

HEURISTICS:

A m s t e l h a e g e

Christiaan | Jos | Tara

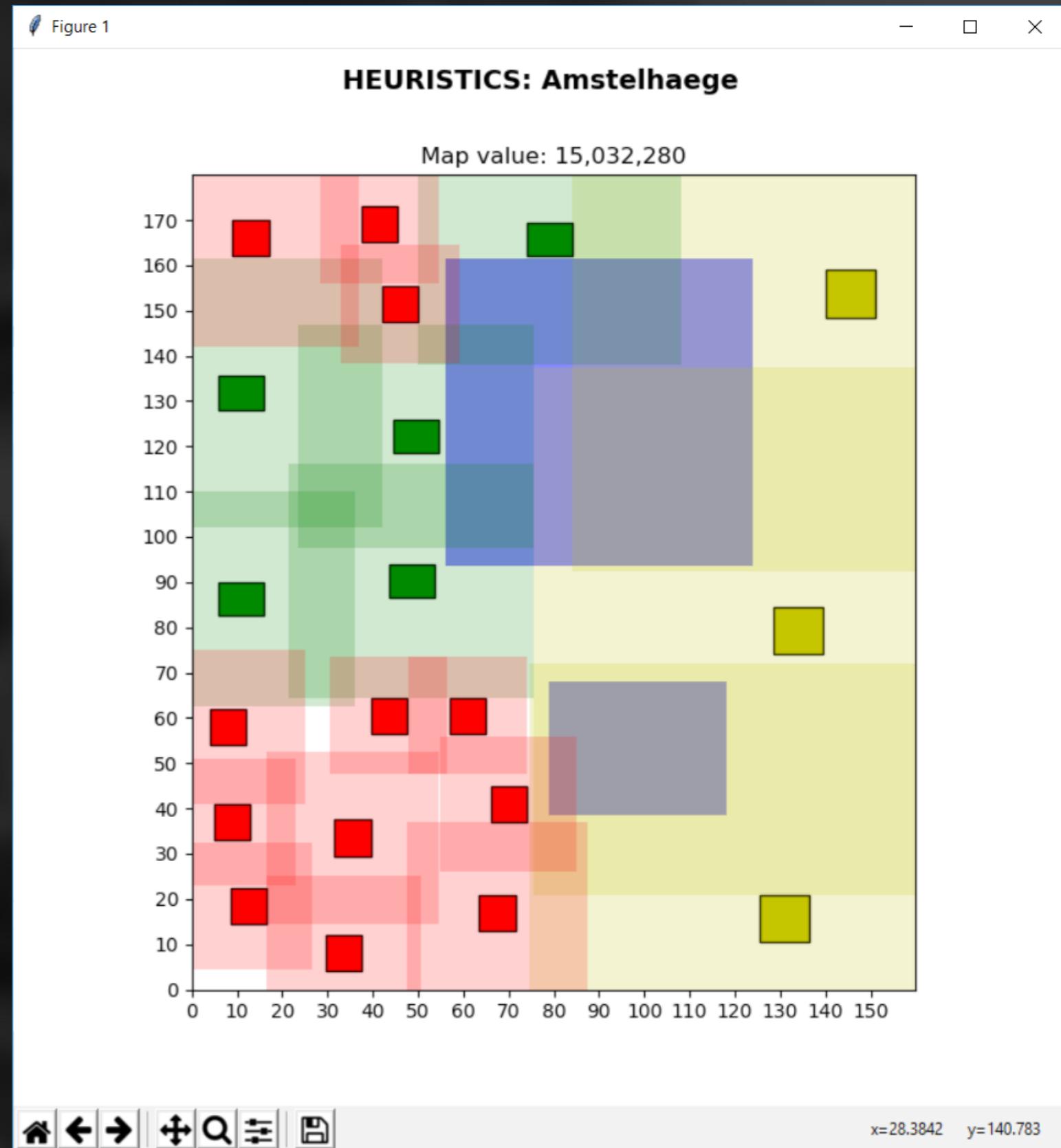
CONTENT:

1. The Case
2. Where we left of
3. Current algorithms
4. TODO

1. THE CASE

Build a map with
the most value
(opbrengst)

Variable to change
to achieve this:
distance in between
Houses



1. THE CASE

Dry info

```
# constances general
AREA = (160, 180)
HOUSE_COUNT = [20, 40, 60]

# water
WATER_PERCENTAGE = 0.20          # percentage of total area covered by water
MAX_BODIES        = 4             # maximum number of bodies
RATIO_UPPER_BOUND = 4             # l/b < x AND b/l < x
RATIO_LOWER_BOUND = 1 / RATIO_UPPER_BOUND
# WATER_COLOUR      = "b"
# STARTING_WATER_ITERATION_SIZE = 1.00

# house constances
NAME            = ["Family", "Bungalow", "Mansion"]
FREQUENCY       = [0.60,      0.25,      0.15]
VALUE           = [285000,    399000,    610000]
SITE            = [(8, 8),     (10, 7.5),   (11, 10.5)]
BASE_RING       = [2,          3,          6,]
RING_INCREMENT = [0.03,      0.04,      0.06]
```

1. THE CASE

NOTE: some given values look variable, but are constant. Examples:

- 6 % of Mansions's static price = static ring increment
- 60 % of all houses are family houses.
20, 40, 60 houses -> 12, 24, 36 F.houses respectively

1. THE CASE: COP

The only actual variable to take in account:

Position of houses

To optimize:

