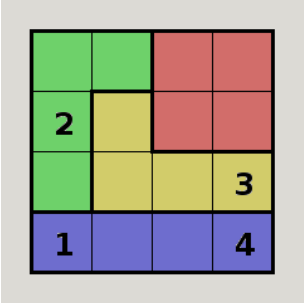
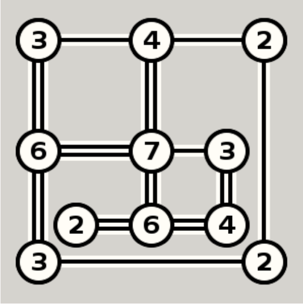
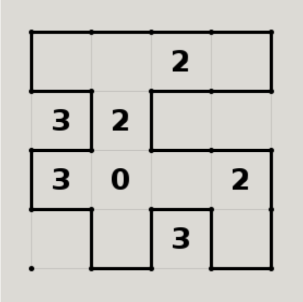
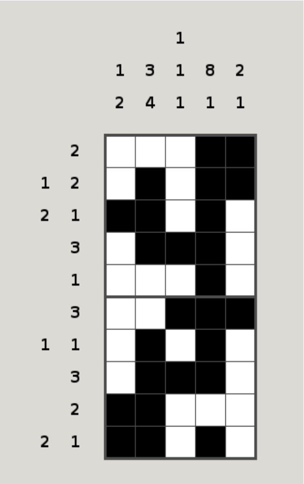
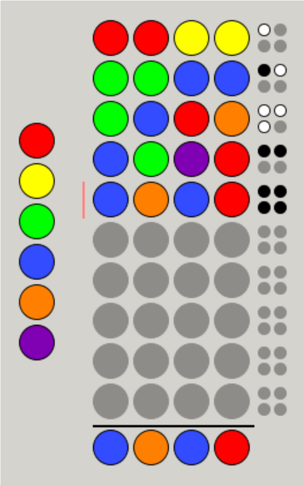


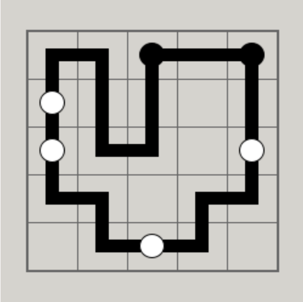


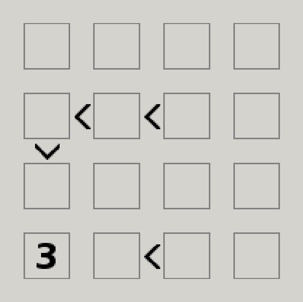




Simon Tatham‘s Portable Puzzle Collection







Steffen Bauer, Linux User Group Frankfurt, 23.02.2021

Puzzles, not jigsaws

This talk is **not** about **Jigsaws** ...

(‚false friend‘ German/English!)

	3							
			1	9	5			
		8					6	
8				6				
4			8					1
				2				
	6					2	8	
			4	1	9			5
							7	

... but about **Logic Puzzles**

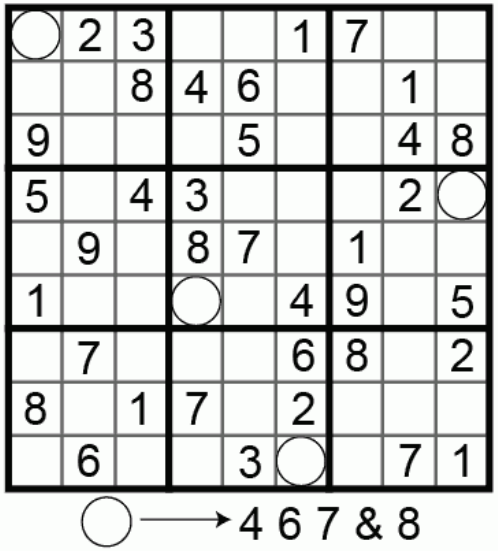
Sudoku

Sūdoku **数独**

Sūji wa dokushin ni kagiru 数字は独身に限る

„The numbers must stay single“

- Inventor: **Howard Garns** (1905 – 1989)
- Original name: *„Number Place“*
- Made popular by **Nikoli** mid-1990s as *„Sudoku“*

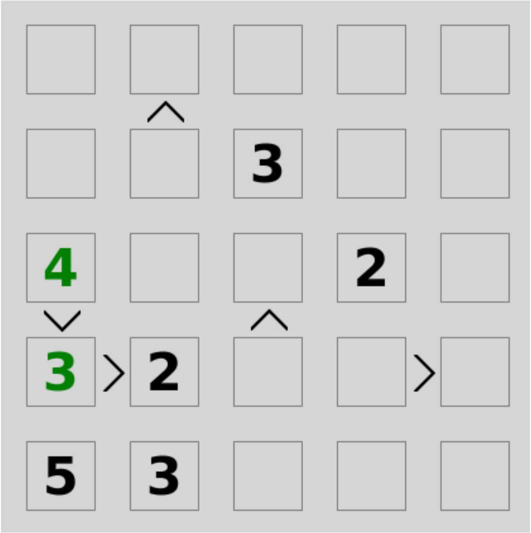


○	2	3			1	7		
		8	4	6			1	
9				5			4	8
5		4	3				2	○
	9		8	7		1		
1			○		4	9		5
	7				6	8		2
8		1	7		2			
	6			3	○		7	1

