HEURISTICS:

Amstelhaege

Christiaan | Jos | Tara

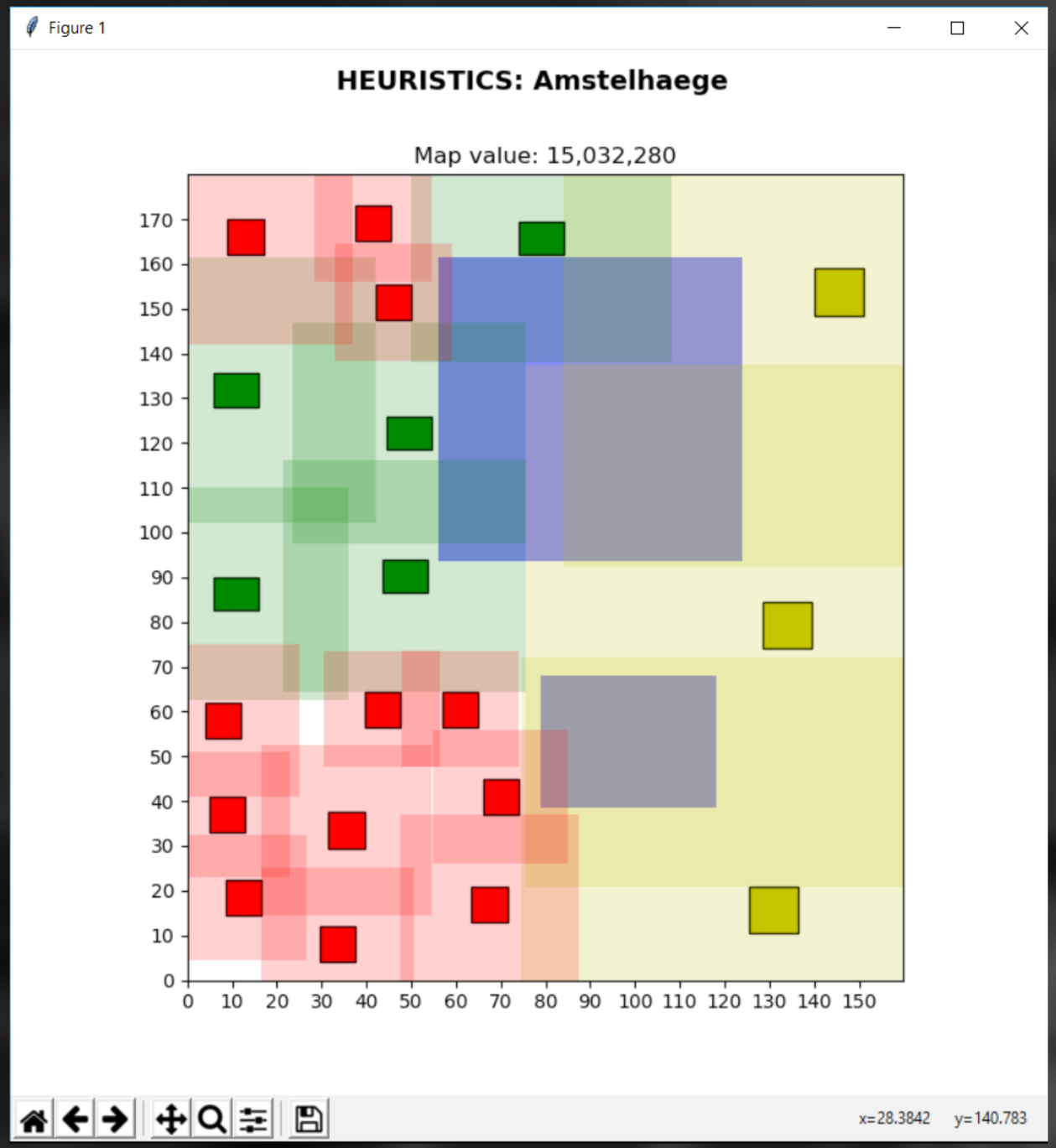
CONTENT:

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

1. THE CASE

Build a map with the most value (opbrengst)