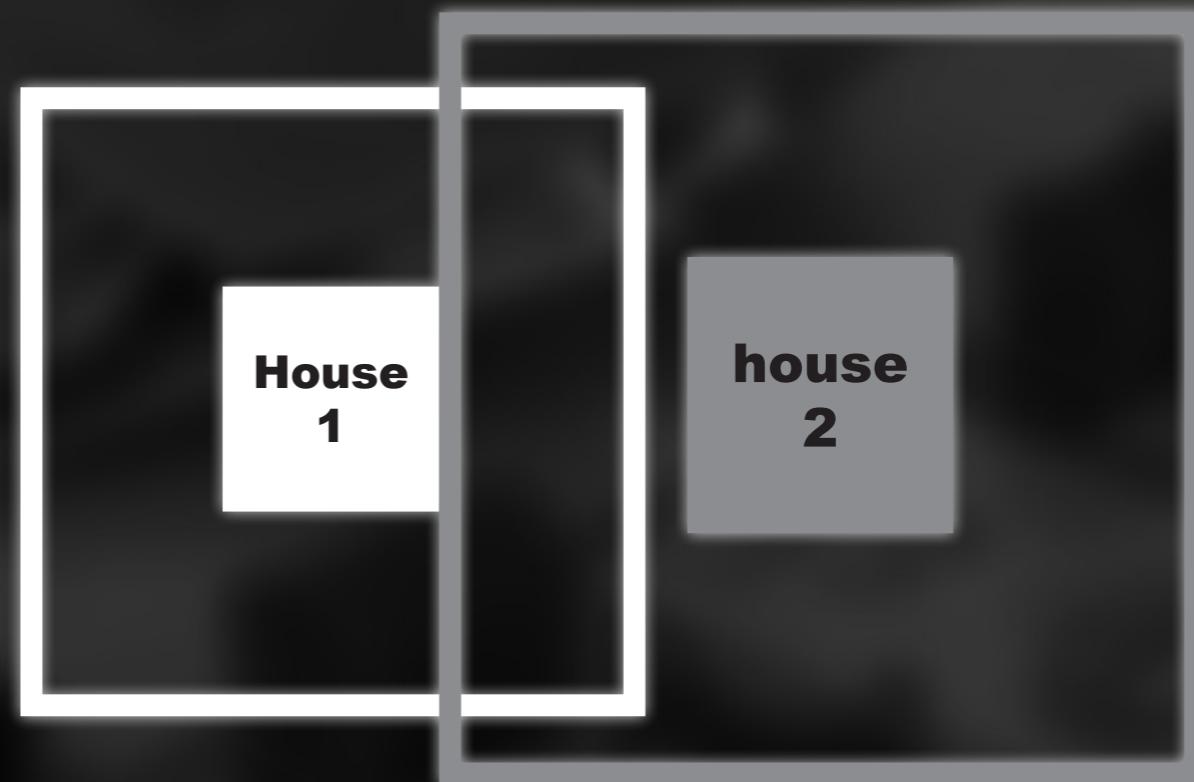
The distance in between houses

2. WHERE WE LEFT OF

- ring approach

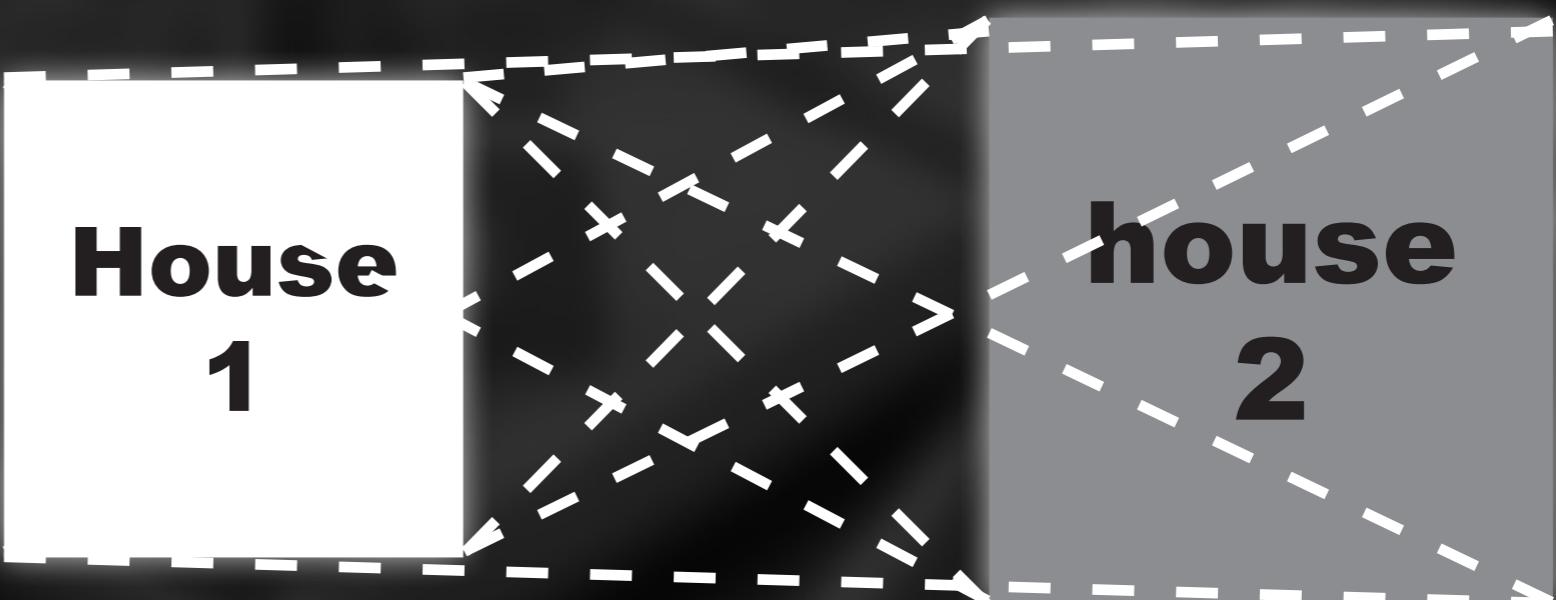
- personal space

- defined per house



- instead of

- Calculating distances



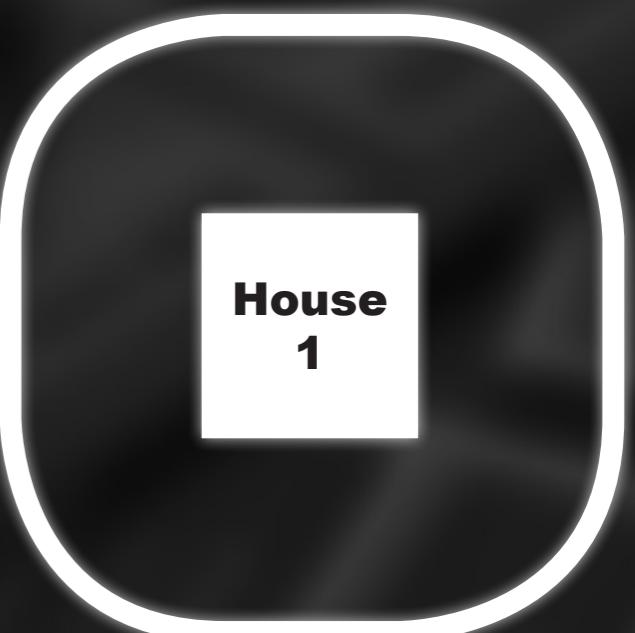
2. WHERE WE LEFT OF

- no “cutting corners”,

- so this:



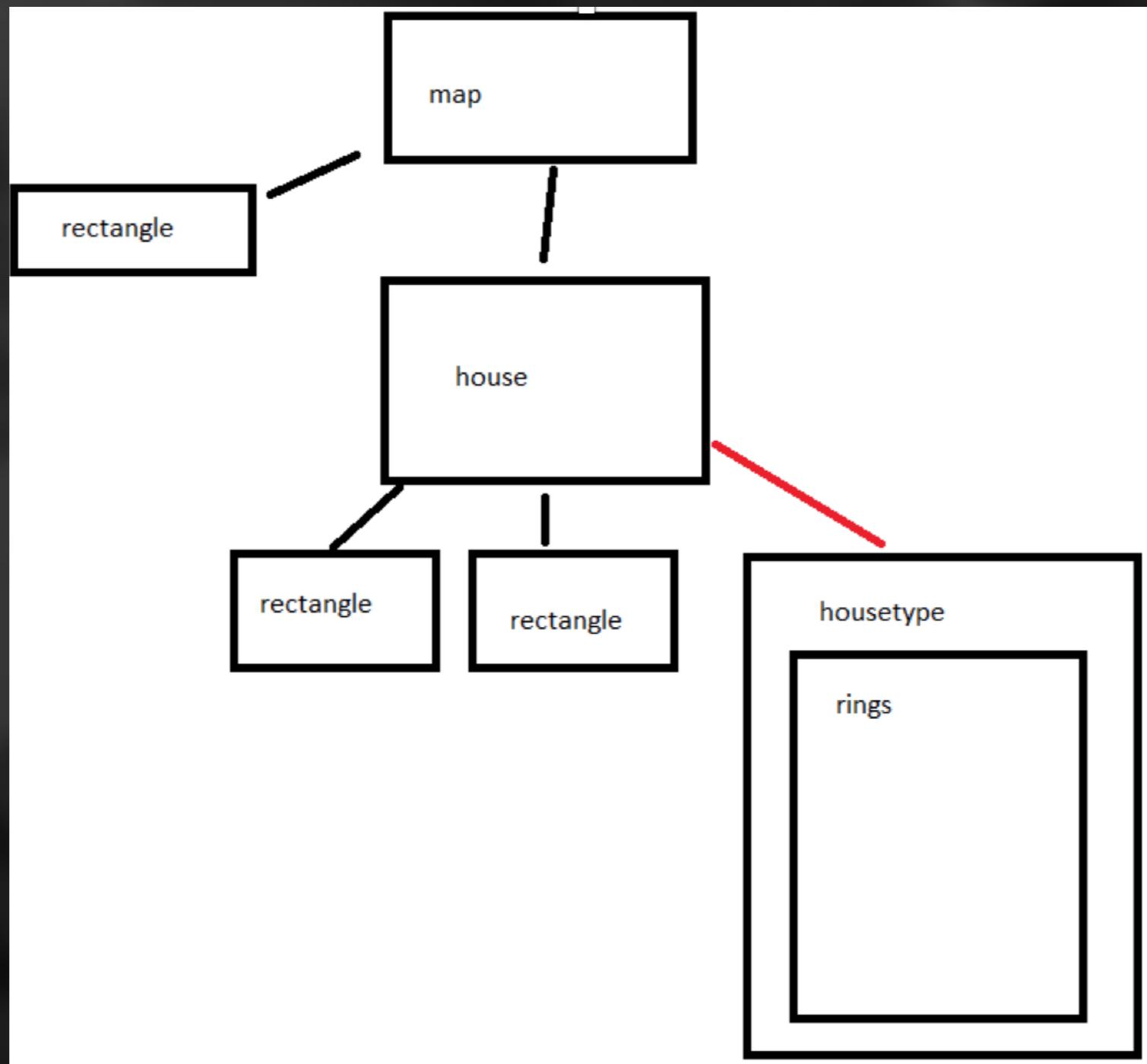
- Instead of this:



