ICFP Programming Contest 2019 Report

Ilya Sergey

https://icfpcontest2019.github.io

Yale NUS College

NUS National University of Singapore

zilliqa

ICFP Programming Contests

- Annual contest, running since **1998**
- Goes for **72 hours** (long weekend), usually with a **24 hour** lightning division
- Modest cash prizes but unlimited bragging rights
  - *Language X is the programming tool of choice for discriminating hackers*
  - *Language X is a fine programming tool for many applications*
  - *Language X is very suitable for rapid prototyping*
  - *Team Z are an extremely cool bunch of hackers*
- Each year the contest is organised by a different group of FP hackers

Recent History

- 2012: Mining lambdas from caves in Scotland
- 2013: Program synthesis
- 2014: Programming ~~Pac~~ LambdaMan
- 2015: Hexagonal Tetris with secret words
- 2016: Origami folding
- 2017: Lambda-punting
- 2018: 3D constructions (and destructions) using nano-bots

ICFP 2017

Hey, would you like to jump at the one-in-a lifetime opportunity to run the ICFP Programming Contest in 2019?

Sure, sounds like a lot of fun!

ICFP 2019 General Chair

ICFP 2017

Hey, would you like to jump at the one-in-a lifetime opportunity to run the ICFP Programming Contest in 2019?

ICFP 2018

Yay, I’ll be running the ICFP Programming Contest next year!

WHEN A FELLOW RESEARCHER HEARS
THAT YOU'LL BE ORGANISING ICFP CONTEST

ICFP Programming Contest 2019

By 2019 Functional Programming has taken over the World

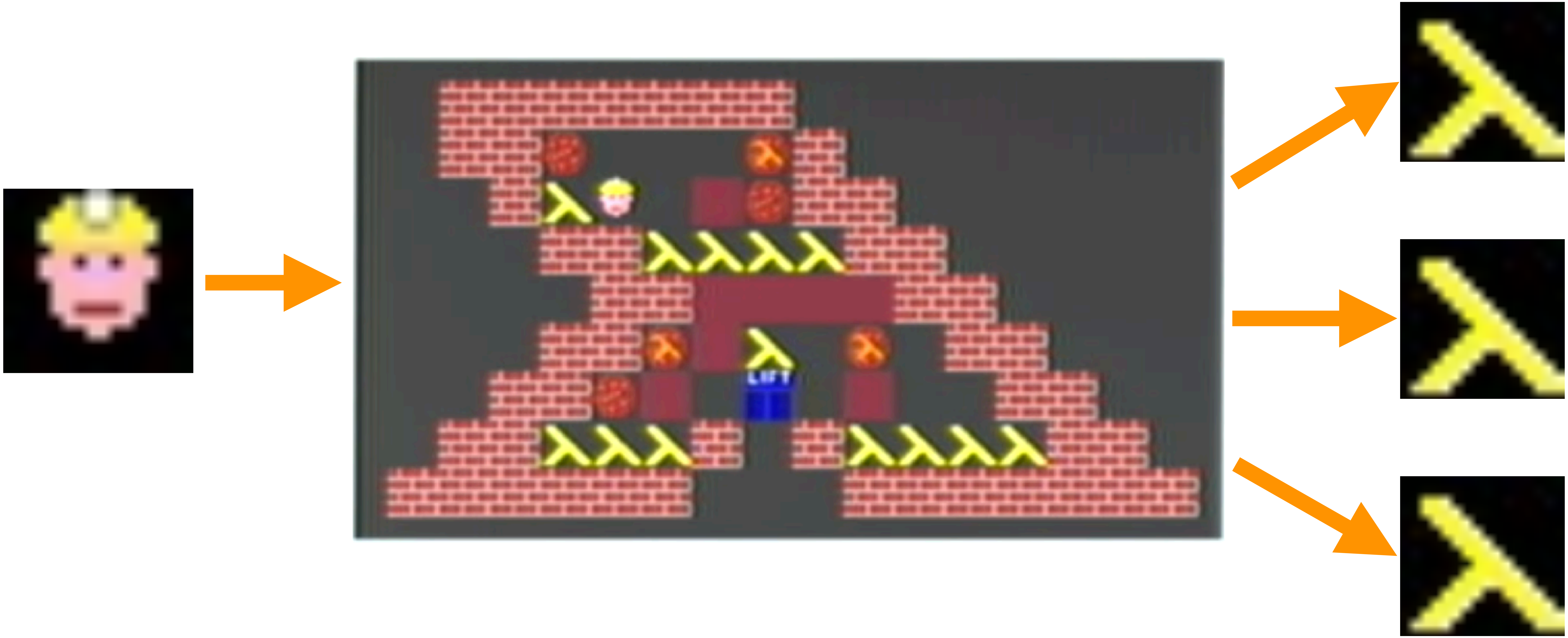
By 2019 Functional Programming has taken over the World

- 2012: Mining lambdas from caves in Scotland
- 2013: Program synthesis
- 2014: Programming ~~Pac~~ LambdaMan
- 2015: Hexagonal Tetris with secret words
- 2016: Origami folding
- 2017: Lambda-punting
- 2018: 3D constructions (and destructions) using nano-bots

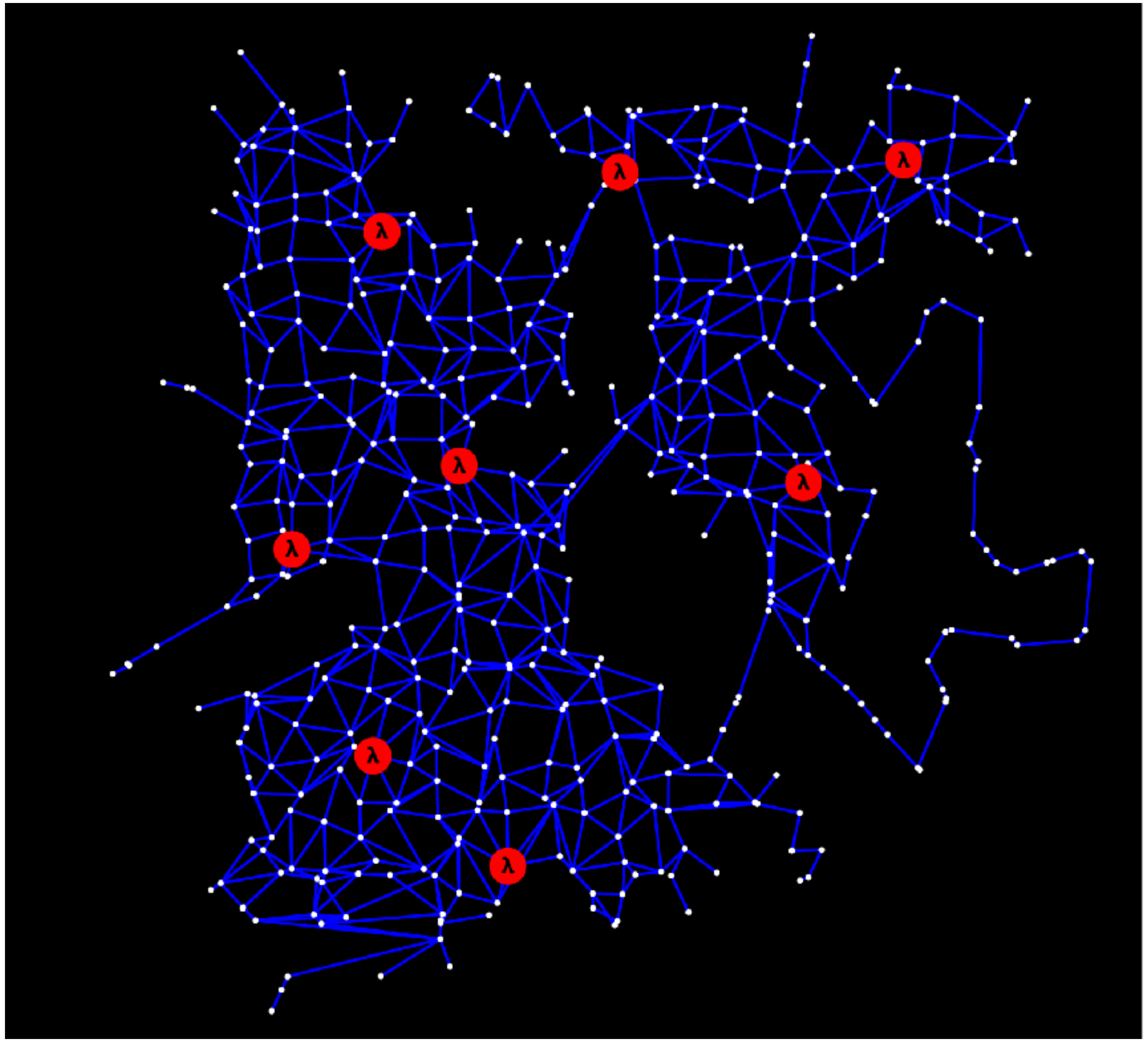
By 2019 Functional Programming has taken over the World

- 2012: Mining lambdas from caves in Scotland
- 2013: Program synthesis
- 2014: Programming ~~Pac~~ LambdaMan
- 2015: Hexagonal Tetris with secret words
- 2016: Origami folding
- 2017: Lambda-punting
- 2018: 3D constructions (and destructions) using nano-bots

2012: Mining lambdas from caves in Scotland



2017: Lambda-Punting

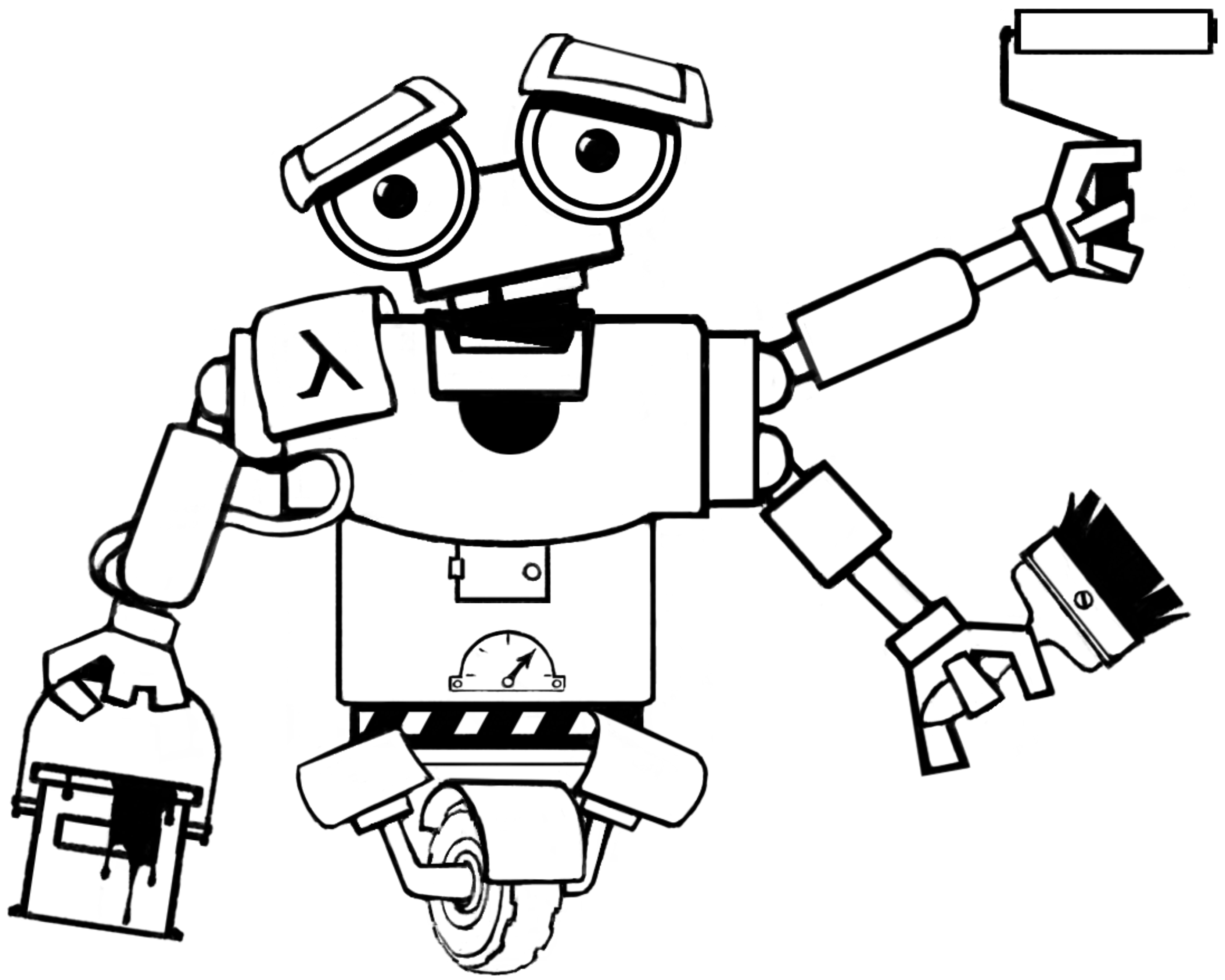


2019: Bit Rotting Problem

- What shall we do with all the legacy code?
- We need to dispose the bit-rotten software…
- Let us use empty mines as waste silos!

Preventing Bit-Rot from Spreading

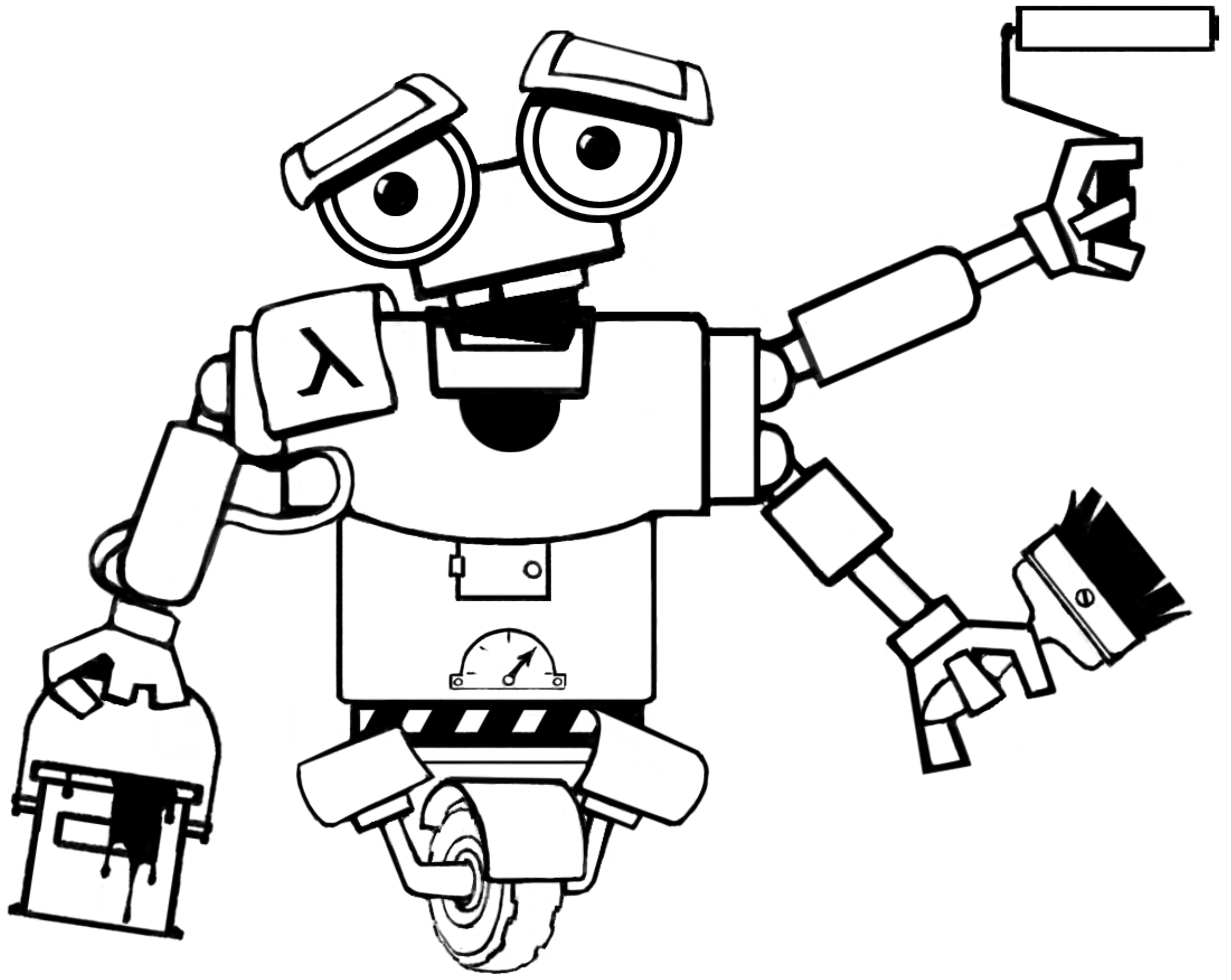
Wrapping the surface of mines in a decay-containing substance.



Worker-Wrapper 2019
(*aka* Wrappy)

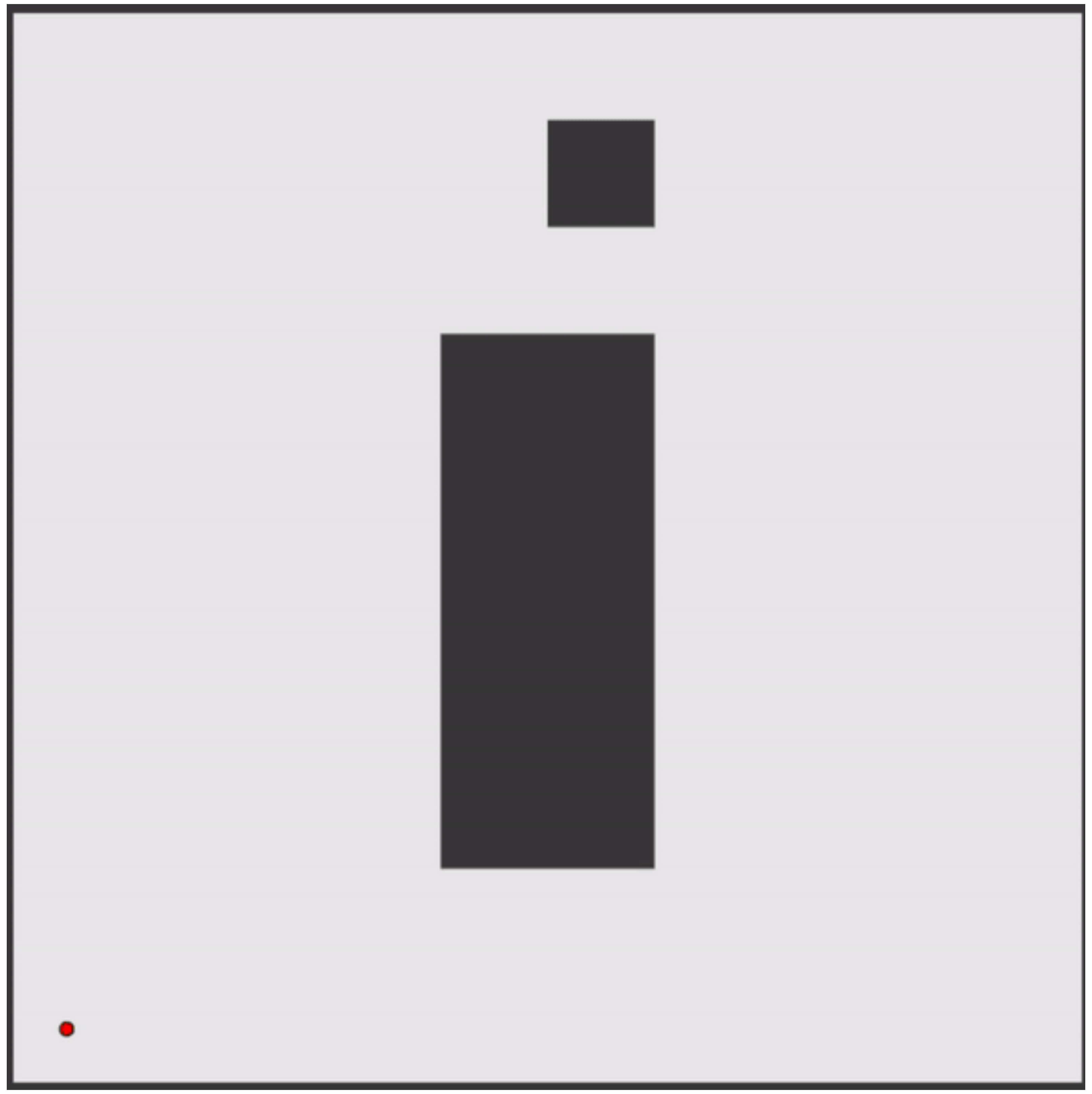
The Task

For a given map, give a Wrappy trace to cover the entire surface, while minimising a number of steps.



The Rules

- Wrappy has three manipulators
- It can move in *four directions* (but not through the walls).
- Wrappy can turn around
- Manipulators can *fold* an *unfold* in narrow parts of a map.



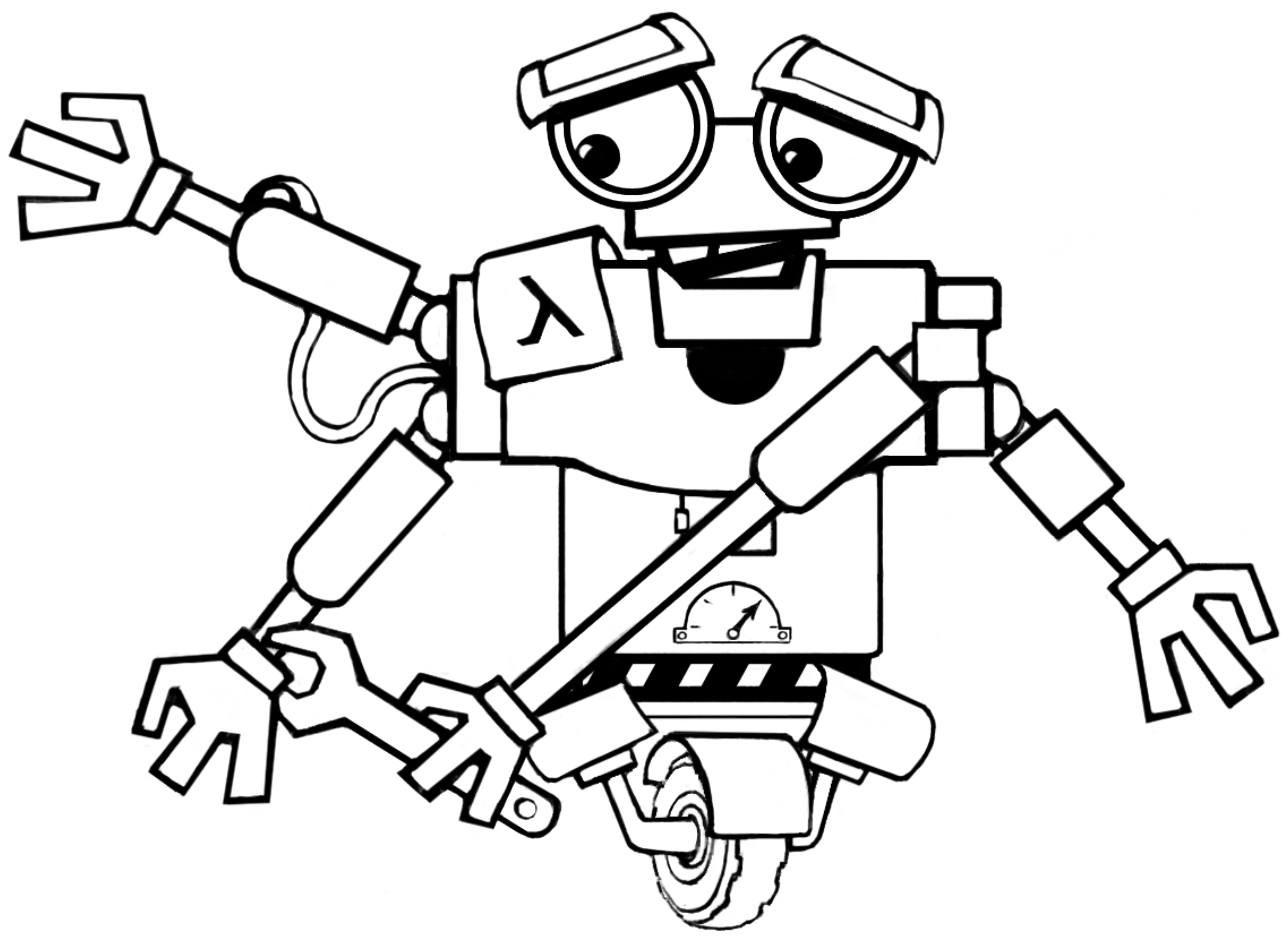
DDDDDDDQWWWWWWWWEDDESSSSSSSAAAAAAEEWWWWWWDDAWWDASQAAAQSSSSSSS

Sounds too Easy?