Variable to change to achieve this: distance in between Houses



HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

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       ]
```

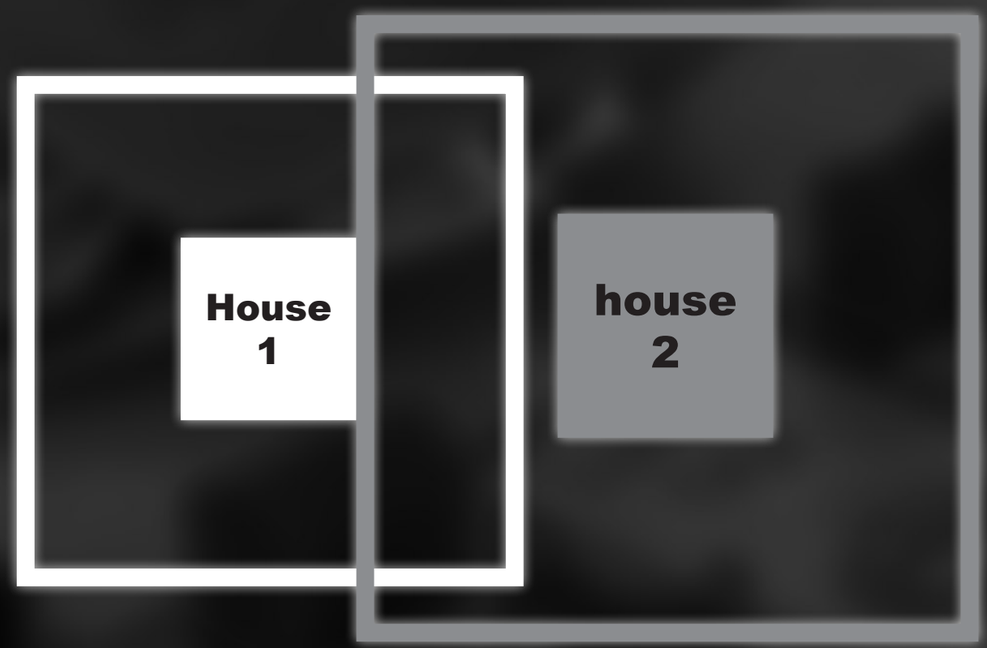
1. THE CASE

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

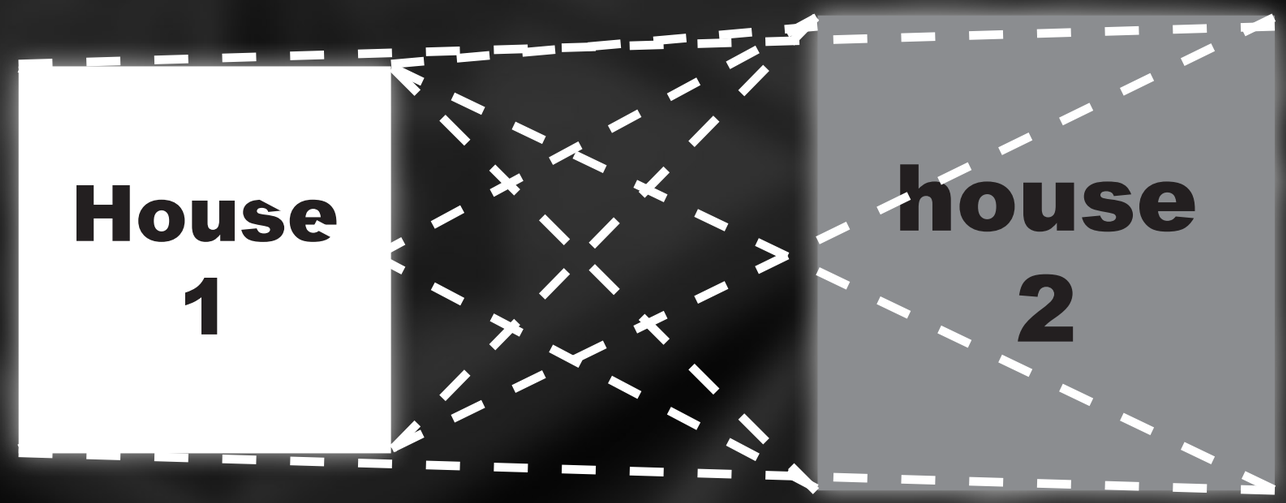
- 6 % of Mansions's static price = static ring increment
- 60 % of all houses are family houses.
  20, 40, 60 houses -> 12, 24, 36 F.houses respectively

2. WHERE WE LEFT OF

- ring approach



- instead of

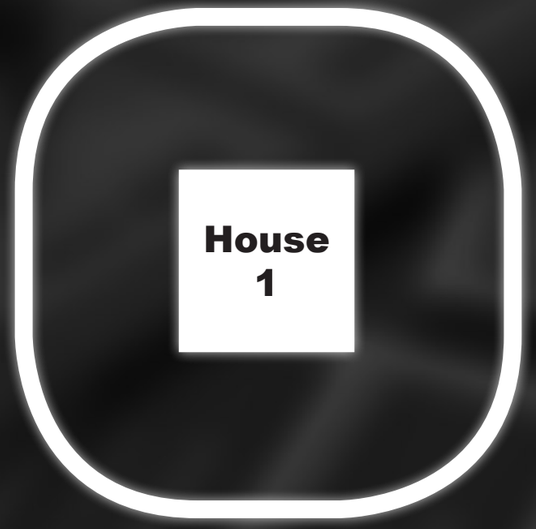