- its allowed, because its worse

2. WHERE WE LEFT OF

- class rectangle functionality



2. HEURISTICS

HEURISTICS NOTES

- [] prefer placing houses with the same type together
- [] prefer placing houses which perfectly fit together
- [] prefer placing houses on edges of other houses or water
- [] prefer placing houses so they fit perfectly in AREA boundaries
- [] prefer to place as much 'extra free space' off the edge of the AREA boundaries
- [] group houses, consider them as 1 (moldable) puzzle piece

3. JOS' ALGORITHM Base concept:

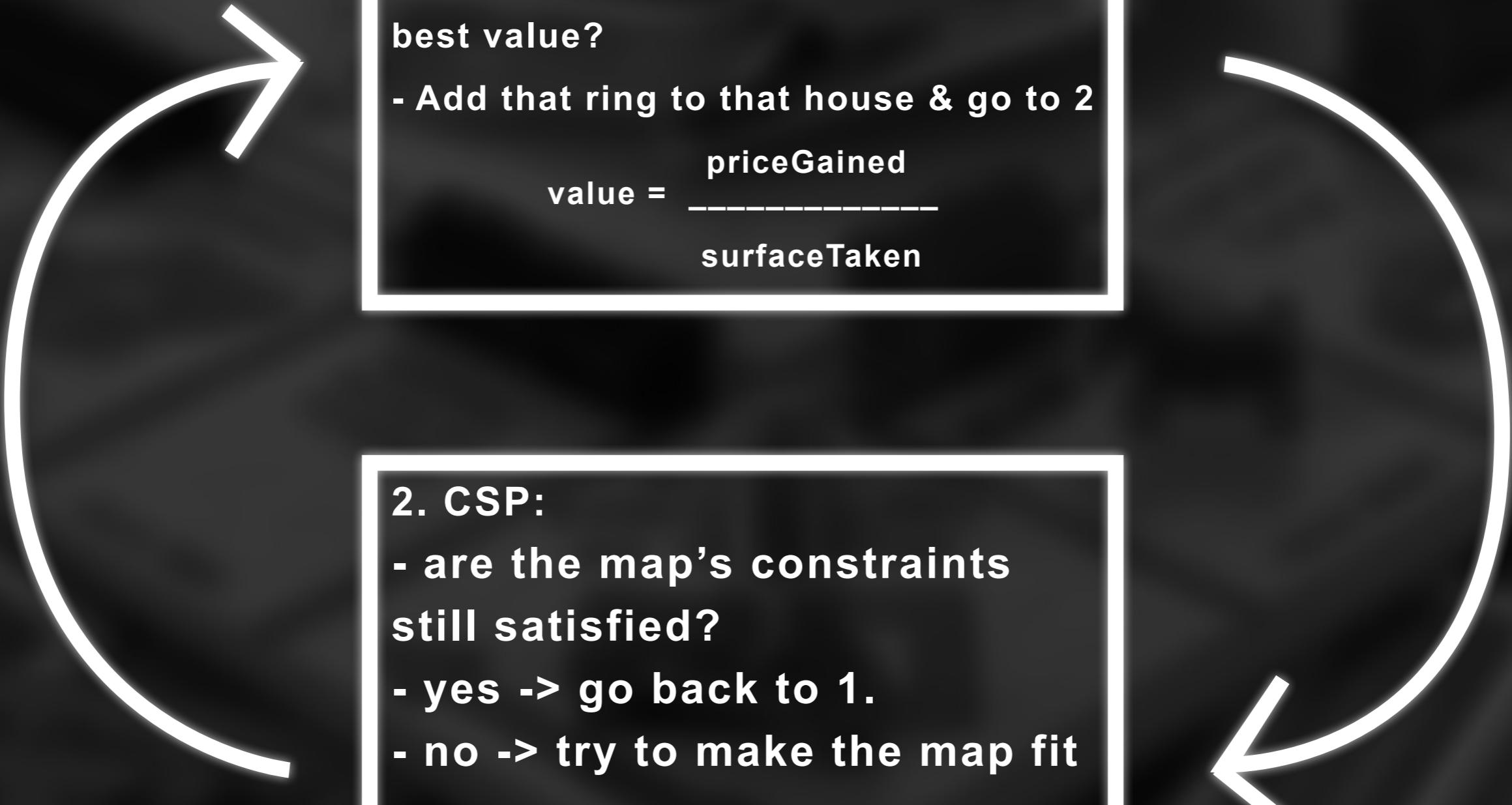
1. Greedy ring increaser:

- which house's next ring has the best value?
- Add that ring to that house & go to 2

$$\text{value} = \frac{\text{priceGained}}{\text{surfaceTaken}}$$

2. CSP:

- are the map's constraints still satisfied?
- yes -> go back to 1.
- no -> try to make the map fit



3. JOS' ALGORITHM V1.1

v 1.1:

- add ring to best house
- if that house doesnt fit anymore
 - jump to random location 1000 times until valid
 - if still not valid, crash