Lambda-miners left ***some stuff*** behind…

Let’s see if we can use it

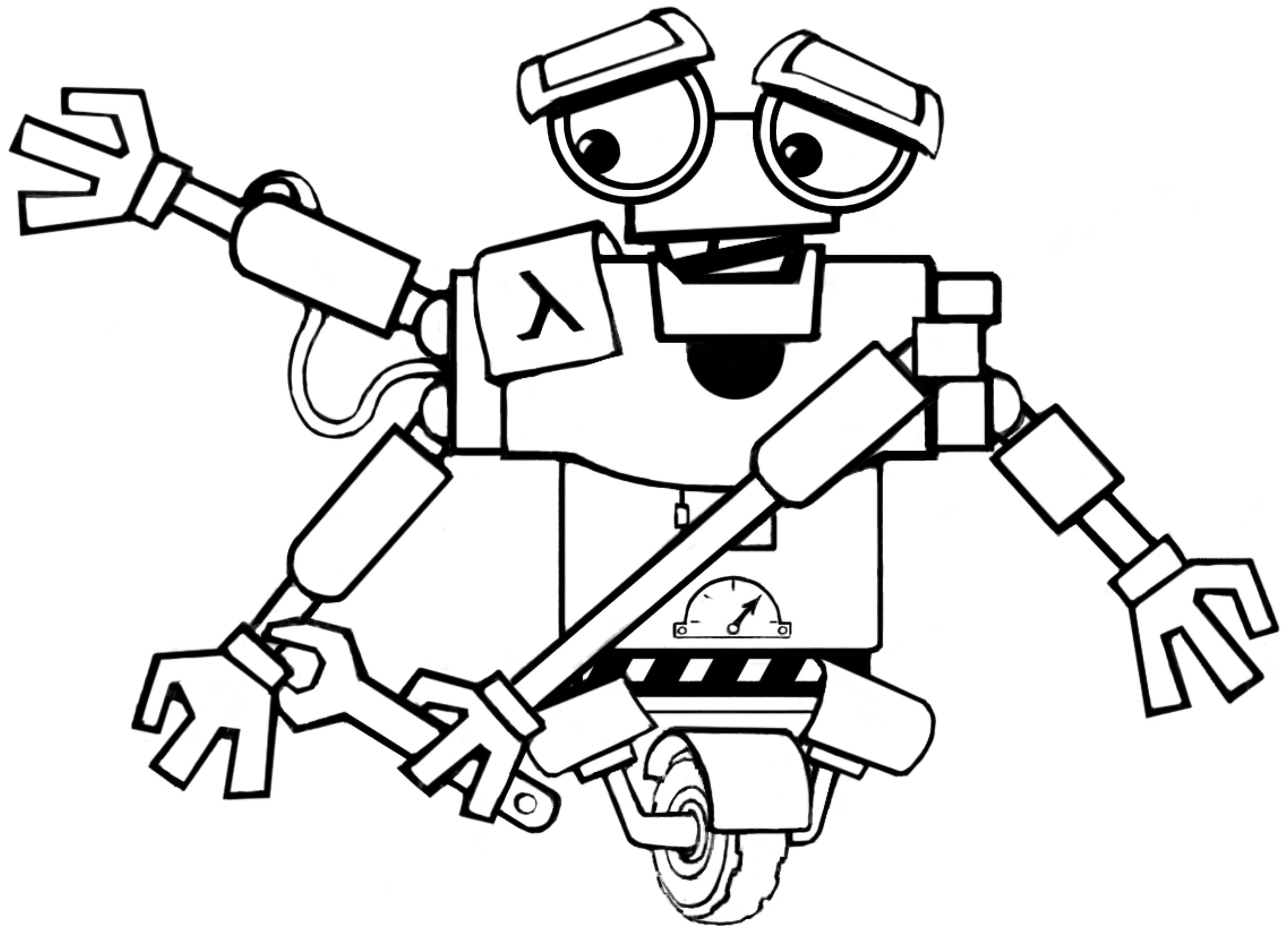
Manipulator Extension

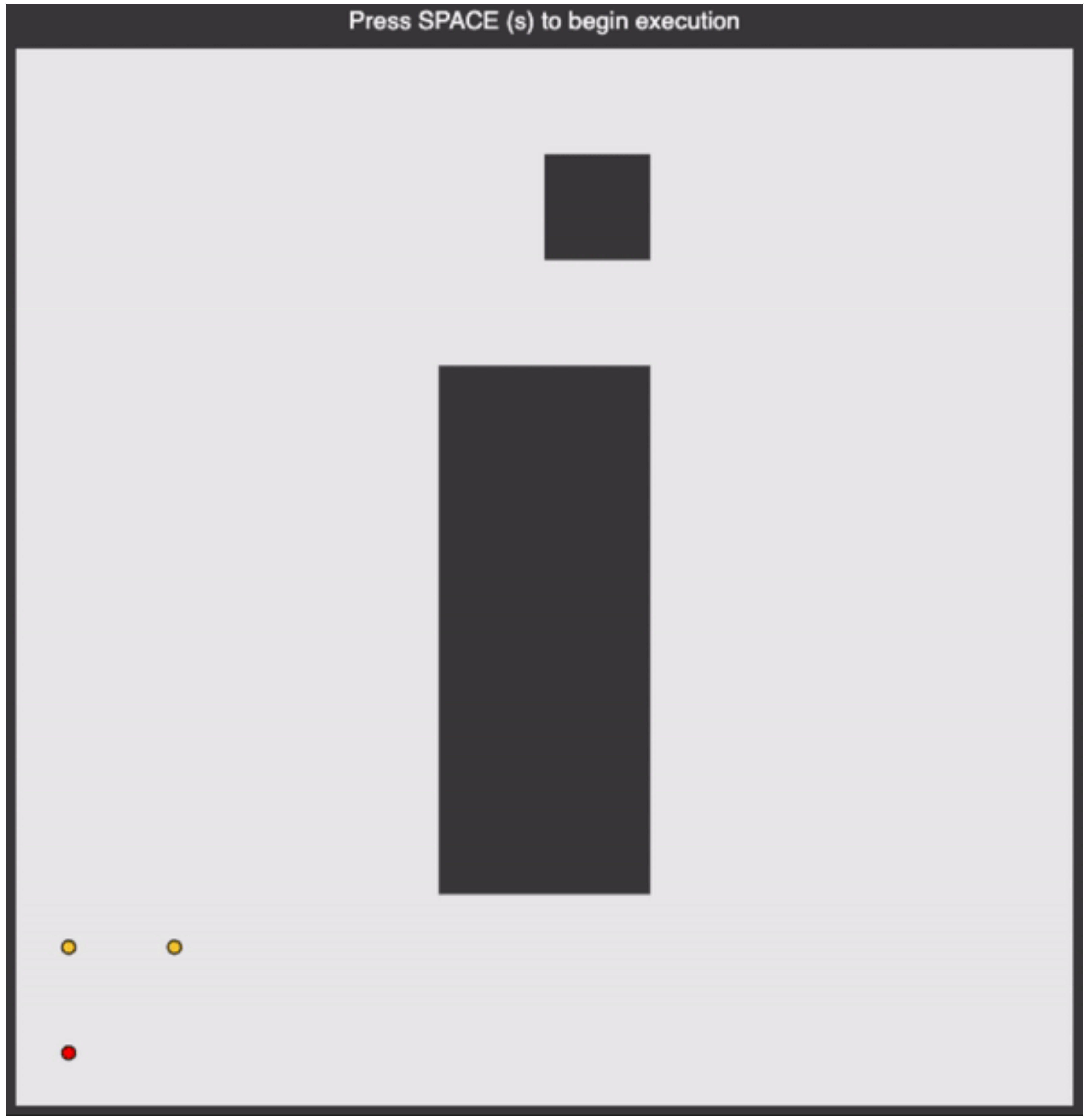


- Wrappy attach an *an additional robotic* hand, extending its range

- Permanent effect

Manipulator Extension



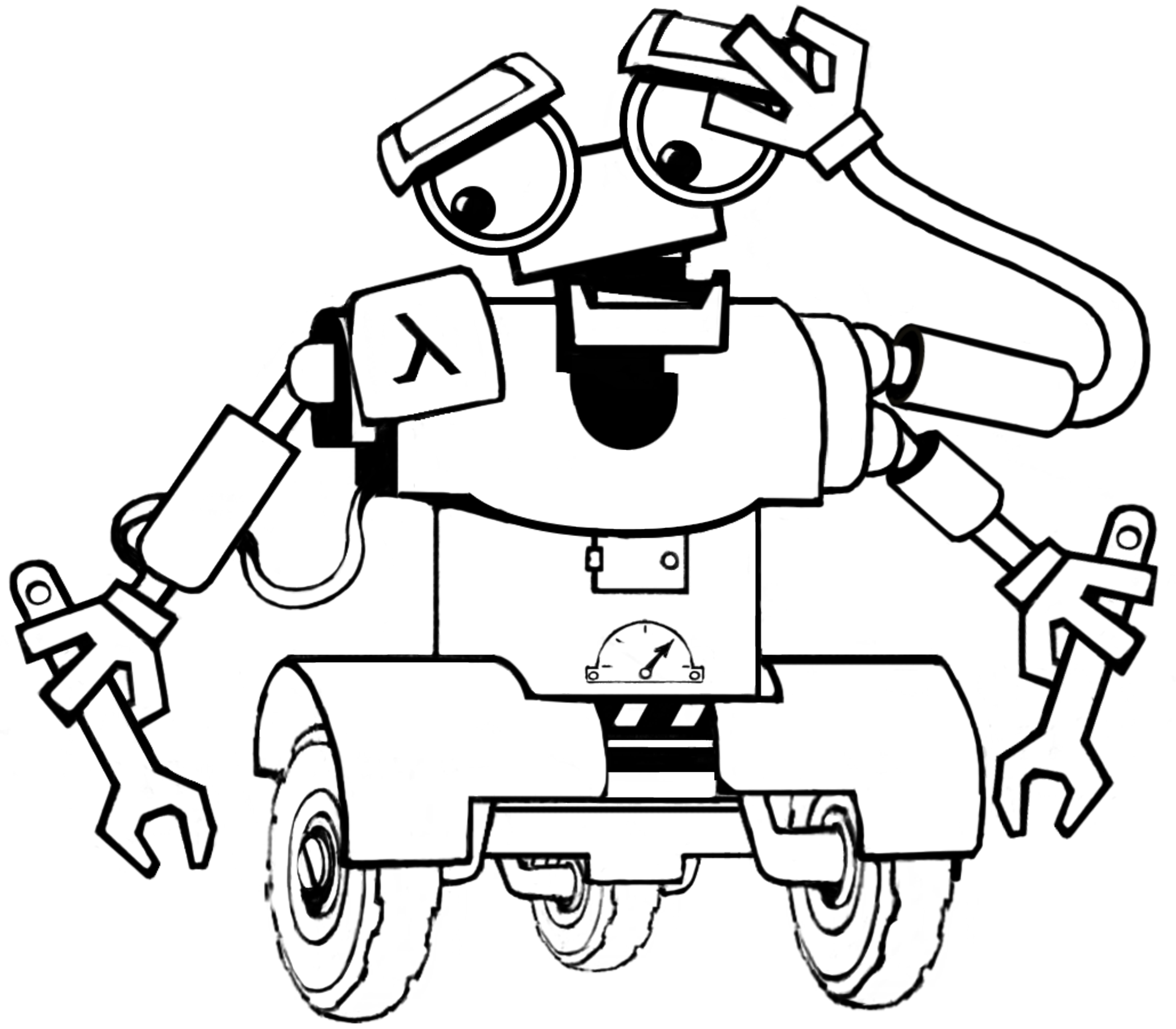


Press SPACE (s) to begin execution

- Wrappy attach an *an additional robotic* hand, extending its range

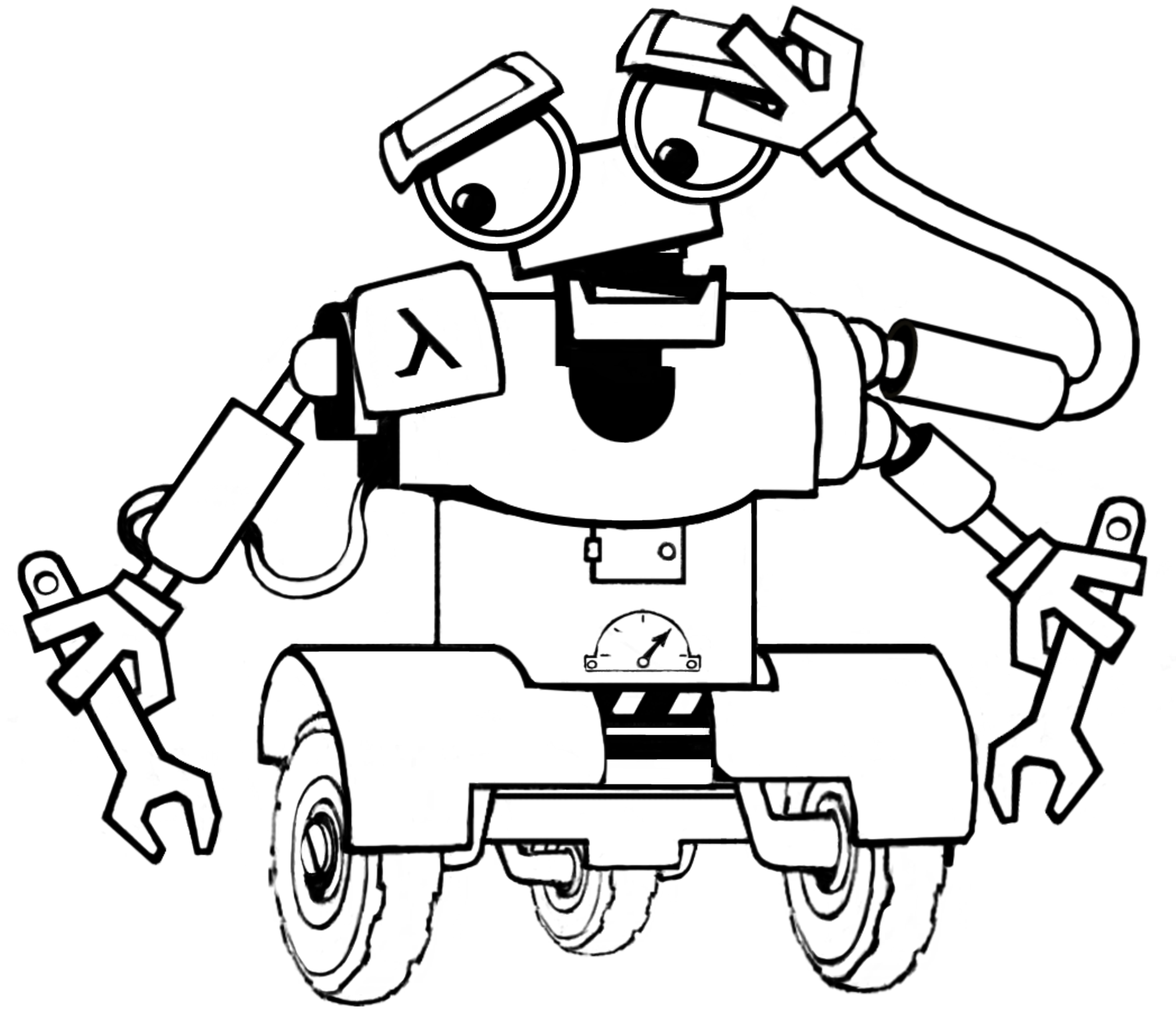
- Permanent effect

Fast Wheels

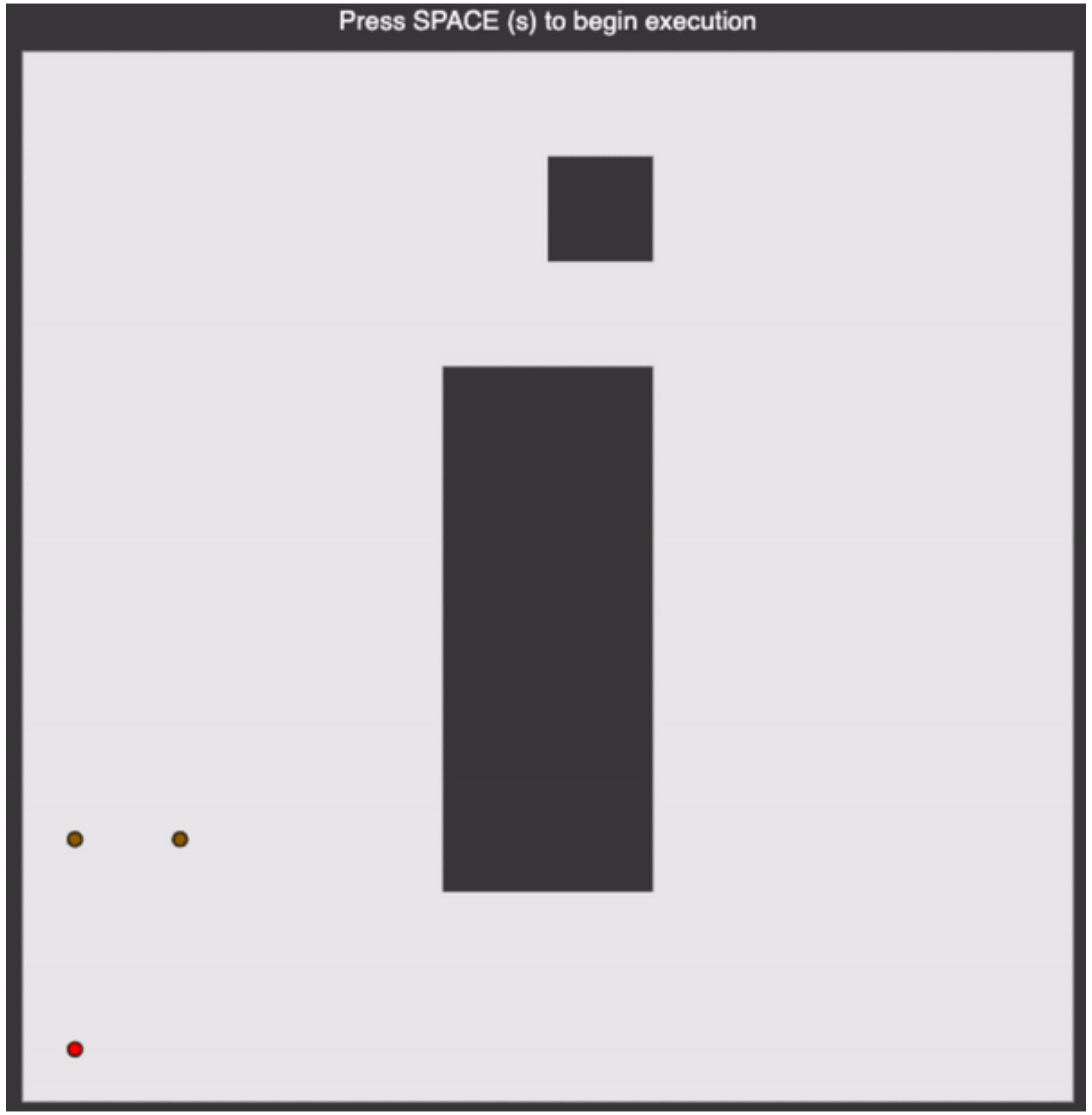


- Wrappy moves *twice as fast* when stepping in either direction
- Effect lasts for 50 time units

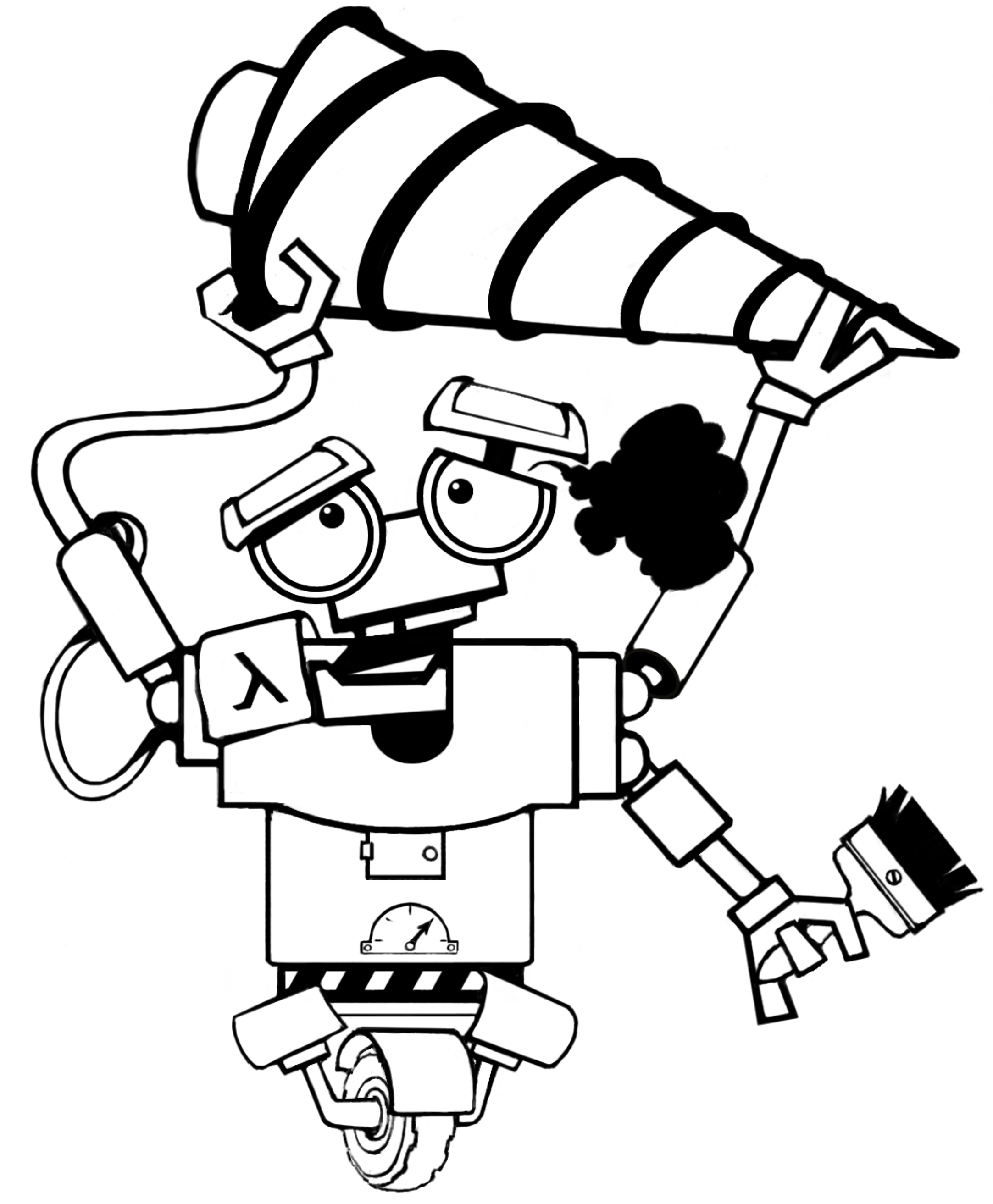
Fast Wheels



- Wrappy moves *twice as fast* when stepping in either direction
- Effect lasts for 50 time units