2. WHERE WE LEFT OF

- no “cutting corners”,

  so this:

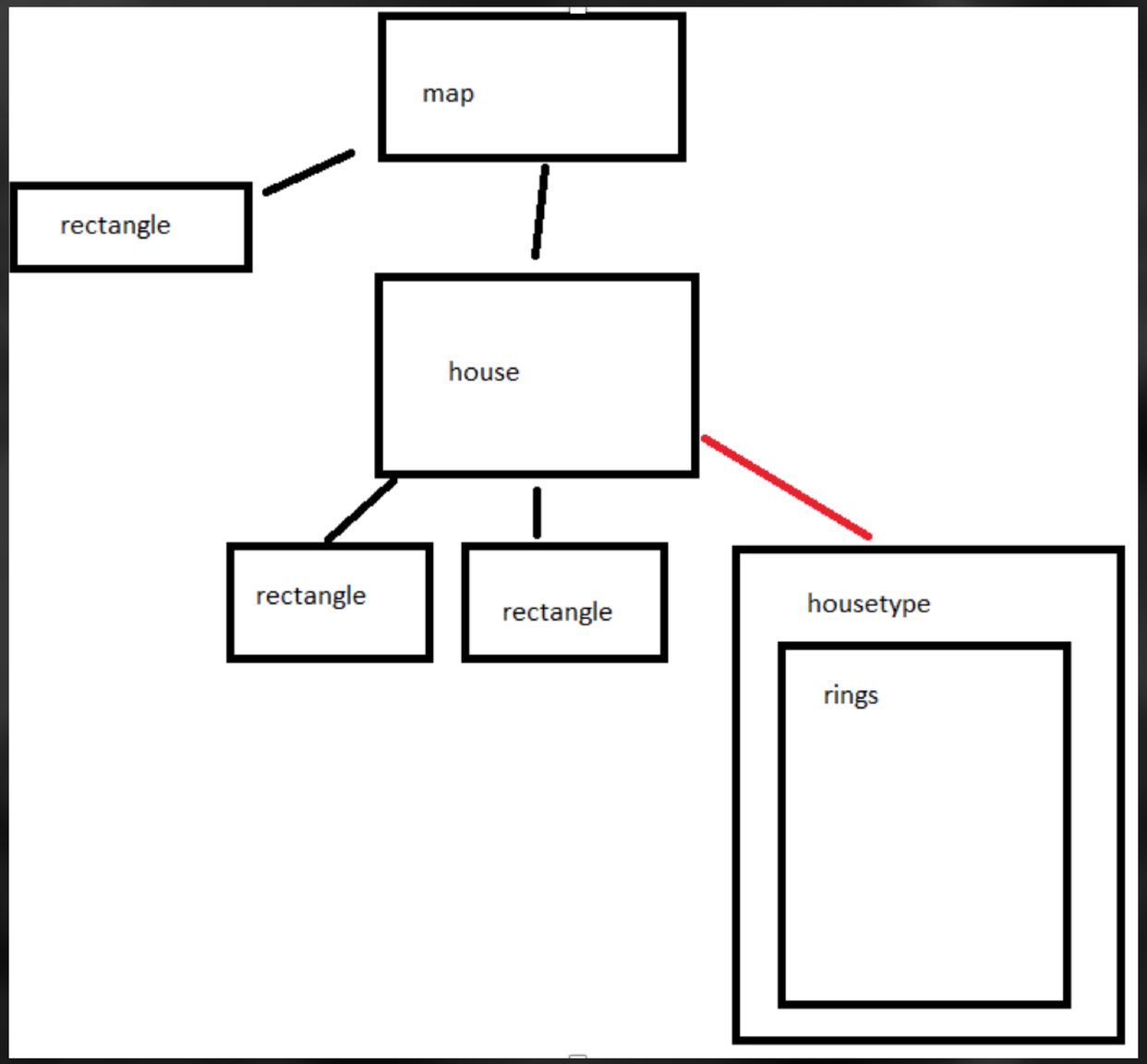
- Instead of this:



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

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

2. CSP:

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

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

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

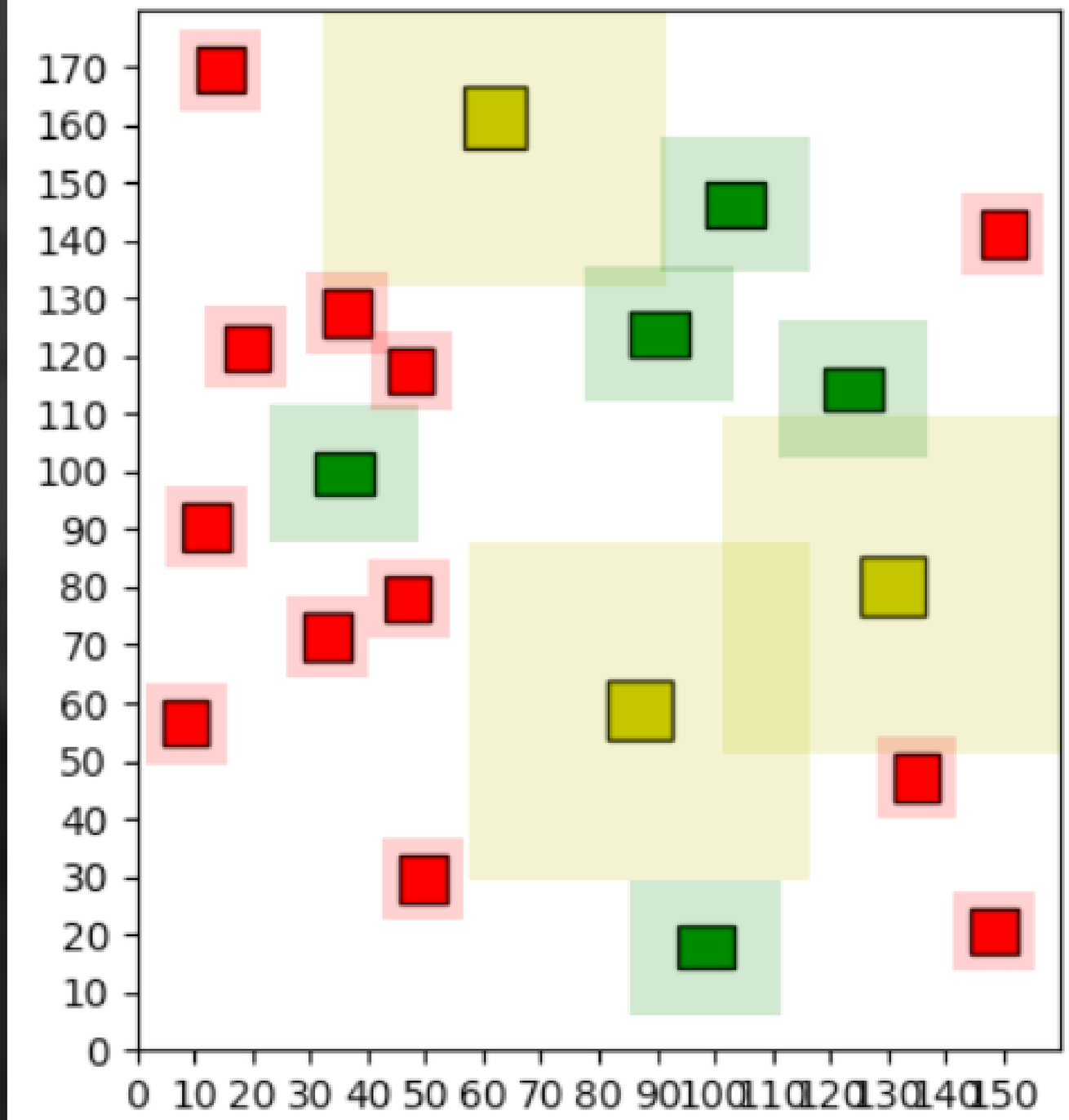
HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

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

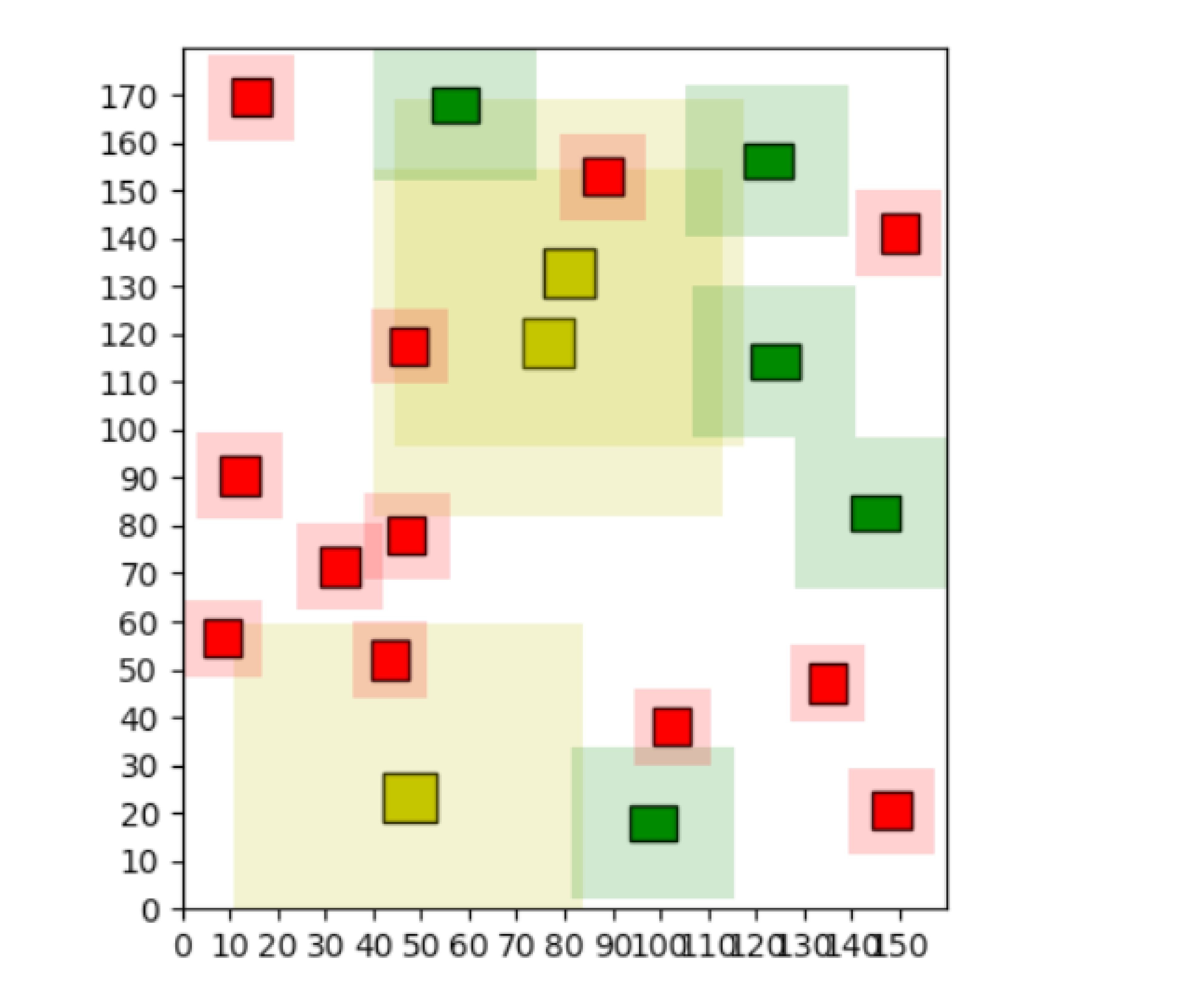



