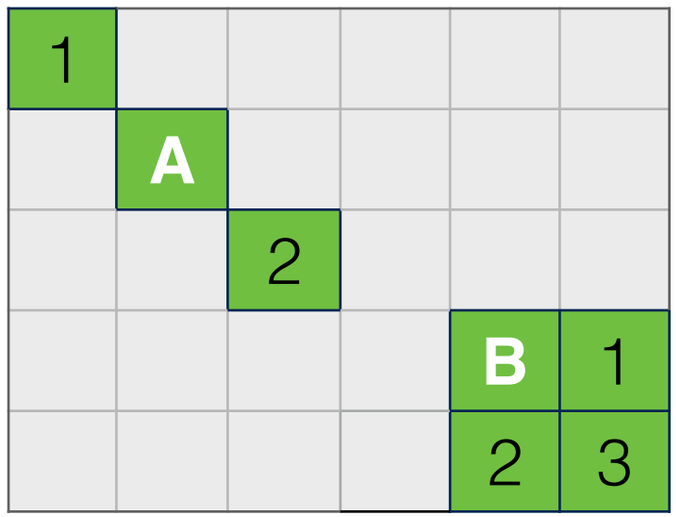



Rules I & II

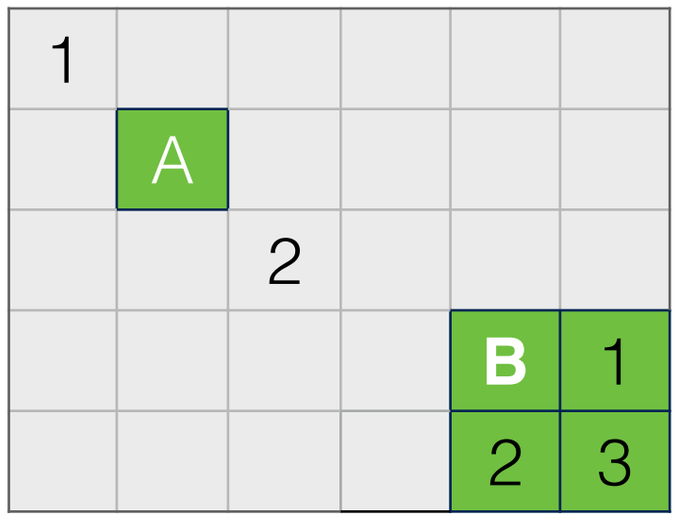


# Game of Life Rules

III. A live cell with 2 or 3 live neighbors lives on.

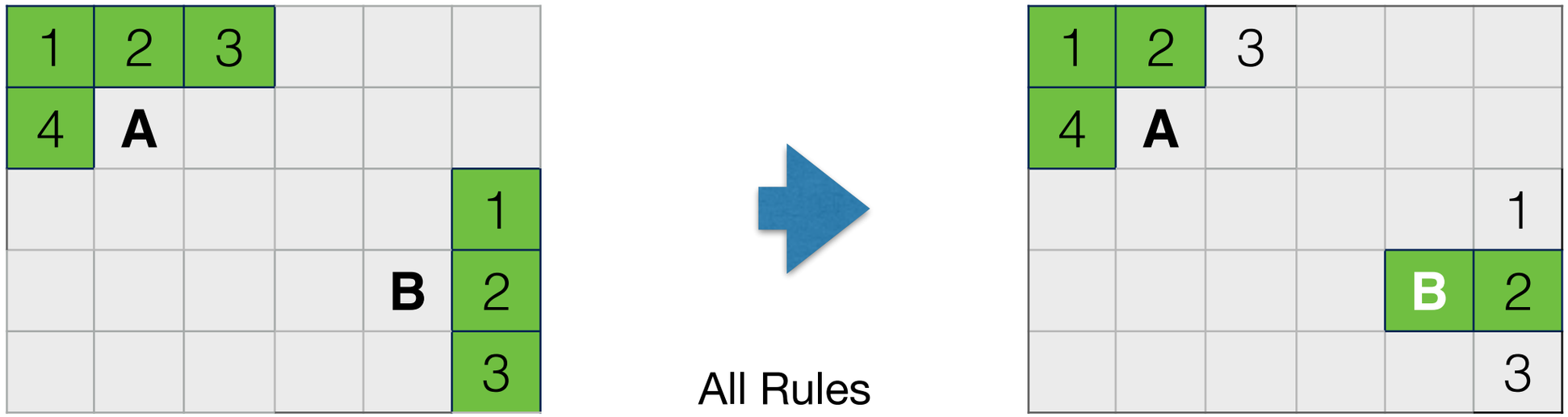



Rules I, II & III



# Game of Life Rules

IV. A dead cell with **exactly** 3 live neighbors becomes alive!



# Game of Life Rules

A live cell with less than 2 live neighbors dies.

A live cell with more than 3 live neighbors dies.

A live cell with 2 or 3 live neighbors lives on.

A square with 3 live neighbors becomes alive.

**Rules ==> Testing !**

# Fork it!

https://github.com/SeaRbSg/workshops

- Example code (with testing),
- visualization code from Ryan Davis,
- and many more links (code, history, etc).

sotoseattle@gmail.com