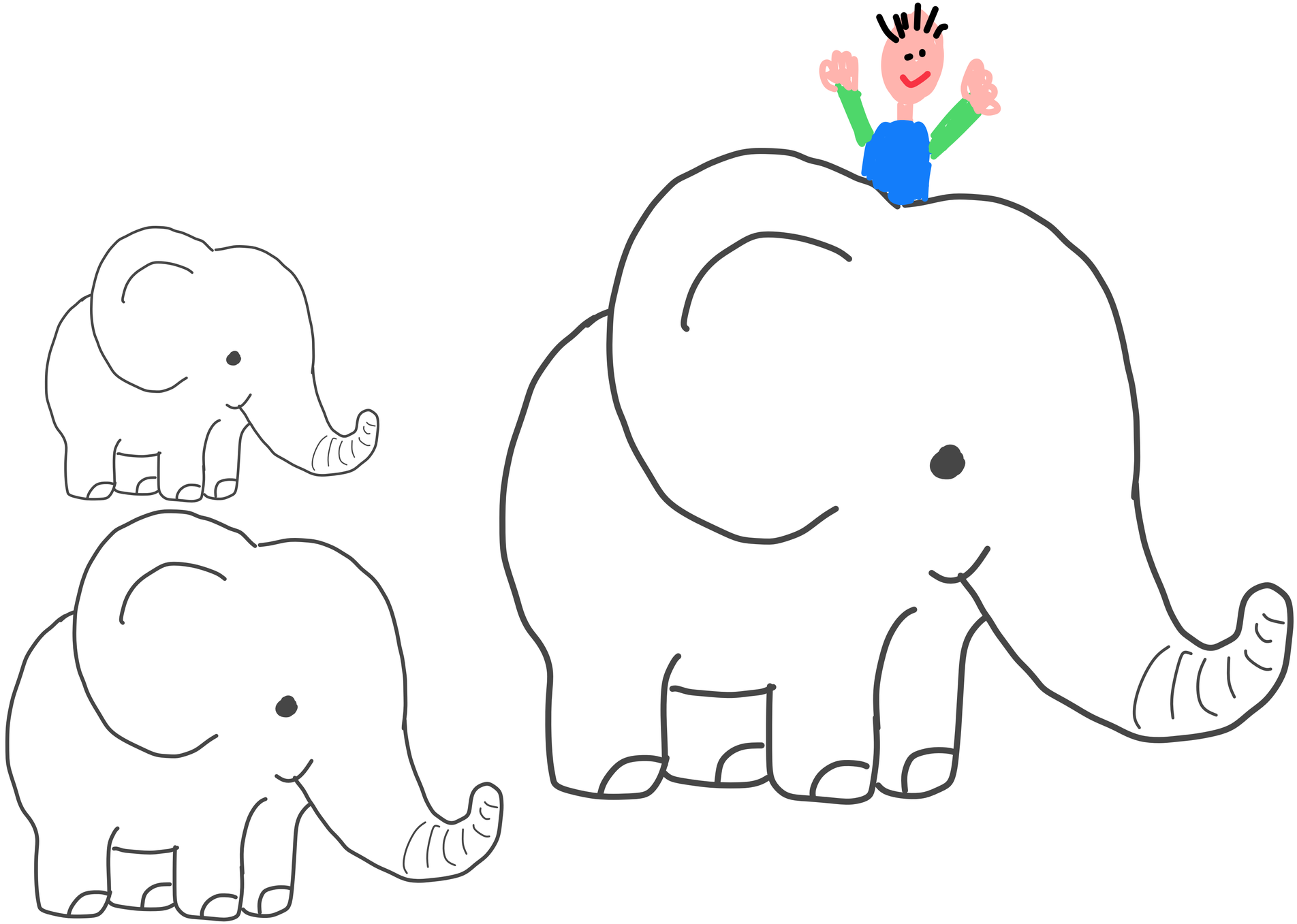
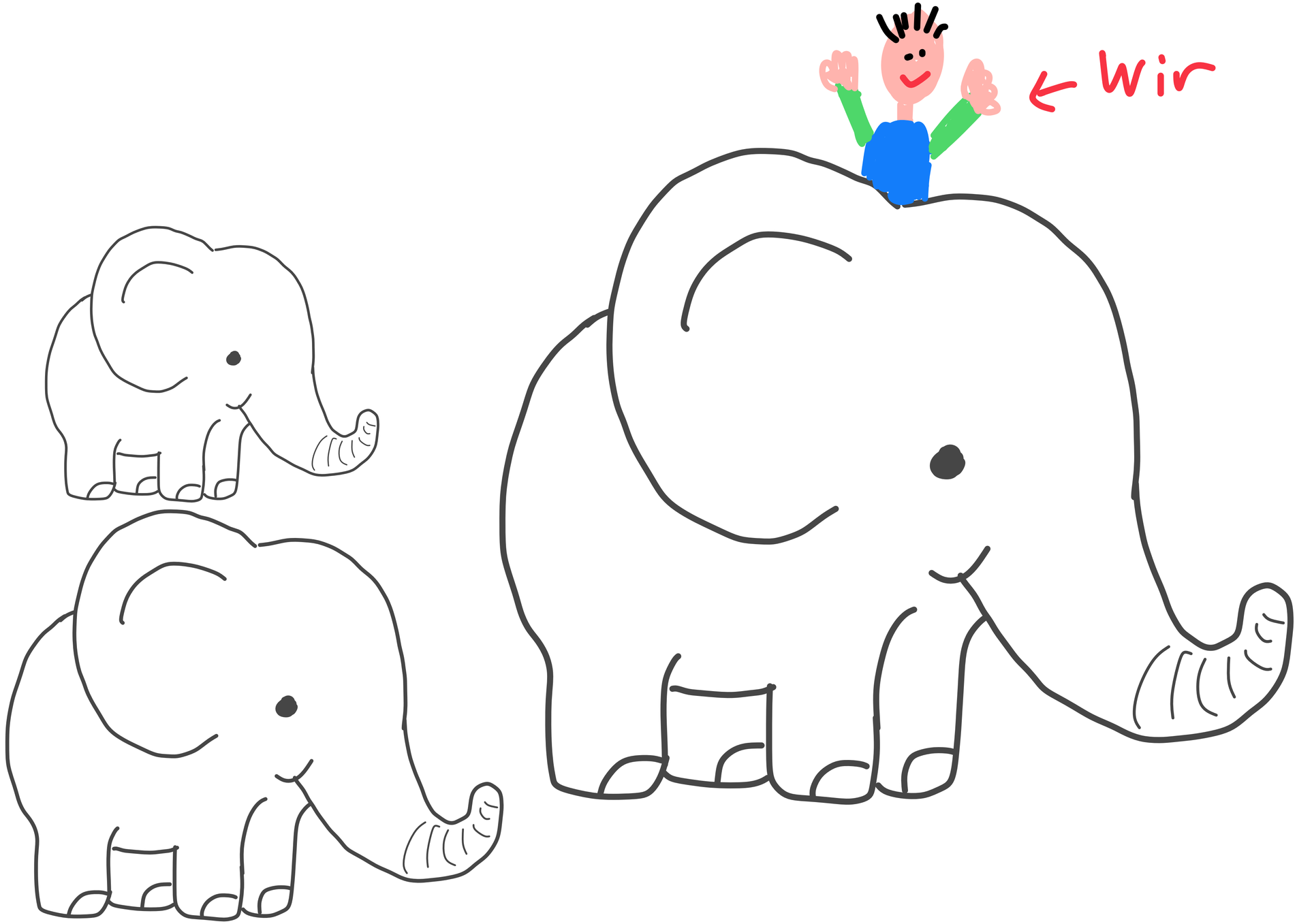
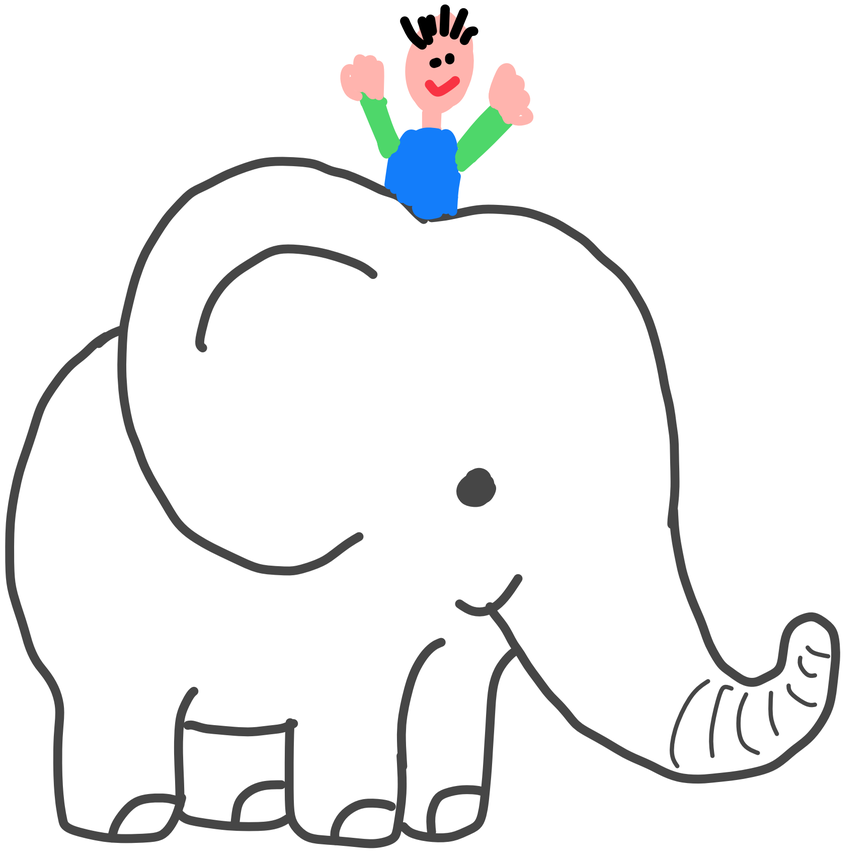
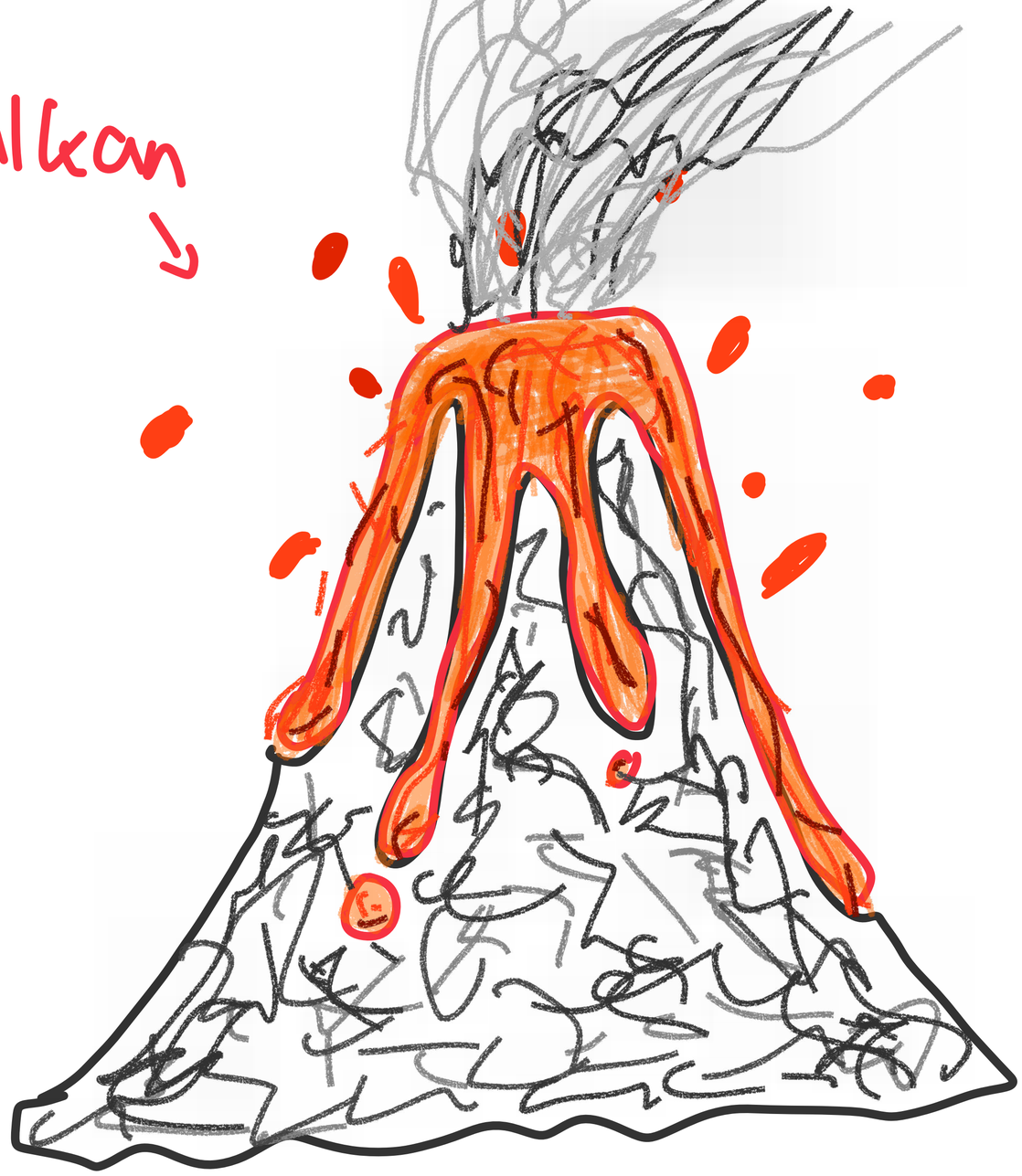
AOC 22/12/18 APL