**TOO BAD TO EVEN SHOW
PICTURES ...**

v1.1

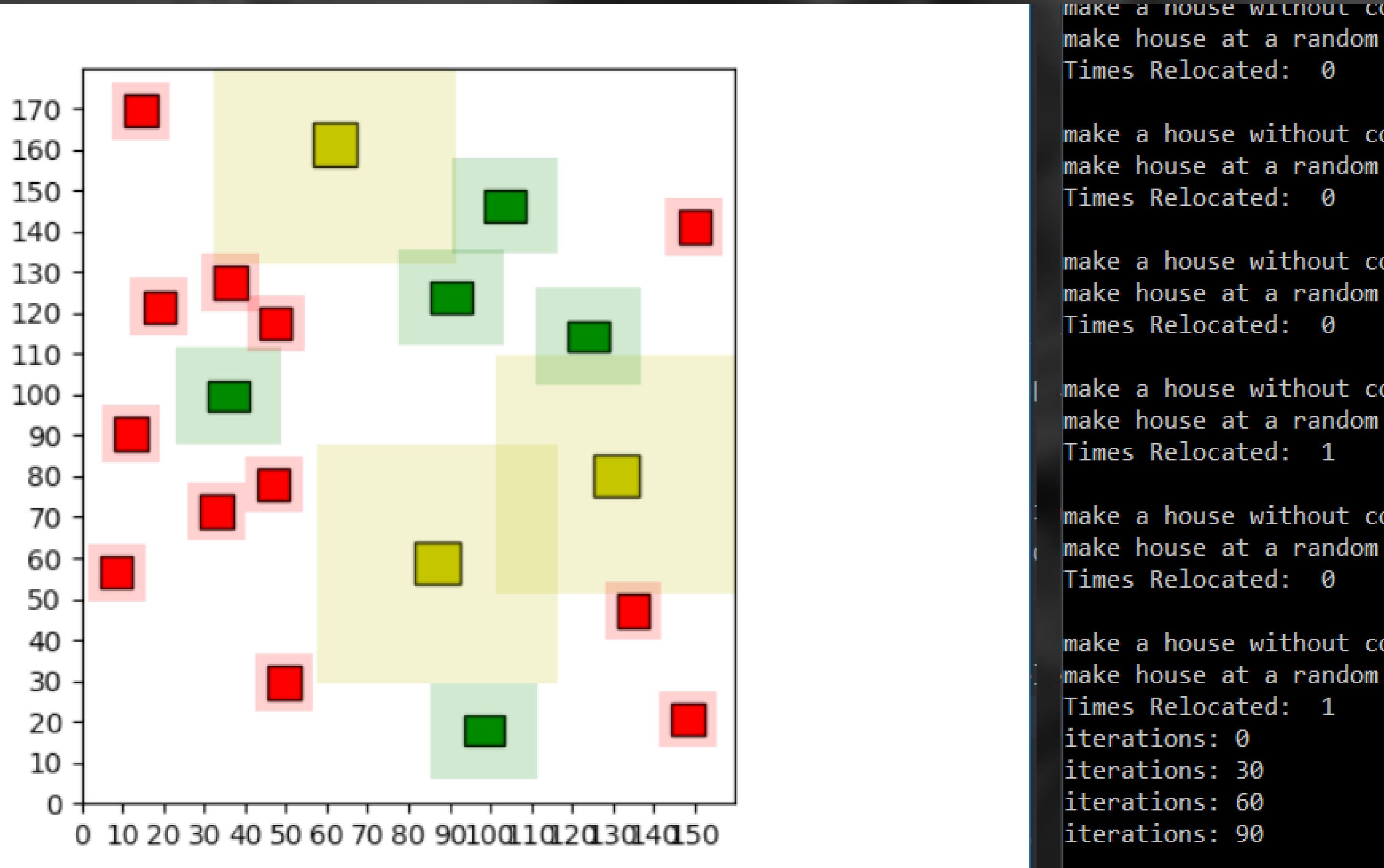
HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

50 iterations

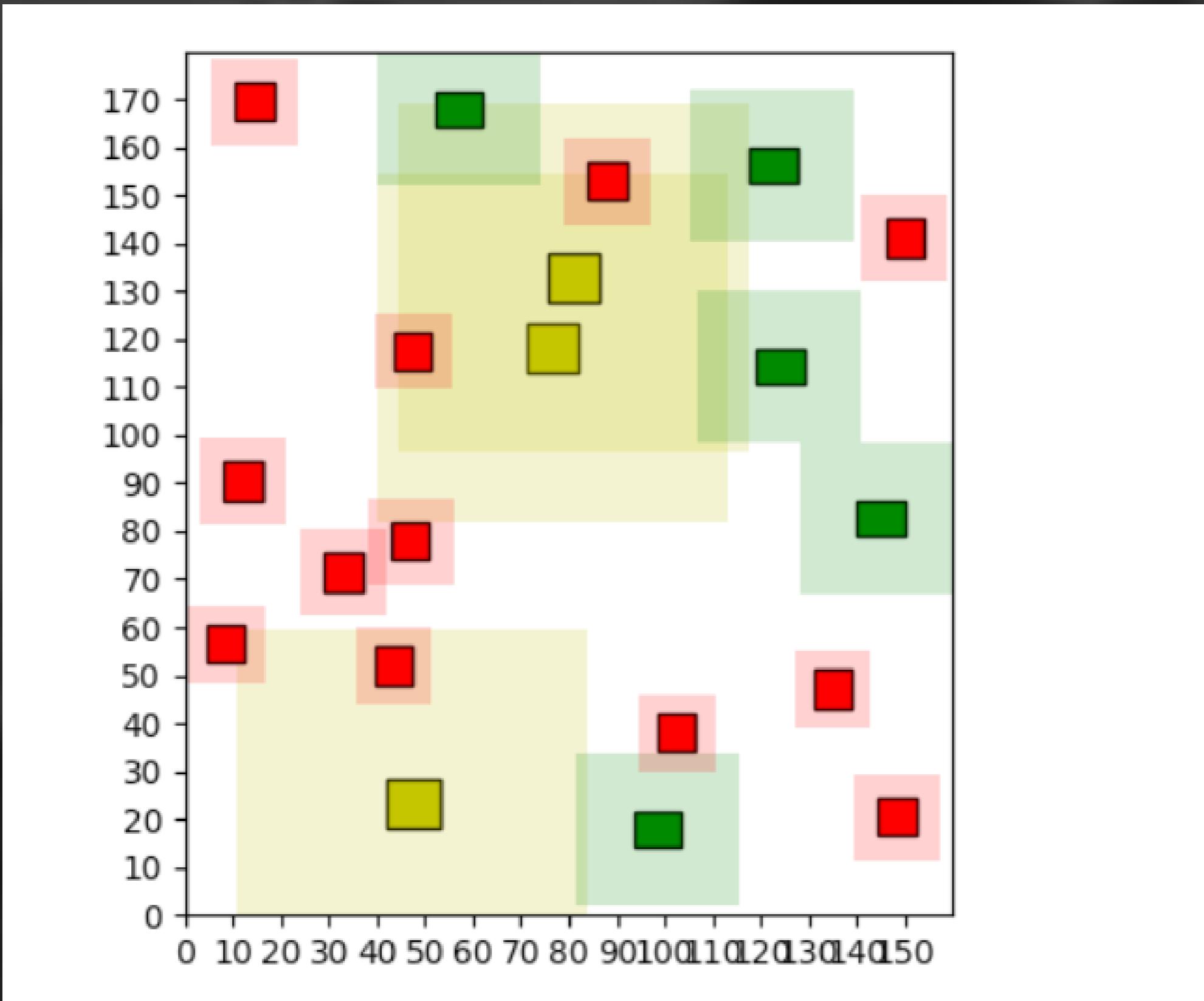
3. JOS' ALGORITHM V1.2

v 1.2:

- add ring to best house
- if one of the OTHER houses does not fit anymore:
 - random jump THEM 1000 times until valid



v1.2



v1.2:

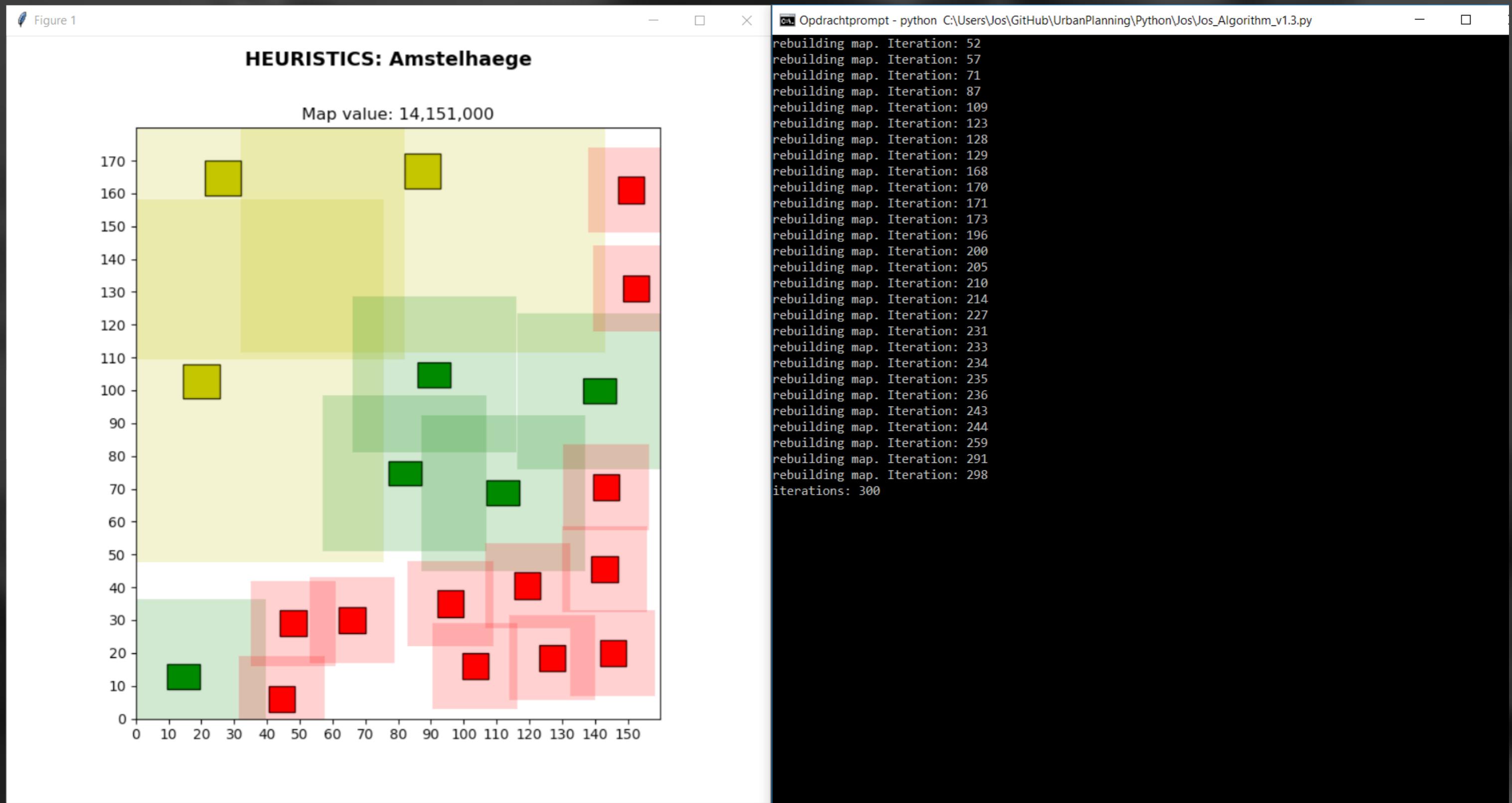
HEURISTICS: Amstelhaege Christiaan | Jos | Tara

Limit: 120 rings

3. JOS' ALGORITHM V1.3

v 1.3:

- add ring to best house
- if MAP CONSTRAINTS are not met:
 - PLOT THE COMPLETE MAP AGAIN
 - if a house of the map builder times out:
 - try building map again,
 - if map builder times out after X iterations, crash



v1.3:

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

Limit: 300 rings

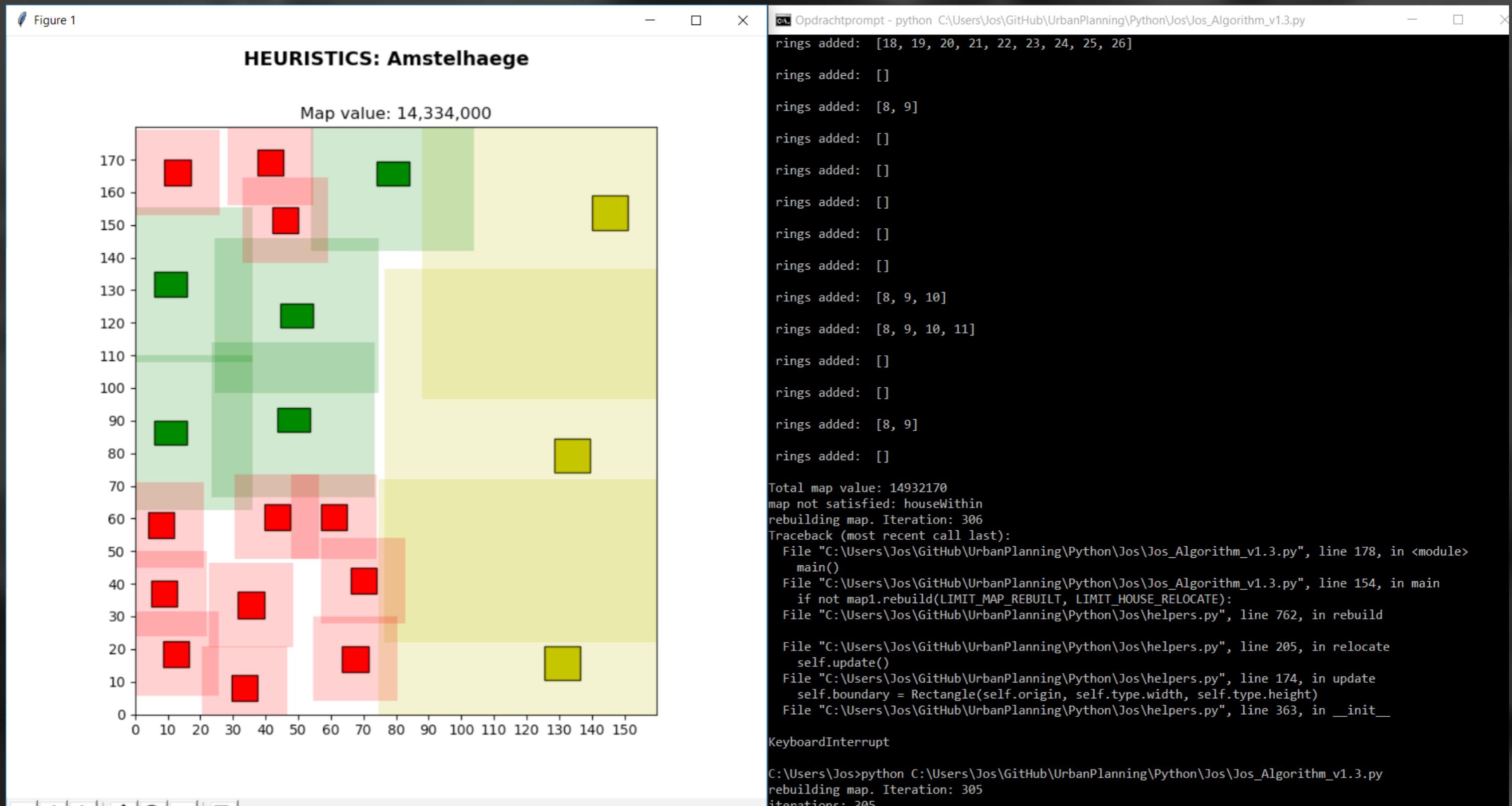
3. JOS' ALGORITHM V1.3.1

v 1.3.1:

in addition to v 1.3

- Up to STARTING_VALUE dont check if correct:
- if iteration count is new & it's map is solvable:
 - STARTING_VALUE that iteration