```
make a house without cc
make house at a random
Times Relocated:  0

make a house without cc
make house at a random
Times Relocated:  0

make a house without cc
make house at a random
Times Relocated:  0

make a house without cc
make house at a random
Times Relocated:  1

make a house without cc
make house at a random
Times Relocated:  0

make a house without cc
make house at a random
Times Relocated:  0

make a house without cc
make house at a random
Times Relocated:  1
iterations: 0
iterations: 30
iterations: 60
iterations: 90
```

v1.2

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

90 iterations




```
Iterations: 120
ERROORORORORORORO
houseClash:Mansion and Mansion
ERROORORORORORORO
houseClash:Mansion and Mansion
ERROORORORORORORO
houseClash:Mansion and Mansion
ERROORORORORORORO
houseClash:Mansion and Mansion
ERROORORORORORORO
houseClash:Mansion and Mansion
ERROORORORORORORO
houseClash:Mansion and Mansion
ERROORORORORORORO
houseClash:Mansion and Mansion
ERROORORORORORORO
houseClash:Mansion and Mansion
ERROORORORORORORO
houseClash:Mansion and Mansion
ERROORORORORORORO
houseClash:Mansion and Mansion
ERROORORORORORORO
houseClash:Mansion and Mansion
ERROORORORORORORO
houseClash:Mansion and Mansion
ERROORORORORORORO
houseClash:Mansion and Mansion
ERROORORORORORORO
houseClash:Mansion and Mansion
ERROORORORORORORO
```

v1.2:

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

Limit: 120 rings

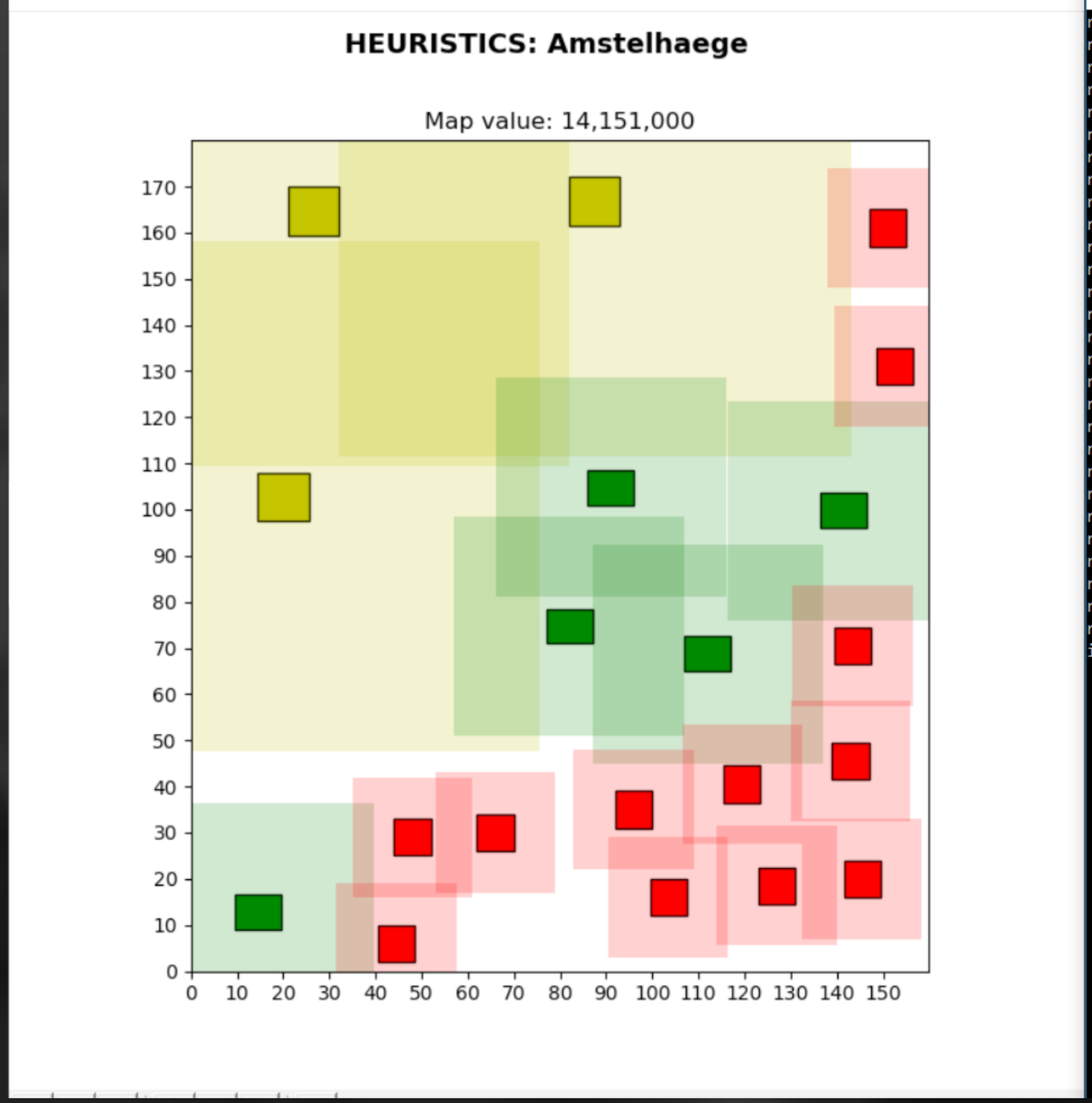
3. JOS' ALGORITHM V1.3

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 itera tions, crash

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

Figure 1



Opdrachtprompt - python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.3.py