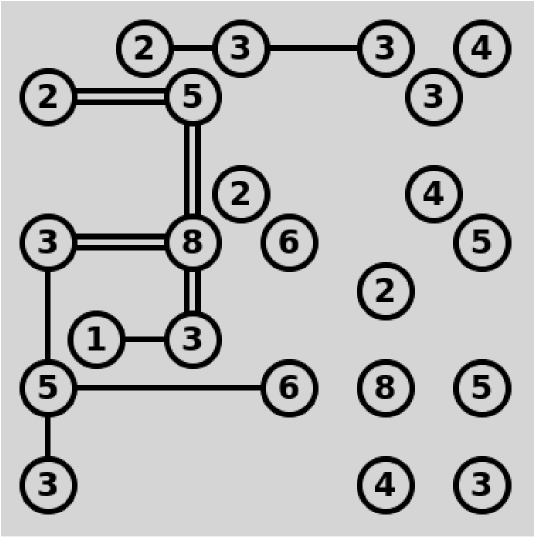
○ → 4 6 7 & 8

Dell Pencil Puzzles & Word Games #16, P.6, 1979

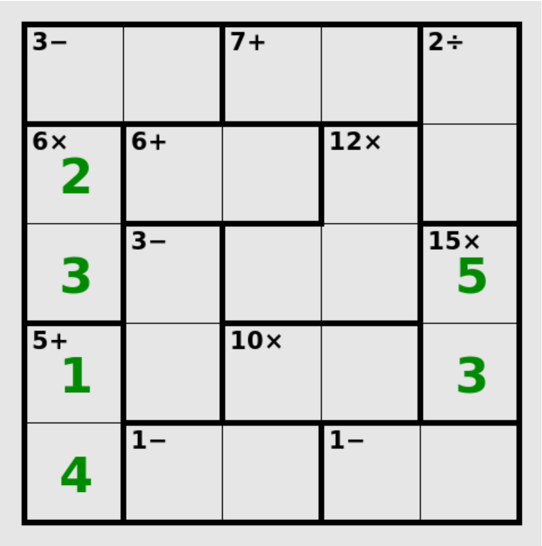
Japanese puzzles



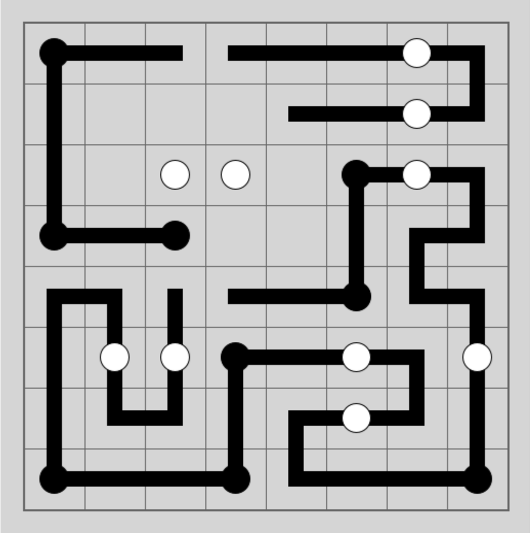
Futoshiki
(‚Inequality‘)



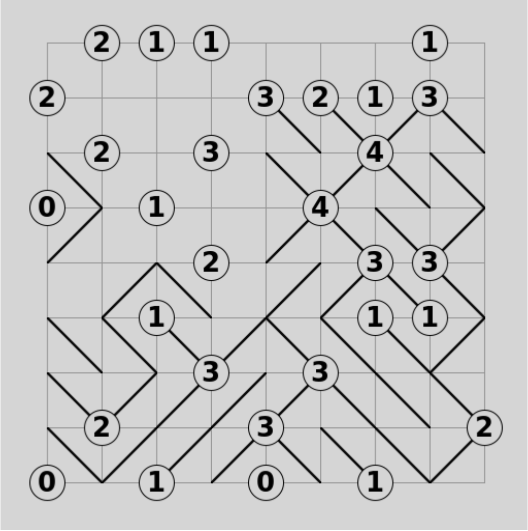
Hashiwokakero
(‚*Building bridges*')



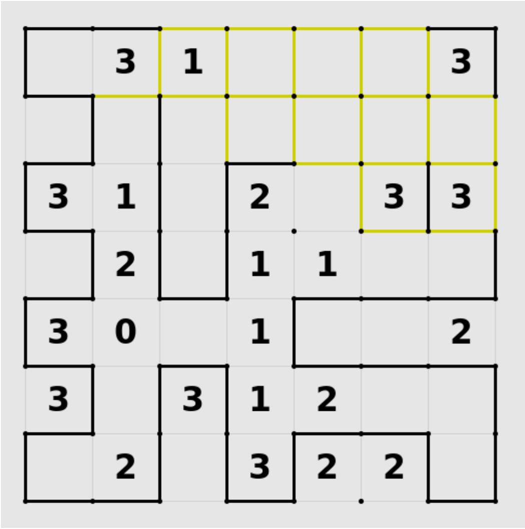
Kendoku
(‚*Smart number*')



Shinju / Masyu
(‚*Pearls*')



Gokigen Naname
(‚*Approaching skewness*')



Suriza rinku
(‚*Slither - Link*')

Simon Tatham

British programmer (*3.5.1977)

Employed at ARM Holdings.

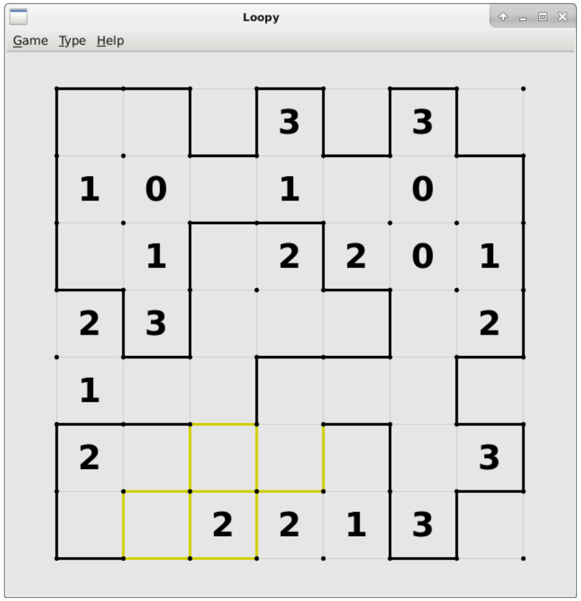
Important enough for a Wikipedia page.