Figure 1



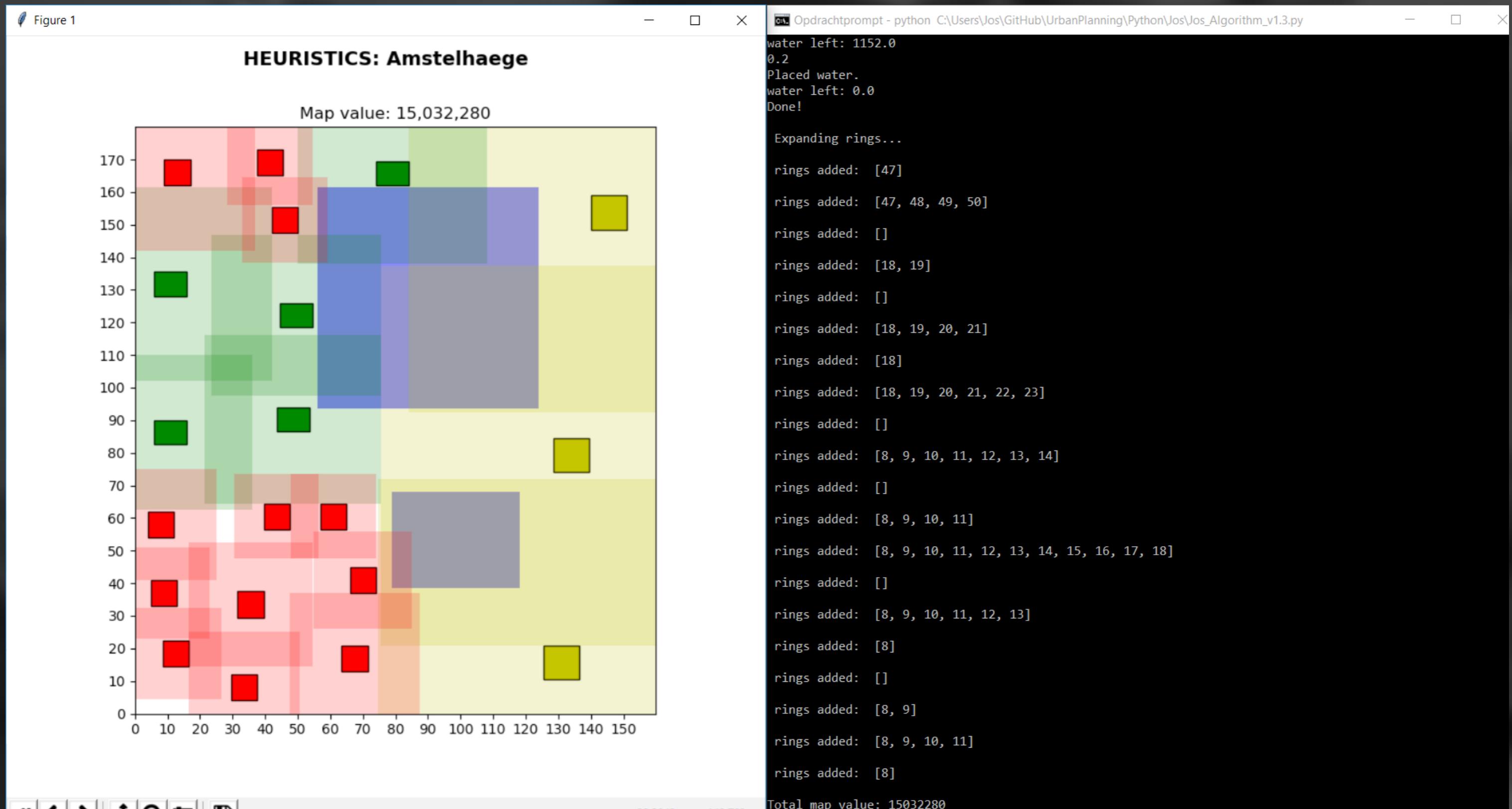
v1.3:

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

before enhancements

Limit: 312 rings

Figure 1



v1.3:

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

after enhancements

Limit: 312 rings

3. JOS' ALGORITHM V1.4

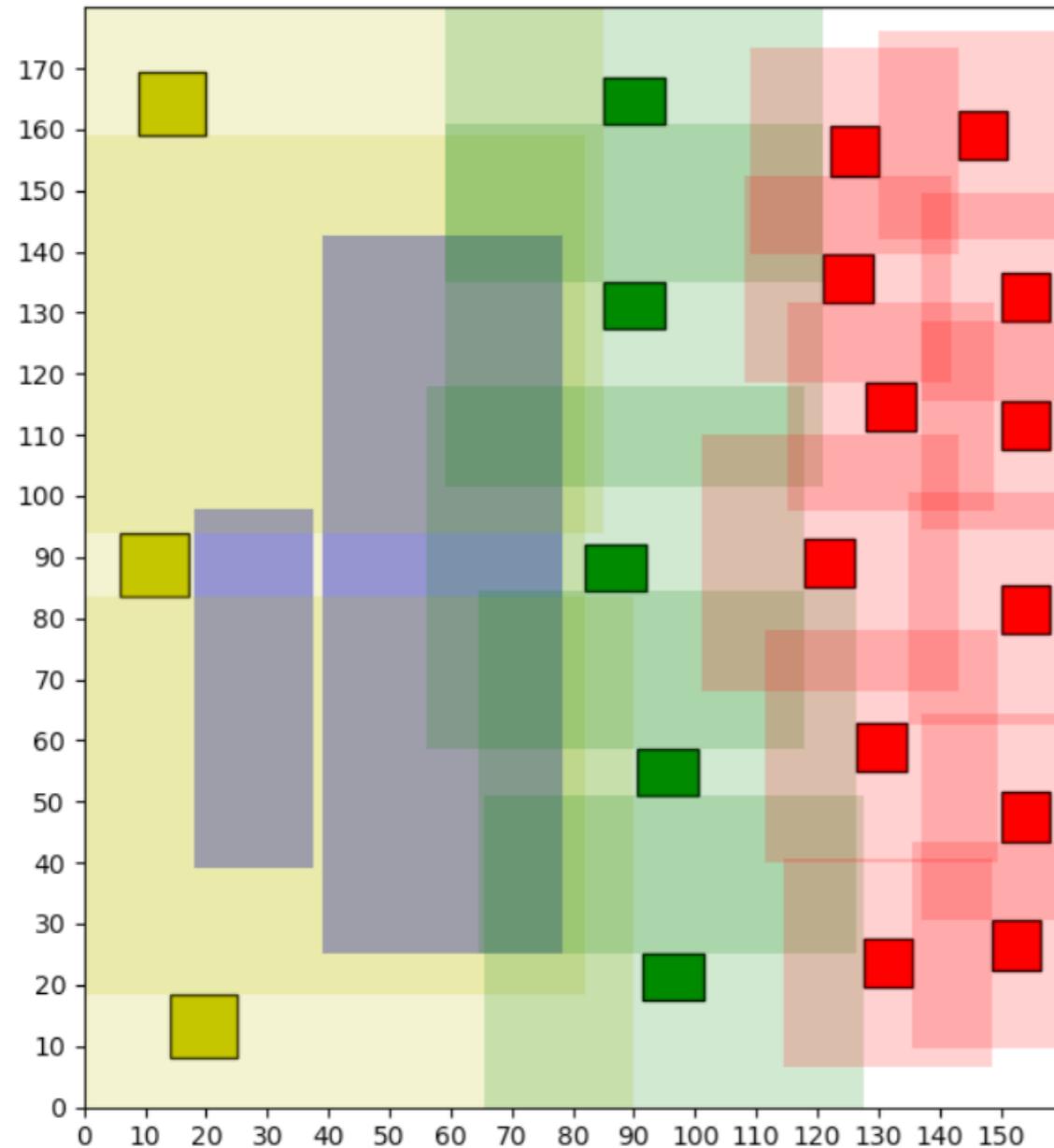
v 1.4

- add ring to best house
- plot the complete map again, IN A
SMARTER FASHION: build on edge of
existing geometry
- if a house times out,
 - do similar things as 1.3.1