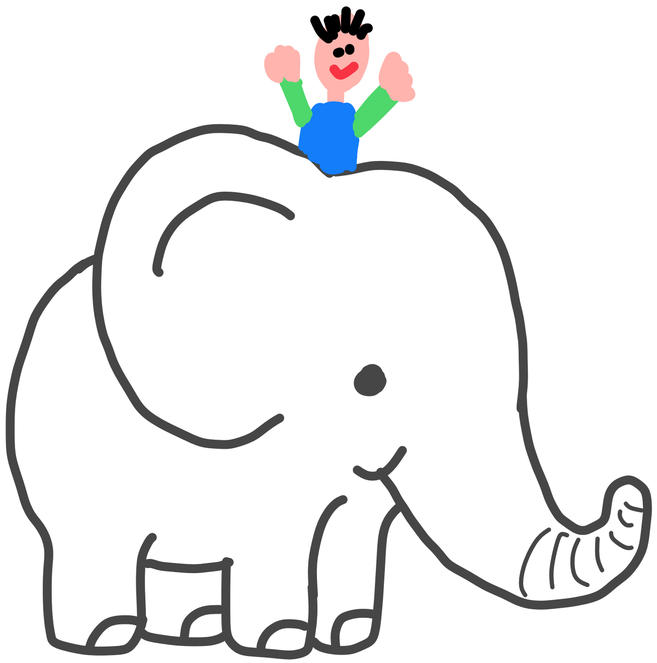
Marvin Borner



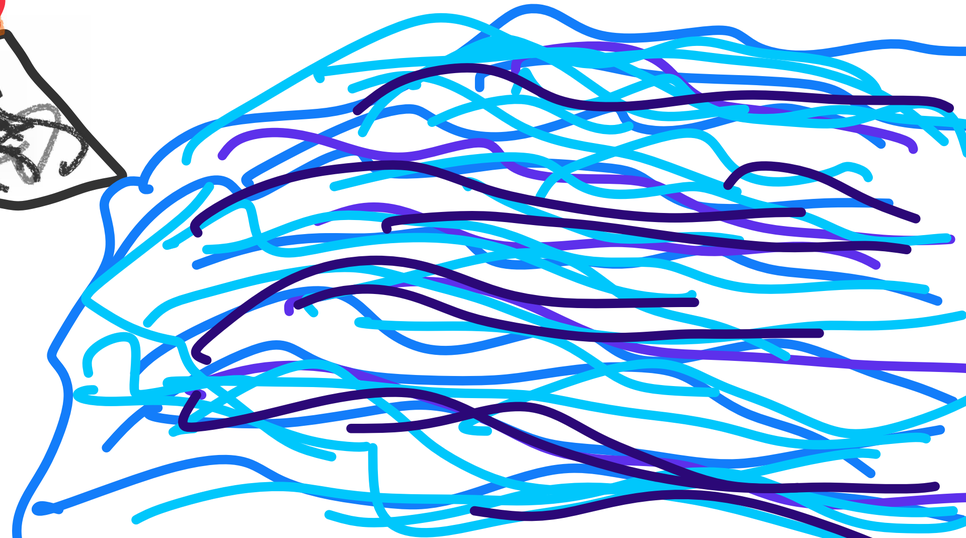

Wir

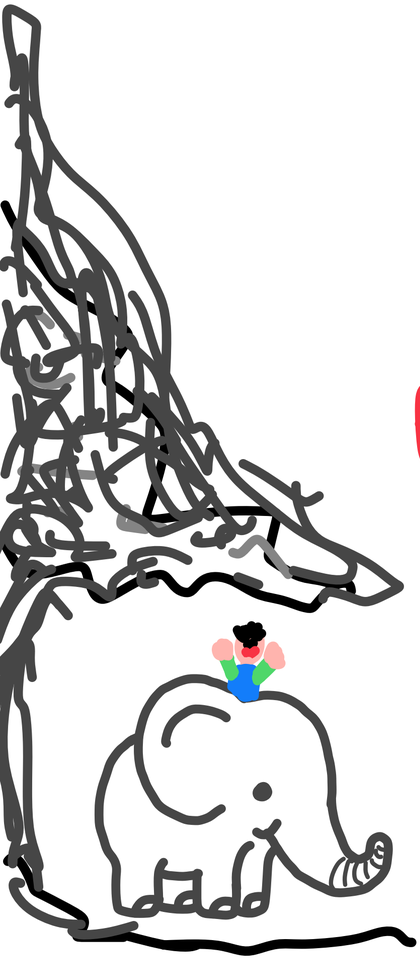



Vulkan

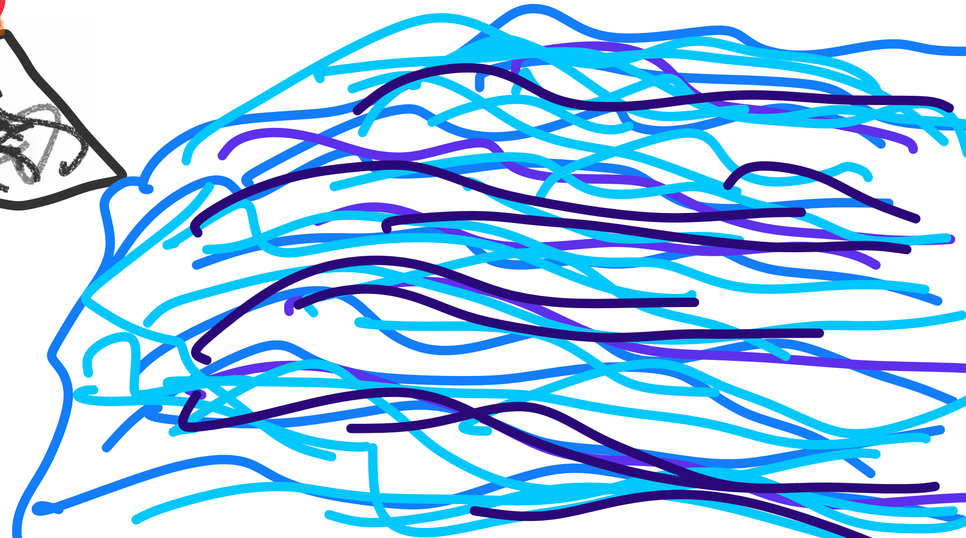


Wasser





Höhle

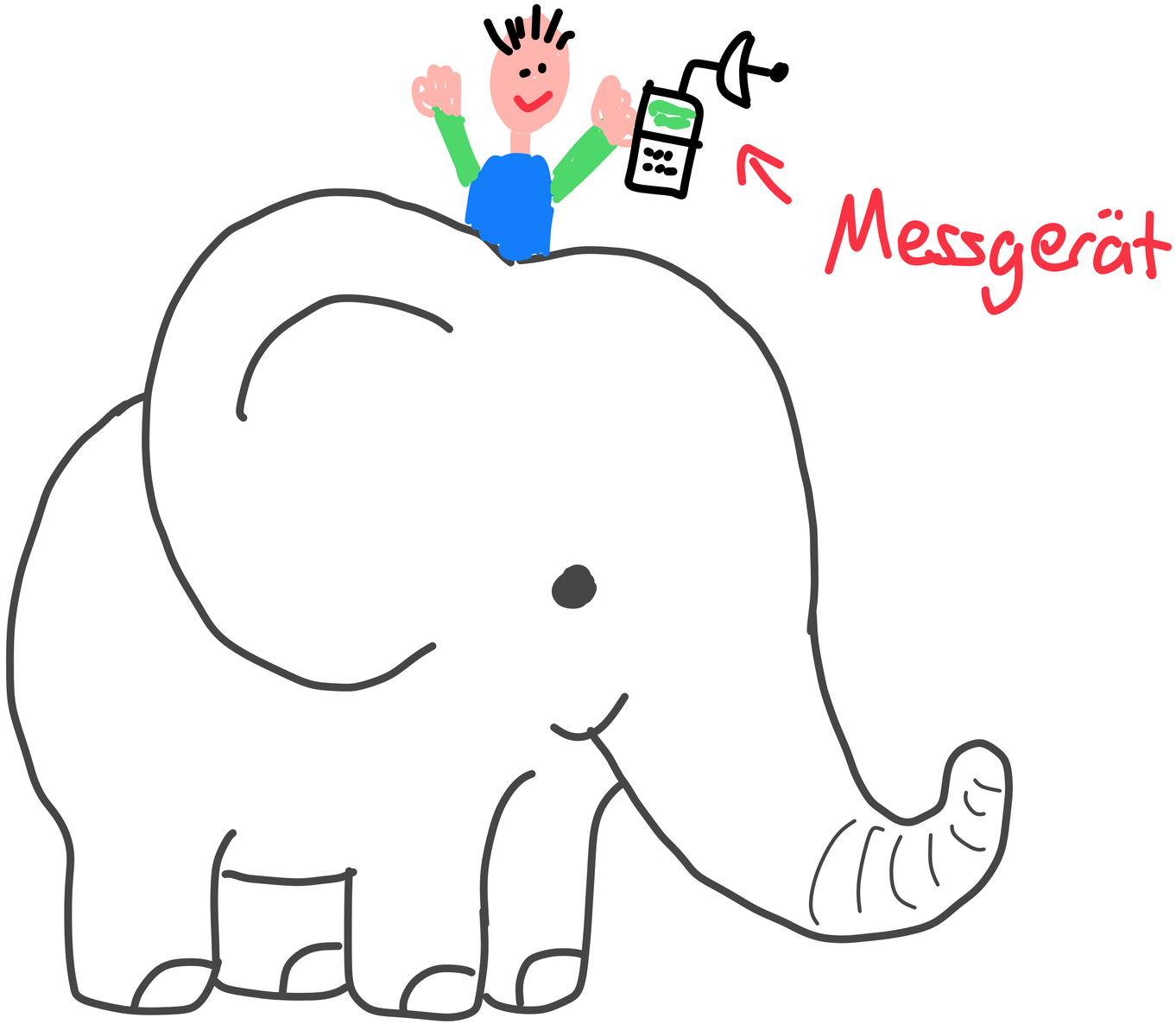


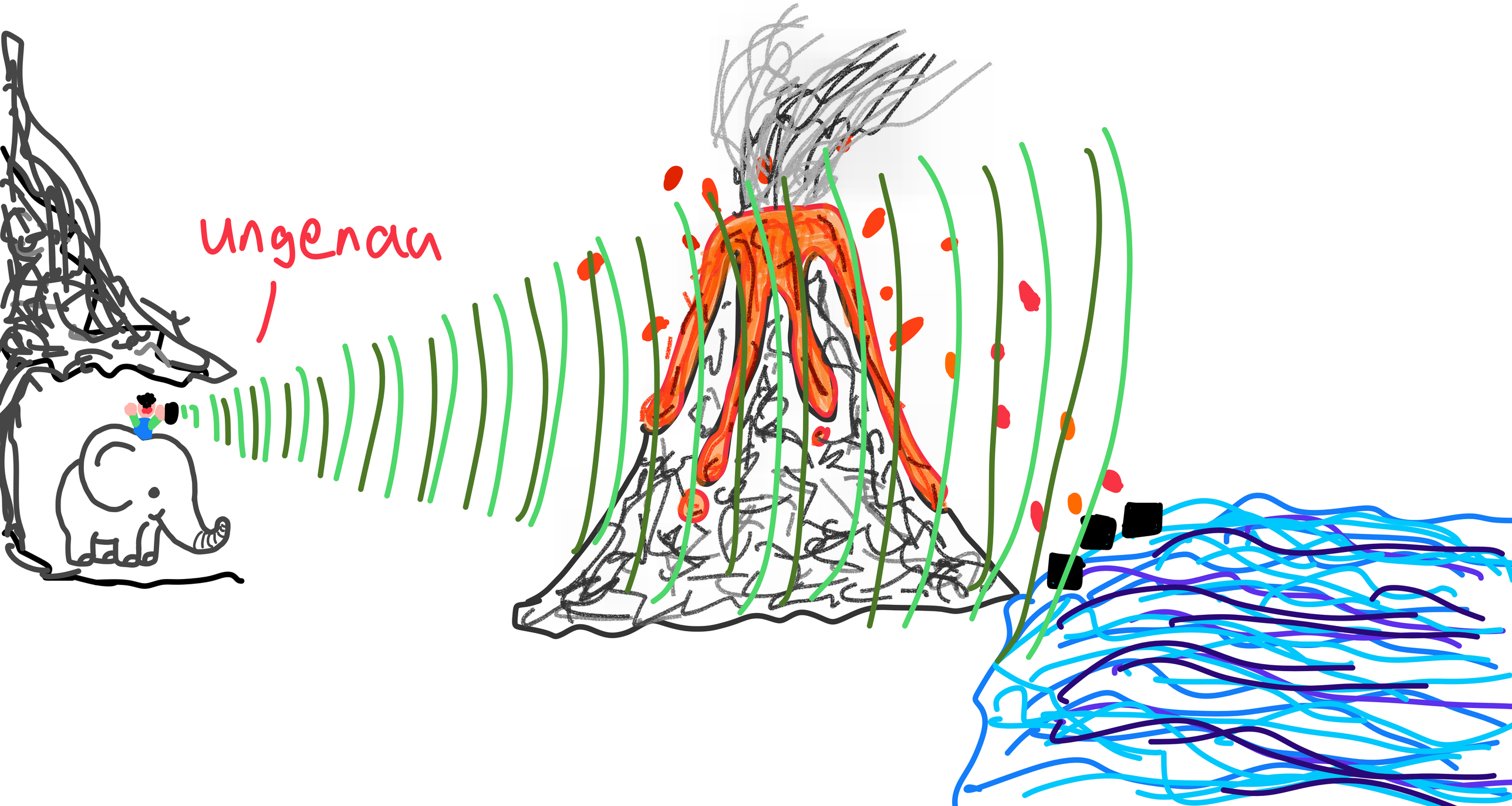

Obsidian?!

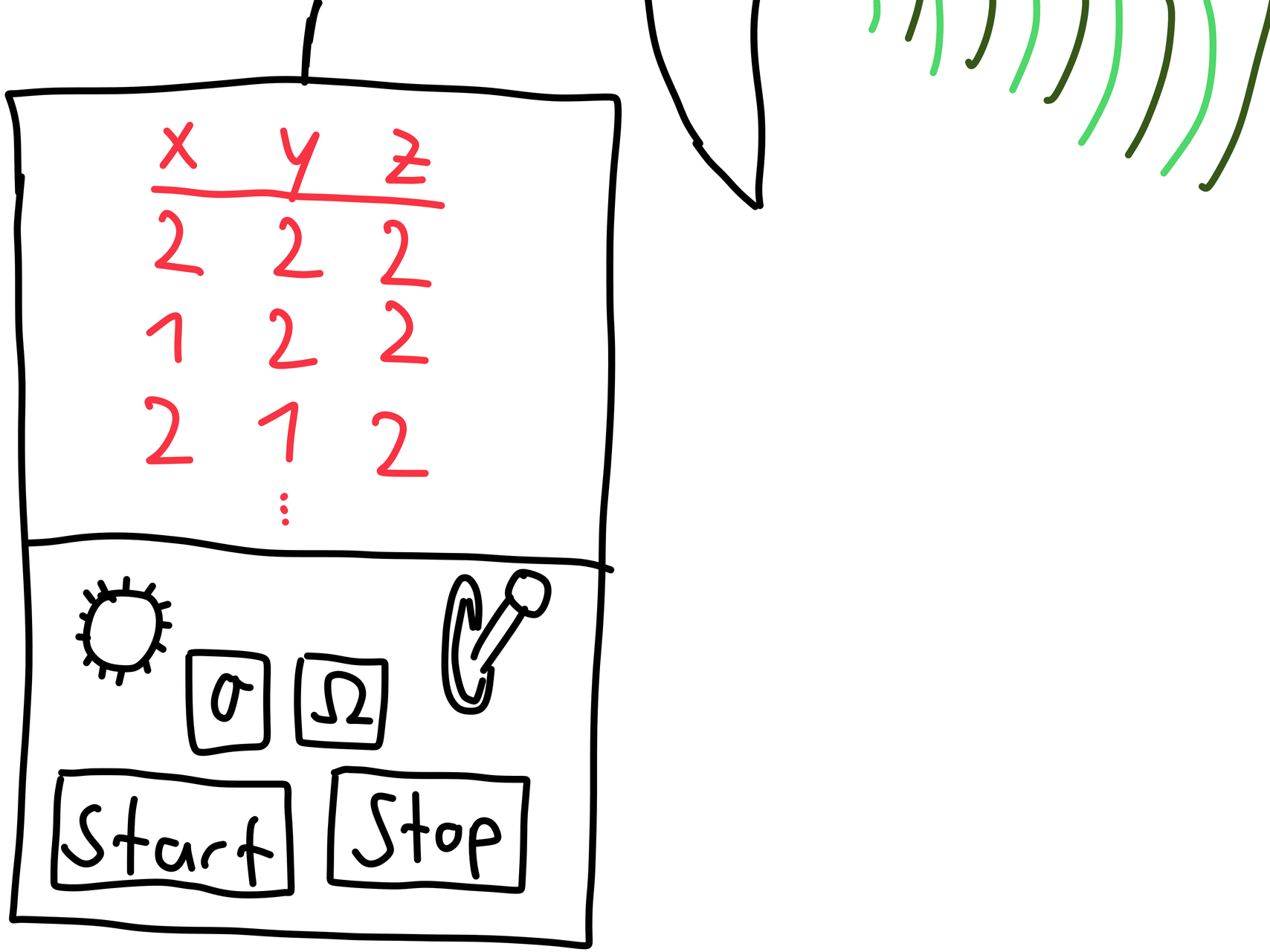
Abhängig
von Kühlrate!