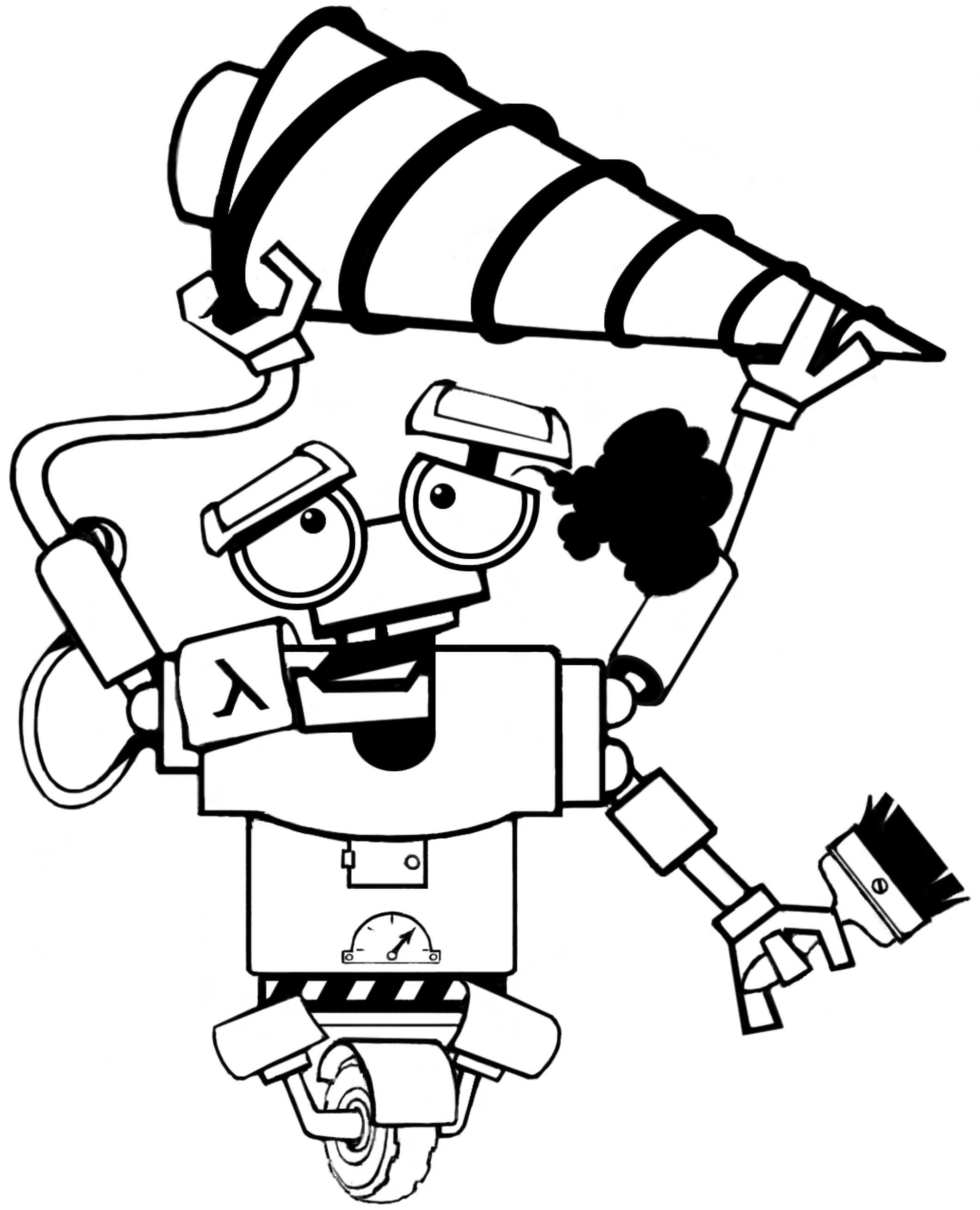
Drill



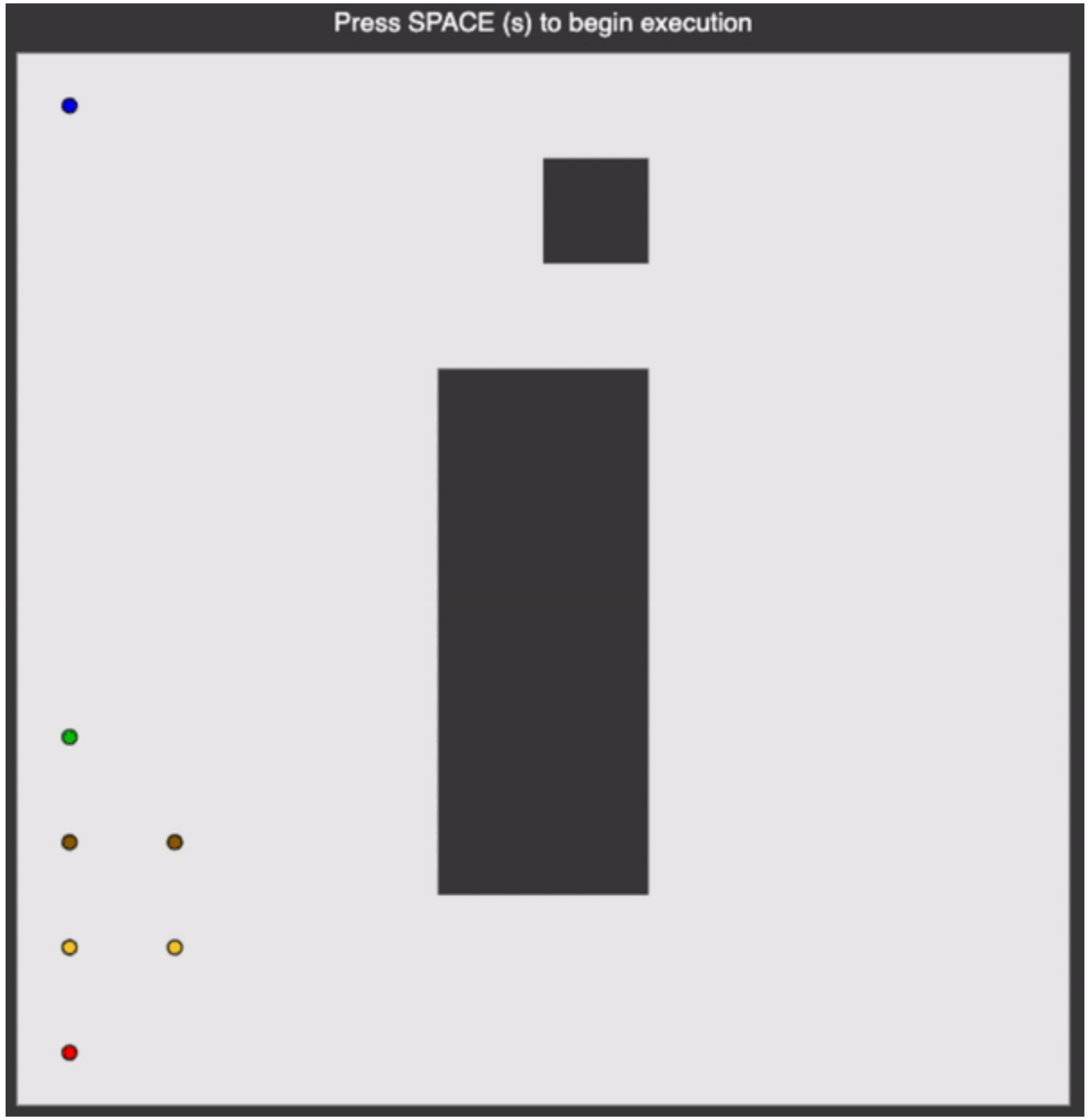
- Wrappy can *make tunnels* in walls and obstacles

- Effect lasts for 30 time units

Drill

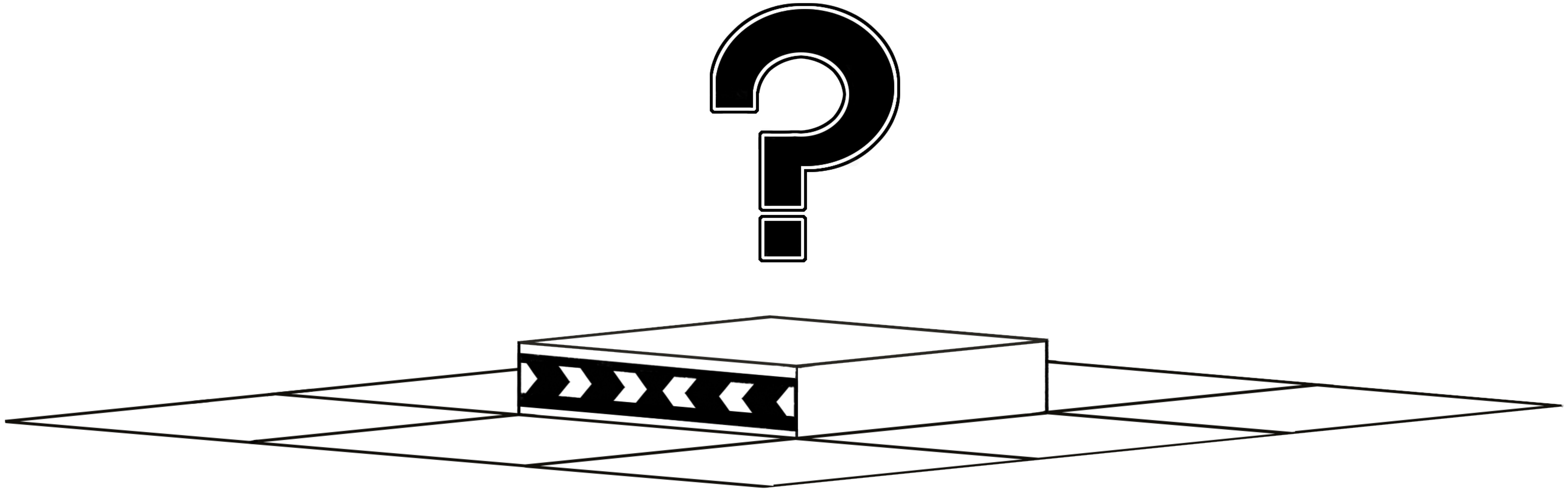


- Wrappy can *make tunnels* in walls and obstacles
- Effect lasts for 30 time units



Press SPACE (s) to begin execution

Mysterious Points



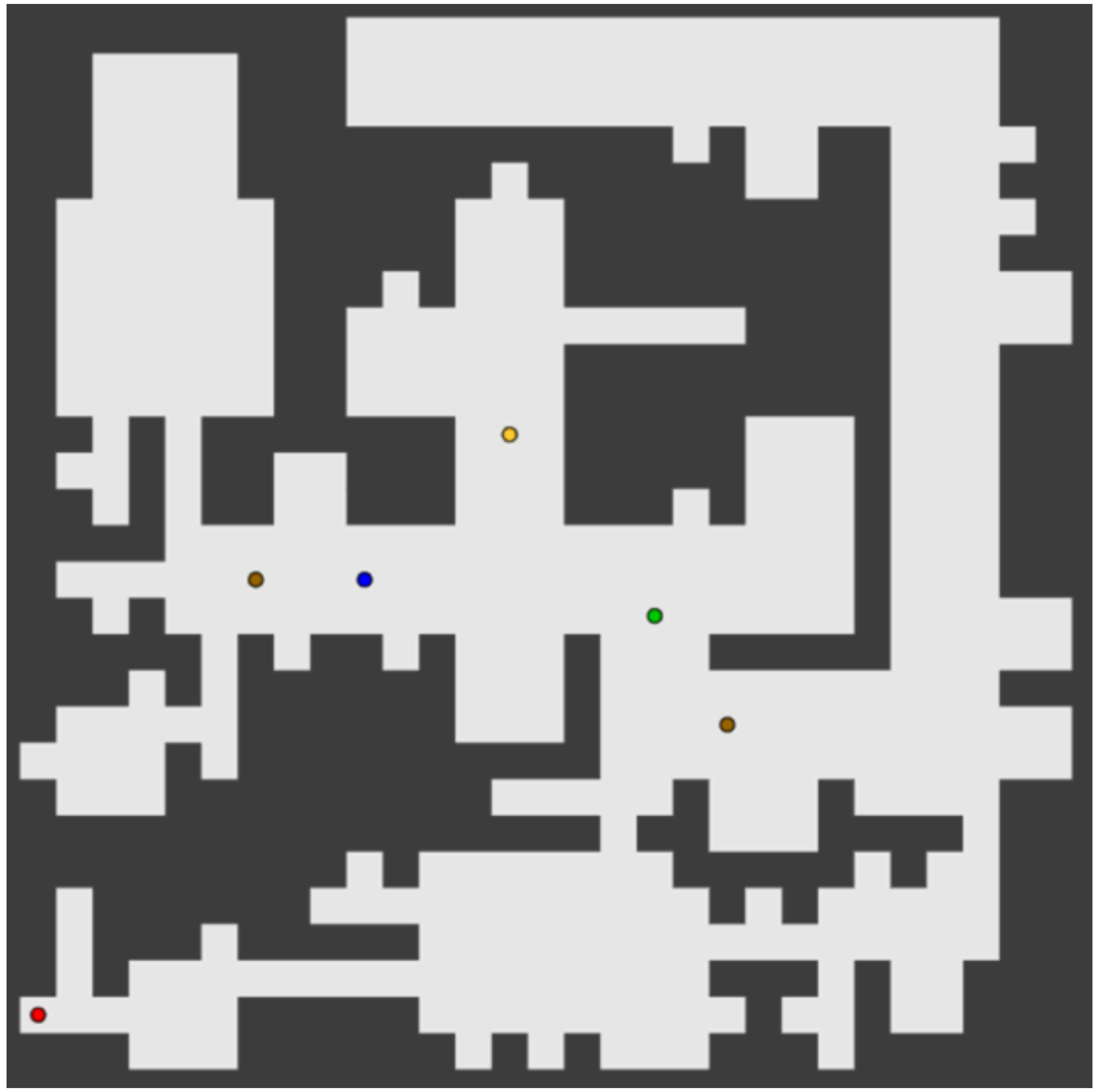
- The purpose is unknown (so far)

Initial Task

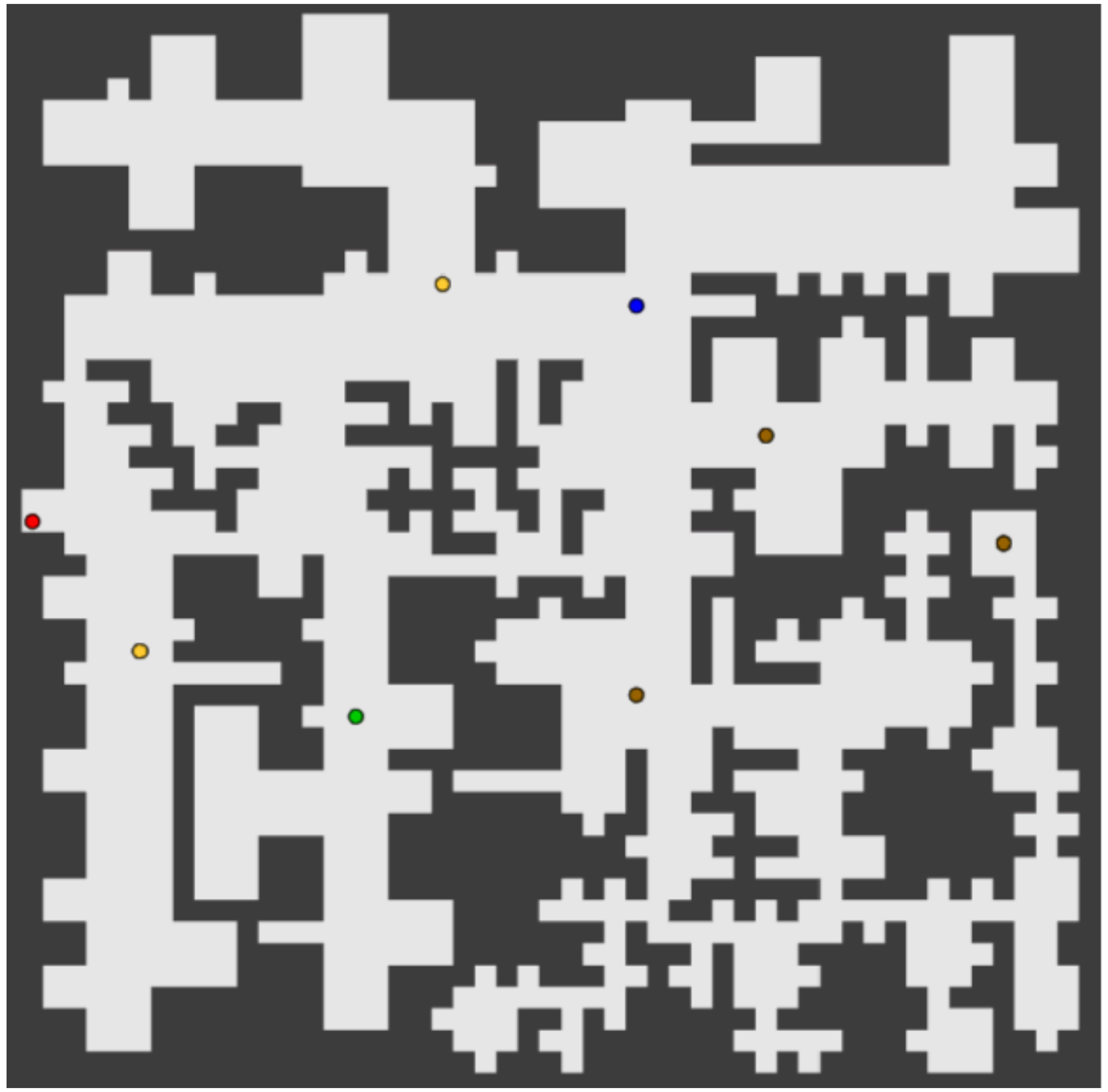
- 150 maps with boosters of different kinds
- Solution is a zip-file with Wrappy traces (text files)
- The shorter the traces, the better

$$score_{team,T} \triangleq \left\lceil 1000 \times \log_2 (X_T \times Y_T) \times \frac{t_{best}}{t_{team}} \right\rceil$$