Figure 1

HEURISTICS: Amstelhaege

Map value: 16,755,600



```
Opdrachtprompt - python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py
Failed to rebuild at iteration 416, Runtime Error, cannot continue...

C:\Users\Jos>python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py
rebuilding map. Iteration: 415
Traceback (most recent call last):
  File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py", line 196, in <module>
    main()
  File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py", line 168, in main
    if not map1.rebuild(LIMIT_MAP_REBUILT, LIMIT_HOUSE_RELOCATE):
  File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\helpers.py", line 848, in rebuild
    house.relocate("random_on_edge", edges)
  File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\helpers.py", line 204, in relocate
    picked = np.random.choice(range(len(edges)))
KeyboardInterrupt

C:\Users\Jos>python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py
rebuilding map. Iteration: 420
iterations: 420

adding 5760.0 m2 of water...
0.8
Placed water.
water left: 1152.0
0.2
Placed water.
water left: 0.0
Done!

Expanding rings...
rings added: []
rings added: [12, 13]
rings added: []
rings added: []
rings added: [12, 13]
rings added: [12, 13, 14, 15]
rings added: [11]
rings added: [11]
rings added: [11]
```

v1.3: 20 houses

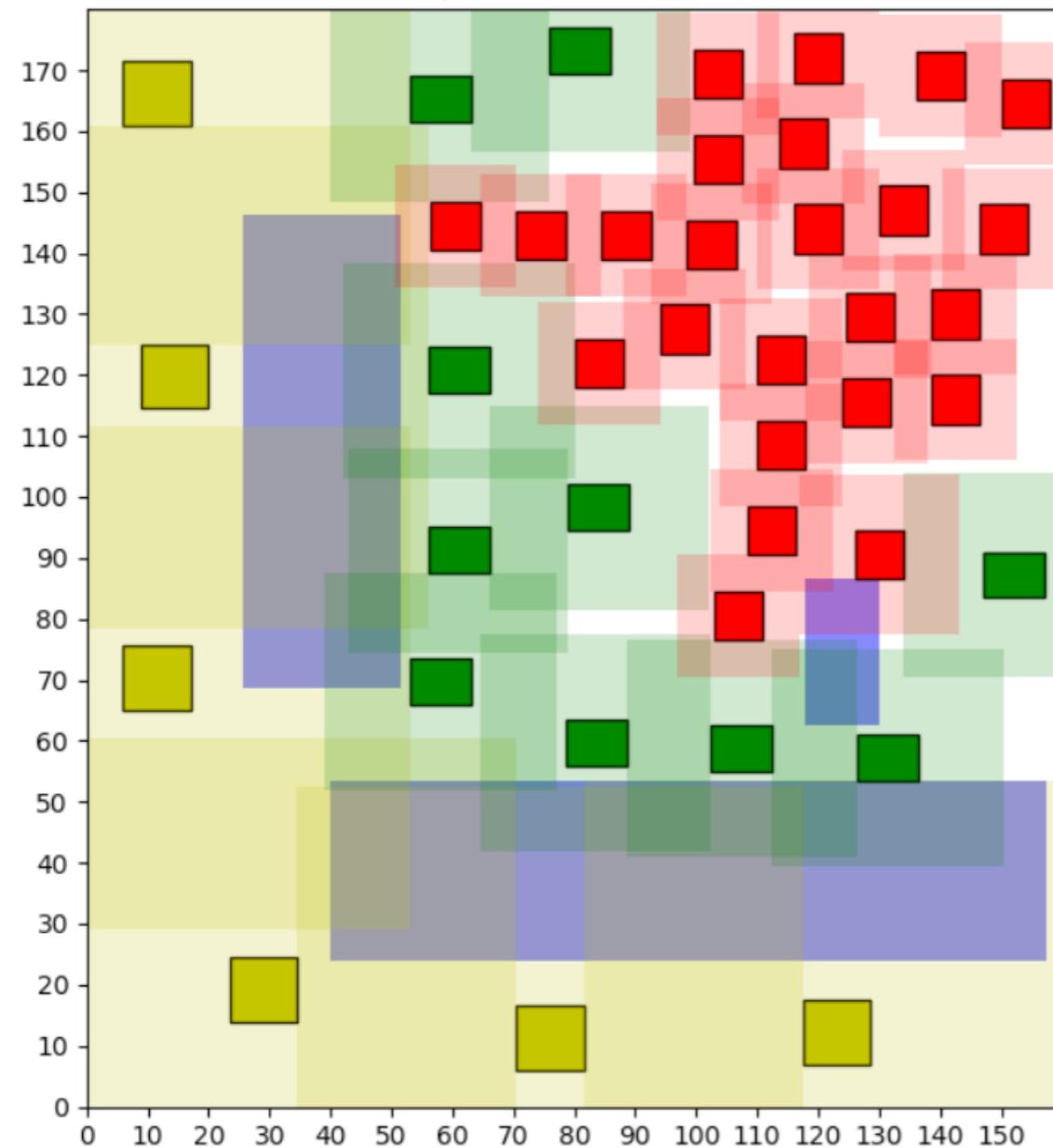
HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

Score: 16.8 mil.

Limit: 420 rings

HEURISTICS: Amstelhaege

Map value: 23,600,250



v1.3: 40 houses

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

```

Opdrachtprompt - python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py
picked = np.random.choice(range(len(edges)))
KeyboardInterrupt

C:\Users\Jos>python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py
rebuilding map. Iteration: 380
iterations: 380

adding 5760.0 m2 of water...
0.8
0.75
0.7
0.65
0.6
Placed water.
water left: 2304.0
0.4
0.35
Placed water.
water left: 288.0
0.05
Placed water.
water left: 0.0
Done!

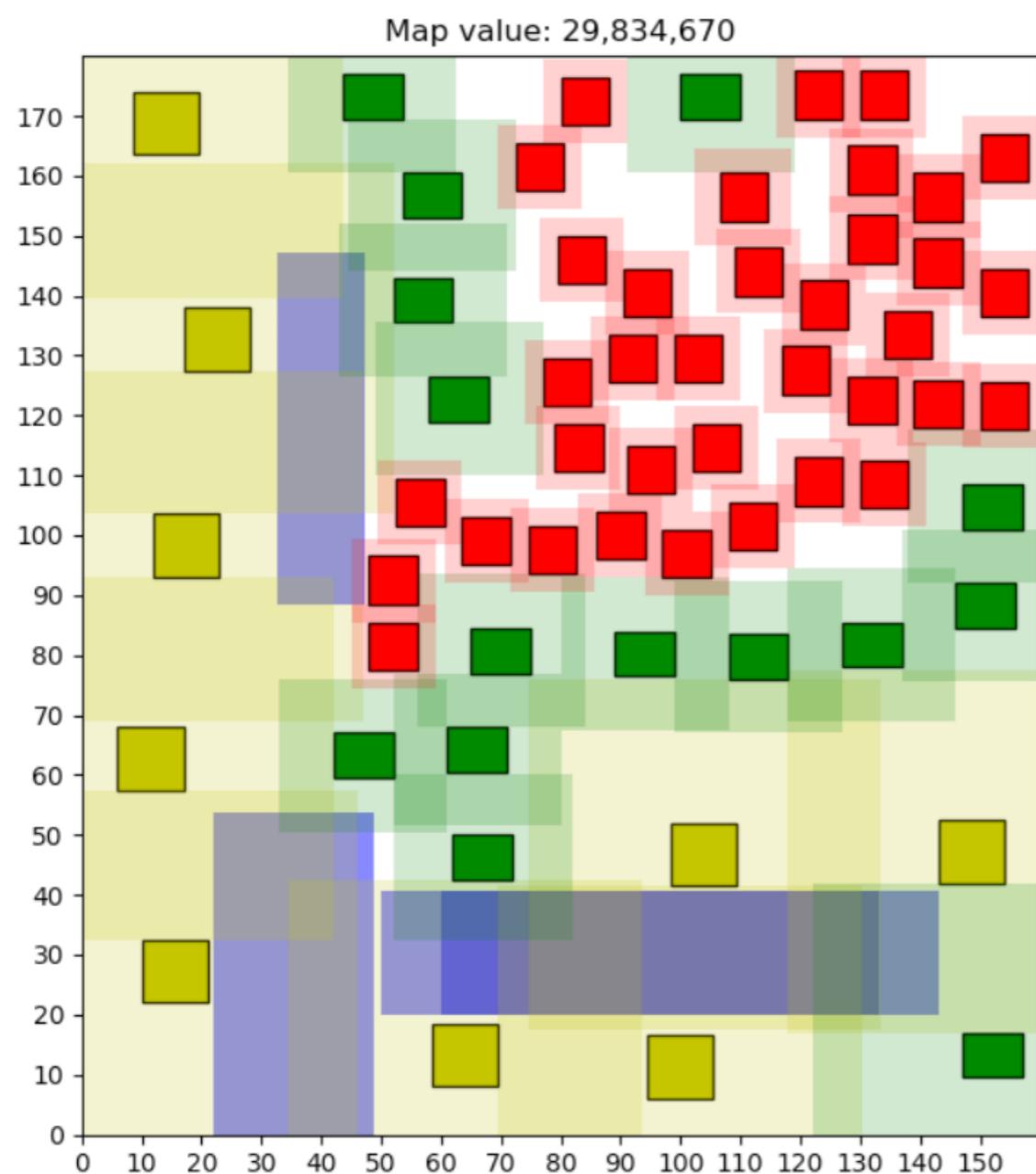
Expanding rings...
rings added: []
rings added: [5, 6, 7]
rings added: []