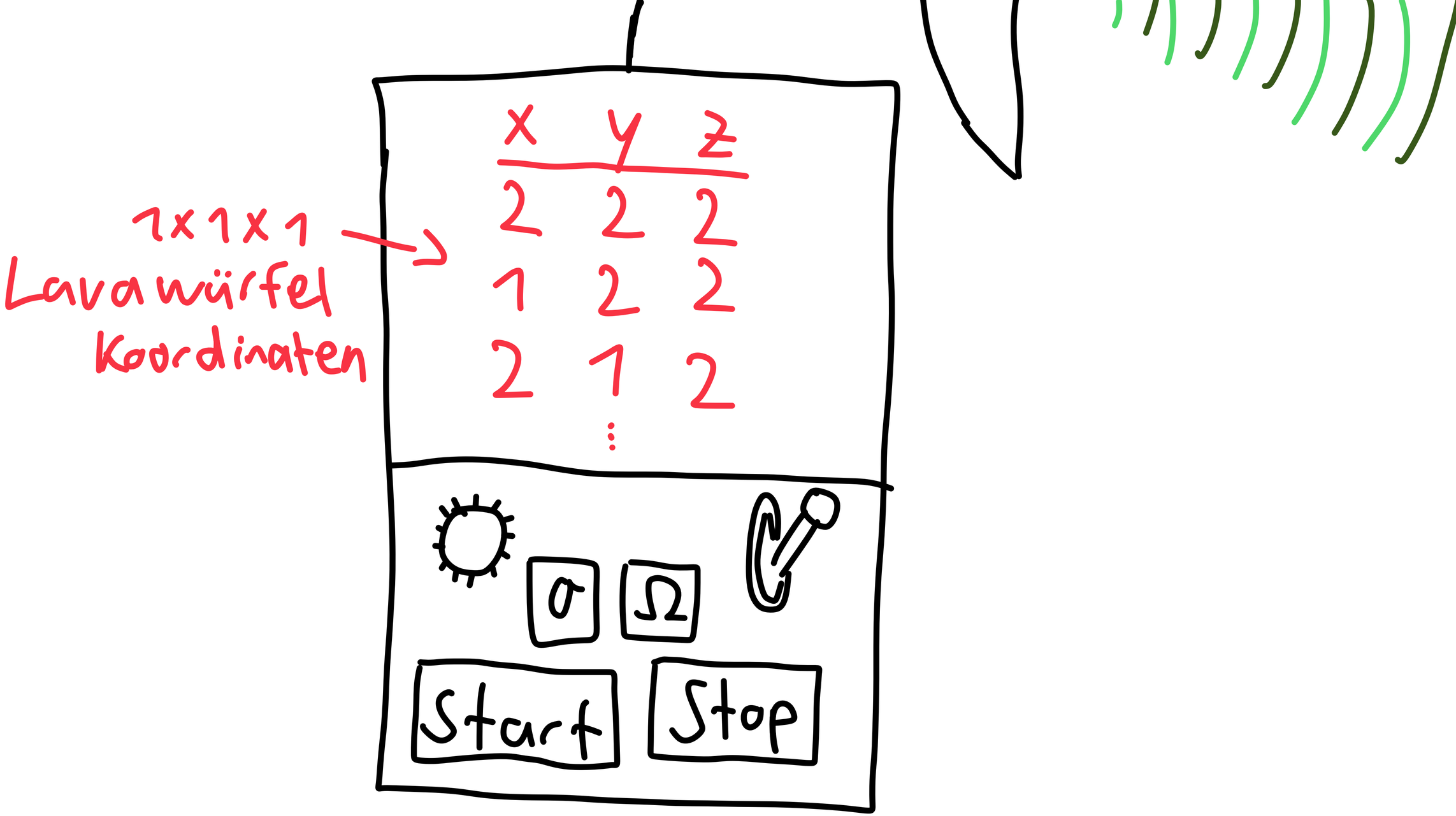
Oberfläche

=> Kühlrate => Obsidian?