The Maps

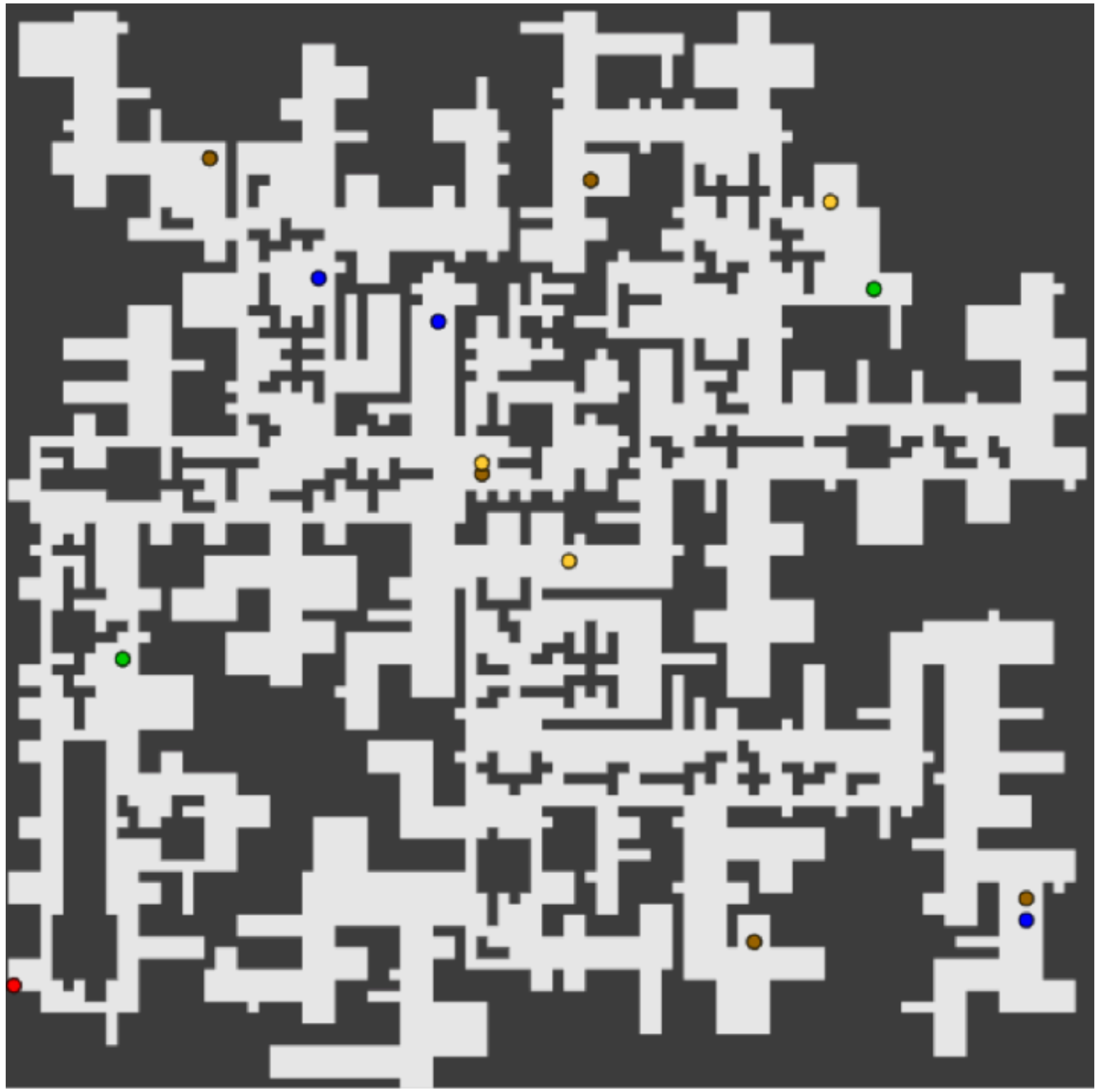


30 x 30

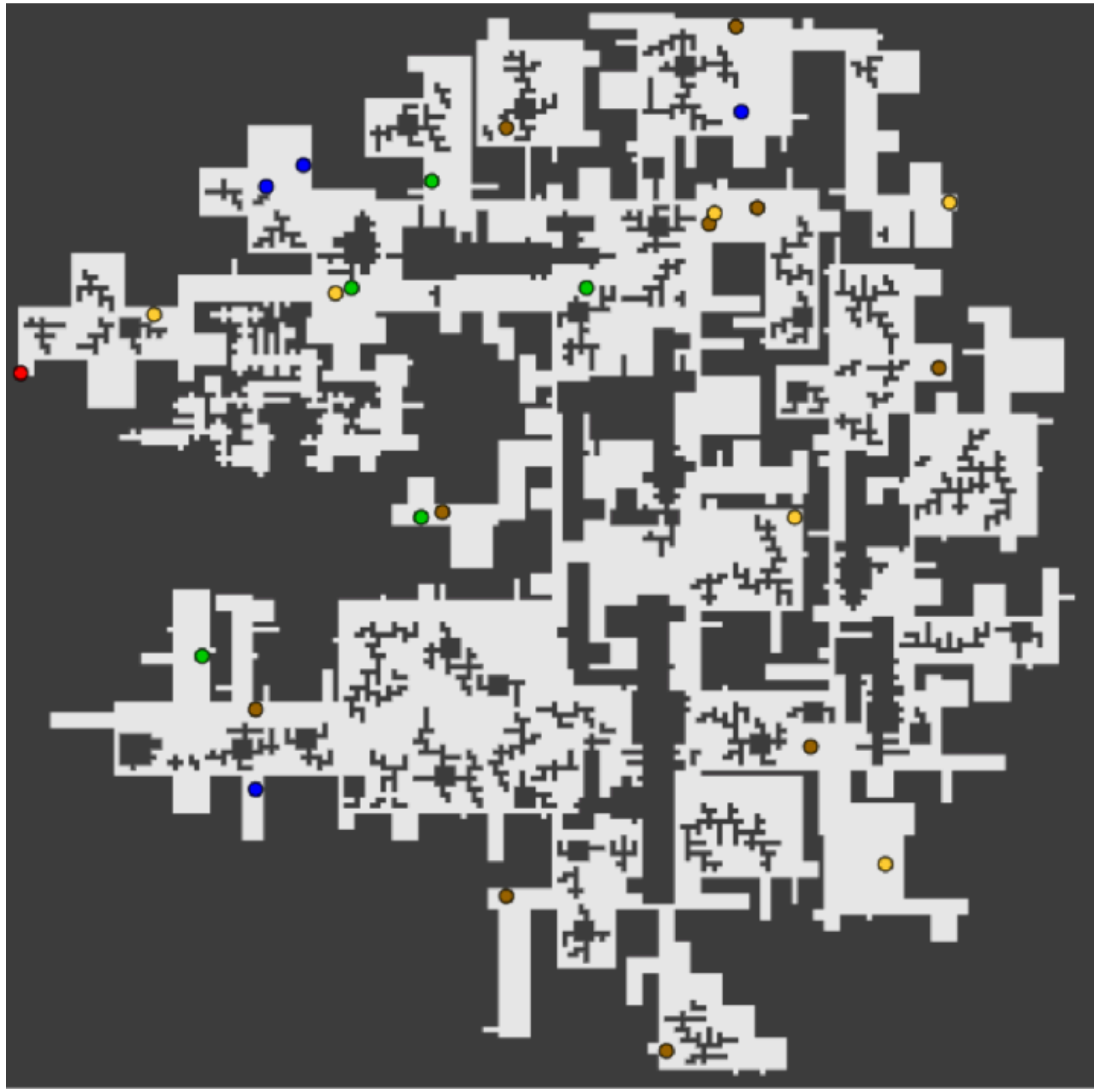


50 x 50

The Maps

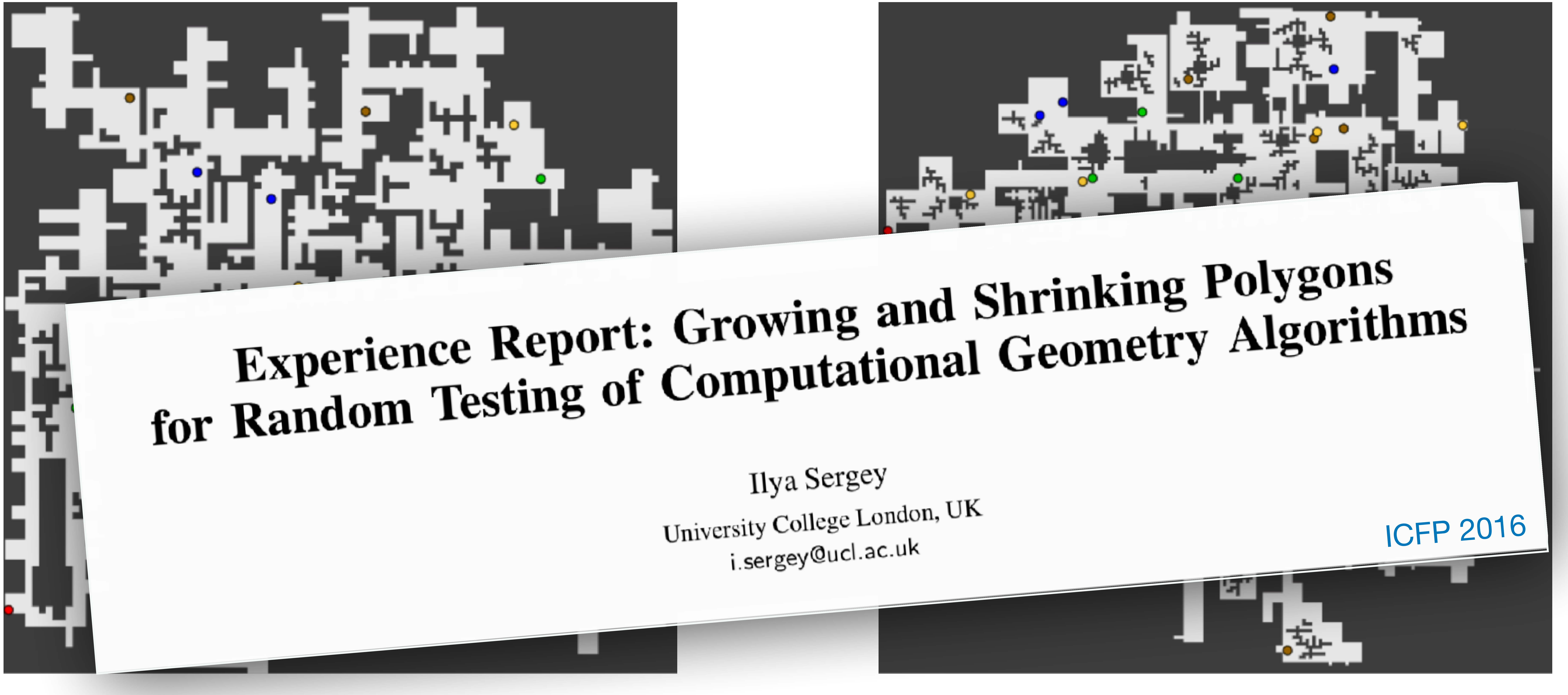


100 x 100



200 x 200

The Maps



Experience Report: Growing and Shrinking Polygons for Random Testing of Computational Geometry Algorithms

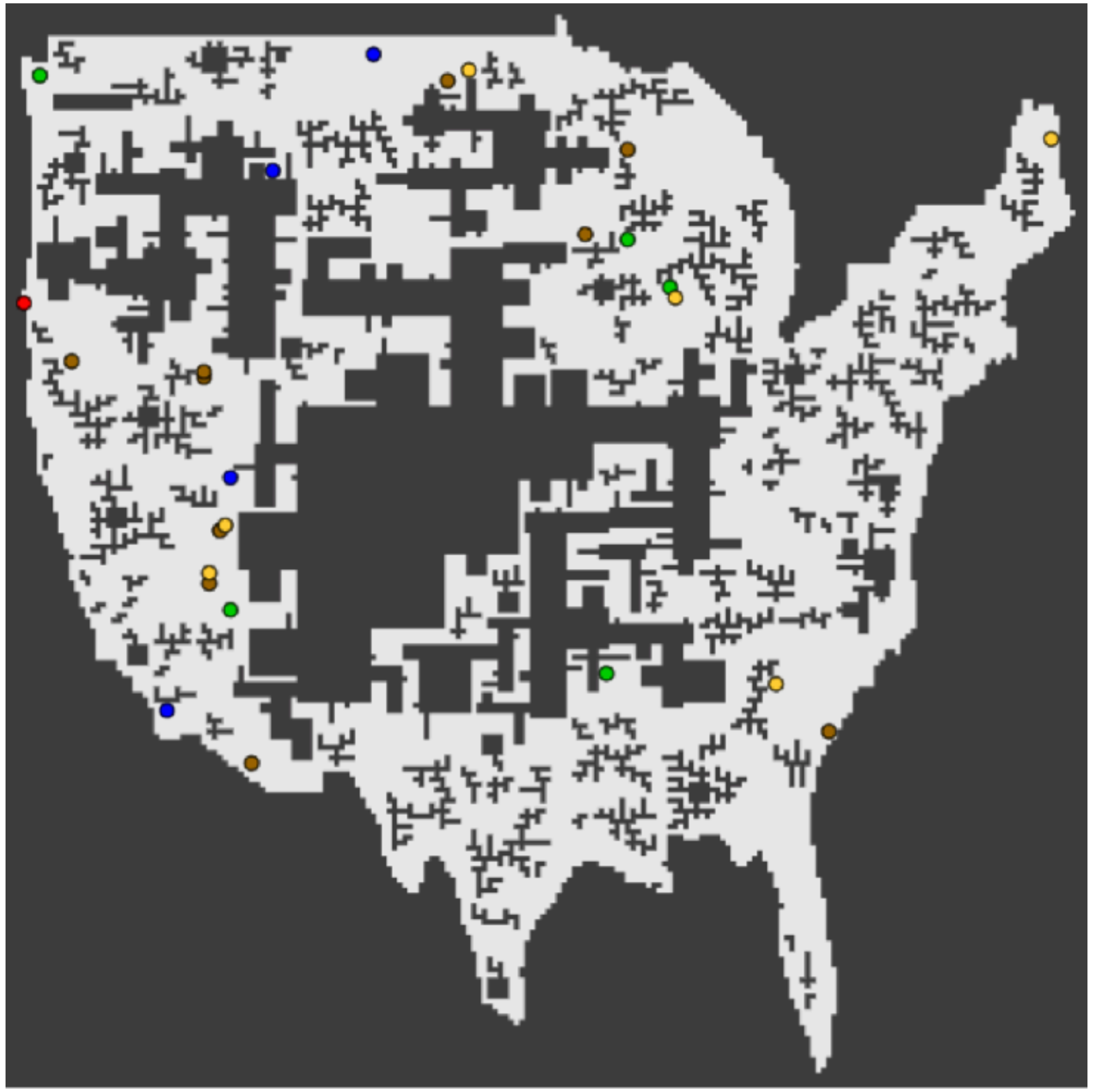
Ilya Sergey
University College London, UK
i.sergey@ucl.ac.uk

ICFP 2016

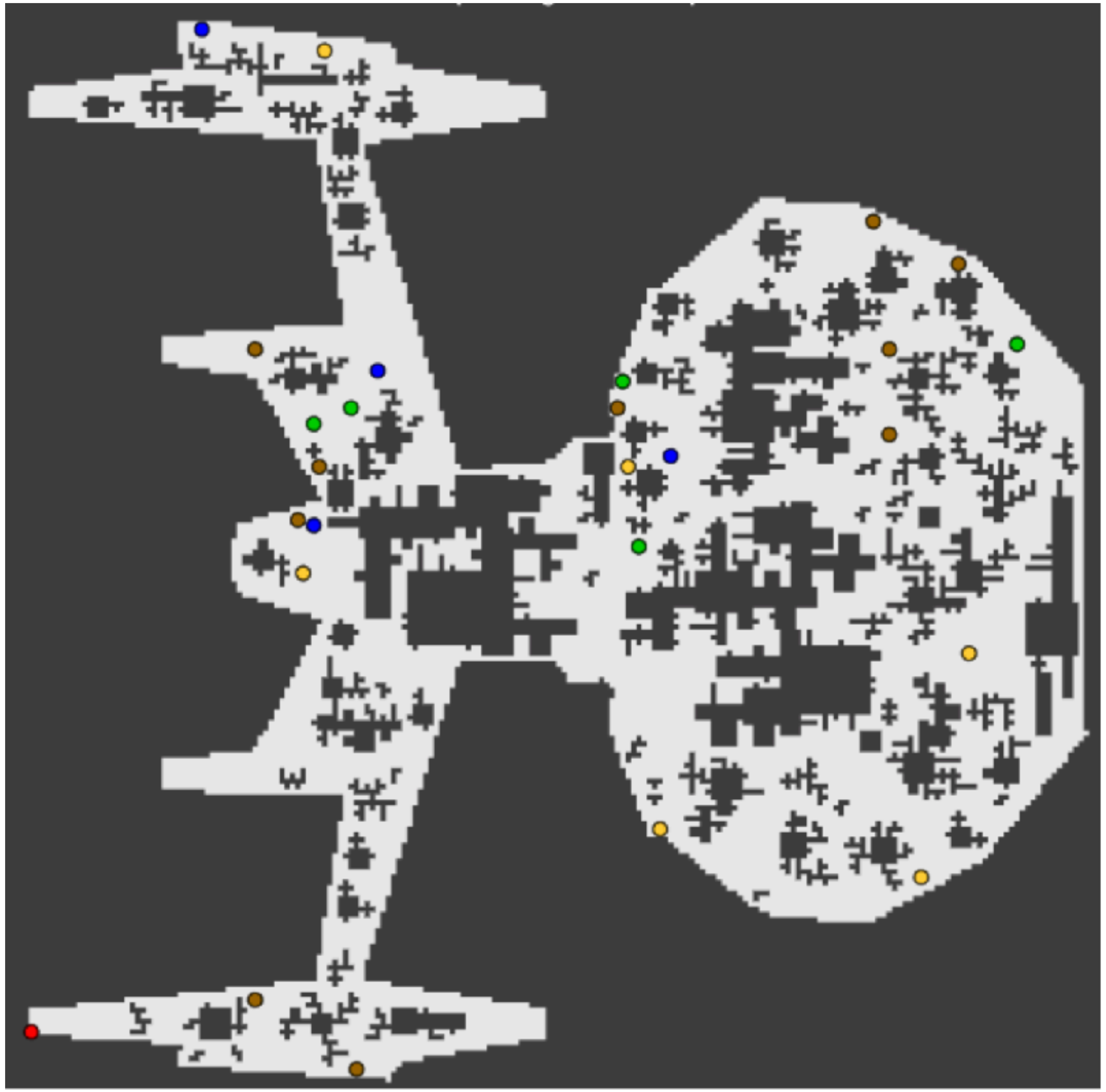
100 x 100

200 x 200

More Maps



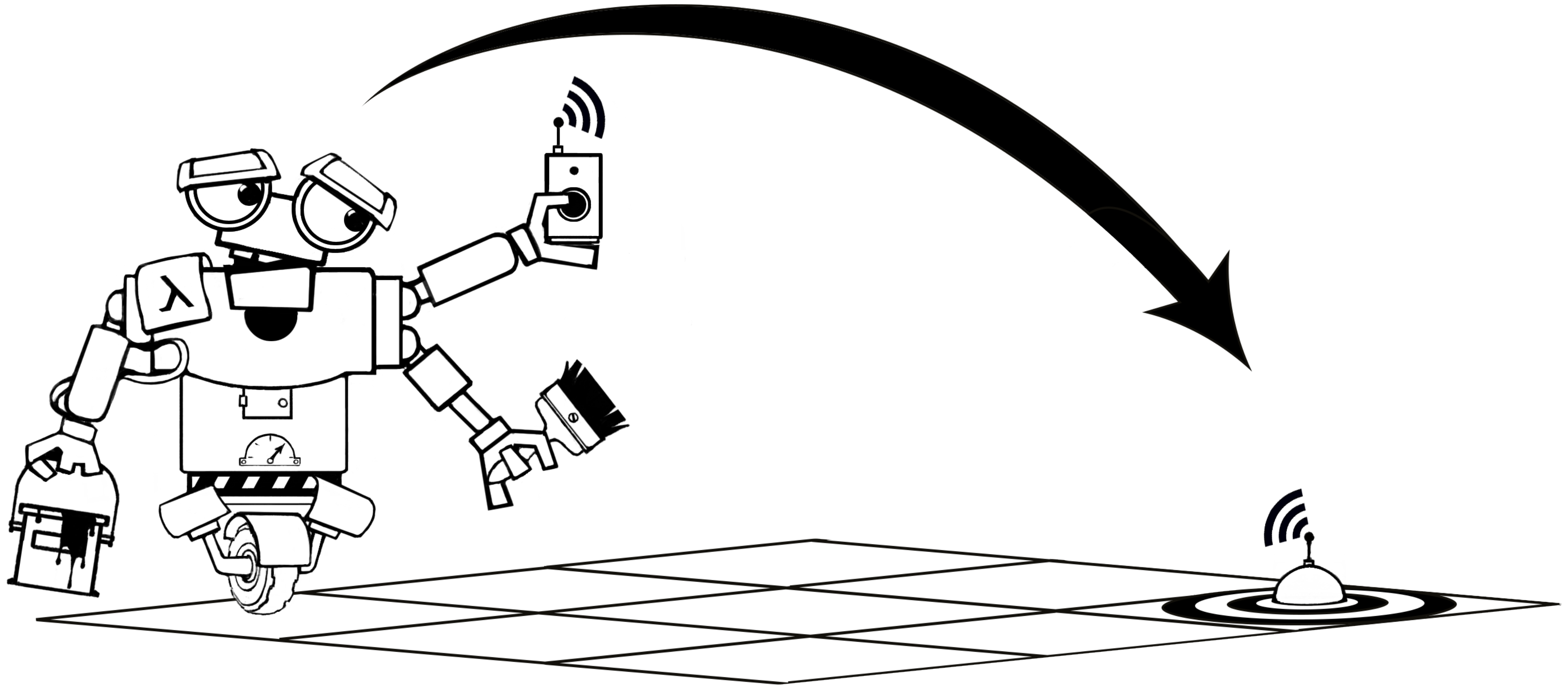
More Maps



Extensions

Teleports

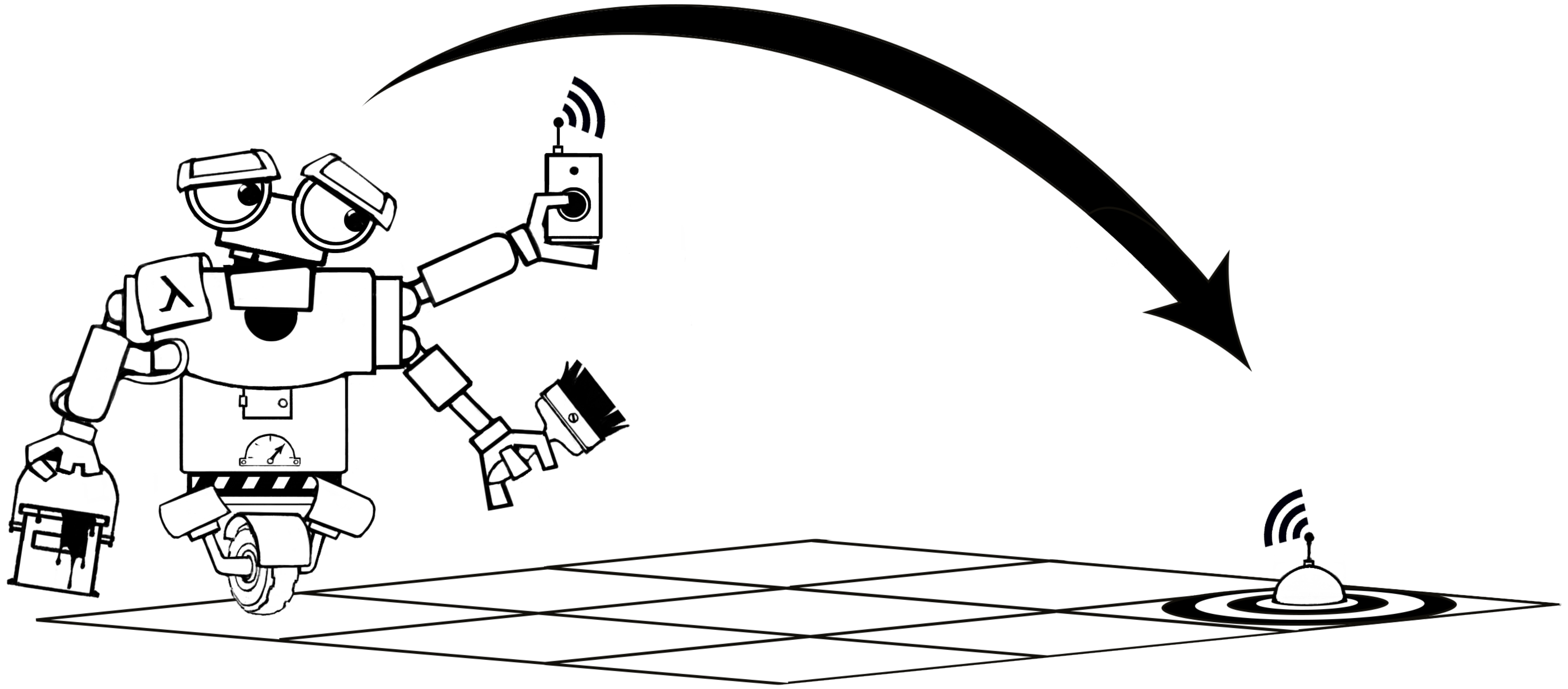
7 hours into the contest

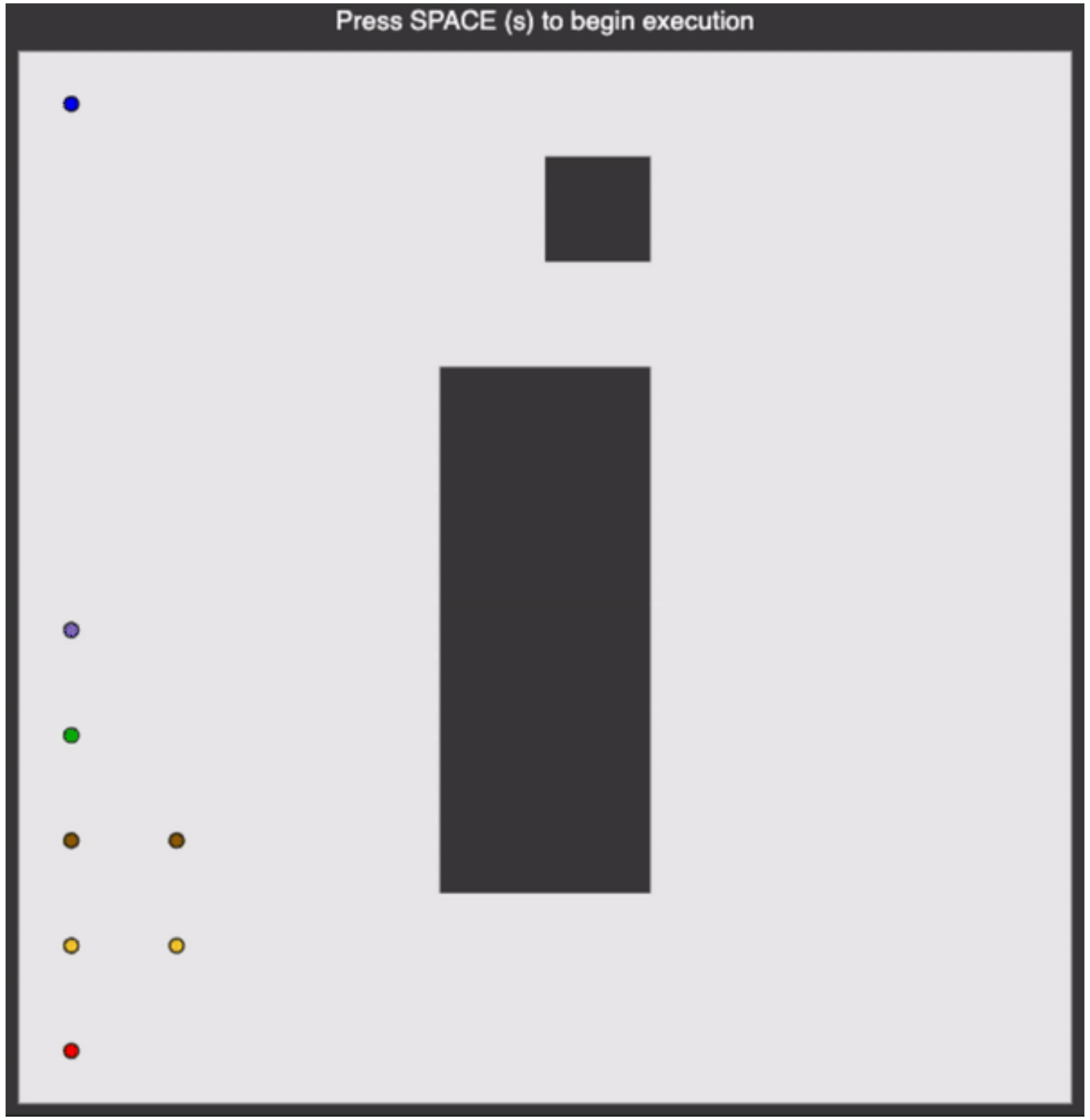


- Wrappy can set up a beacon at any non-wall location on a map
- It can *teleport to the beacon* from any point
- Can do as many teleportations as necessary

Teleports

7 hours into the contest



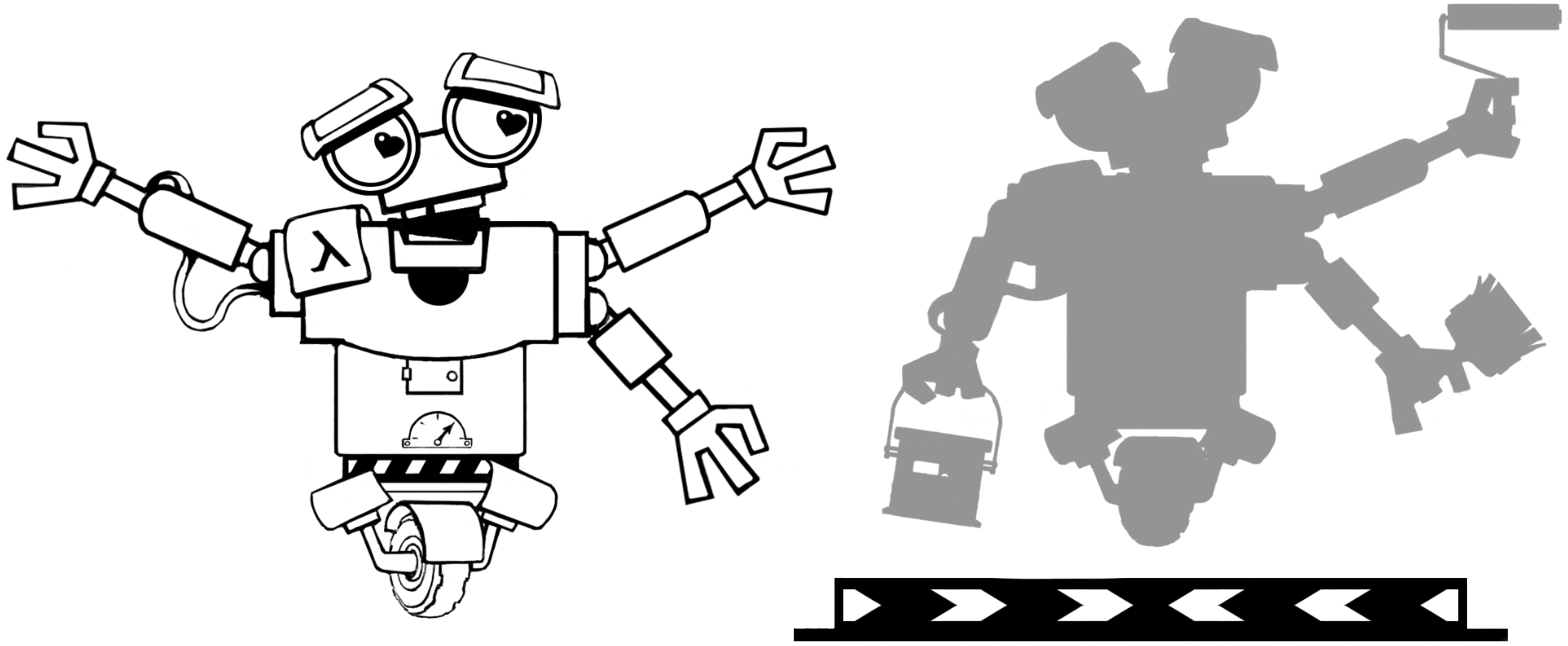


Press SPACE (s) to begin execution

- Wrappy can set up a beacon at any non-wall location on a map
- It can *teleport to the beacon* from any point
- Can do as many teleportations as necessary