Known for:

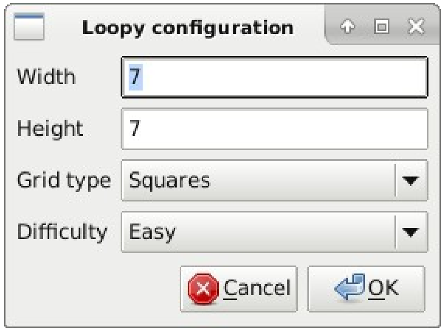
- **PuTTY** (Terminal console / SSH)
- Initiator of **NASM** (Netwide Assembler)
- Often quoted essay „How to report bugs effectively“
- Linux Kernel: Contributed **Console colour codes**
- Collection of small puzzle games (**SGT Puzzles**)

Simon Tatham‘s Portable Puzzle Collection

- Started in 2004 by Simon Tatham
- Motivation: Collection of little games to waste some time
- **Main feature**: *Configurable* random *generated* levels (unlimited games)
- Contains (as of 02.2021) 39 official and ~17 inoffical games
- Ported to dozens of platforms, especially *portable* devices
- Released under MIT license



			3		3	
1	0		1		0	
	1		2	2	0	1
2	3					2
1						
2						3
		2	2	1	3	



Width	7
Height	7
Grid type	Squares
Difficulty	Easy

https://www.chiark.greenend.org.uk/~sgtatham/puzzles/

MIT License (most popular permissive free software license)

Copyright (c) <YEAR> <COPYRIGHT HOLDER>

Portions copyright <COPYRIGHT HOLDERS>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

TL;DR: You can do pretty much anything you like with the game binaries or the code, except pretending you wrote them yourself, or suing me if anything goes wrong.

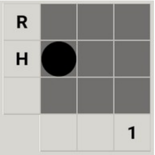
Platforms

	Available in repositories of most Linux distributions: **sgt-puzzles** (Debian, Ubuntu) **puzzles** (Fedora, Arch)
	https://f-droid.org/en/packages/name.boyle.chris.sgtpuzzles/ https://chris.boyle.name/projects/android-puzzles/
iOS	https://hewgill.com/puzzles/
palm	https://www.chiark.greenend.org.uk/~jharvey/puzzles/
PocketBook	https://github.com/SteffenBauer/PocketPuzzles

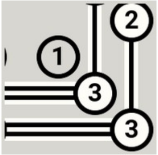
. . . and

Java JS maemo™ SAILFISH OS symbian ROCKbox open source jukebox firmware

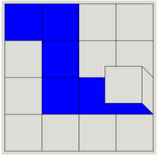
Official Puzzles



BlackBox

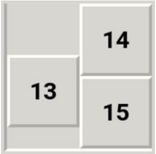


Bridges

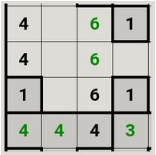


Cube

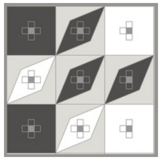
Dominosa



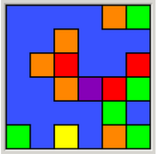
Fifteen



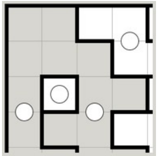
Filling



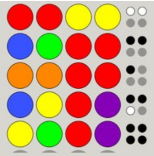
Flip



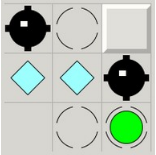
Flood



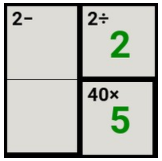
Galaxies



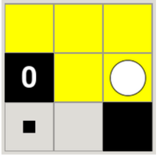
Guess



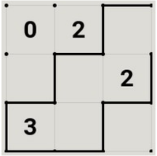
Inertia



Keen

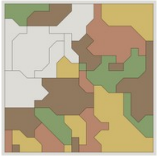


LightUp

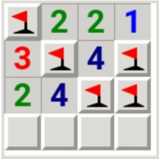


Loopy

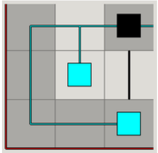
Magnets



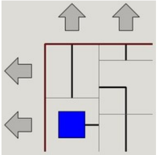
Map



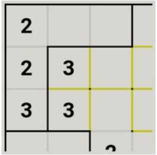
Mines



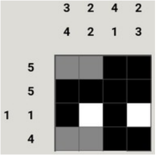
Net



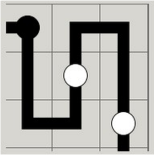
Netslide



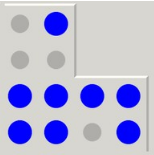
Palisade



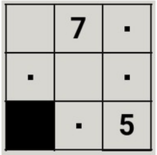
Pattern



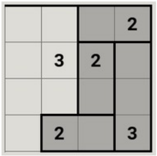
Pearl



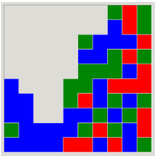
Pegs



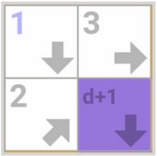
Range



Rect

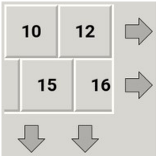


SameGame

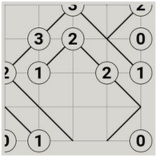


Signpost

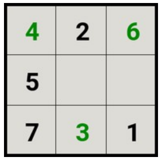
Singles



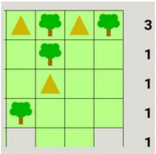
Sixteen



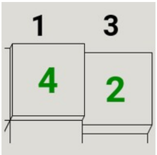
Slant



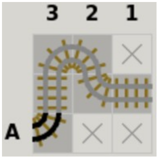
Solo



Tents

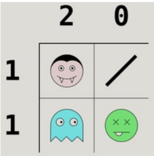


Towers

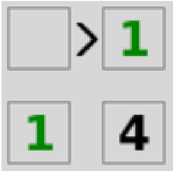


Train Tracks

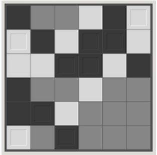
Twiddle



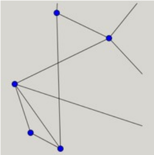
Undead



Unequal



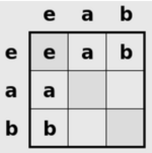
Unruly



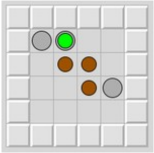
Untangle

Inofficial Puzzles

Unreleased ‚official‘ puzzles

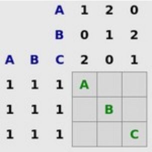

Group

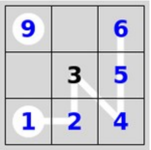
Slide

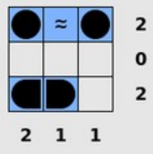

Sokoban

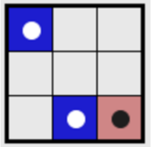
Lennard Sprongs collection

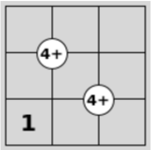
https://github.com/x-sheep/puzzles-unreleased