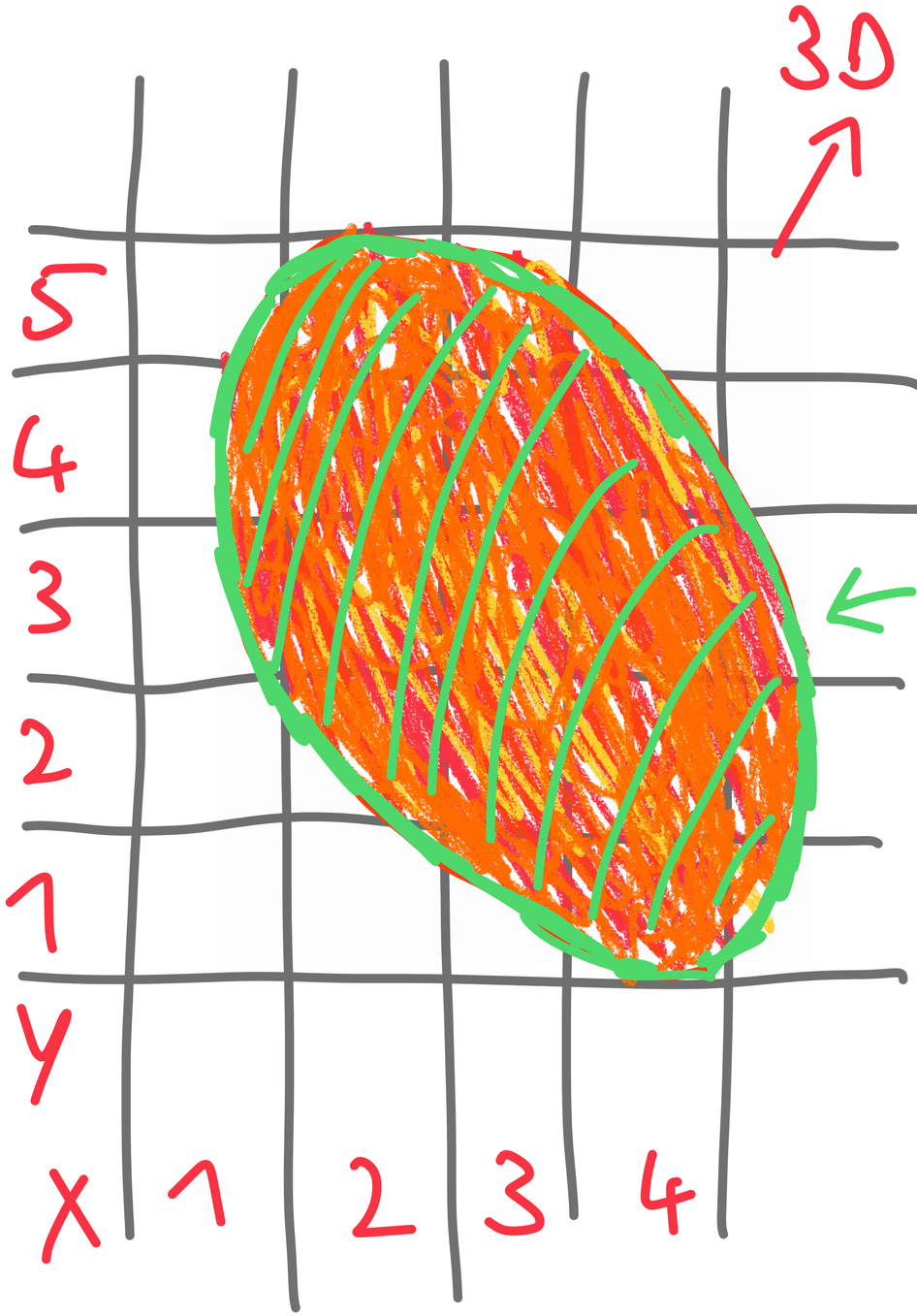

Messgerät


ungenau

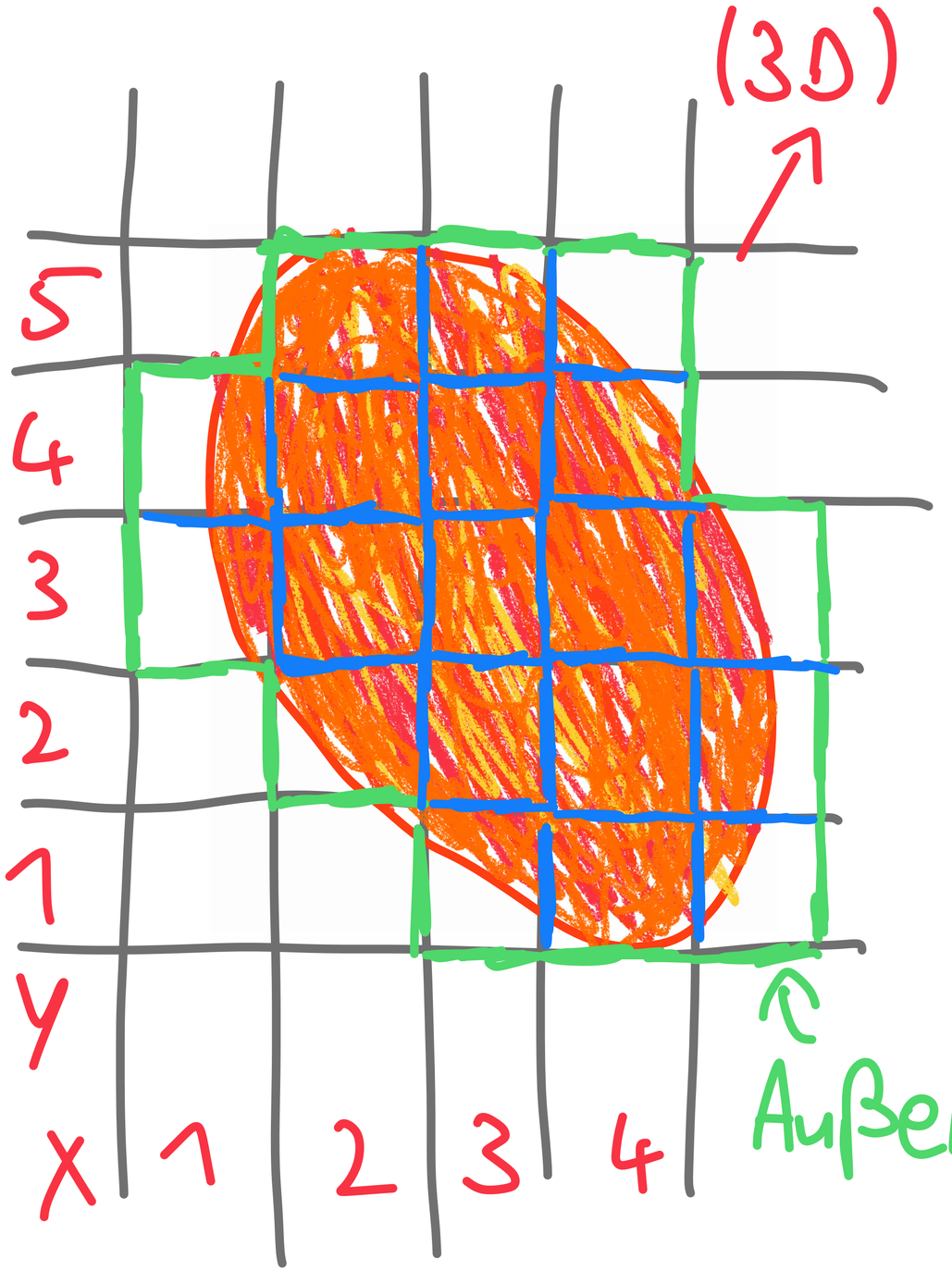

x y z
2 2 2
1 2 2
2 1 2
σ
Ω
Start
Stop


1x1x1
Lavawürfel
Koordinaten
x y z
2 2 2
1 2 2
2 1 2
Start
Stop

x	y	z
1	3	2
1	4	2
2	2	2
2	3	1
2	4	1
2	5	2
3	1	1
⋮		

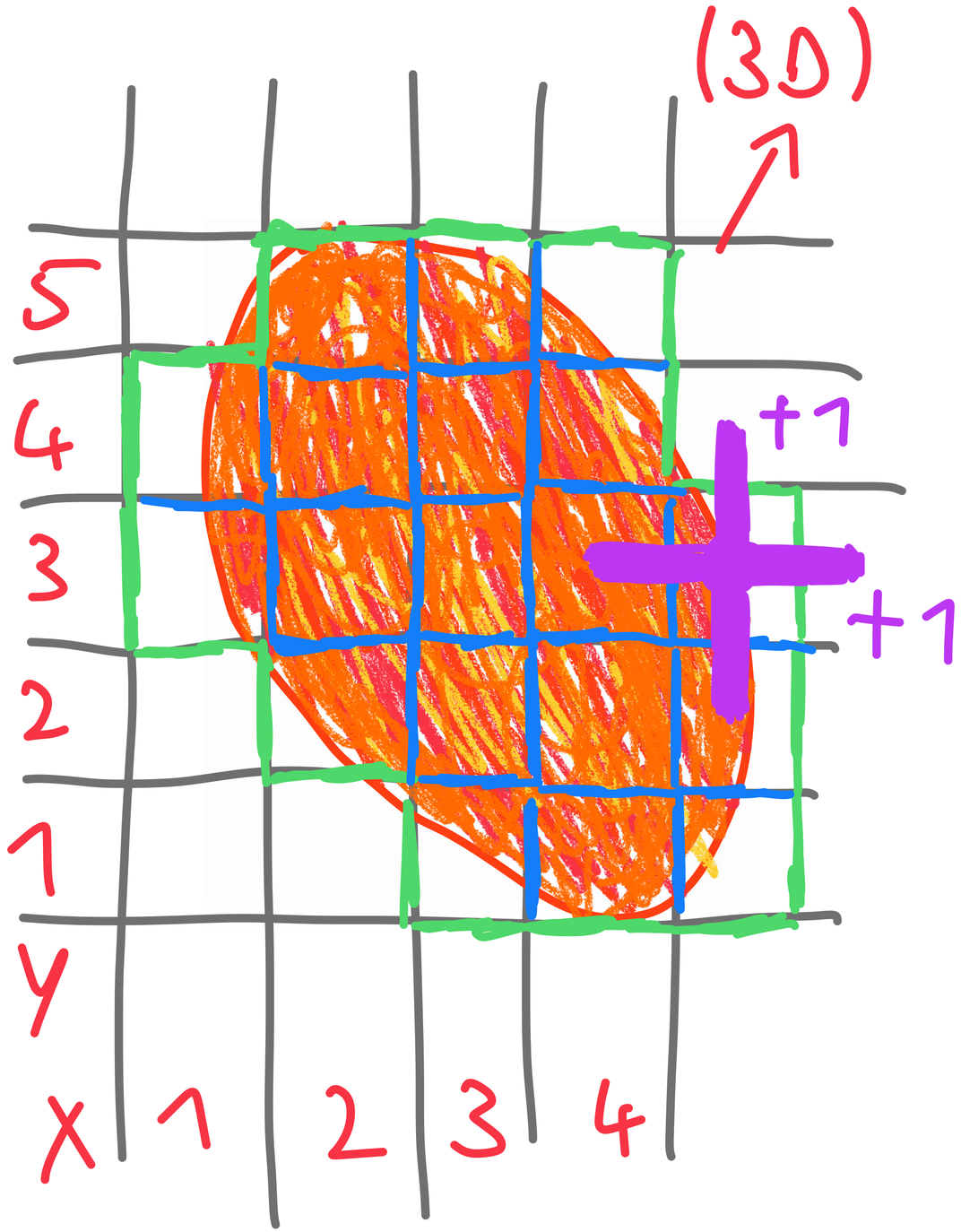


X	Y	Z
1	3	2
1	4	2
2	2	2
2	3	1
2	4	1
2	5	2
3	1	1
⋮		



x	y	(z)
1	3	
1	4	
2	2	
2	3	