```

Score: 23.5 mil.
Limit: 380 rings

Figure 1

HEURISTICS: Amstelhaege



```
Opdrachtprompt - python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py
File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\helpers.py", line 483, in isTouching
    or B.x1 < A.x1 < B.x2      # one of your x's are within my x's
KeyboardInterrupt

C:\Users\Jos>python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py
File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\helpers.py", line 179
    while(map1.addWater())
           ^
SyntaxError: invalid syntax

C:\Users\Jos>python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py
File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\helpers.py", line 179
    while(map1.addWater())
           ^
SyntaxError: invalid syntax

C:\Users\Jos>python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py
rebuilding map. Iteration: 290
iterations: 290

adding 5760.0 m2 of water...
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
Placed water.
water left: 4032.0
0.3
Placed water.
water left: 2304.0
0.3
0.25
Placed water.
water left: 864.0
0.15
Placed water.
water left: 0.0
Done!

Expanding rings...
rings added: []
rings added: []
```

v1.3: 60 houses

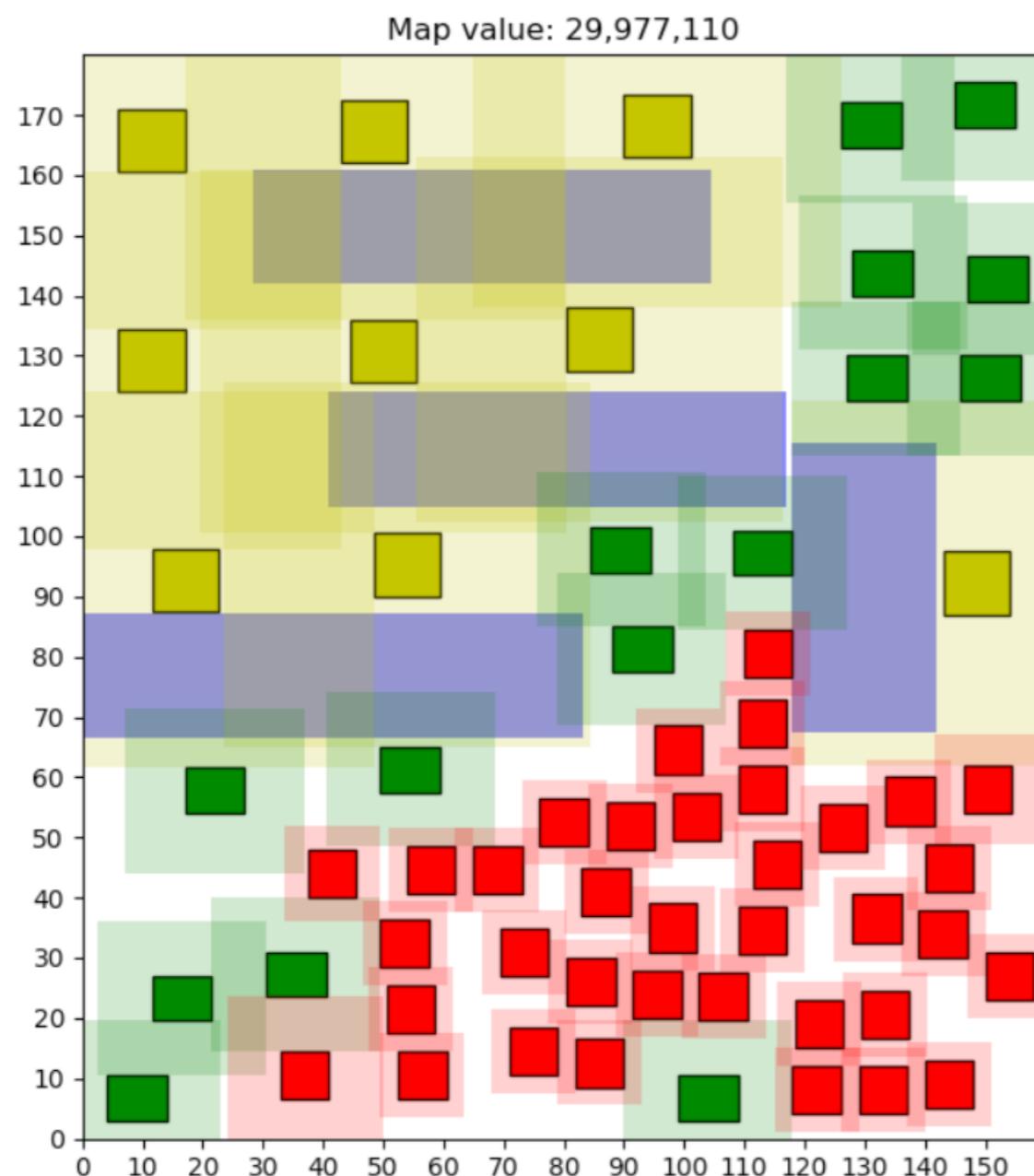
HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

not the best but PATTERNS

Limit: 290 rings

Figure 1

HEURISTICS: Amstelhaege



```
Opdrachtprompt - python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py
0.35
0.3
0.25
Failed to add water...

Expanding rings...
rebuilding map. Iteration: 301
Traceback (most recent call last):
  File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py", line 203, in <module>
    if __name__ == "__main__":
  File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py", line 169, in main
    if not map1.rebuild(LIMIT_MAP_REBUILT, LIMIT_HOUSE_RELOCATE):
  File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\helpers.py", line 837, in rebuild
    while(house.boundary.isTouching(compareRingBoundaries) or
  File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\helpers.py", line 489, in isTouching
    or B.y1 < A.y2 < B.y2)):
KeyboardInterrupt

C:\Users\Jos>python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.4.py
rebuilding map. Iteration: 300
iterations: 300

adding 5760.0 m2 of water...
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
Placed water.
water left: 4032.0
0.3
0.25
Placed water.
water left: 2592.0
0.25
Placed water.
water left: 1152.0
0.2
Placed water.
water left: 0.0
Done!

Expanding rings...
```

v1.3: 60 houses

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

Score: 29.9 mil.

Limit: 300 rings

3. CHRISTIAAN'S ALGORITHM

- Grid approach
- aligns houses
- increases personal space most valuable house
- repeat until it is impossible to add personal space to houses

4. TODO

HEURISTICS NOTES

- [:)] prefer placing houses with the same type together
- [:|] prefer placing houses which perfectly fit together
- [:)] prefer placing houses on edges of other houses or water
- [:|] prefer placing houses so they fit perfectly in AREA boundaries
- [:)] prefer to place as much 'extra free space' off the edge of the AREA boundaries
- [:(| group houses, consider them as 1 (moldable) puzzle piece

4. TODO

Saving & Loading maps using Tara's methods

Making the hillclimber automatic, not manual

Make Jos Algorithm implement: "simulated annealing"

Make Jos Algorithm change existing maps, instead of redoing them over and over again

Build a "Shape Judger"

THANK YOU FOR YOUR TIME !

QUESTIONS?