Clones

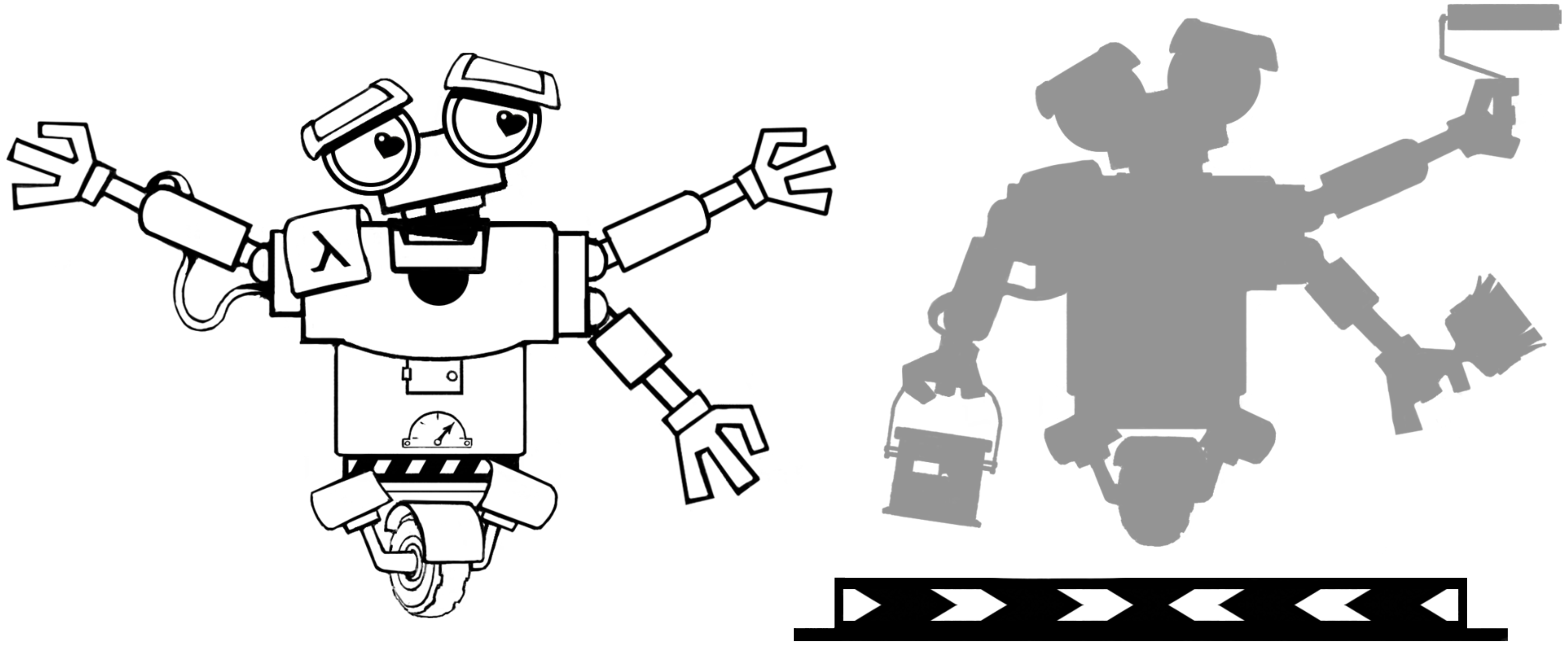
14 hours into the contest

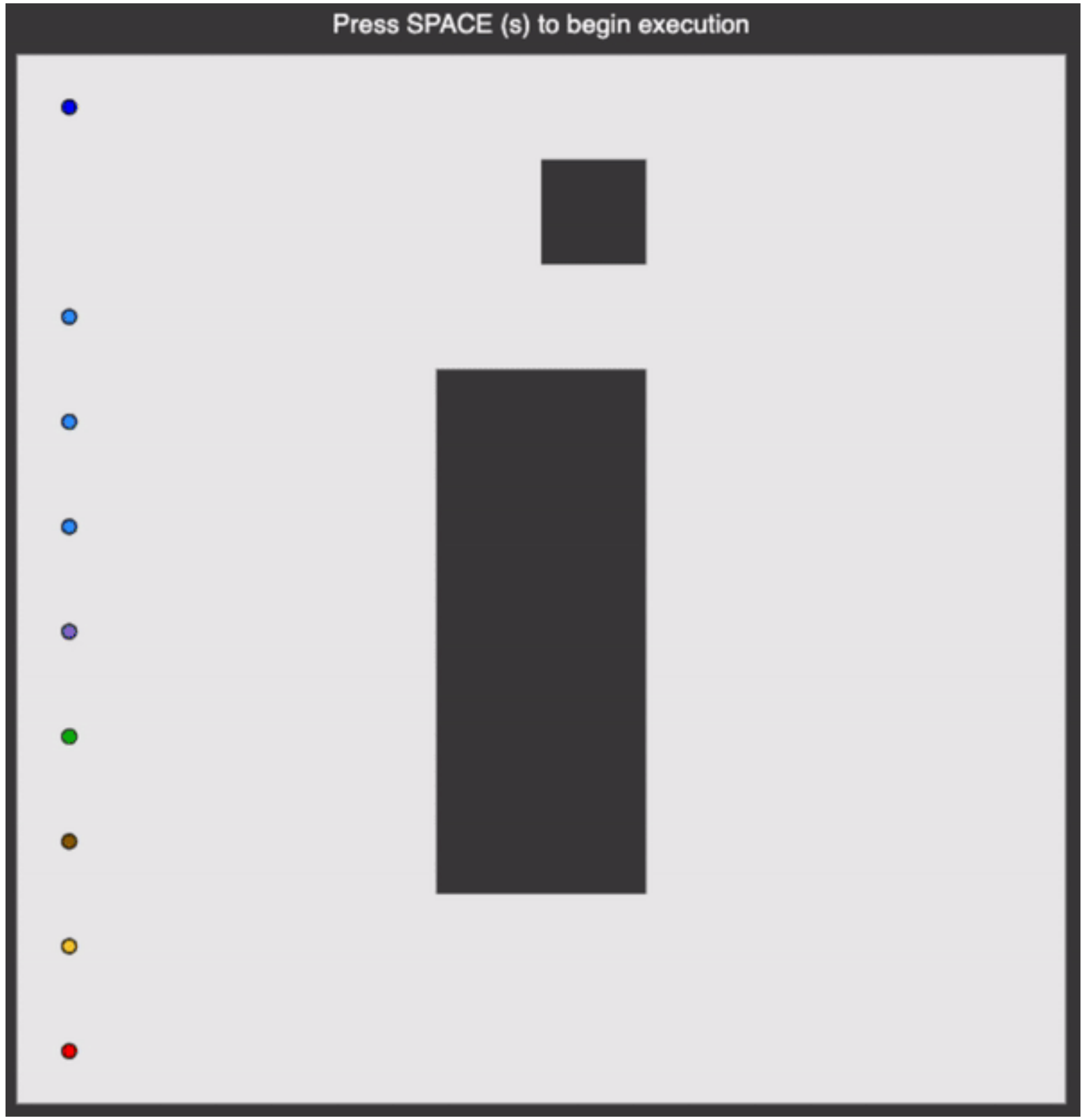


- Can create a clone at one of the “mystery points”
- Clones can do the wrapping independently
- Collected boosters are shared between all clones

Clones

14 hours into the contest





- Can create a clone at one of the "mystery points"
- Clones can do the wrapping independently
- Collected boosters are shared between all clones

Extended Task

- **150** maps with boosters of different kinds
- Solution is a zip-file with Wrappy traces (text files)
- The shorter the traces, the better

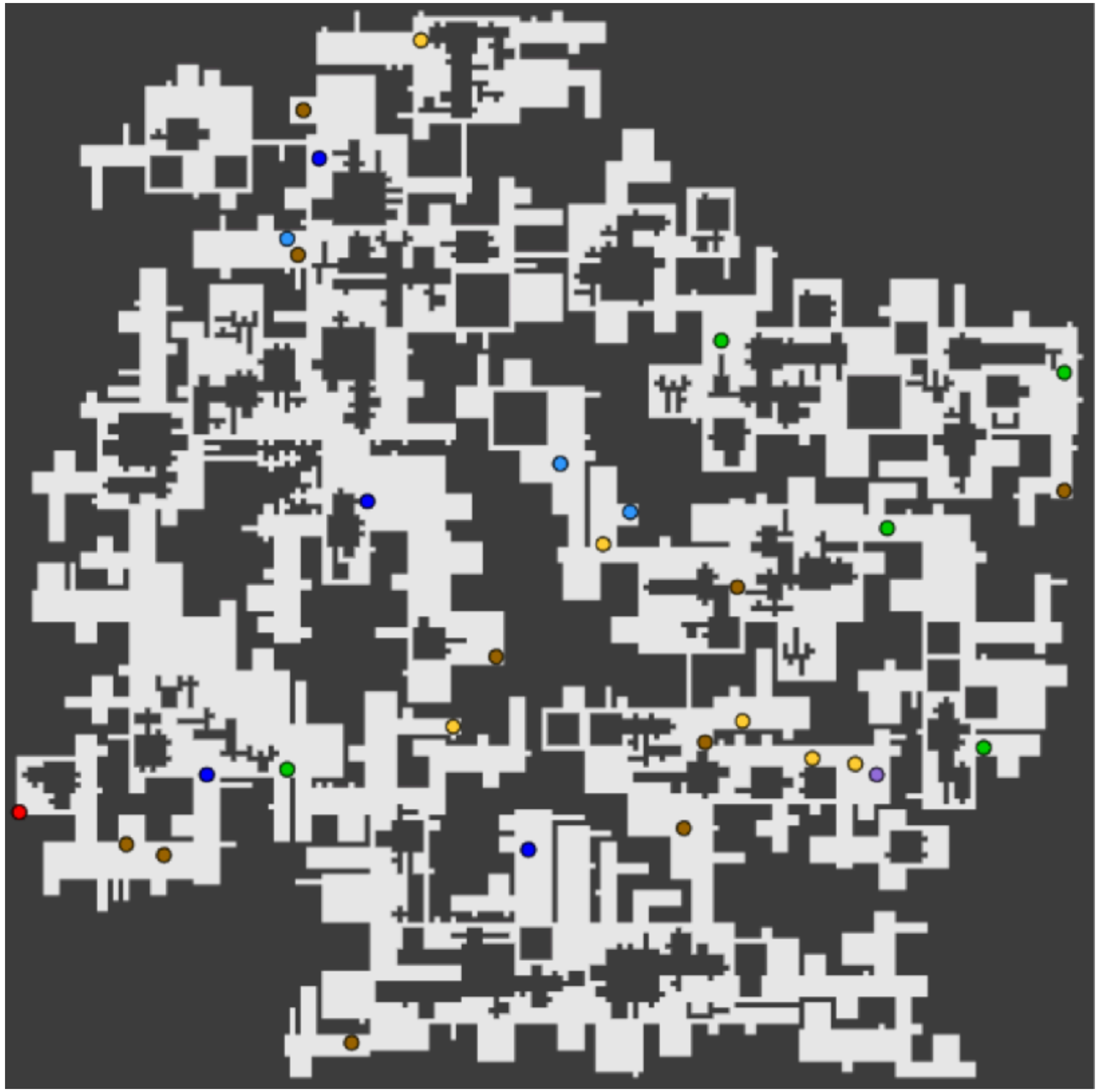
$$score_{team,T} \triangleq \left\lceil 1000 \times \log_2 \left(X_T \times Y_T\right) \times \frac{t_{best}}{t_{team}} \right\rceil$$

Extended Task

- ~~150~~ **300** maps with boosters of different kinds
- Solution is a zip-file with Wrappy traces (text files)
- The shorter the traces, the better

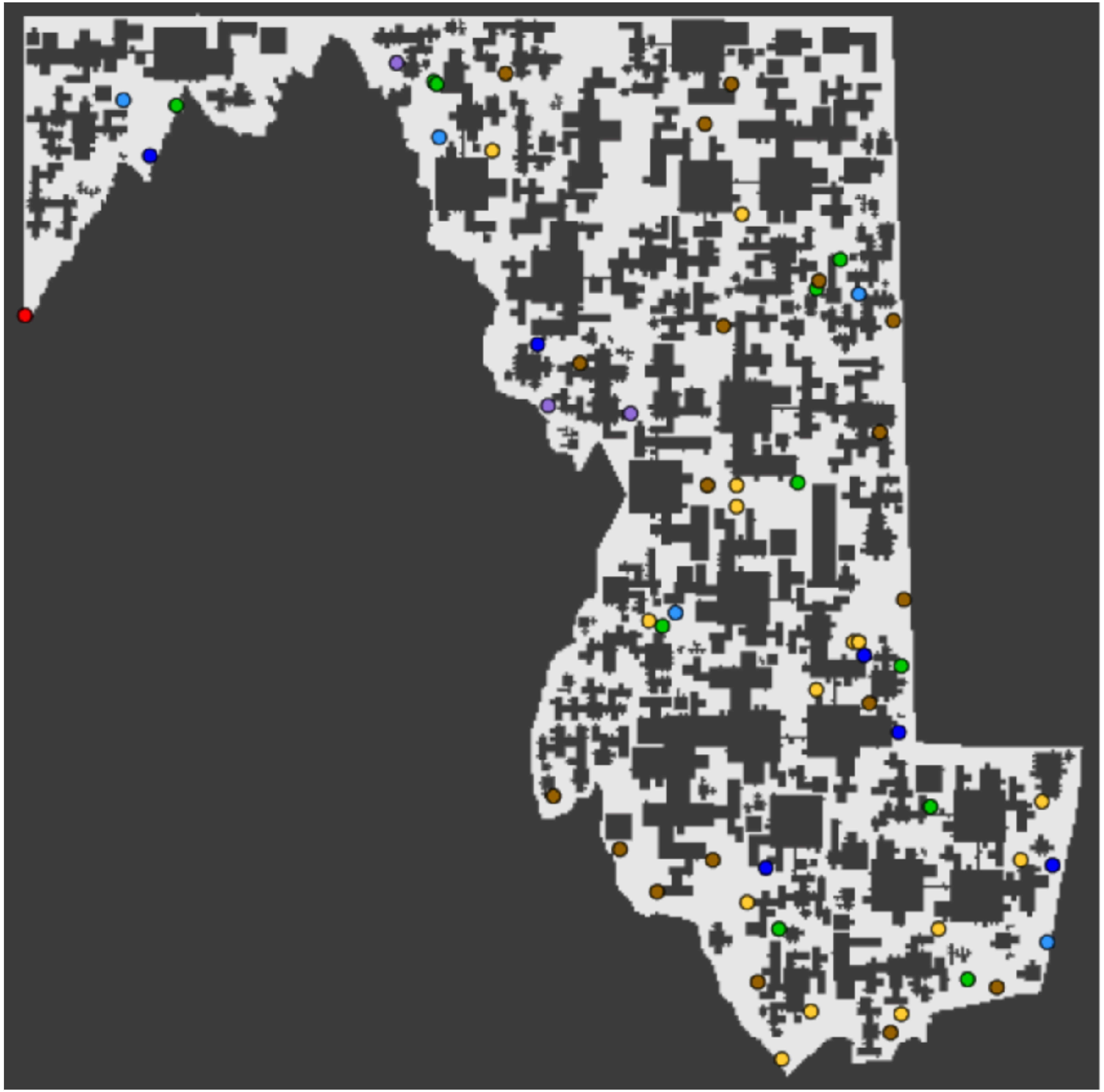
$$score_{team,T} \triangleq \left\lceil 1000 \times \log_2 (X_T \times Y_T) \times \frac{t_{best}}{t_{team}} \right\rceil$$