2	4	
2	5	
3	1	
⋮		

Part 1

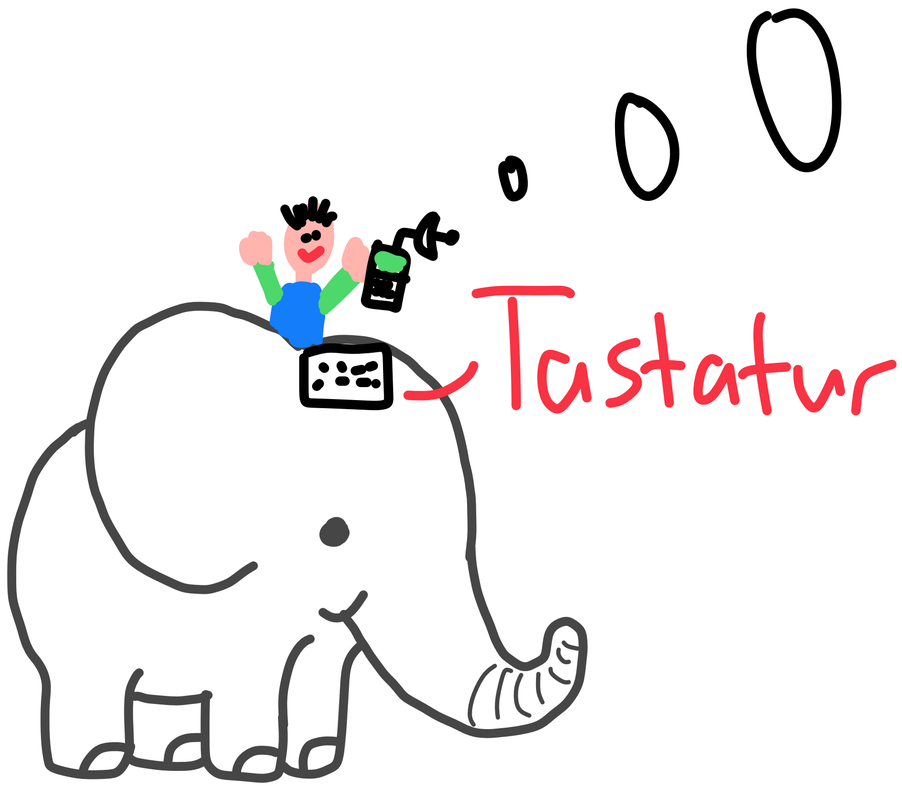

(3D)
5
4
3
2
1
y
x
1
2
3
4
+1
+1

```
n = 0
for cube in cubes:
  for neighbor in neighbors(cube):
    if neighbor not in cubes:
      n += 1
print(n)
```

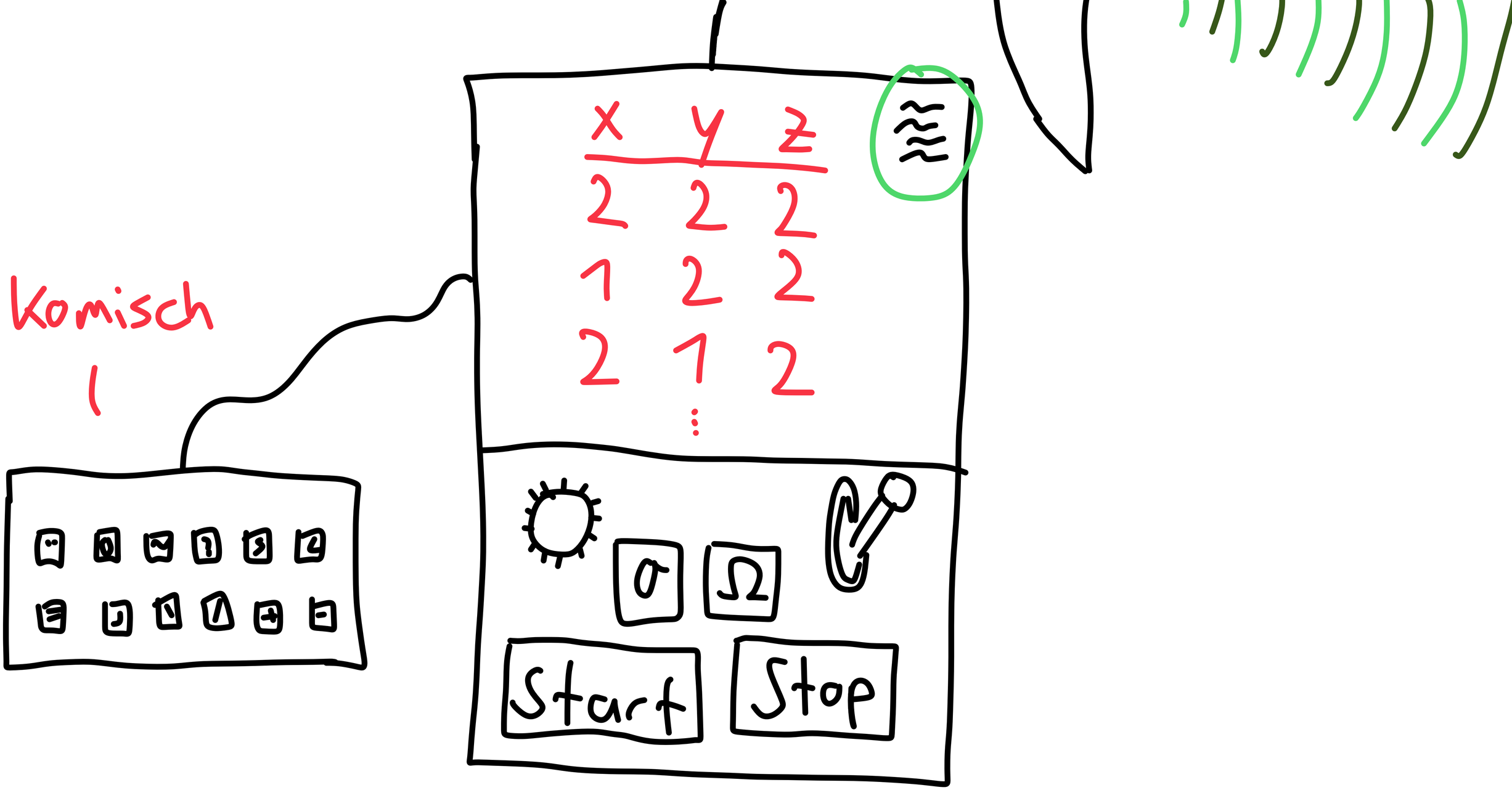
```
n = 0
for cube in cubes:
  for neighbor in neighbors(cube):
    if neighbor not in cubes:
      n += 1
print(n)
```