```
rebuilding map. Iteration: 52
rebuilding map. Iteration: 57
rebuilding map. Iteration: 71
rebuilding map. Iteration: 87
rebuilding map. Iteration: 109
rebuilding map. Iteration: 123
rebuilding map. Iteration: 128
rebuilding map. Iteration: 129
rebuilding map. Iteration: 168
rebuilding map. Iteration: 170
rebuilding map. Iteration: 171
rebuilding map. Iteration: 173
rebuilding map. Iteration: 196
rebuilding map. Iteration: 200
rebuilding map. Iteration: 205
rebuilding map. Iteration: 210
rebuilding map. Iteration: 214
rebuilding map. Iteration: 227
rebuilding map. Iteration: 231
rebuilding map. Iteration: 233
rebuilding map. Iteration: 234
rebuilding map. Iteration: 235
rebuilding map. Iteration: 236
rebuilding map. Iteration: 243
rebuilding map. Iteration: 244
rebuilding map. Iteration: 259
rebuilding map. Iteration: 291
rebuilding map. Iteration: 298
iterations: 300
```

v1.3:

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

Limit: 300 rings

3. JOS' ALGORITHM V1.3.1

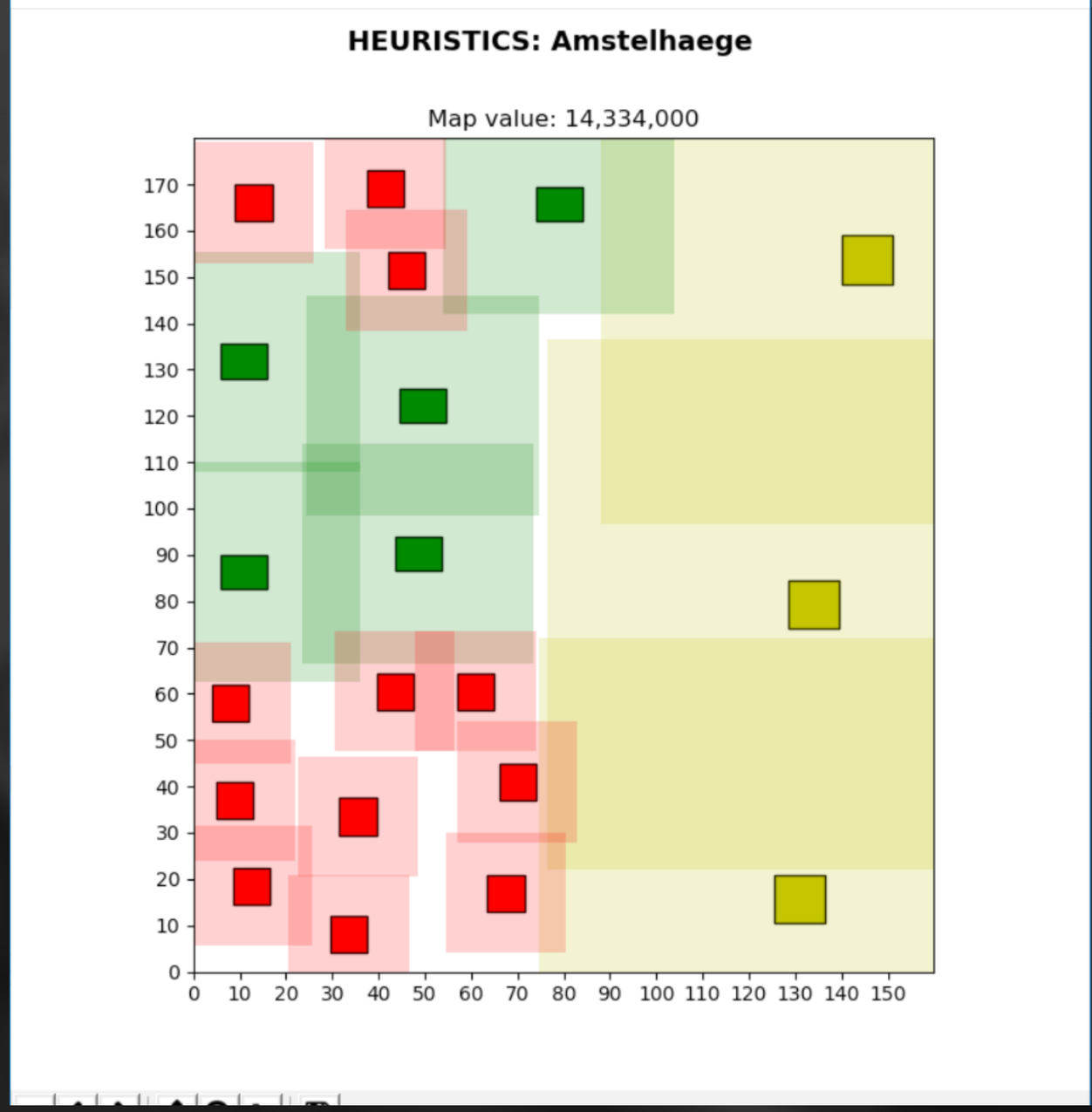
v 1.3.1:

in addition to v 1.3

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

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

Figure 1



Opdrachtprompt - python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.3.py

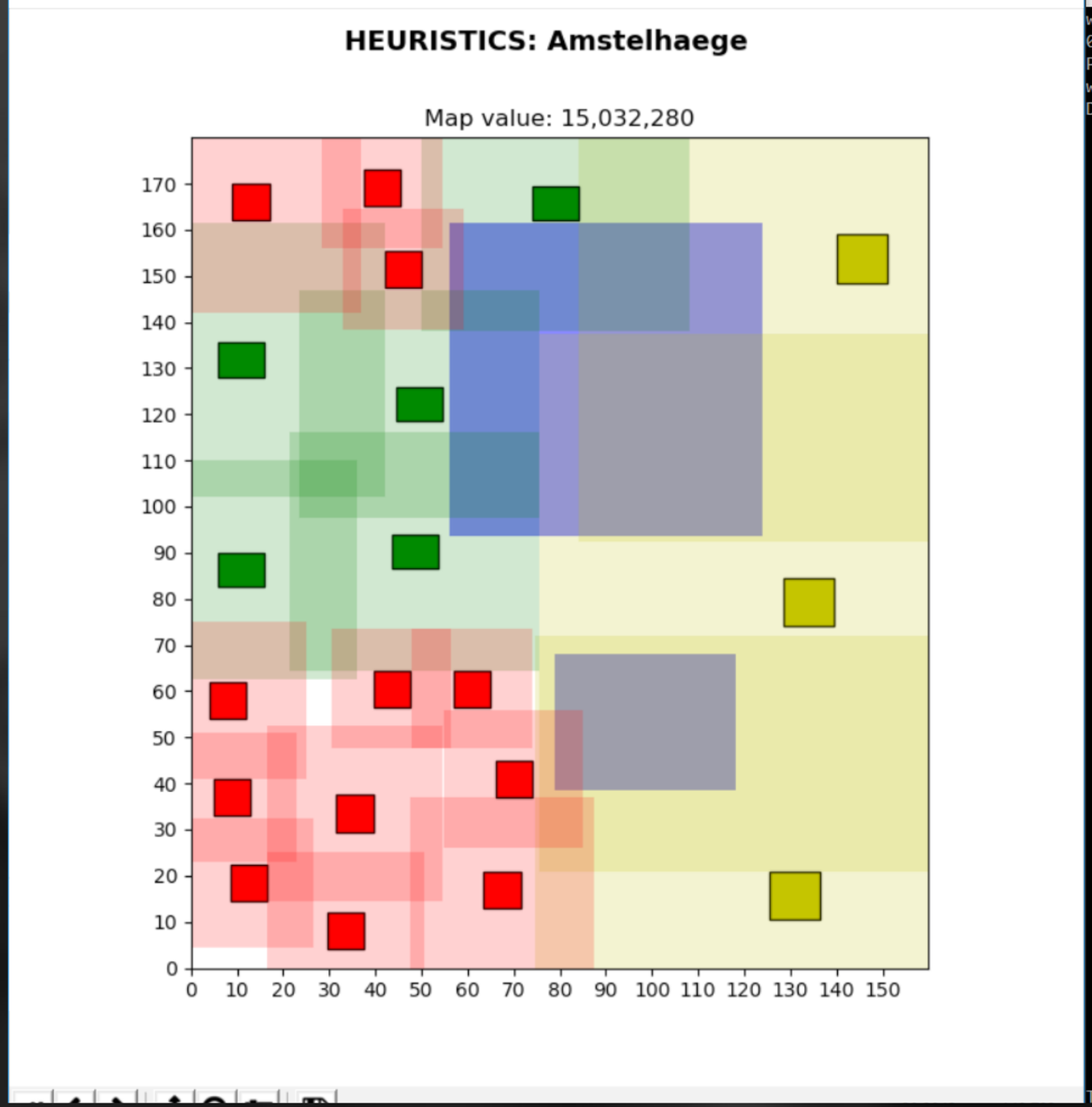
```
rings added:  [18, 19, 20, 21, 22, 23, 24, 25, 26]