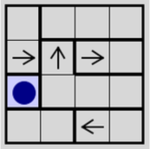

ABCD

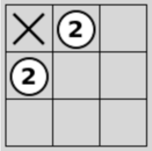

Ascent

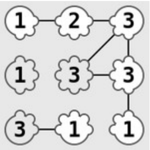

Boats

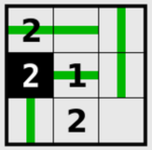

Clusters


Mathrax


Rome

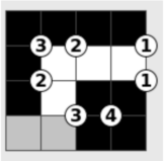

Salad

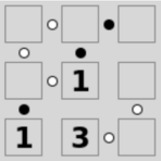

Spokes

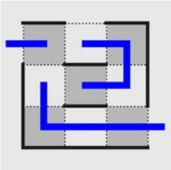

Sticks

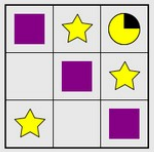
Steffen Bauers collection

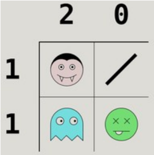
https://github.com/SteffenBauer/sgtpuzzles-extended


Creek


Kropki


Walls


Stellar


Undead++

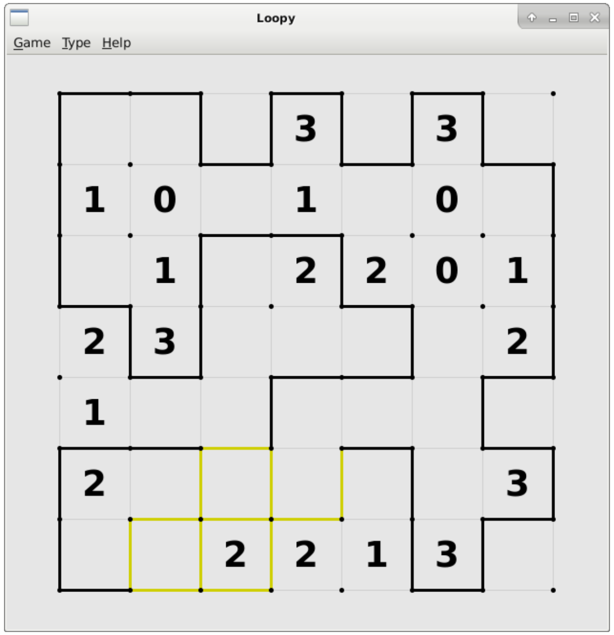
Solo+

Didi Kohens ‚Mosaic‘

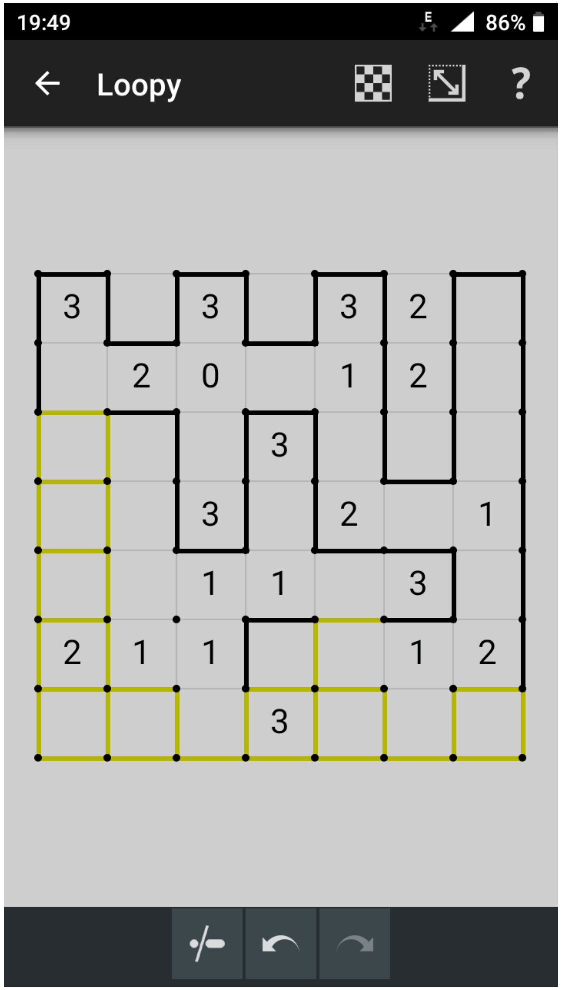
https://github.com/kohend/simon-puzzles

Mosaic

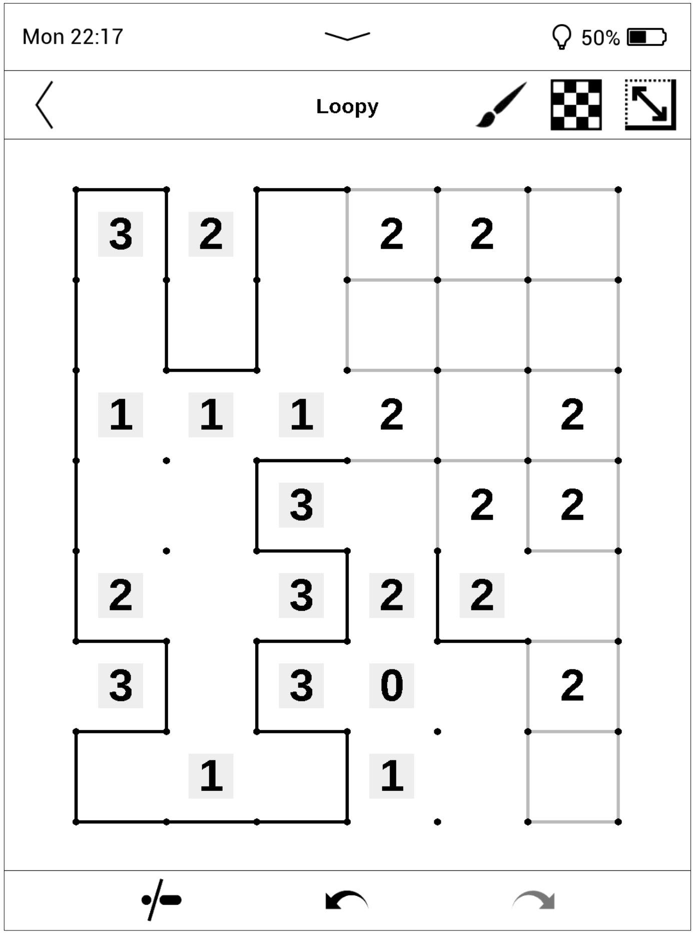
Live demo



			3		3	
1	0		1		0	
	1		2	2	0	1
2	3					2
1						
2						3
		2	2	1	3	



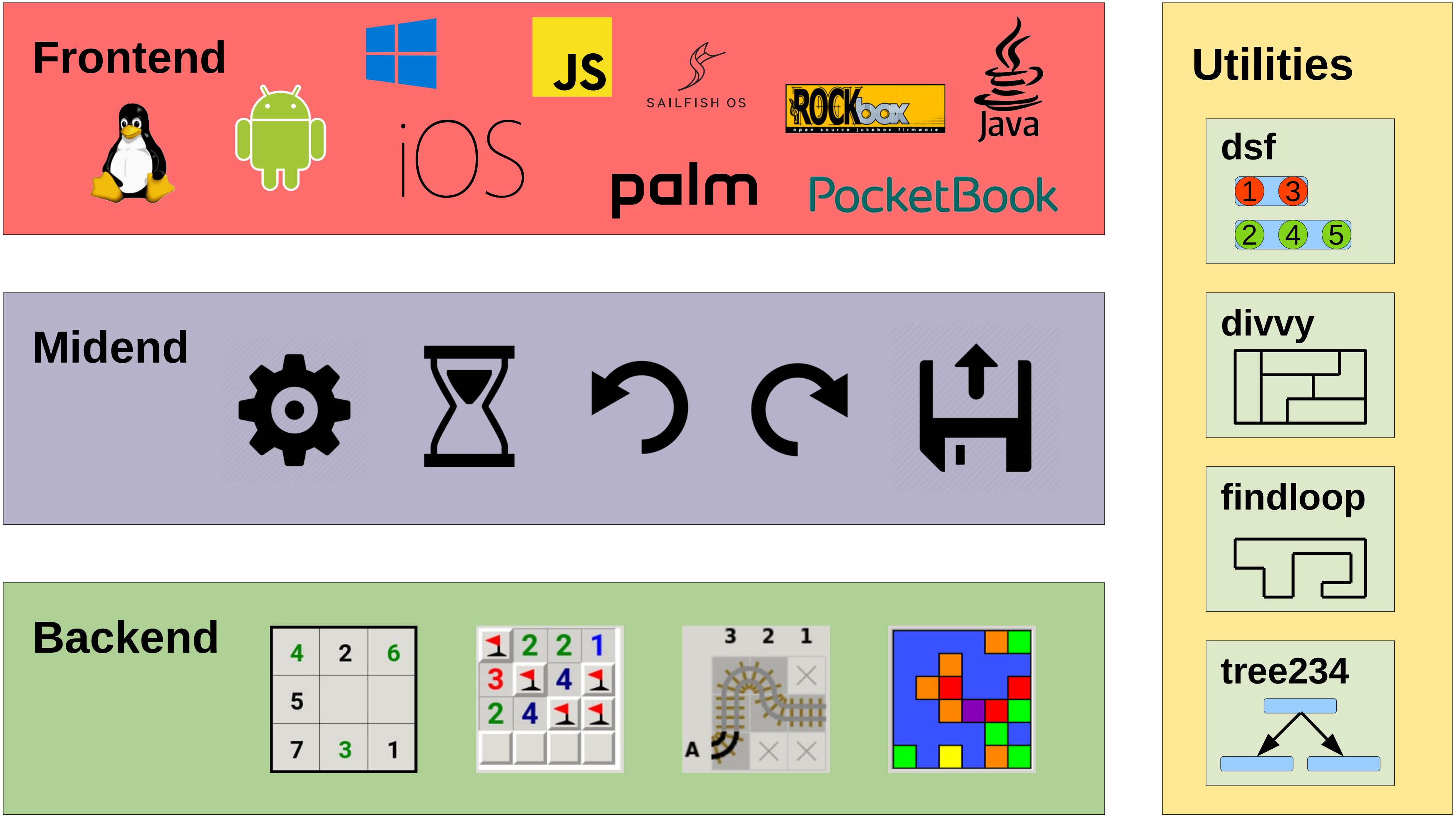
3		3		3	2	
	2	0		1	2	
			3			
		3		2		1
		1	1		3	
2	1	1			1	2
			3			