x, y, z

x±1, y±1, z±1

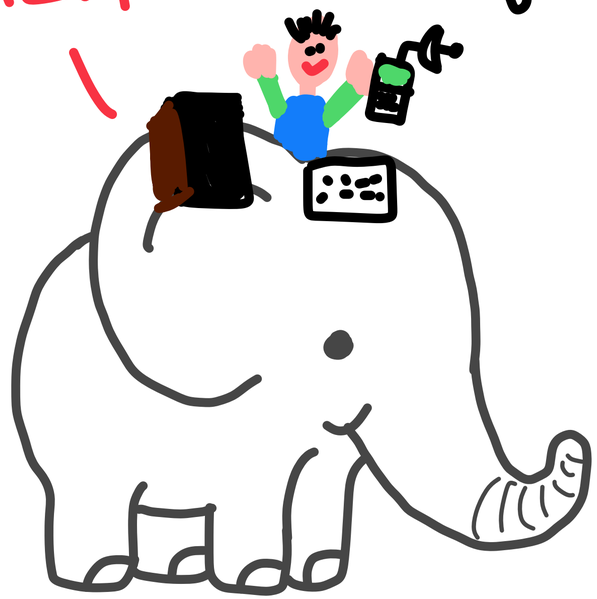



```python
n = 0
for cube in cubes:
  for neighbor in neighbors(cube):
    if neighbor not in cubes:
      n += 1
print(n)
```


Komisch
x y z
2 2 2
1 2 2
2 1 2
Start
Stop


APL only!
it's the best
X
y
z
2 2 2
1 2 2

Buch



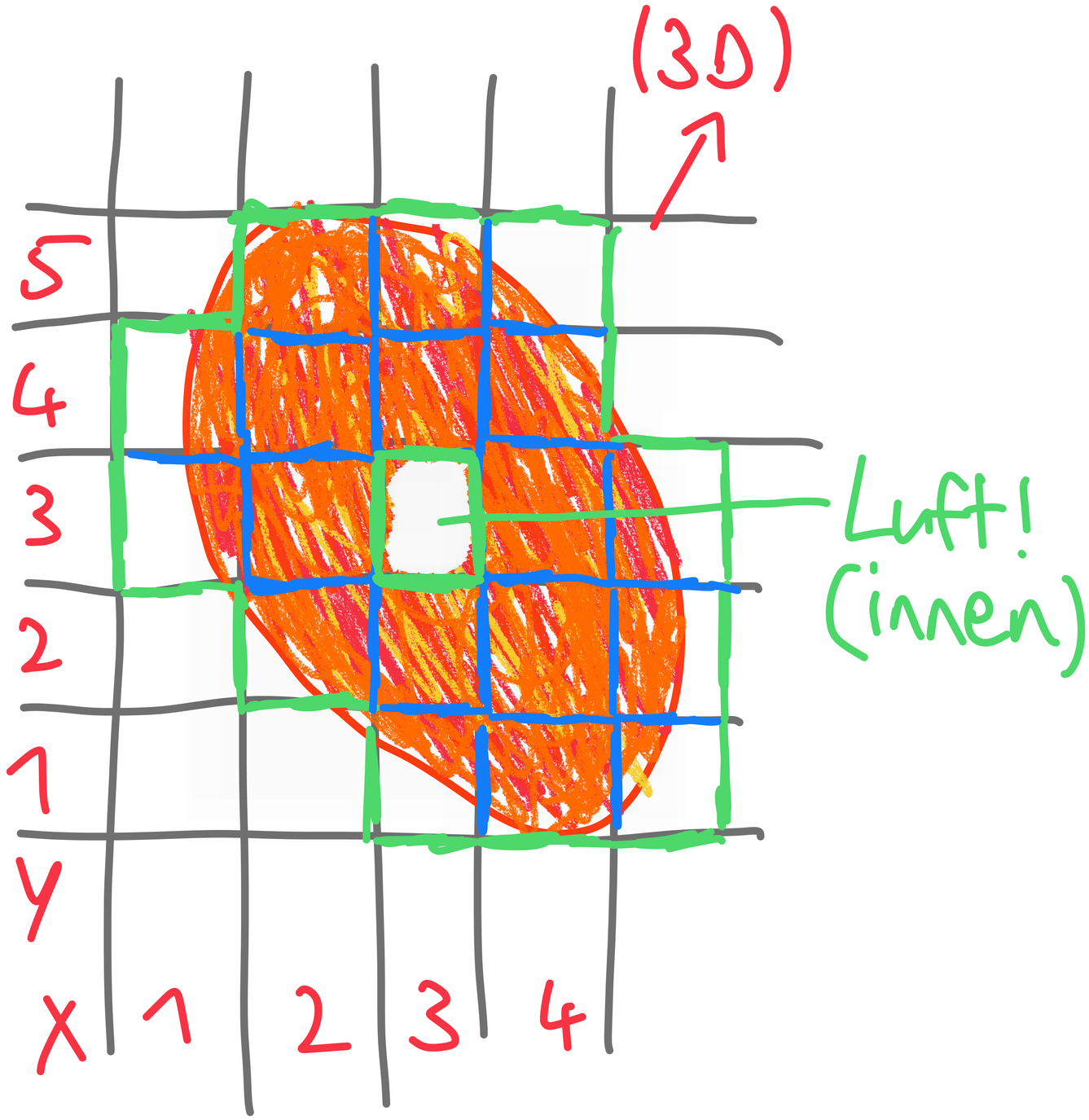
,,oo APL2!

Buch: APL Hints

- Von rechts nach links lesen
- Komische Unicode Zeichen
- Operatoren auf Arrays (monadisch/dyadisch)