rings added:  []

rings added:  [8, 9]

rings added:  []

rings added:  []

rings added:  []

rings added:  []

rings added:  []

rings added:  []

rings added:  [8, 9, 10]

rings added:  [8, 9, 10, 11]

rings added:  []

rings added:  []

rings added:  [8, 9]

rings added:  []

Total map value: 14932170
map not satisfied: houseWithin
rebuilding map. Iteration: 306
Traceback (most recent call last):
  File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.3.py", line 178, in <module>
    main()
  File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.3.py", line 154, in main
    if not map1.rebuild(LIMIT_MAP_REBUILT, LIMIT_HOUSE_RELOCATE):
  File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\helpers.py", line 762, in rebuild

  File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\helpers.py", line 205, in relocate
    self.update()
  File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\helpers.py", line 174, in update
    self.boundary = Rectangle(self.origin, self.type.width, self.type.height)
  File "C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\helpers.py", line 363, in __init__

KeyboardInterrupt

C:\Users\Jos>python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.3.py
rebuilding map. Iteration: 305
iterations: 305
```

before enhancements

v1.3: Limit: 312 rings

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

Figure 1



Opdrachtprompt - python C:\Users\Jos\GitHub\UrbanPlanning\Python\Jos\Jos_Algorithm_v1.3.py

```
water left: 1152.0
0.2
Placed water.
water left: 0.0
Done!

 Expanding rings...

 rings added:  [47]

 rings added:  [47, 48, 49, 50]

 rings added:  []

 rings added:  [18, 19]

 rings added:  []

 rings added:  [18, 19, 20, 21]

 rings added:  [18]

 rings added:  [18, 19, 20, 21, 22, 23]

 rings added:  []

 rings added:  [8, 9, 10, 11, 12, 13, 14]

 rings added:  []

 rings added:  [8, 9, 10, 11]

 rings added:  [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]

 rings added:  []

 rings added:  [8, 9, 10, 11, 12, 13]

 rings added:  [8]

 rings added:  []

 rings added:  [8, 9]

 rings added:  [8, 9, 10, 11]

 rings added:  [8]