3	2		2	2	
1	1	1	2		2
		3		2	2
2		3	2	2	
3		3	0		2
	1		1		

PocketBook

Software architecture



Frontend

Midend

Backend

Utilities

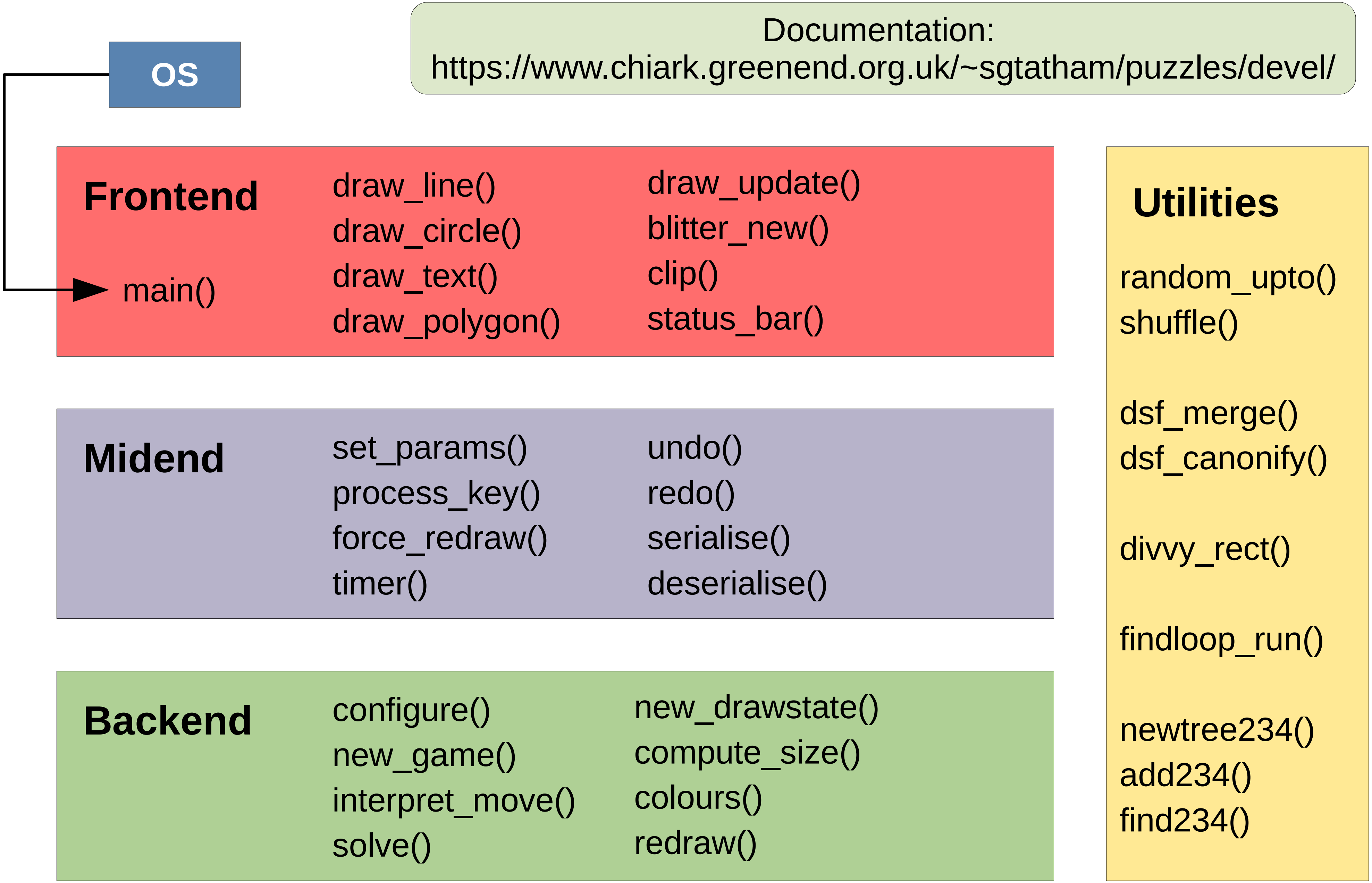
dsf

divvy

findloop

tree234

Software architecture



Documentation:
https://www.chiark.greenend.org.uk/~sgtatham/puzzles/devel/

OS

Frontend

main()

draw_line()
draw_circle()
draw_text()
draw_polygon()

draw_update()
blitter_new()
clip()
status_bar()

Midend

set_params()
process_key()
force_redraw()
timer()

undo()
redo()
serialise()
deserialise()

Backend

configure()
new_game()
interpret_move()
solve()

new_drawstate()
compute_size()
colours()
redraw()

Utilities

random_upto()
shuffle()

dsf_merge()
dsf_canonify()

divvy_rect()

findloop_run()

newtree234()
add234()
find234()

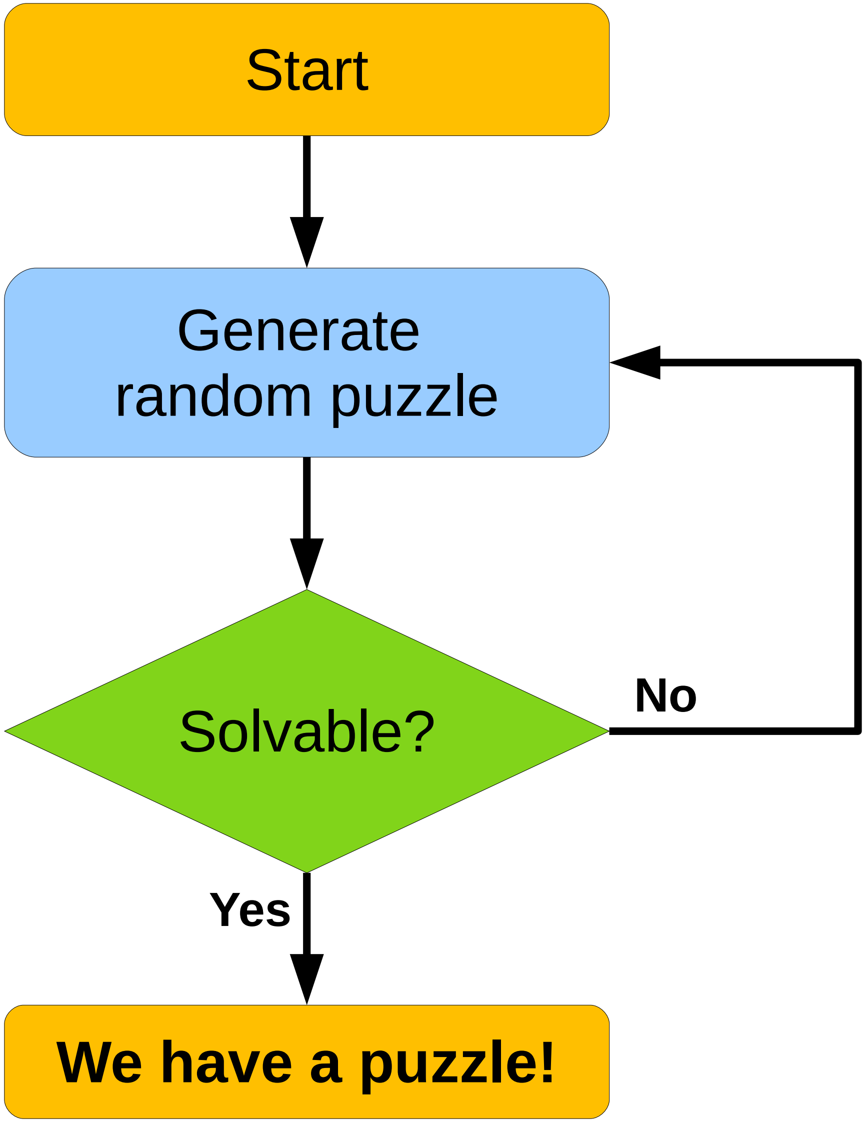
Puzzle generators

Requirements for a puzzle generator:

- The generated puzzle must be **solvable**
- The solution of a puzzle should be **unique**
- The puzzle should be **configurable**;
  especially the **difficulty level**
- A puzzle should be generated **fast** (few seconds maximum);
  even on limited (*mobile*!) hardware

→ 6 years development time for the first Sudoku generator!

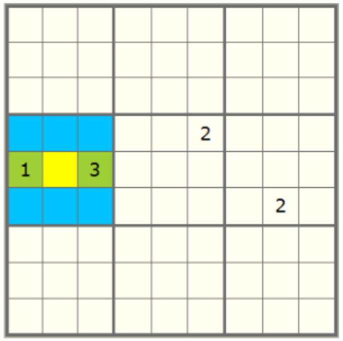
Puzzle generators



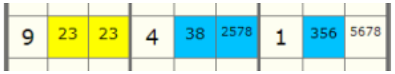
- A puzzle *Generator* is based on a *Solver*
- Modern hardware can test 100s of puzzles per second; even on mobile
- Success rate < 10% is okay
- *Preprocessing* in generation step can increase success rate significantly

Solving methods

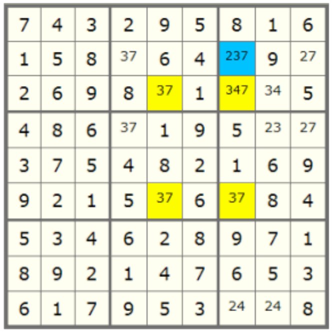
Hidden singles



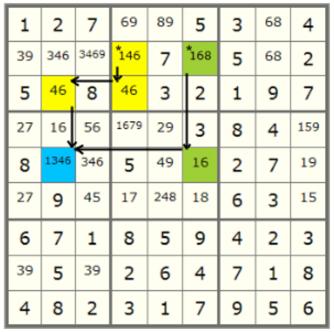
Naked Pair



X-Wing



Coloring

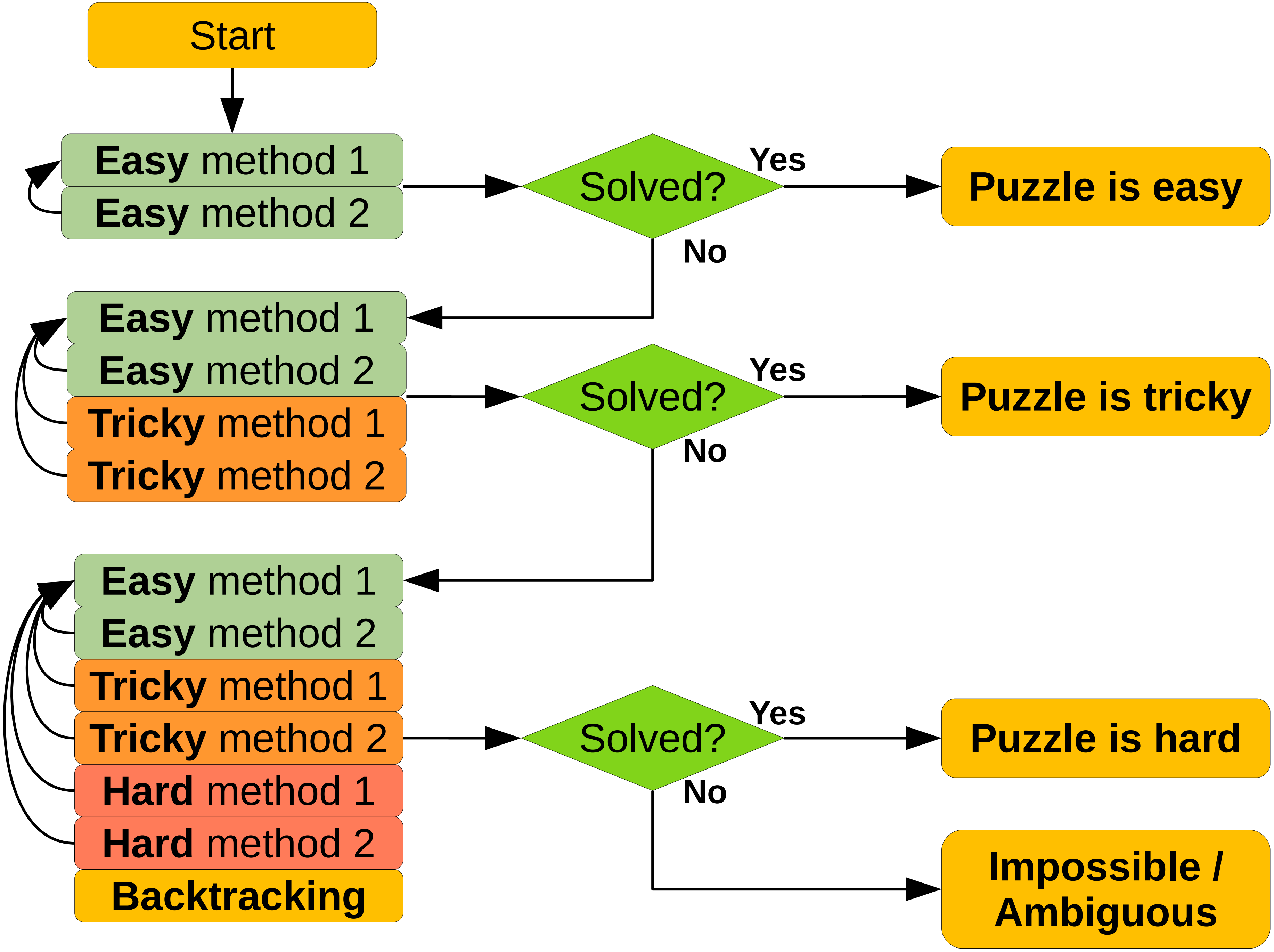


Increasing difficulty

Mathematical background:

Logic puzzles are **‚Constraint Satisfaction Problems‘**

Puzzle generators – Algorithmic



Start

Easy method 1
Easy method 2

Solved?
Yes
No

Puzzle is easy

Easy method 1
Easy method 2
Tricky method 1
Tricky method 2

Solved?
Yes
No

Puzzle is tricky

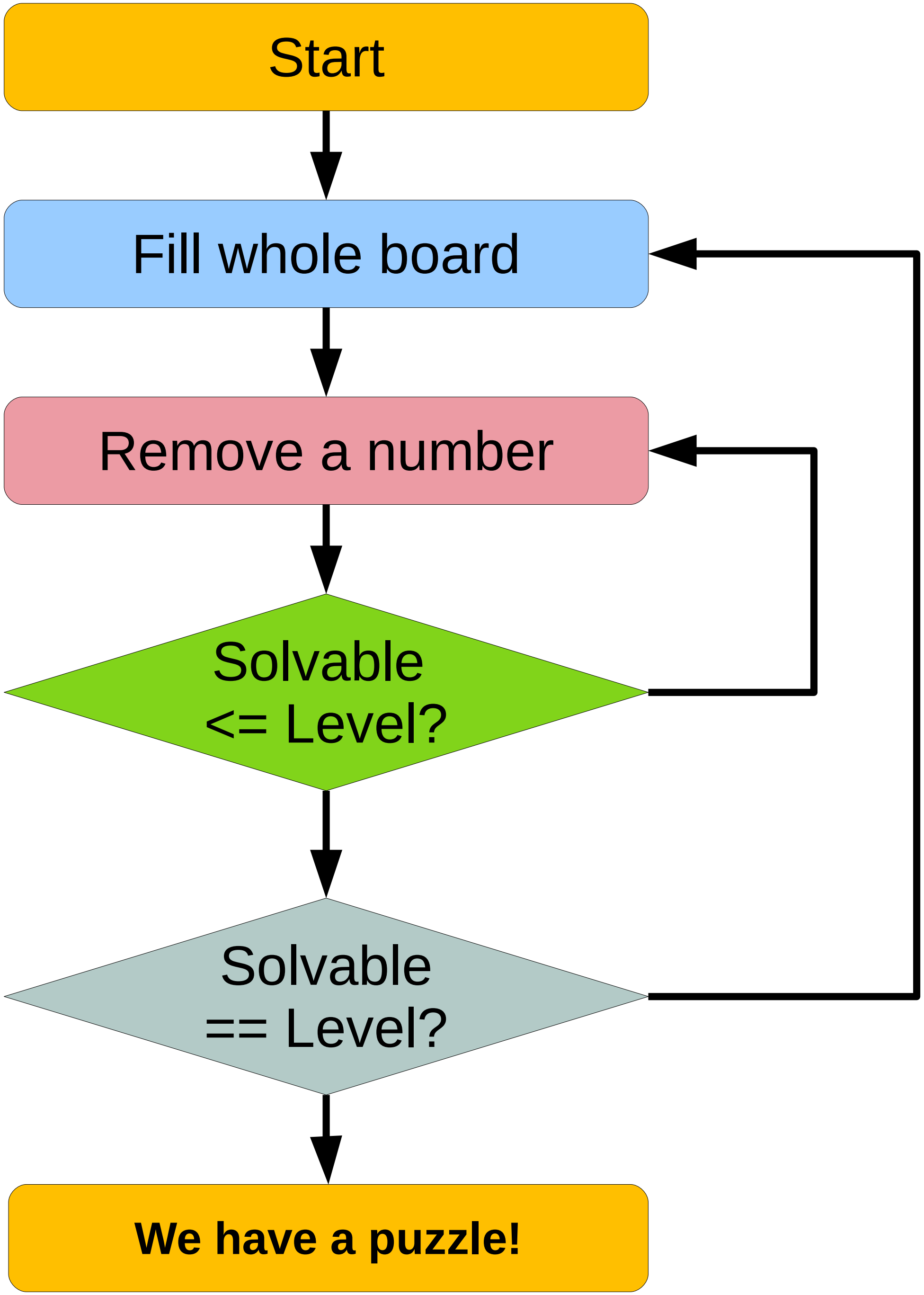
Easy method 1
Easy method 2
Tricky method 1
Tricky method 2
Hard method 1
Hard method 2
Backtracking

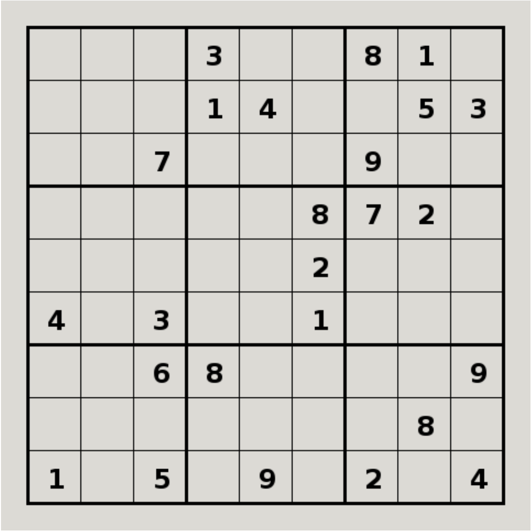
Solved?
Yes
No

Puzzle is hard

Impossible / Ambiguous

Puzzle generators – Sudoku





			3			8	1	
			1	4			5	3
		7				9		
					8	7	2	
					2			
4		3			1			
		6	8					9
							8	
1		5		9		2		4

Sudoku