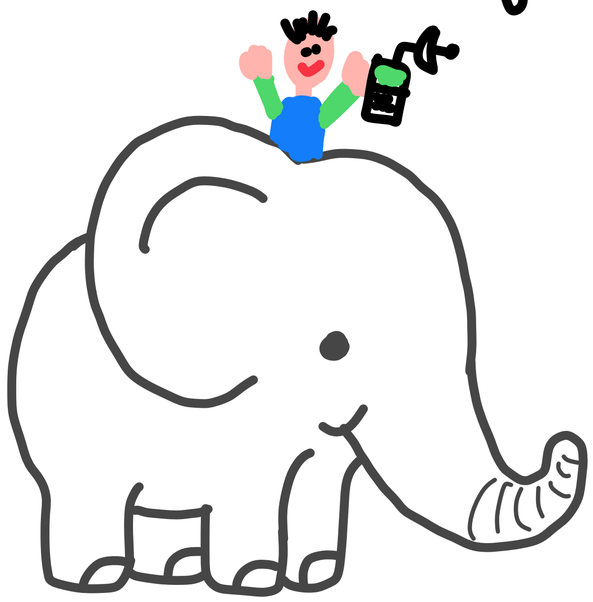
APL IDE

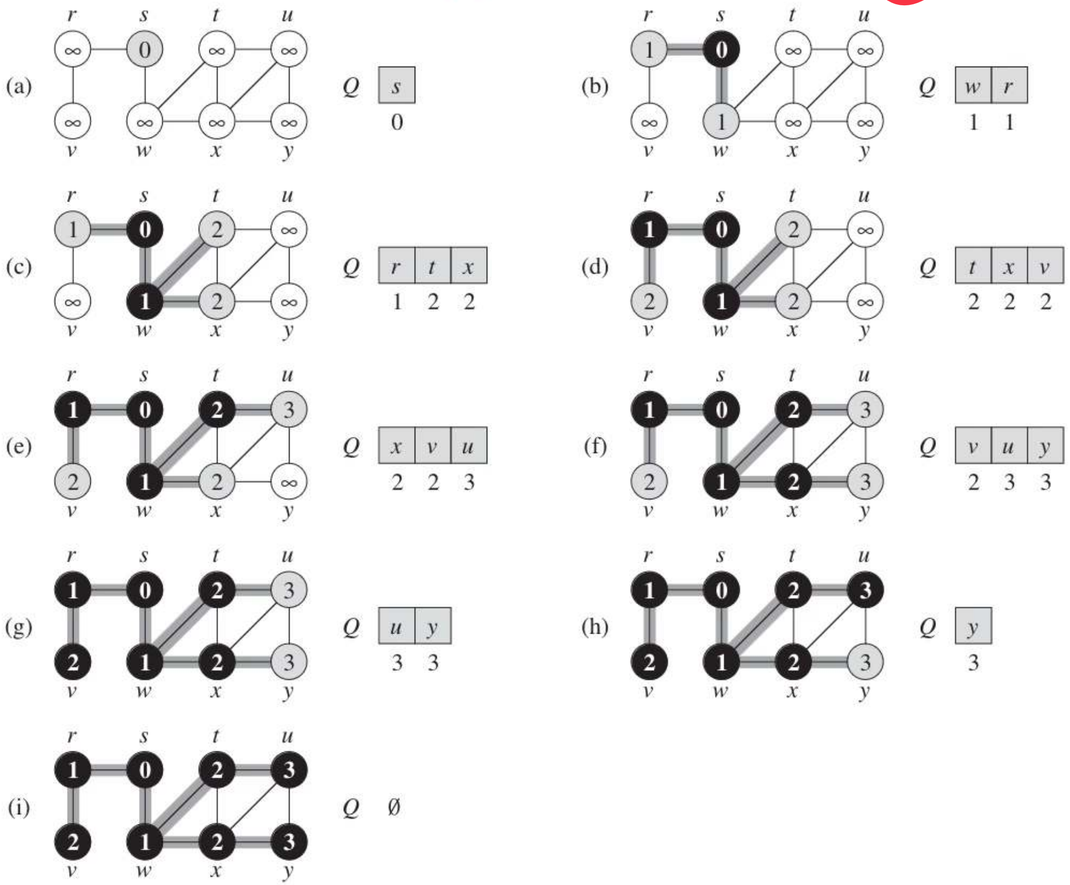
Part 2

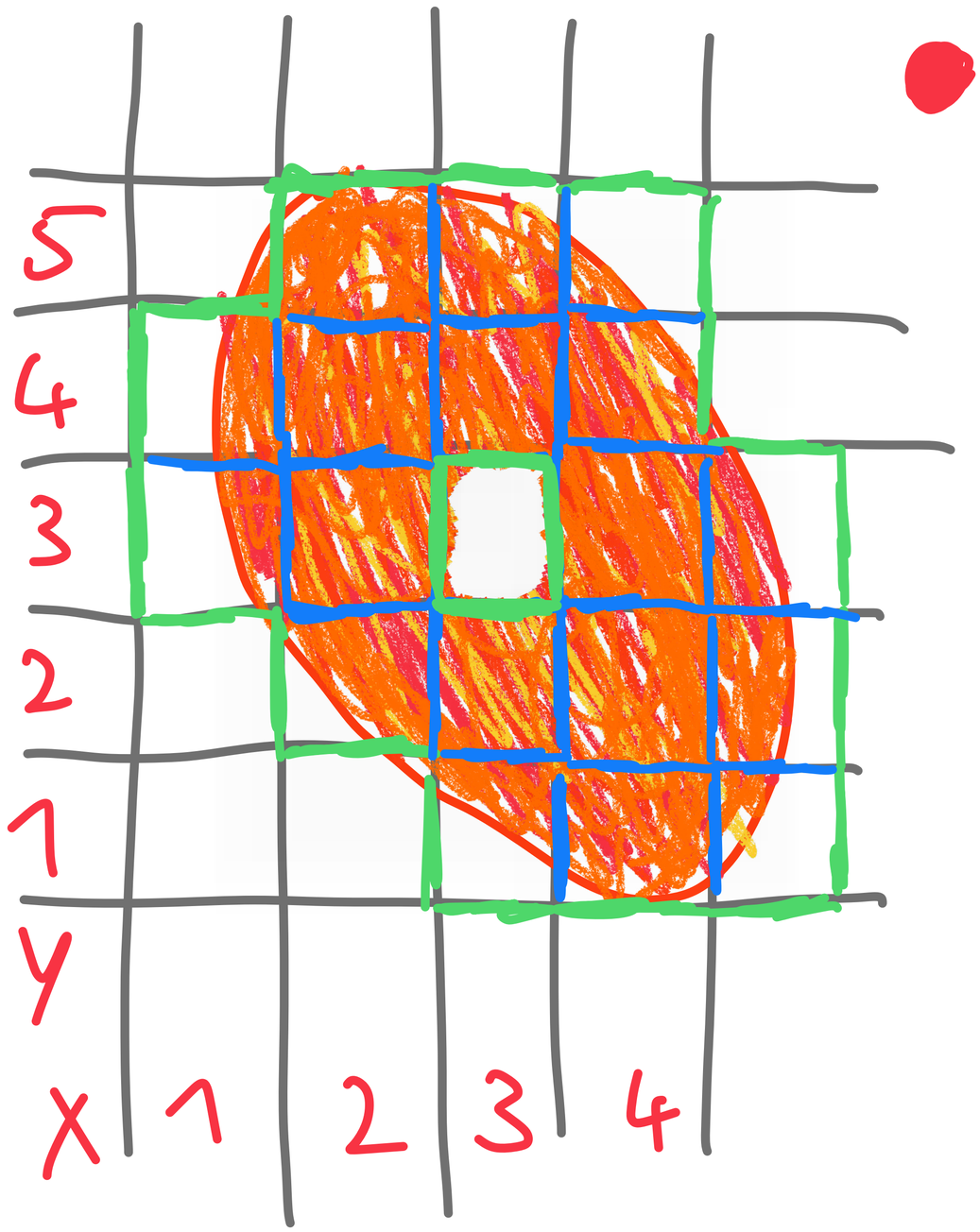

(3D)
5
4
3
2
1
y
x
1
2
3
4
Luft!
(innen)

Wie groß ist die Fläche, die von außen allein durch ⬅️➡️⬆️⬇️ erreicht werden kann?

Algo Vorlesung !