Total map value: 15032280
```

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

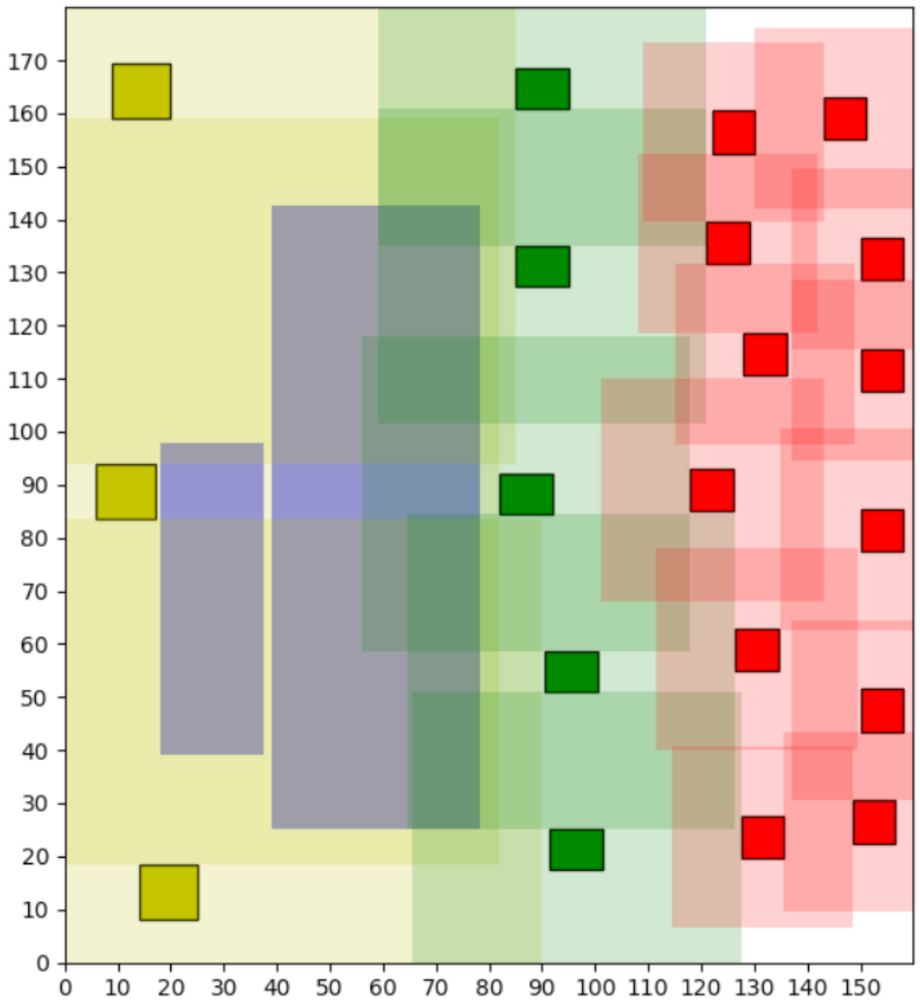
- if a house times out,
  - do similar things as 1.3.1

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

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:  []
 rings added:  []
 rings added:  []
 rings added:  []
 rings added:  []
 rings added:  []
 rings added:  []
 rings added:  []
 rings added:  []
 rings added:  []
 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]
```

Score: 16.8 mil.