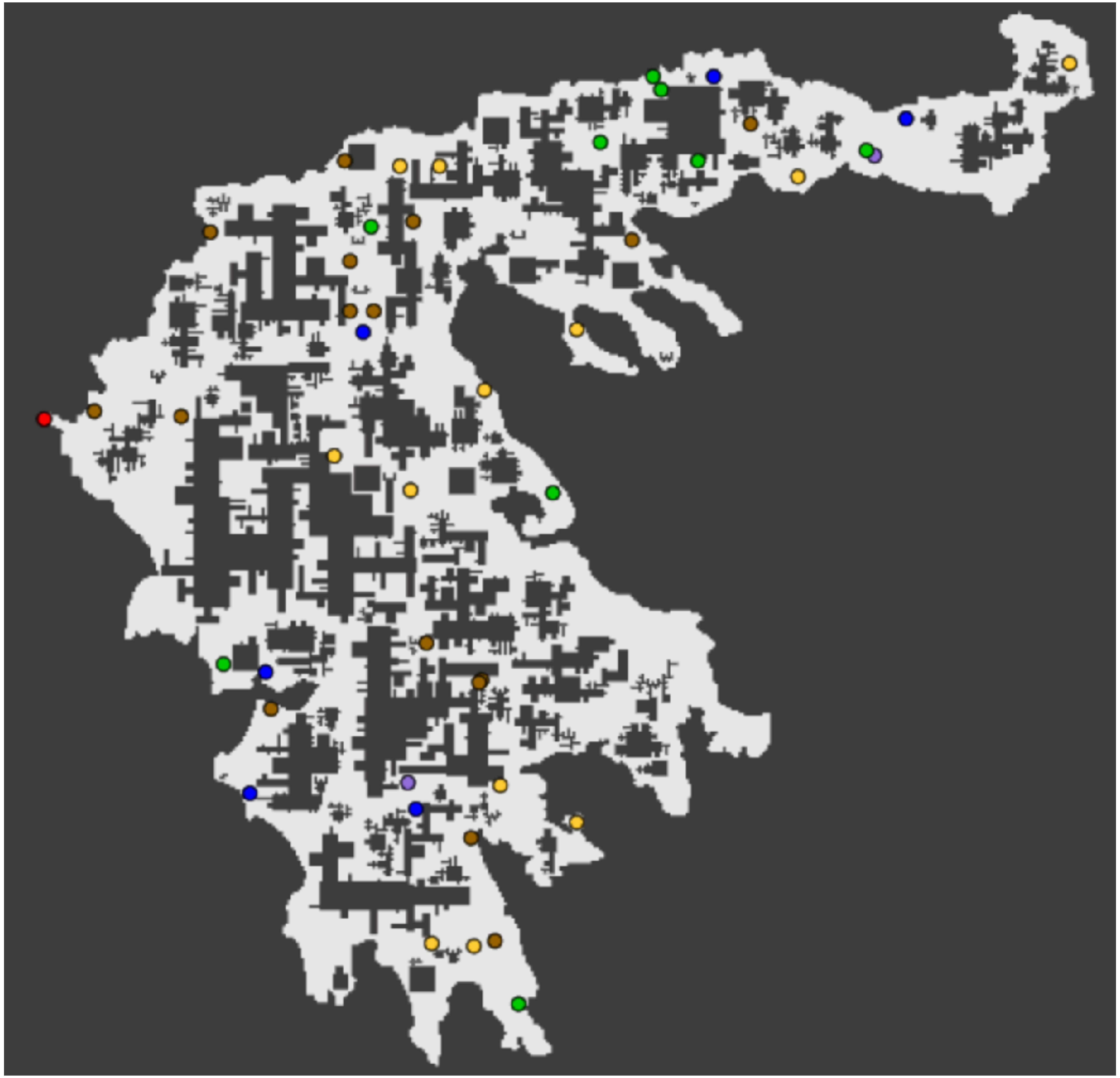
More Maps

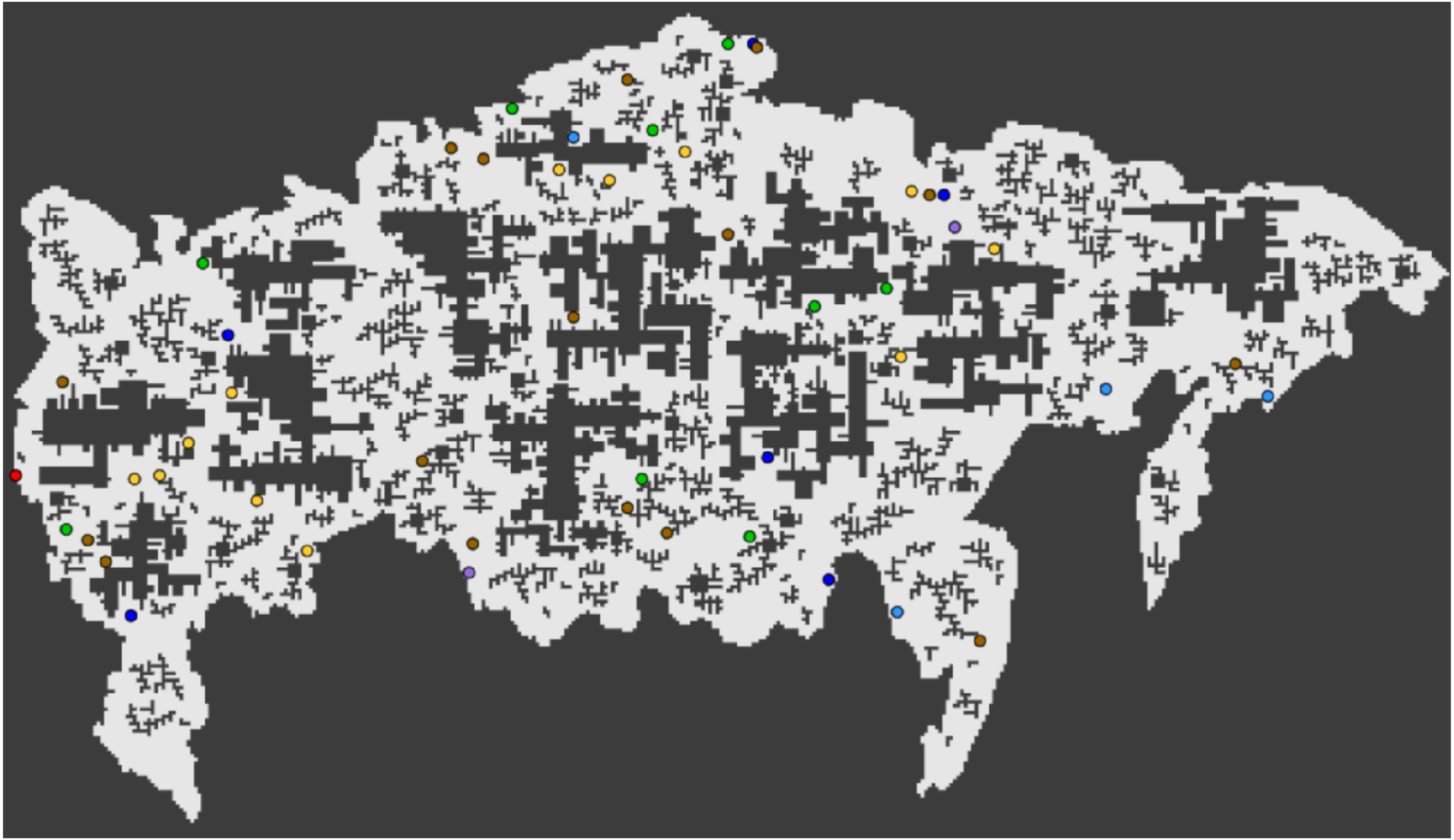


200 x 200

400 x 400

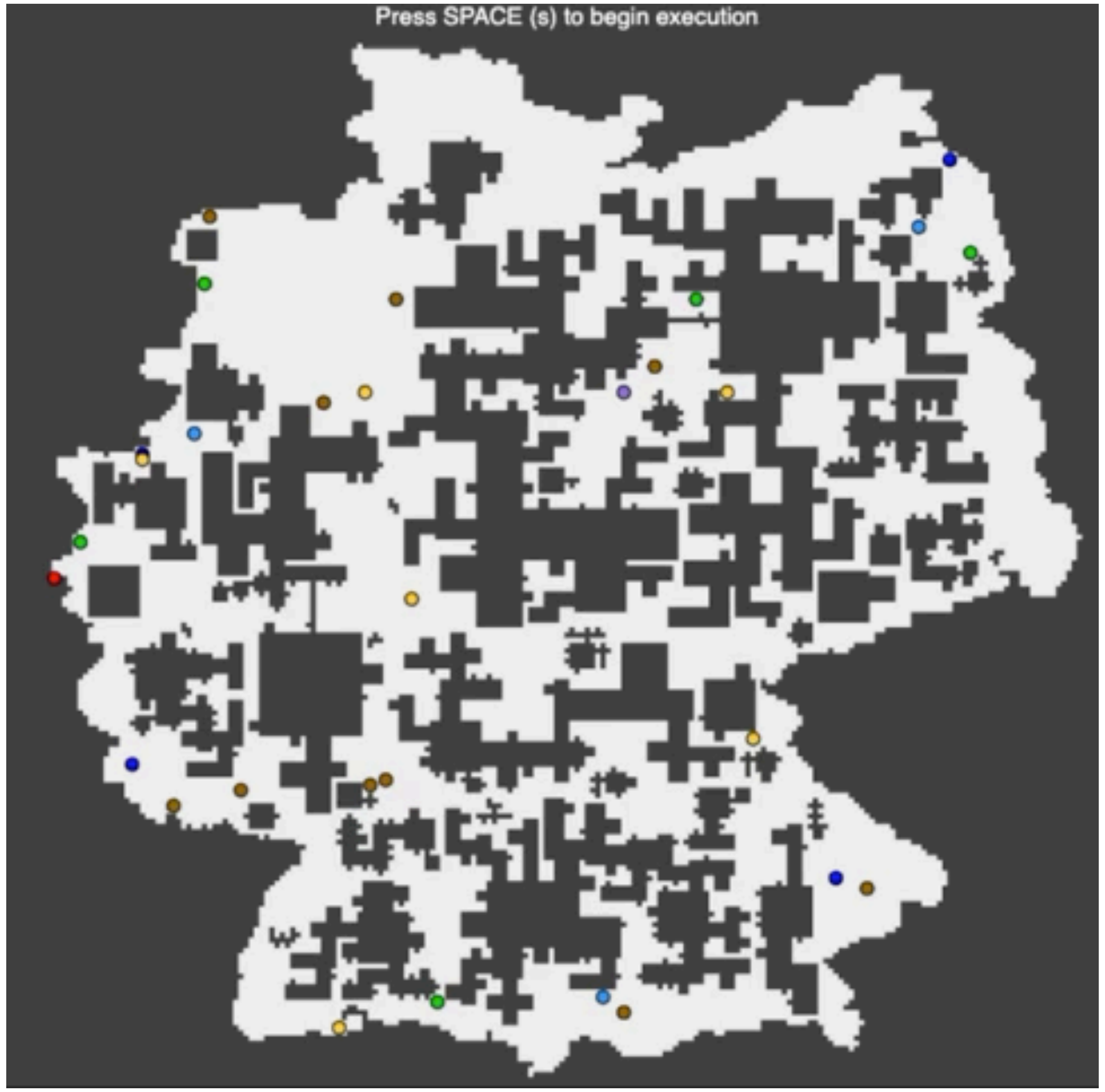






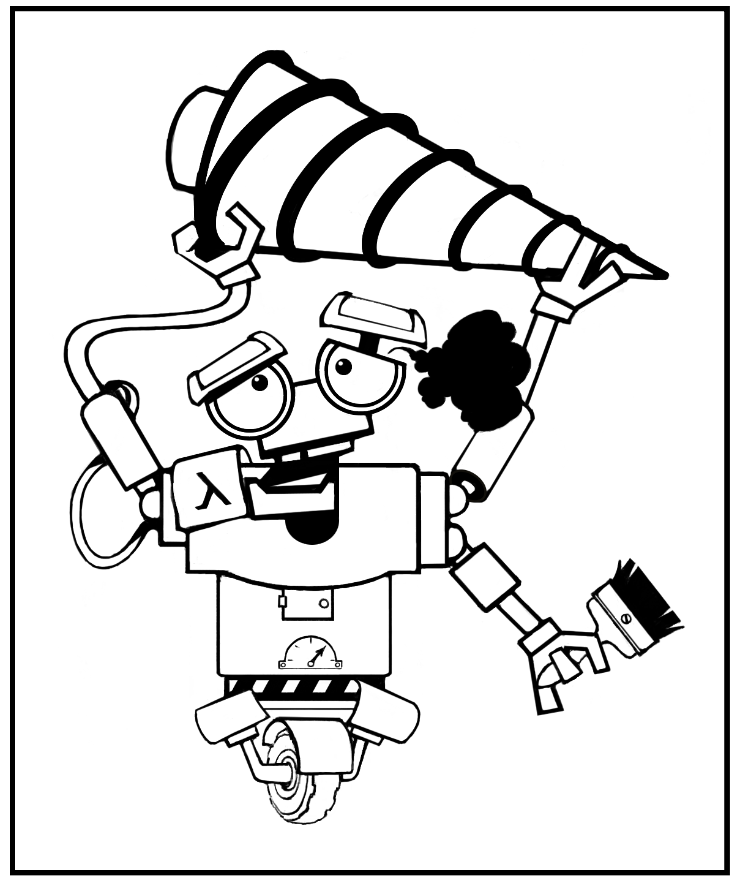
Provided Infrastructure

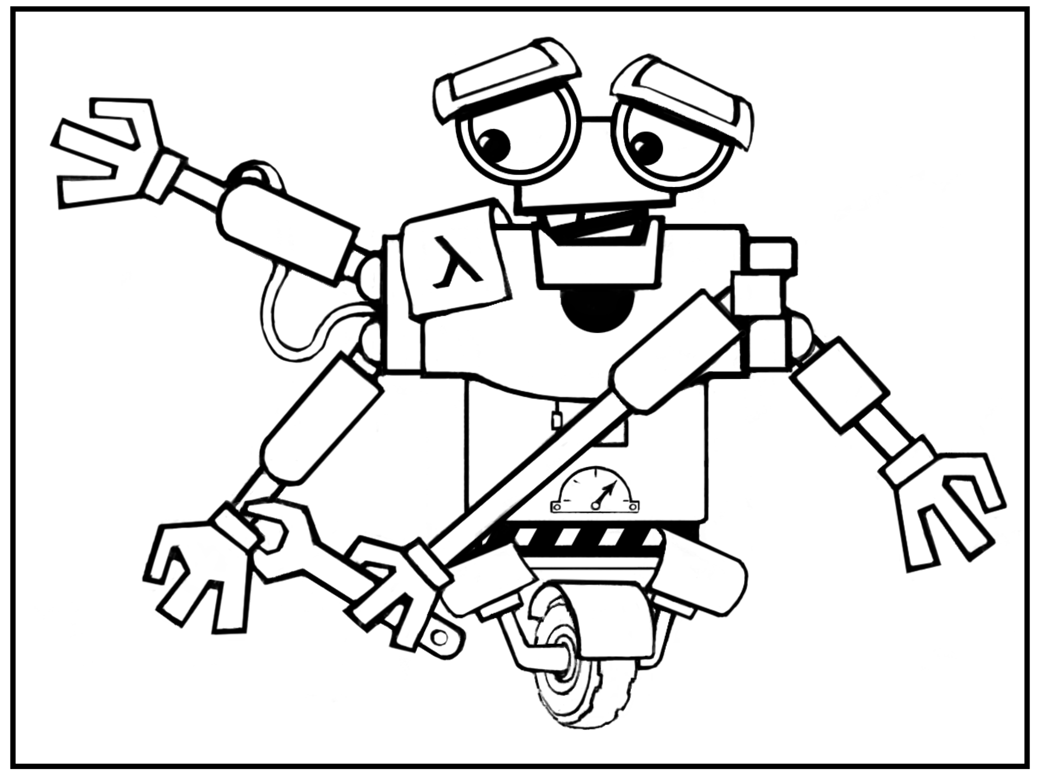
- Solution is a zip-file with Wrappy traces (text files)
-
  ```
  curl -F “private_id=34fbde"\
  -F "file=@solutions.zip" \
  https://monadic-lab.org/submit
  ```
- A JavaScript visualiser (running in the browser)
- Live Rankings (updated every five minutes)
- A *very crappy* reference solver (not released)

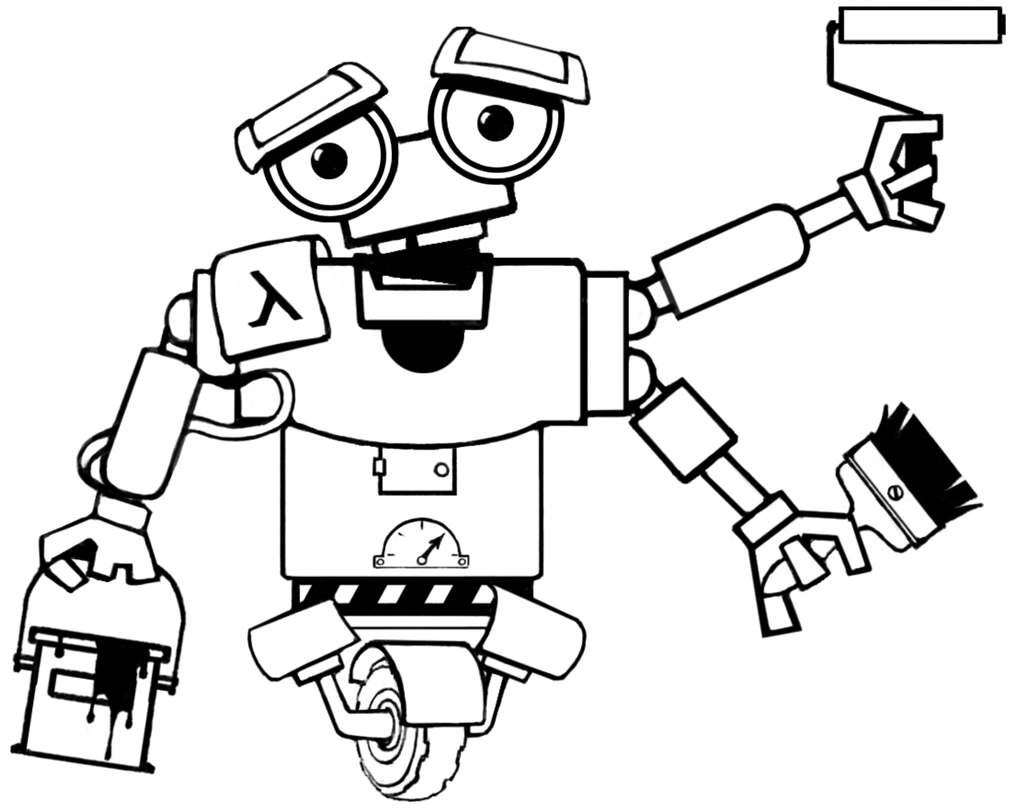


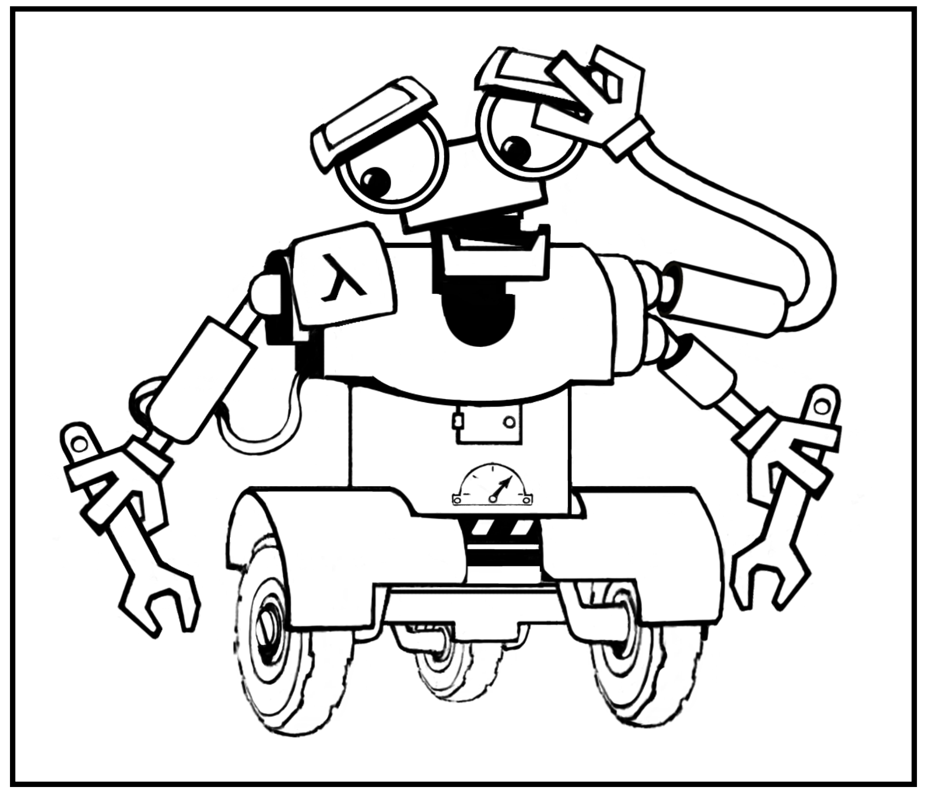
And with that we finished the Lightning Division…

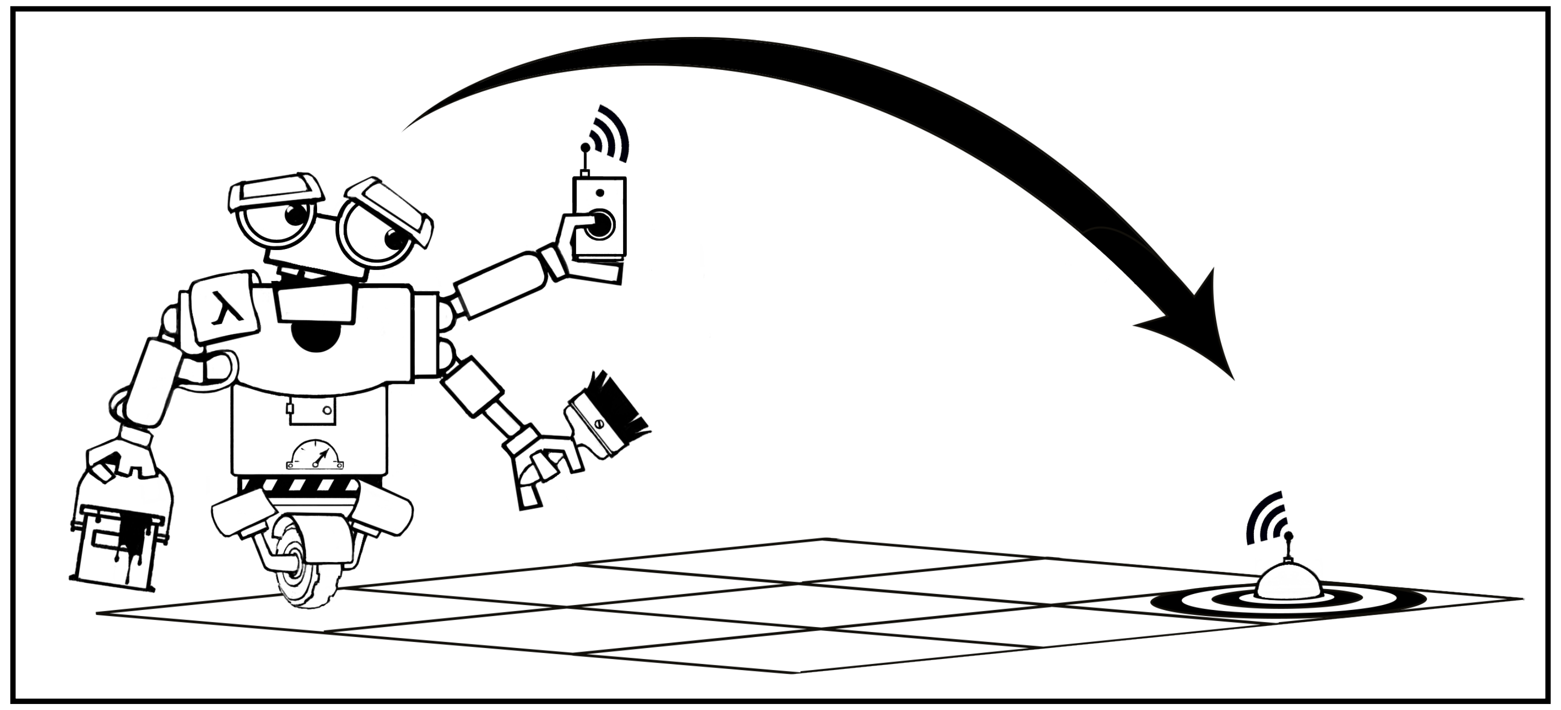
… but what kind of
ICFP Contest would that be
without a *twist*?