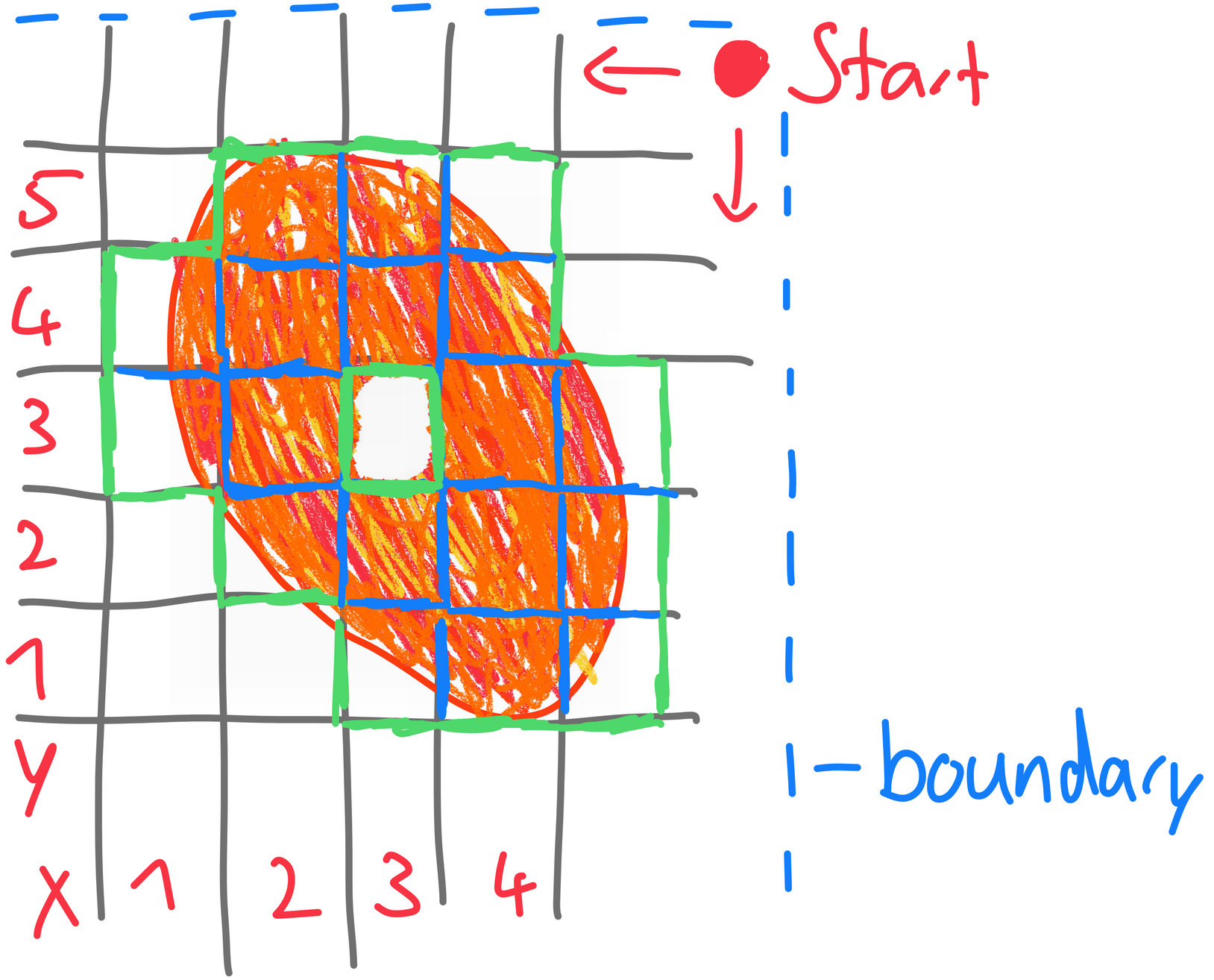


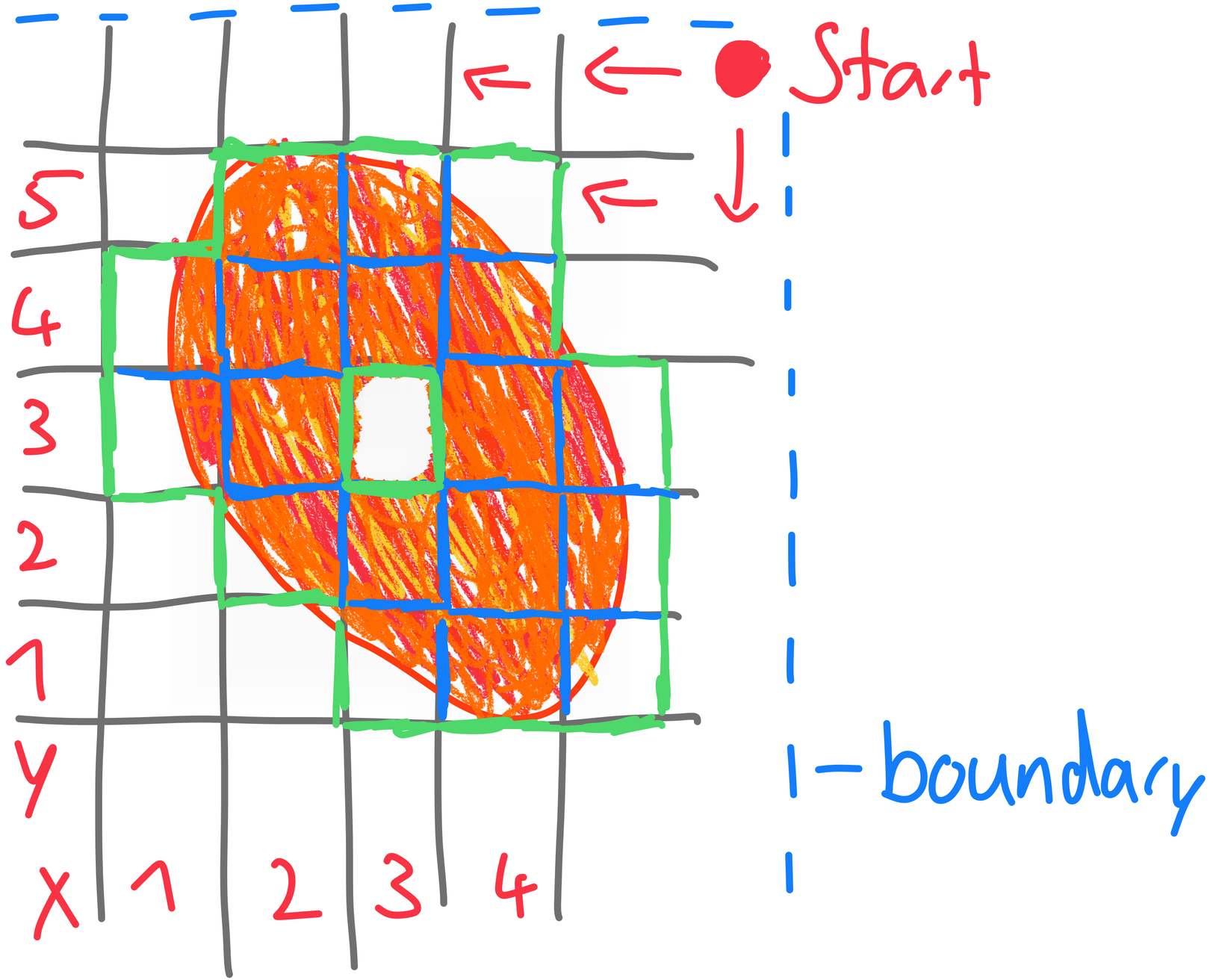


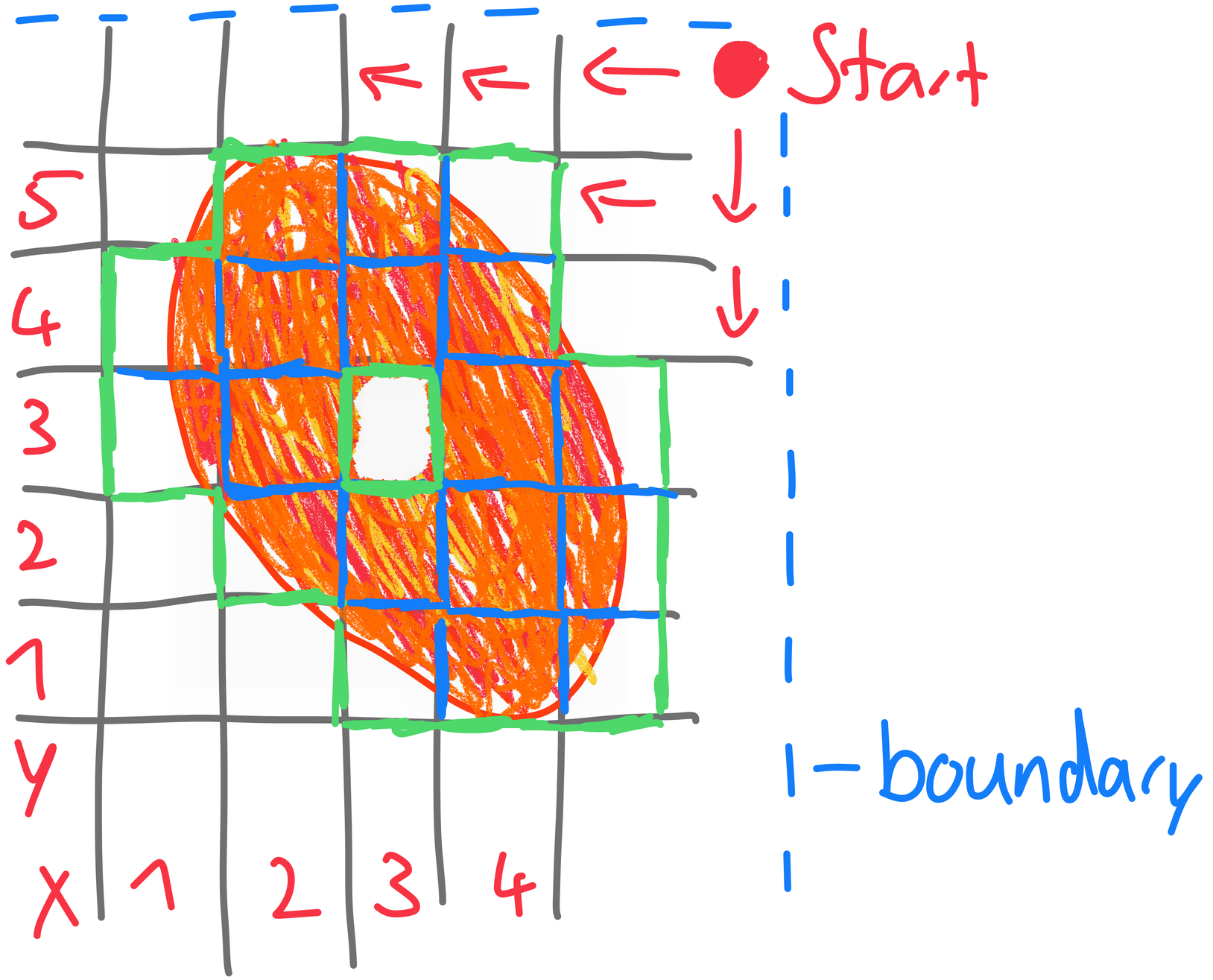


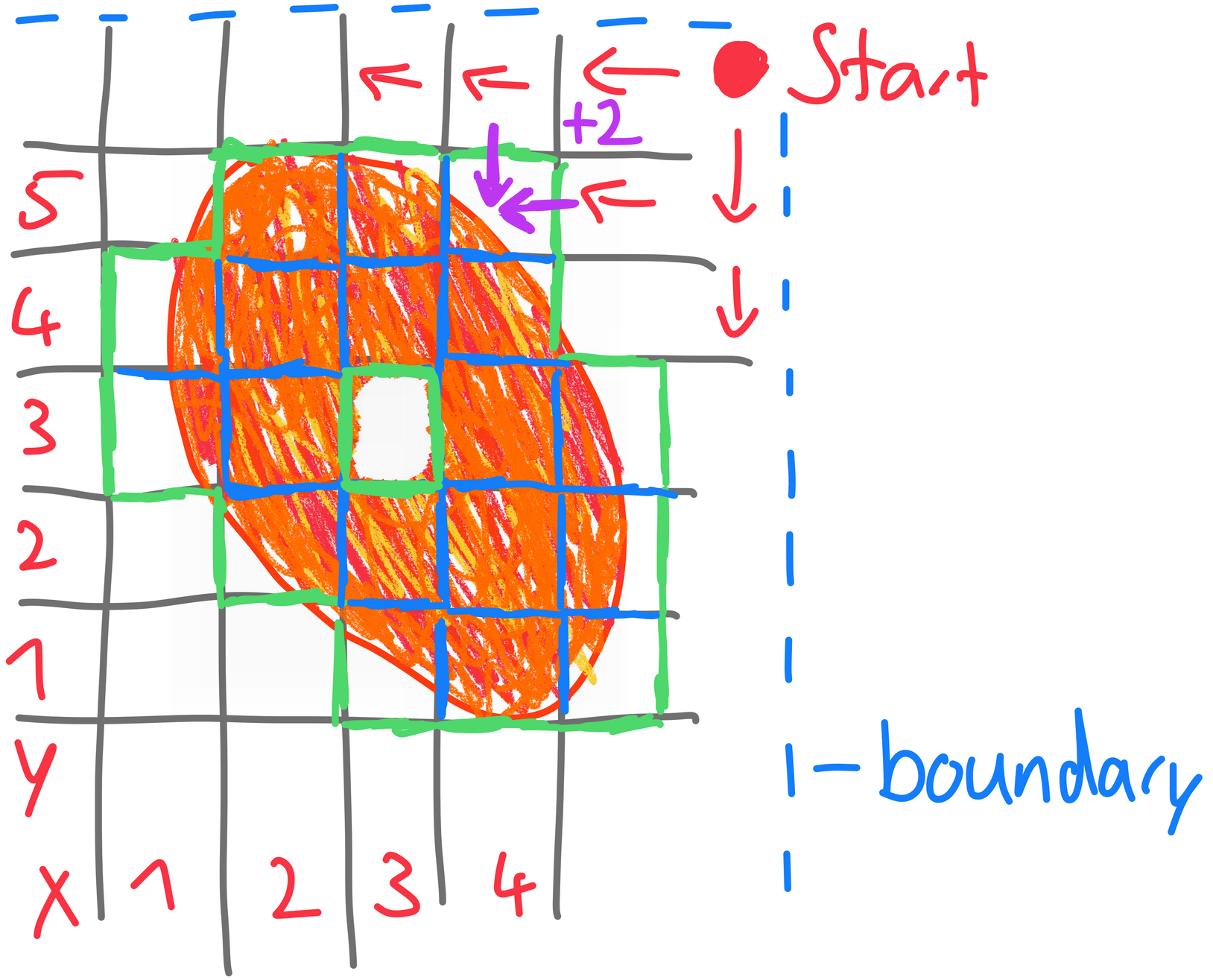
- Start: $\begin{pmatrix} \max x \\ \max y \\ \max z \end{pmatrix} + \underline{1}$

→ boundary analog


Start
5
4
3
2
1
y
x
1
2
3
4
boundary


Start
5
4
3
2
1
y
x
1
2
3
4
boundary


Start
5
4
3
2
1
y
x
1
2
3
4
boundary


Start
+2
5
4
3
2
1
y
x
1
2
3
4
boundary

```python
n = 0
visited = set()
queue = [(xmax, ymax, zmax)]
while queue:
  cube = queue.pop(0)
  for neighbor in neighbors(cube):
    if not in_boundary(neighbor):
      continue
    if neighbor in cubes:
      n += 1
    if neighbor in visited:
      continue
    visited.add(neighbor)
    queue.append(neighbor)
print(n)
```

APL IDE

Diskussion

APL?!

Such-Algorithmen

Andere Ansätze

Ende