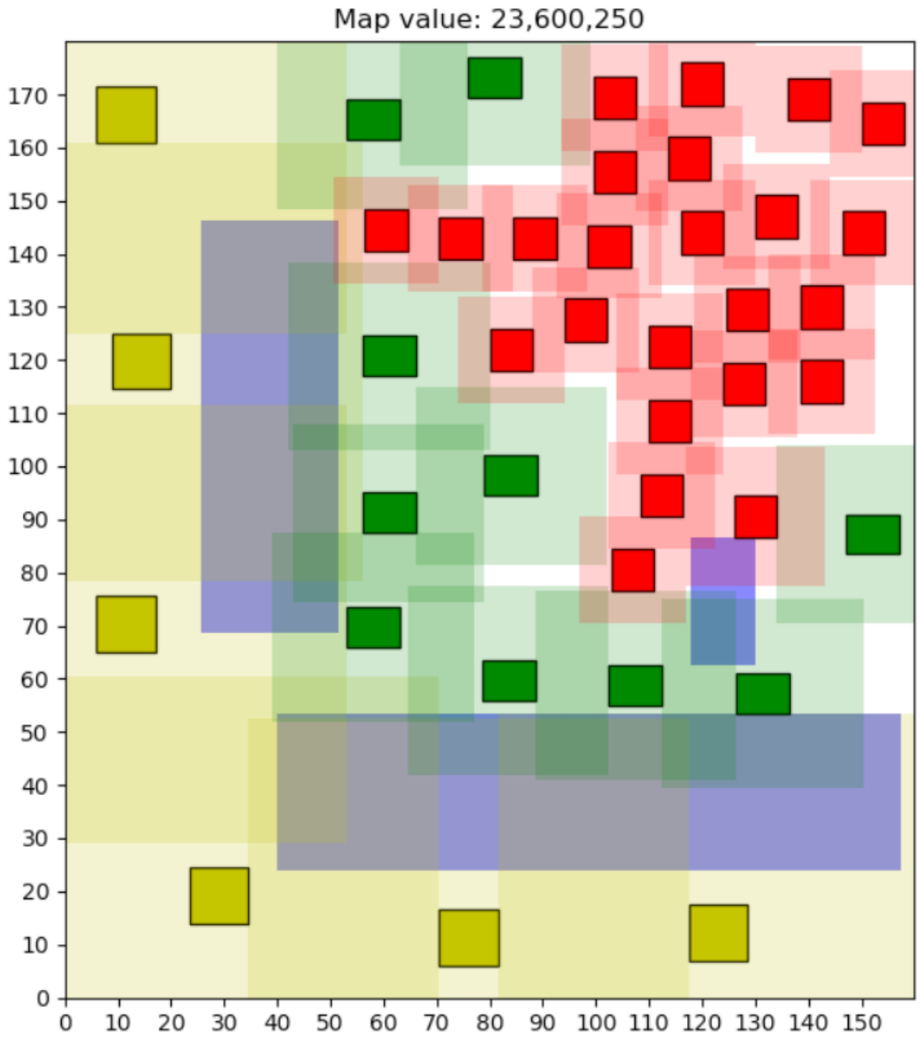
Limit: 420 rings

v1.3: 20 houses

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

Figure 1

HEURISTICS: Amstelhaege



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:  []
 rings added:  []
 rings added:  []
 rings added:  []
 rings added:  []
 rings added:  []
rings added:  []
rings added:  []
rings added:  []
rings added:  []
rings added:  []
rings added:  []
rings added:  []
rings added:  []
rings added:  []
rings added:  []
rings added:  [5, 6, 7]
rings added:  []
rings added:  []
rings added:  []
rings added:  []
```

v1.3: 40 houses

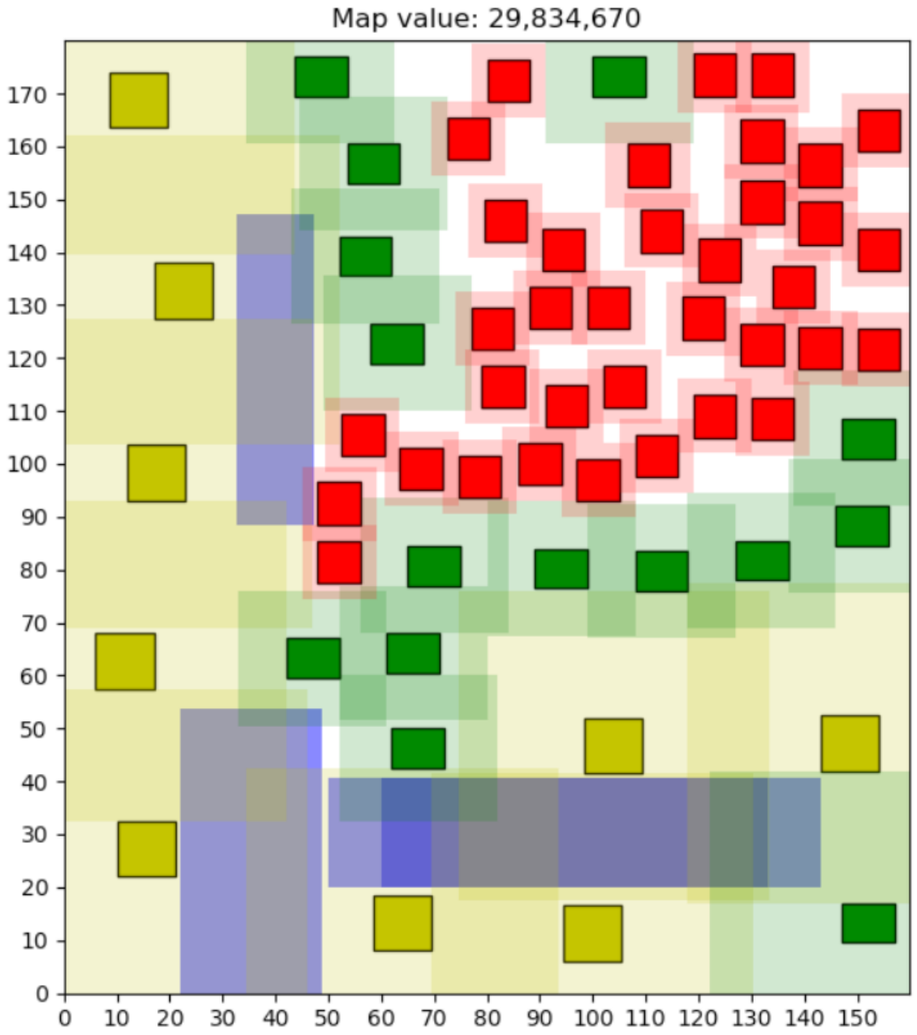
HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

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\Jos_Algorithm_v1.4.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\Jos_Algorithm_v1.4.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:  []
```

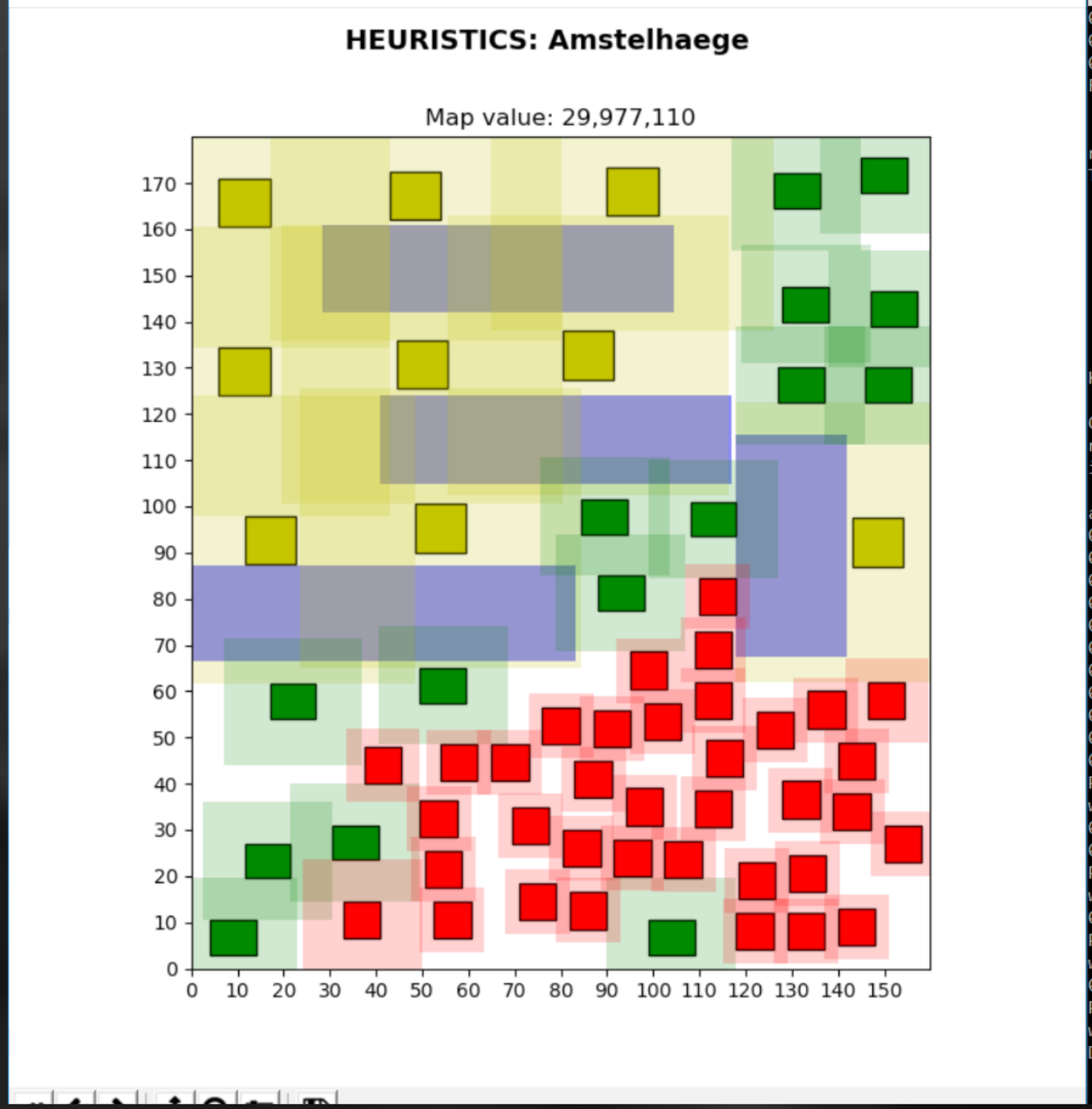
not the best but PATTERNS

v1.3: 60 houses

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

Limit: 290 rings

Figure 1



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...
```

Score: 29.9 mil.

Limit: 300 rings

v1.3: 60 houses

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

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

Implement Christiaan Algorithm into Jos Algorithm

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 !

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara

QUESTIONS?

HEURISTICS:
Amstelhaege
Christiaan | Jos | Tara