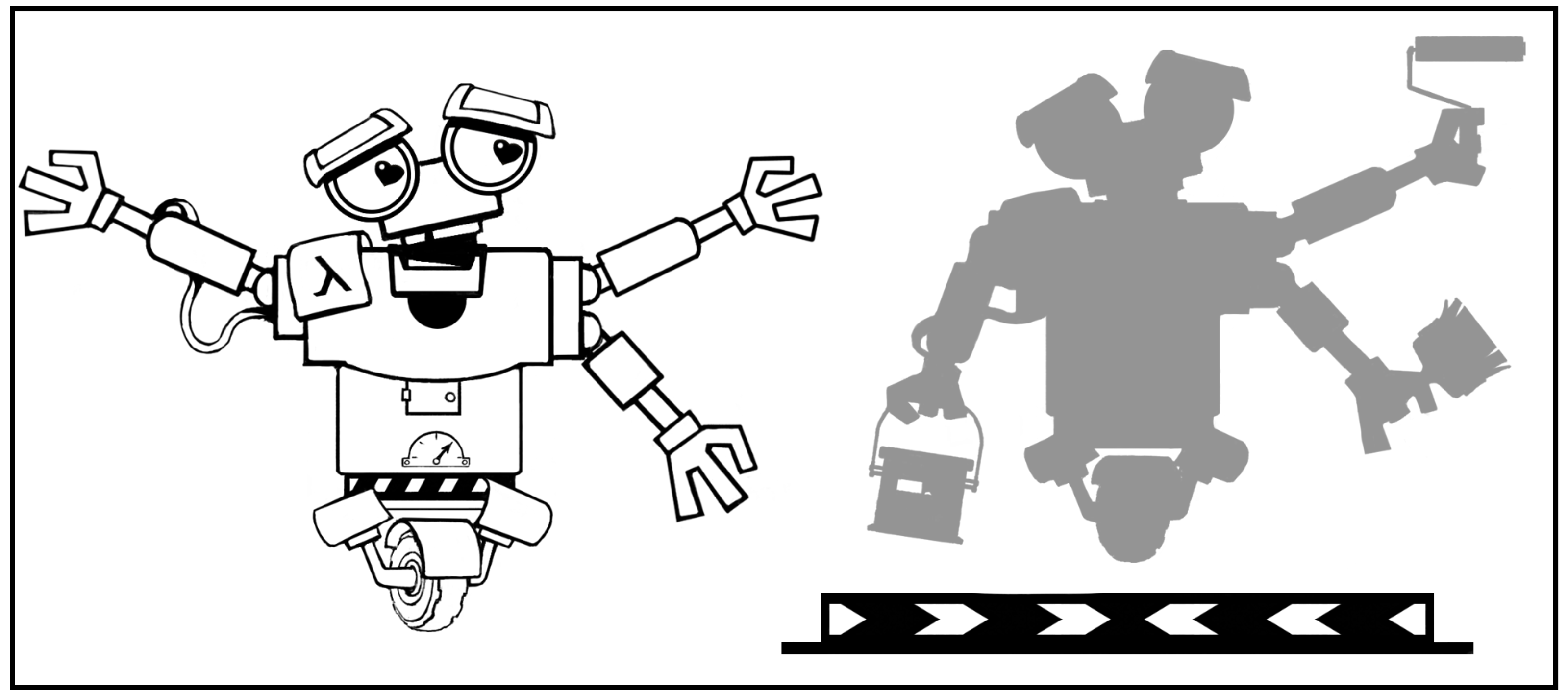


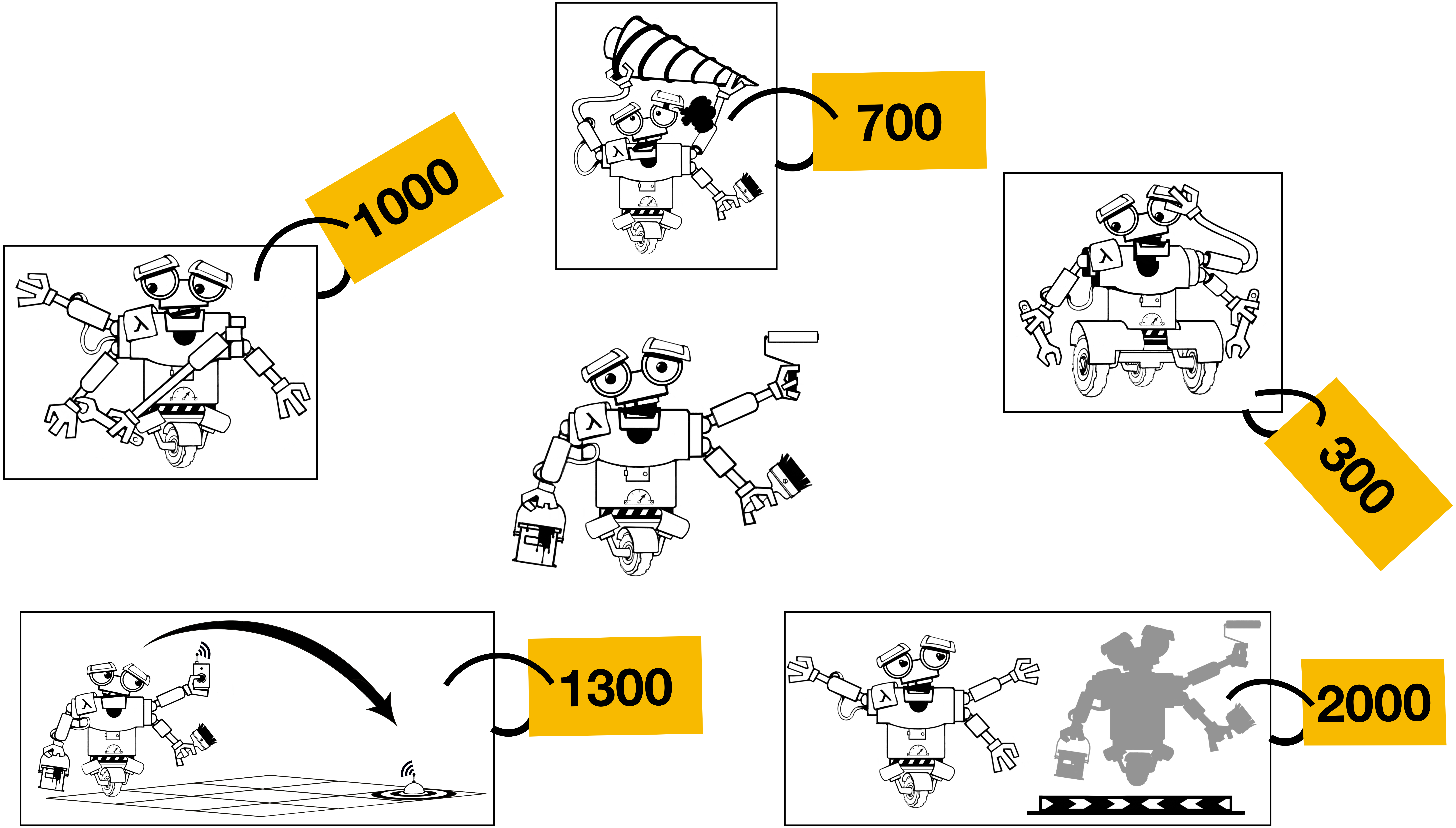












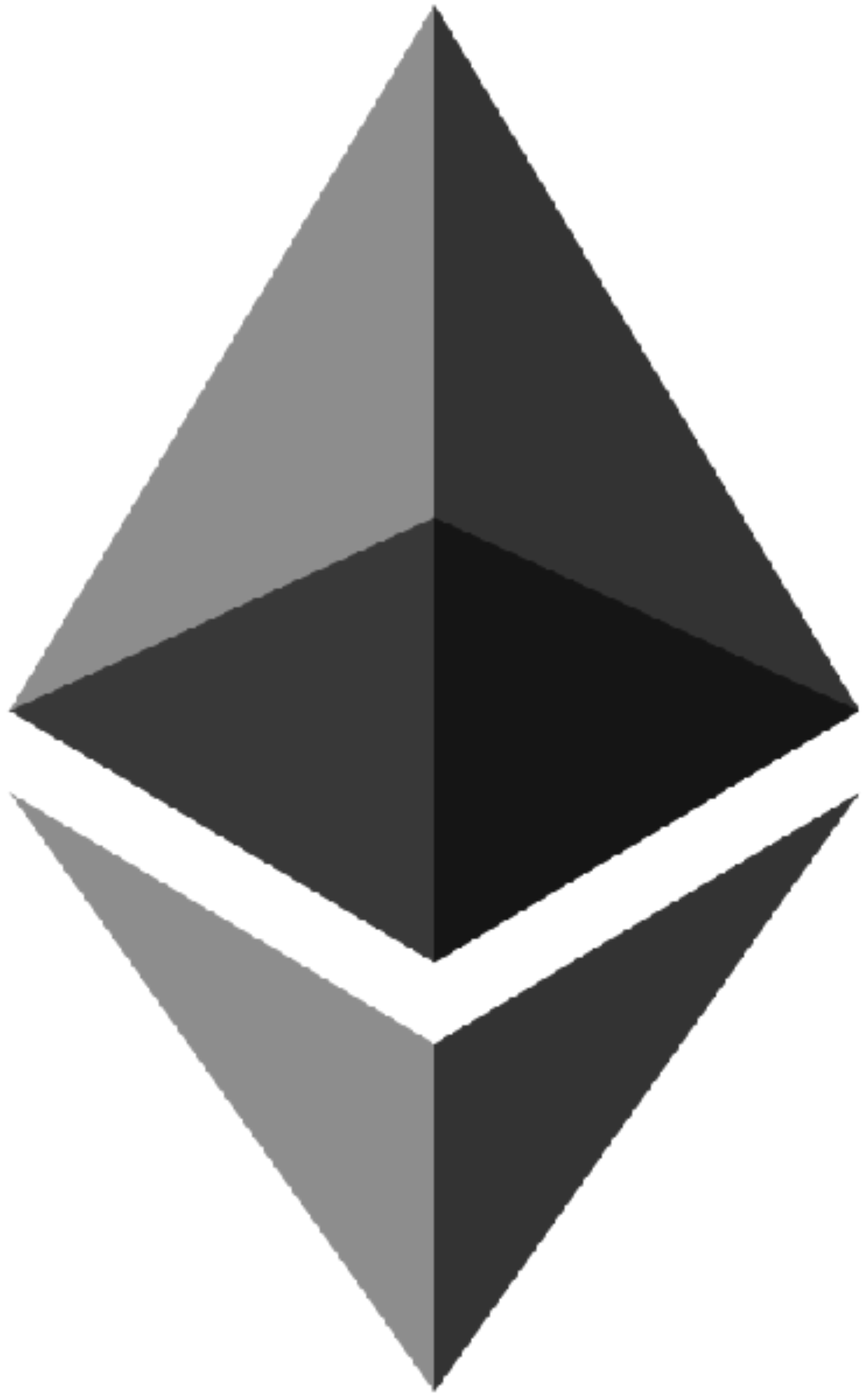
1000

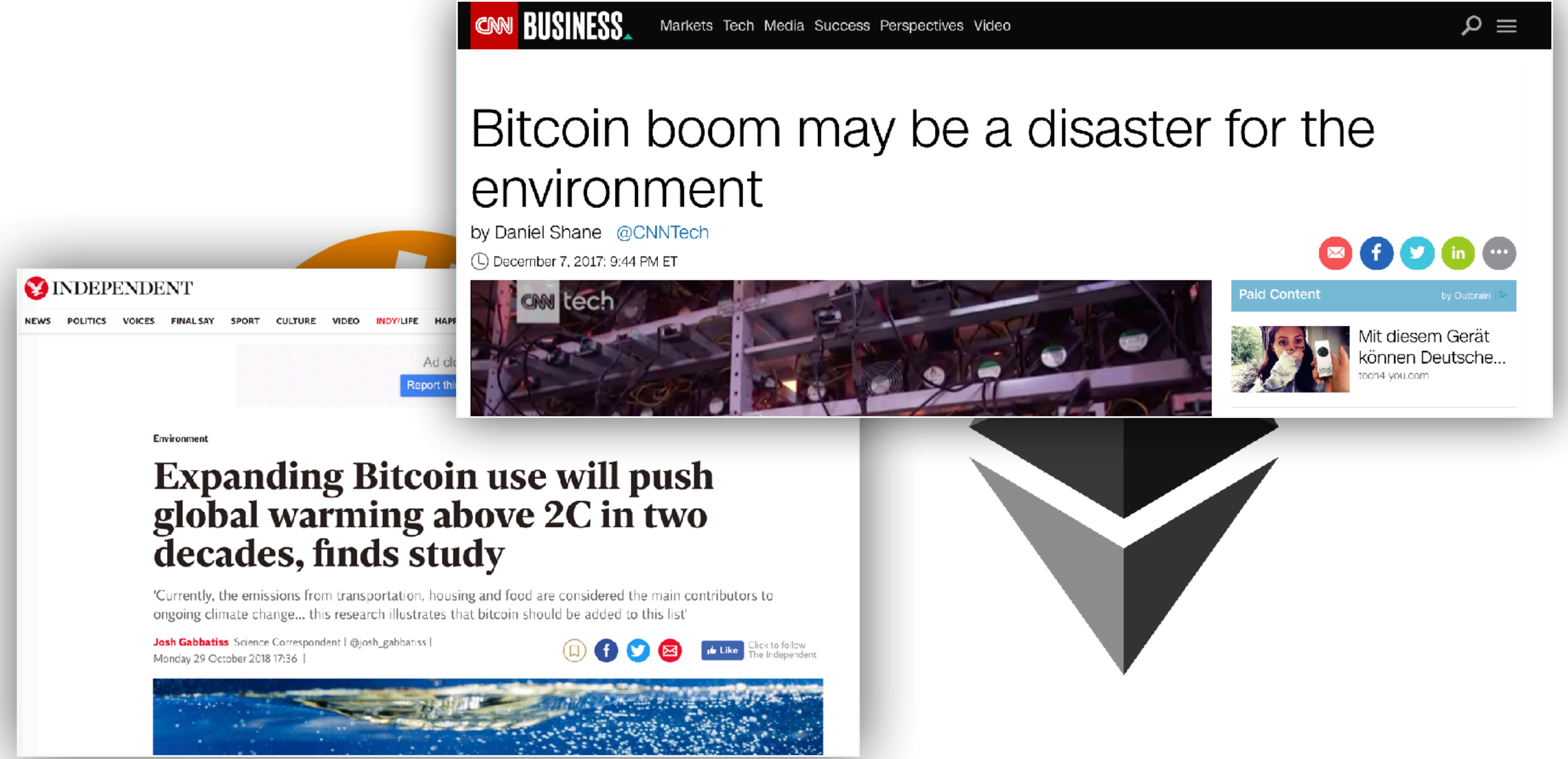
700

300

1300

2000





CNN BUSINESS
Markets Tech Media Success Perspectives Video

Bitcoin boom may be a disaster for the environment

by Daniel Shane @CNNTech

December 7, 2017: 9:44 PM ET

INDEPENDENT
NEWS POLITICS VOICES FINAL SAY SPORT CULTURE VIDEO INDY/LIFE HAPP

Environment

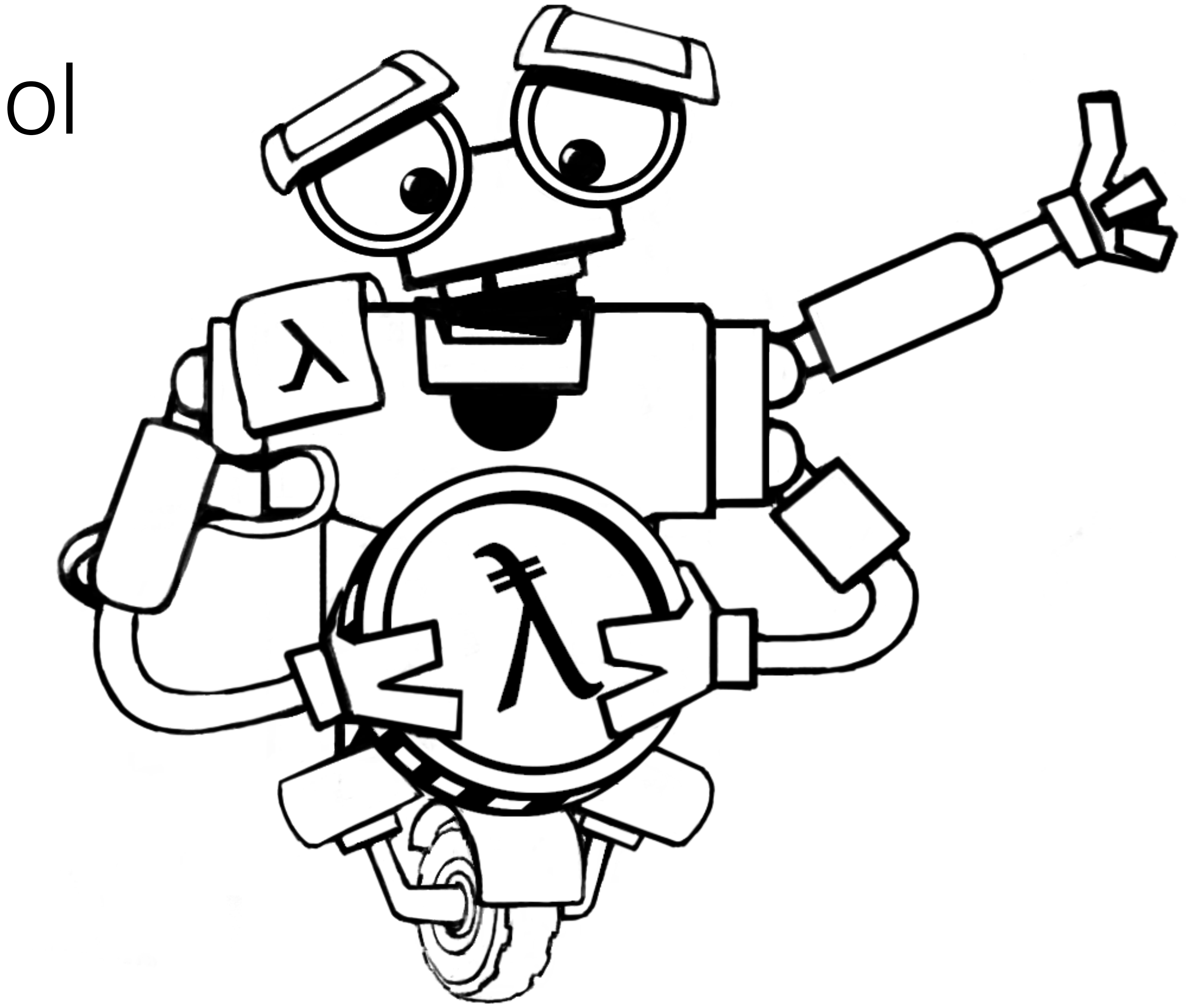
Expanding Bitcoin use will push global warming above 2C in two decades, finds study

'Currently, the emissions from transportation, housing and food are considered the main contributors to ongoing climate change... this research illustrates that bitcoin should be added to this list'

Josh Gabbatiss Science Correspondent | @josh_gabbatiss |
Monday 29 October 2018 17:36 |

Lambda-Coin (LAM)

- Awarded for participating in the *Lambda-chain* protocol
- Lambda-chain mining is environmentally friendly
- By doing it, one helps to fight the Bit-Rot Problem
- Powered by a ***novel consensus protocol***



(Also, it’s an anagram of **Monadic Lab**)

~~Proof-of-Work~~

Proof-of-Wrap

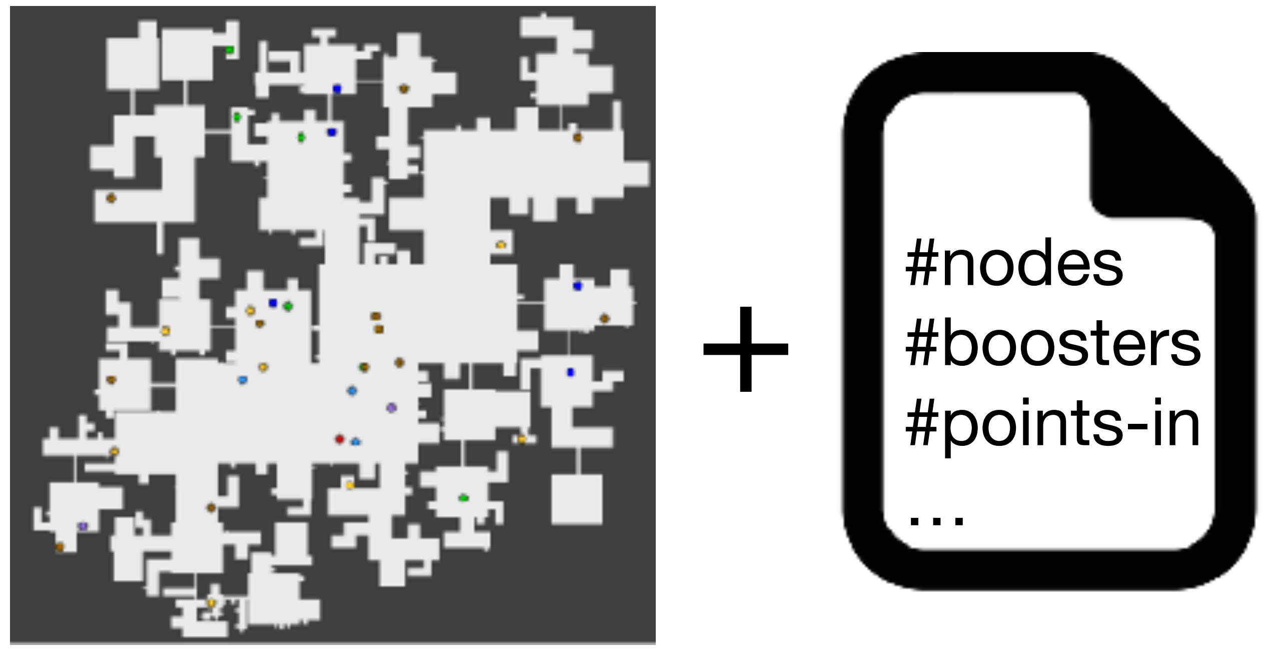
Proof-of-Wrap in a Nutshell

- Lambda-chain is *a* ***synchronous distributed*** (in fact, not) **consensus protocol**

- Initially, a *genesis block* is announced. It contains:

  (1) A *map for wrapping* (similar to a task from the main contest)

  (2) A *proof-of-wrap* puzzle (new map specification)



- For the next 15 minutes, the contest server expects solutions for the PoW block:

  (1) A *solution* for the wrapping task

  (2) A *new task (map)* satisfying *the specification in the puzzle*

- **100,000 LAM** are split between the teams with top **25 best-ranked solutions** for (1)

- A ***contributed map*** (2) from one of the **top 10 ranked teams** is chosen for the ***next round's task*** (1)
  - This team is awarded additional **2,000 LAM** and is *banned* for the next round

- The cycle *repeats* with the new map and a new puzzle (generated by organisers)

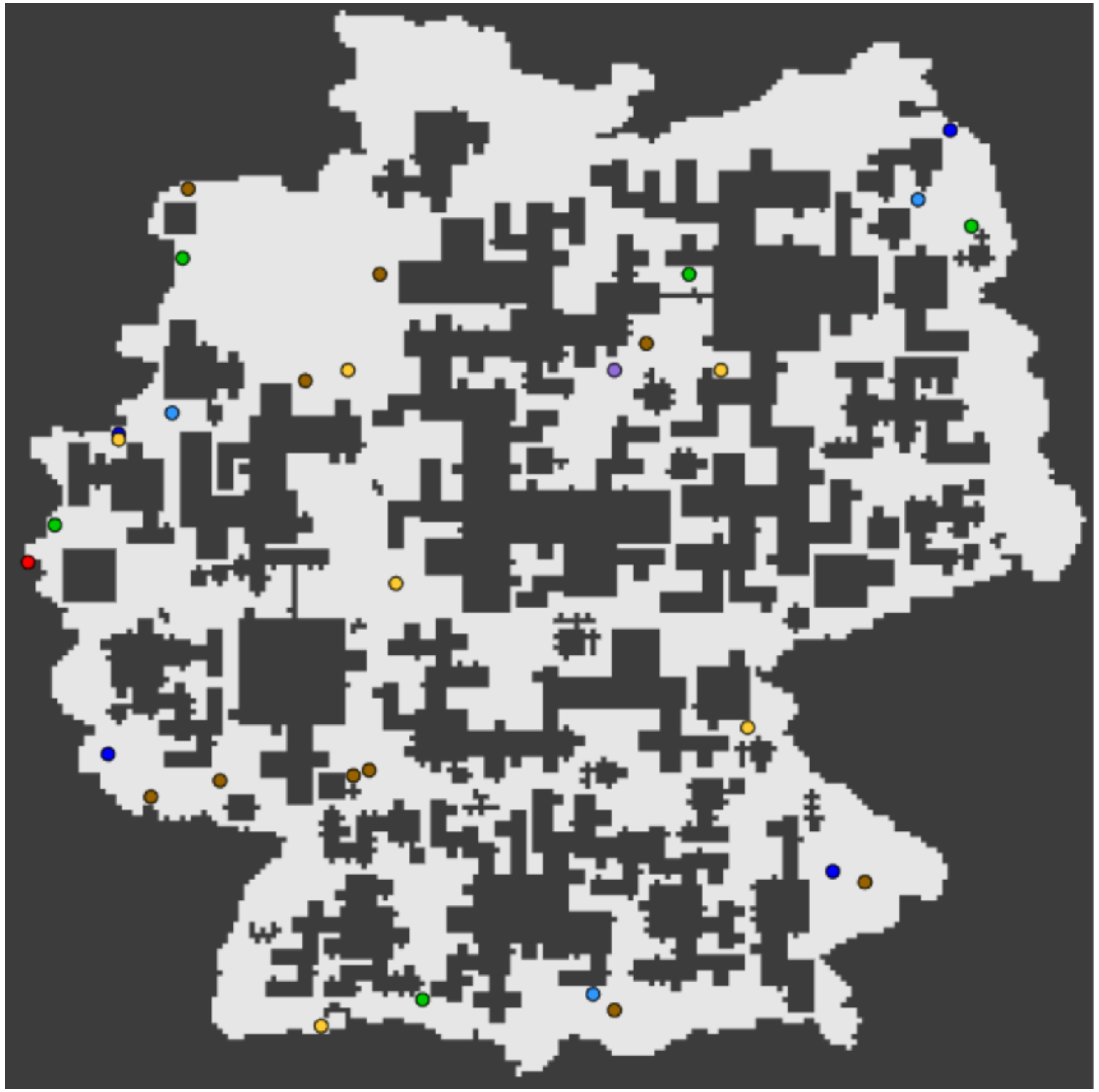
The Essence

- To get awarded LAM, one has to ***solve current block's task*** and ***propose a map for the next block's task***

- The ***block reward*** (in LAM) is ***proportional to the quality*** of the solution

- If a team's map is chosen as a task for the *next block*, ***it cannot participate*** in the next round of mining, but ***gets an extra reward***

Genesis Block Task



Rules for Contributed Maps

- Each block's *puzzle* ***specifies parameters*** for the next block map proposal:
  - Dimensions of the map
  - A distribution of boosters in the puzzle
  - A number of coordinates that should be ***within it***
  - A number of coordinates that should be ***outside of it***, *etc*.
- Suggested maps shouldn't bee ***too sparse*** or ***too small*** (size condition)
- They also shouldn't bee ***too boring*** or ***too heavy***

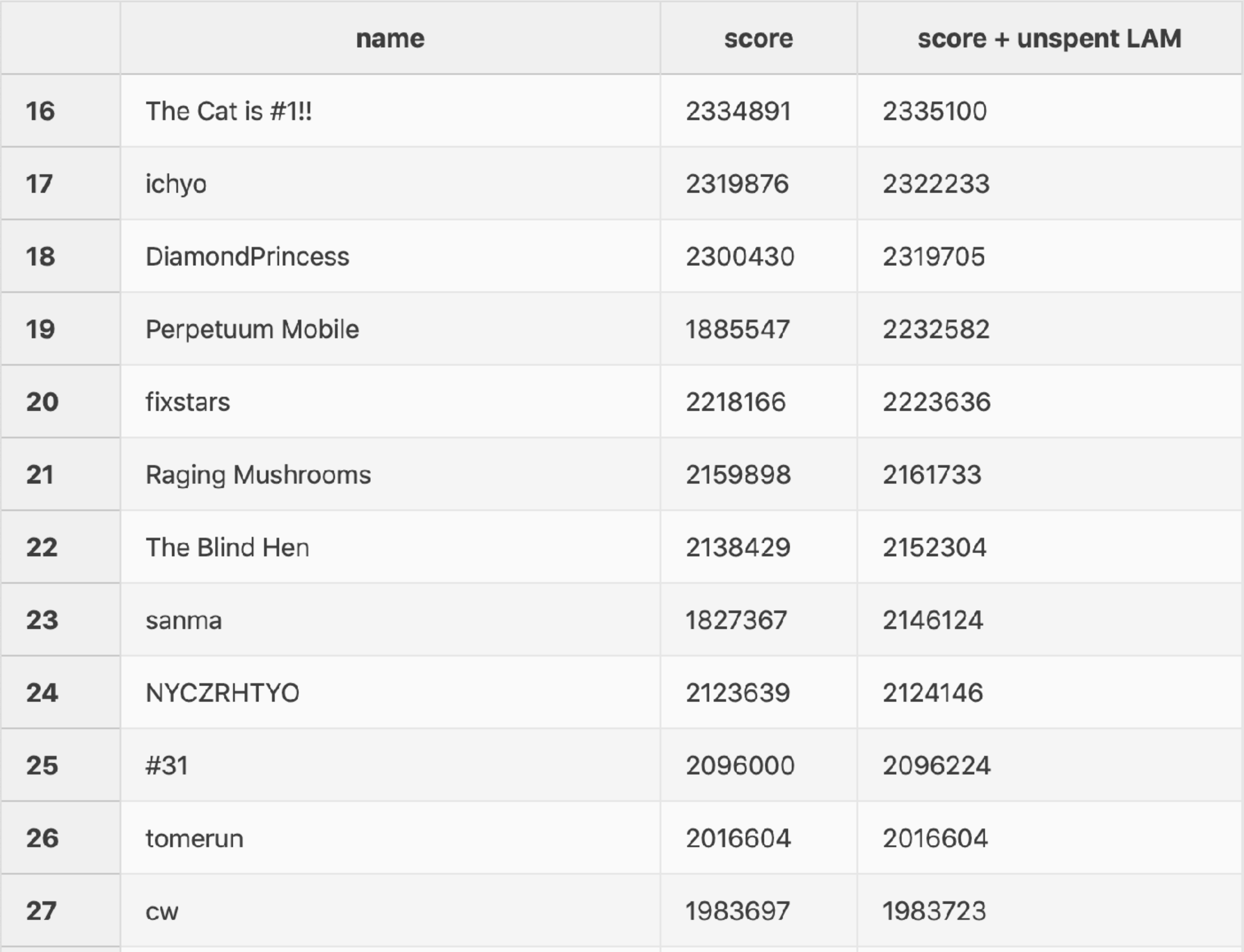
Spending Lambda-coins

- A team can spend awarded LAM ***on boosters***
- LAM can be spent ***multiple times*** in resubmissions
- Alternatively, LAM can be exchanged to score points in the ration **1 to 1**

Item	Price (LAM)
Manipulator	1000
Fast Wheels	300
Drill	700
Teleport	1200
Cloning	2000

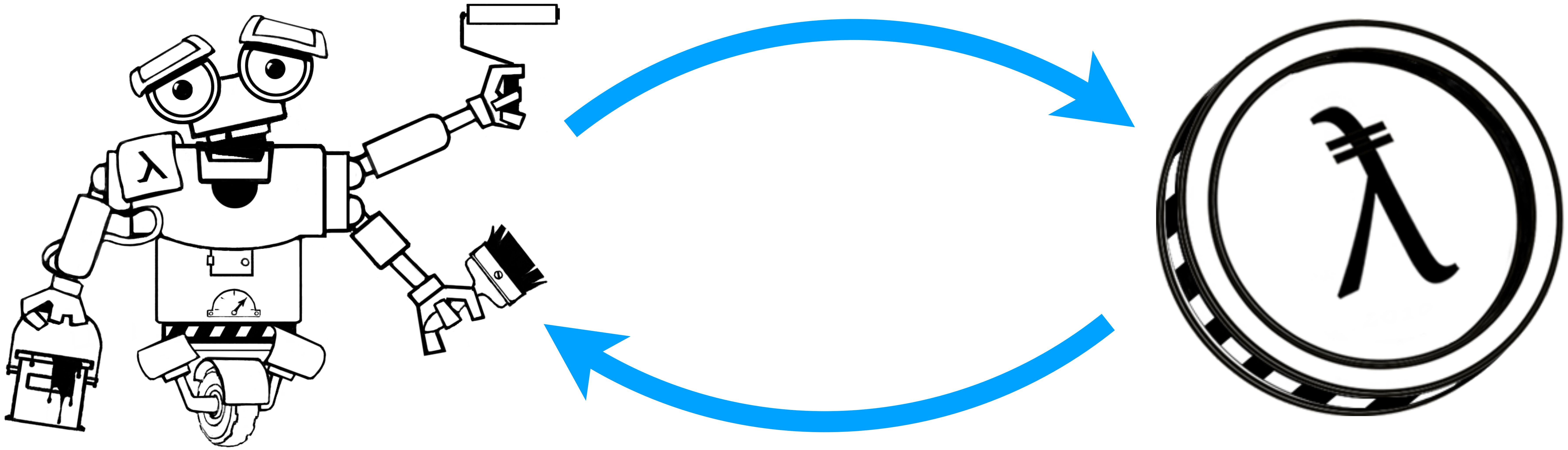
Provided Infrastructure for Lambda-coin Mining

- A Python ***script to submit*** block solutions
- Command-line block explorer
- ***Real-time monitoring*** of submitted solution
- Rankings with and without ***unspent*** LAM



	name	score	score + unspent LAM
16	The Cat is #1!!	2334891	2335100
17	ichyo	2319876	2322233
18	DiamondPrincess	2300430	2319705
19	Perpetuum Mobile	1885547	2232582
20	fixstars	2218166	2223636
21	Raging Mushrooms	2159898	2161733
22	The Blind Hen	2138429	2152304
23	sanma	1827367	2146124
24	NYCZRHTYO	2123639	2124146
25	#31	2096000	2096224
26	tomerun	2016604	2016604
27	cw	1983697	1983723

The Contest



Statistics

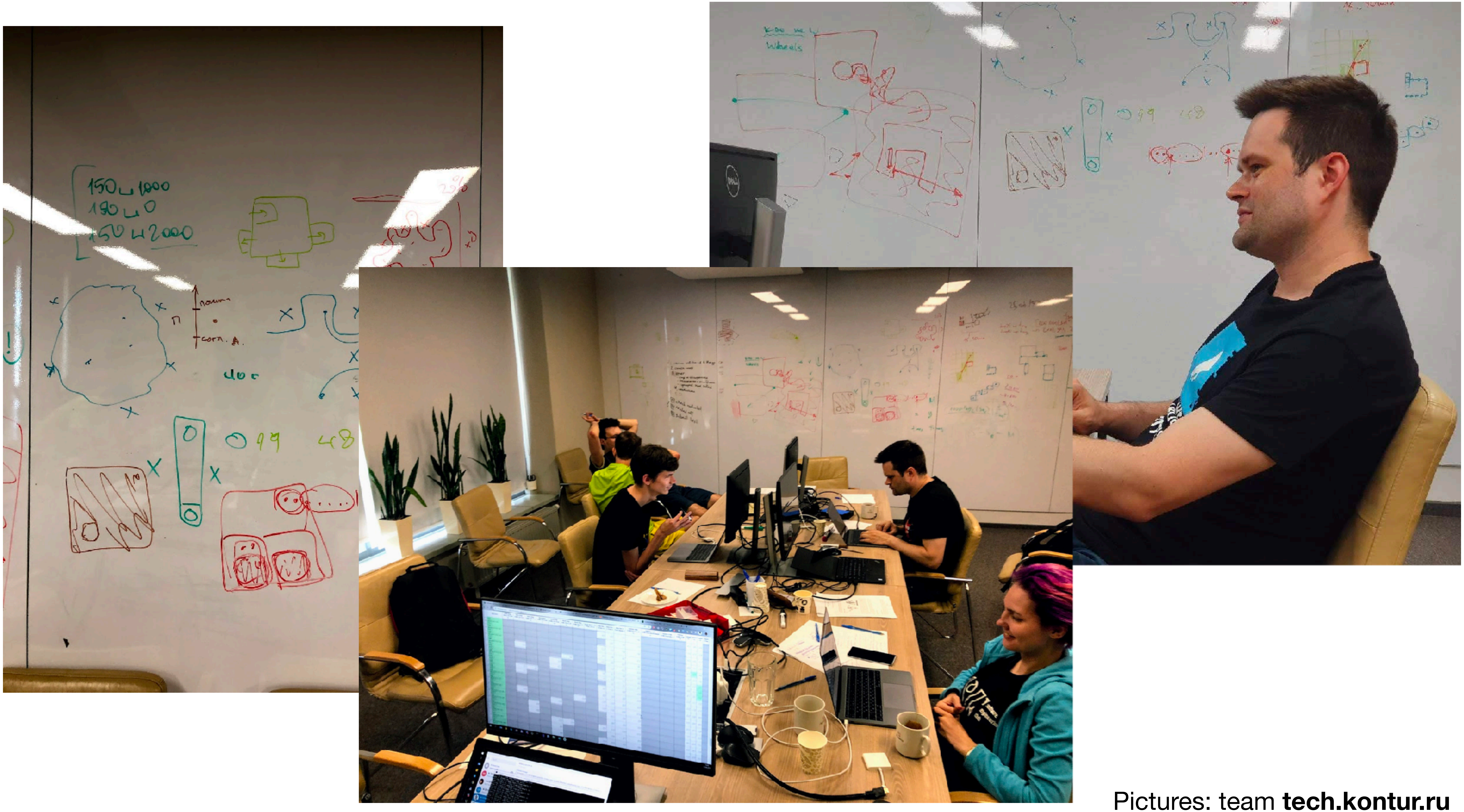
Regular Contest

- **194** registered teams
- **105** complete profiles
- Lightning division
  - **774** submissions
  - **106** final submissions with non-zero scores
- Main contest
  - **2745** submissions
  - **142** final submissions with non-zero scores
- CPU time spent grading: **25h 27m**
- Average grading time: **33.37 sec**

19
1
6
2
5
3
1
1
1
25
29
3
1
1
1
2
1

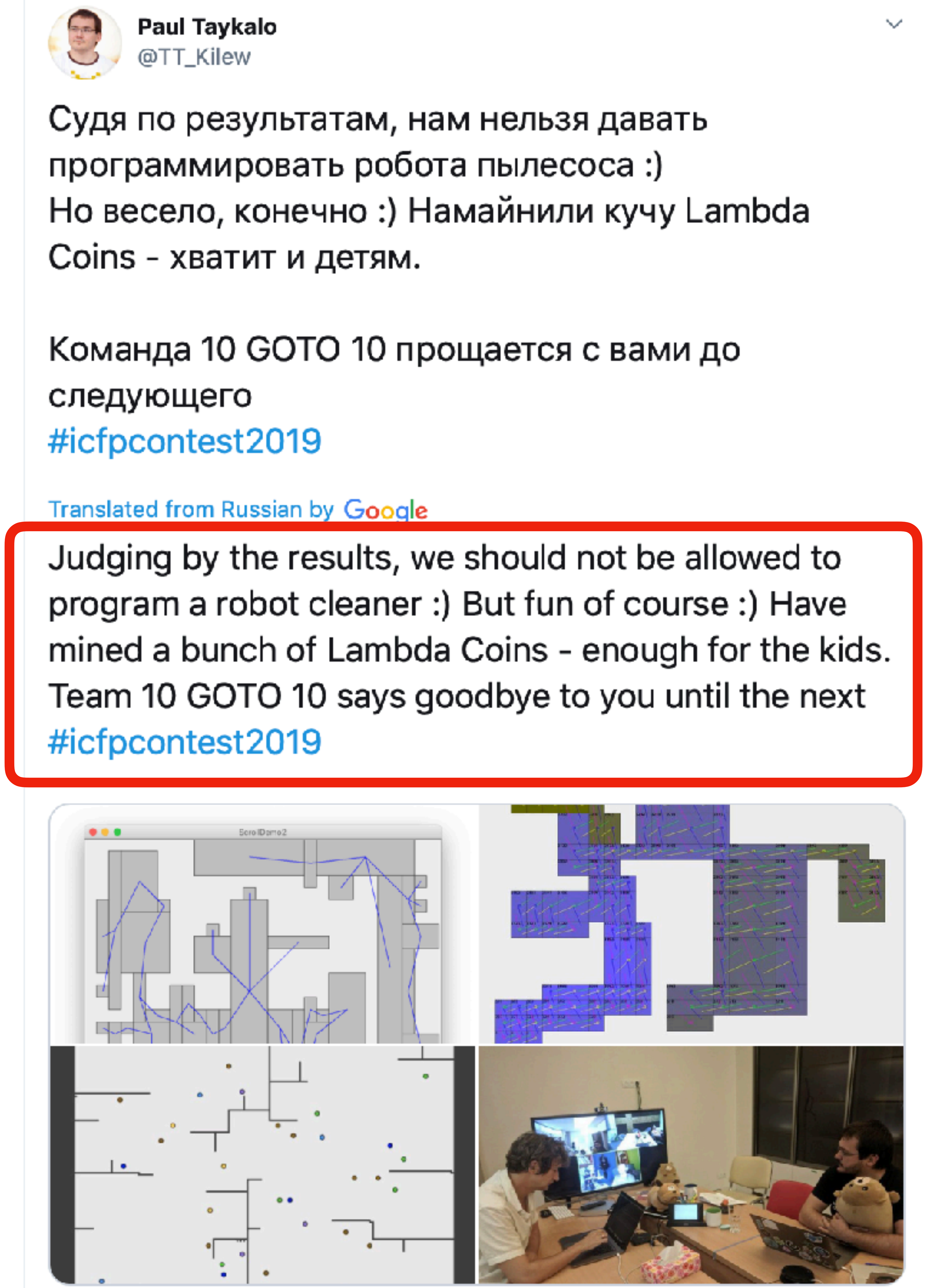
Countries (as in Profiles)

Japan	29	Switzerland	1
USA	25	Colombia	1
Russia	19	United Kingdom	1
Germany	6	Indonesia	1
Ukraine	5	Iceland	1
France	3	Italy	1
India	3	Latvia	1
Australia	2	Malaysia	1
Denmark	2	The Netherlands	1
Poland	2	New Zealand	1
Austria	1	Portugal	1
Bahamas	1	Singapore	1
Canada	1	No Info	6



Pictures: team **tech.kontur.ru**

Picture credit: **Niki Vazou**



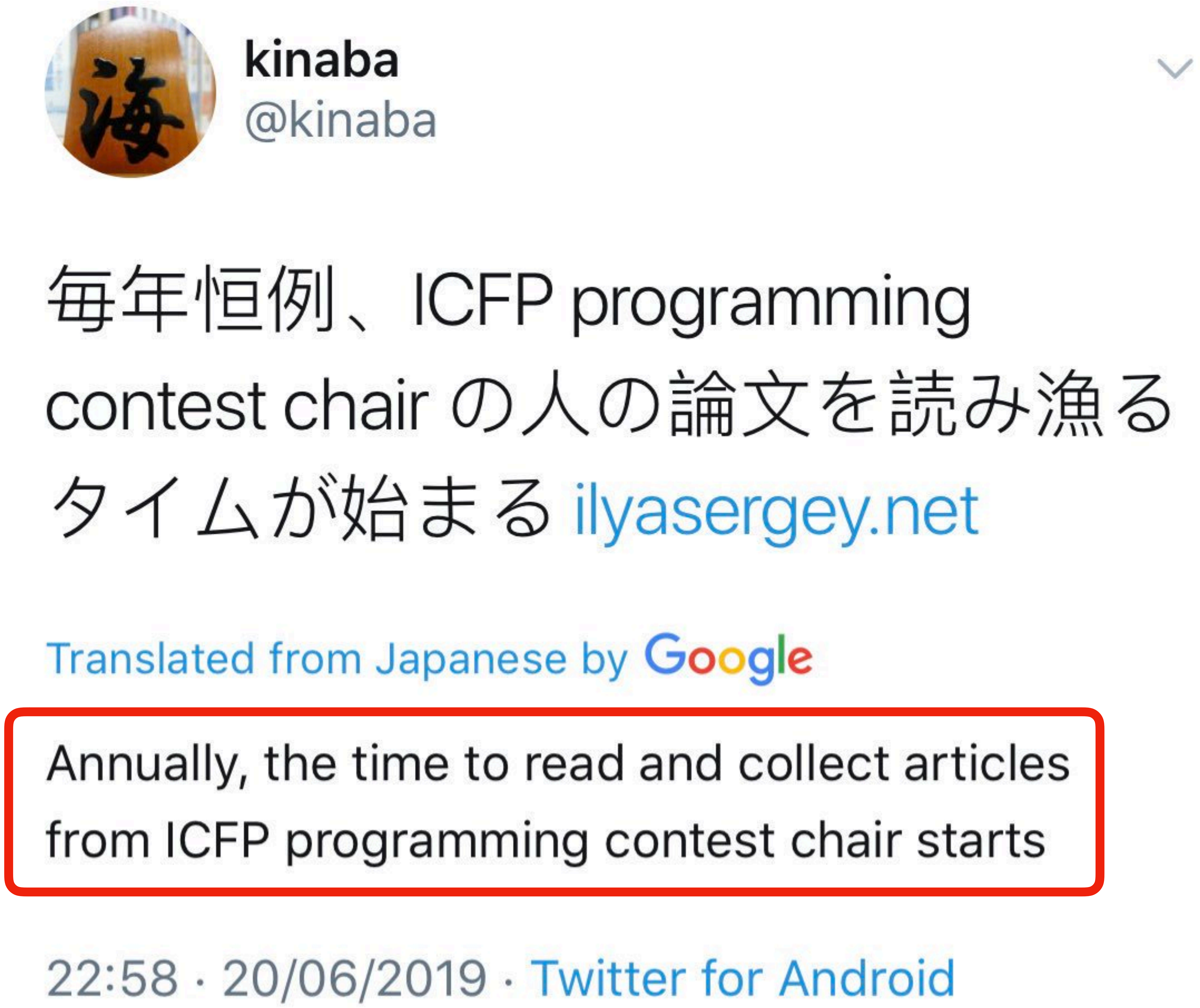
Paul Taykalo
@TT_Kilew

Судя по результатам, нам нельзя давать программировать робота пылесоса :)
Но весело, конечно :) Намайнили кучу Lambda Coins - хватит и детям.

Команда 10 GOTO 10 прощается с вами до следующего
#icfpcontest2019

Translated from Russian by Google

Judging by the results, we should not be allowed to program a robot cleaner :) But fun of course :) Have mined a bunch of Lambda Coins - enough for the kids. Team 10 GOTO 10 says goodbye to you until the next #icfpcontest2019



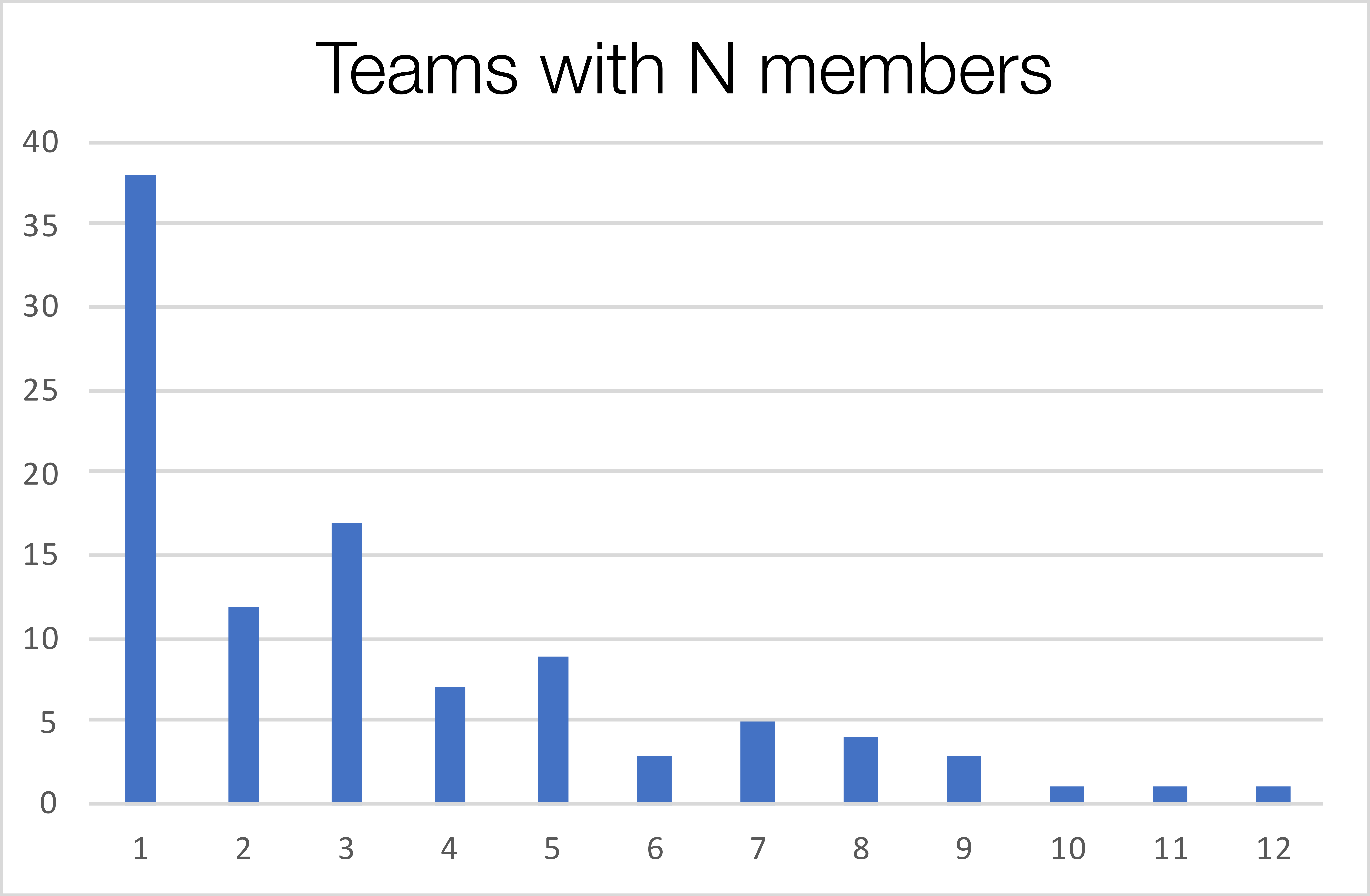
kinaba
@kinaba

毎年恒例、ICFP programming contest chair の人の論文を読み漁るタイムが始まる ilyasergey.net

Translated from Japanese by Google

Annually, the time to read and collect articles from ICFP programming contest chair starts

22:58 · 20/06/2019 · Twitter for Android


Teams with N members
40
35
30
25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
11
12

30
25
20
15
10
5
0
29
27
14
12
10
9
7
6
6
5
5
4
3
3
2
2
2
2
2
1
1
1
1
1
1
1
1
1
1
c++
python
rust
haskell
bash
javascript
ruby
kotlin
ocaml
c#
scala
go
java
sh
c
clojure
f#
lisp
typescript
common lisp
crystal
hand
perl
powershell
r
shellscript
smt-lib
sql
vue.js

Submissions per hour of the contest

160
140
120
100
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

Lambda-chain Statistics

- **31** hour of mining
- **4177** puzzle solutions submitted
- **3769** of them are *valid* (90.23%)
- **121** block mined
- **12,136,480** LAM awarded

Block submissions for each block
50
40
30
20
10
0
-10
0
20
40
60
80
100
120
total_subs
good_subs

Slow adoption:

had to postpone mining for 10 hours

Brave Early Adopters

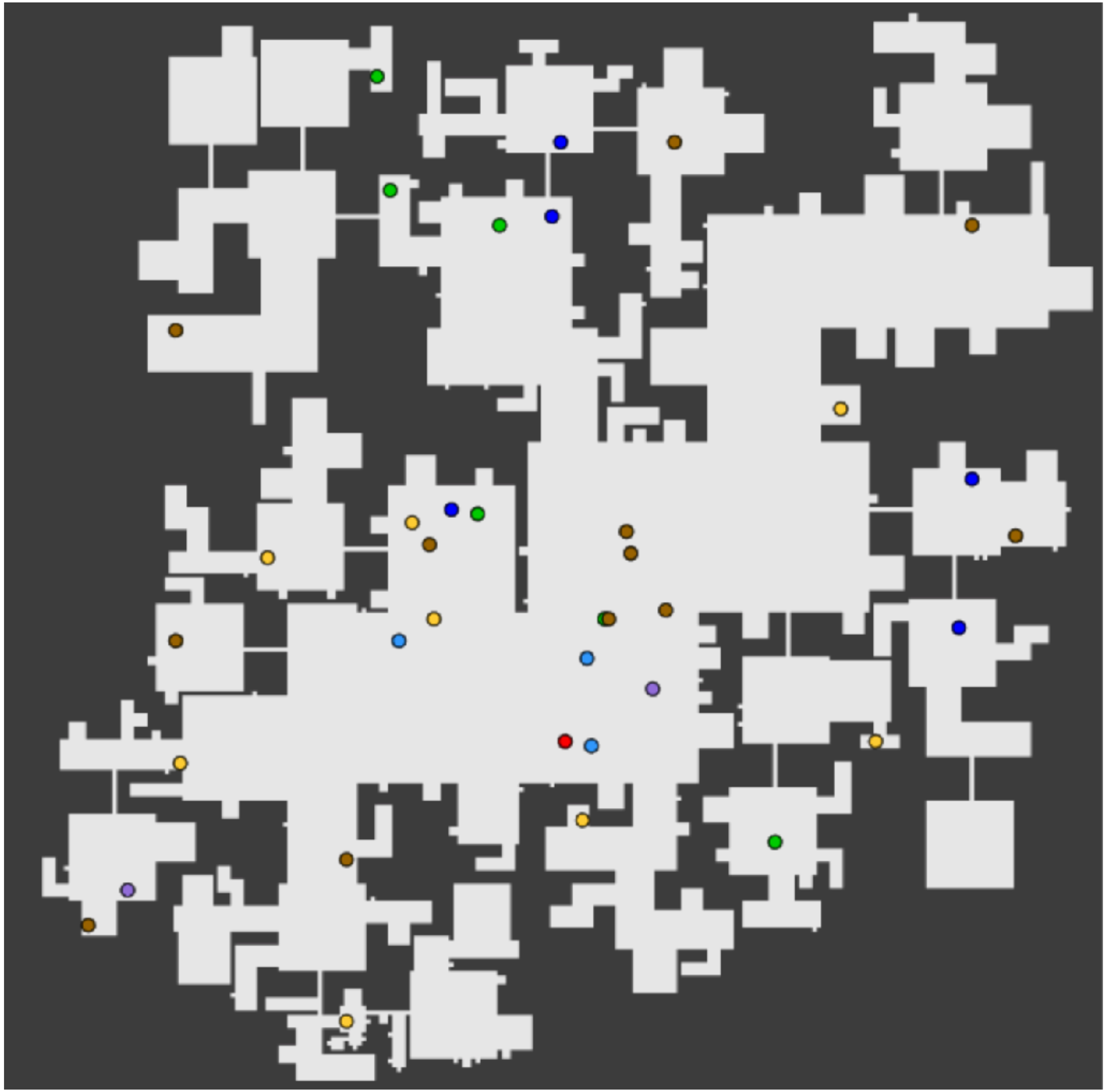
- Unagi
- All your lambda are belong to us
- Sound! TypeSystem
- 1kg cheese

Block submissions for each block

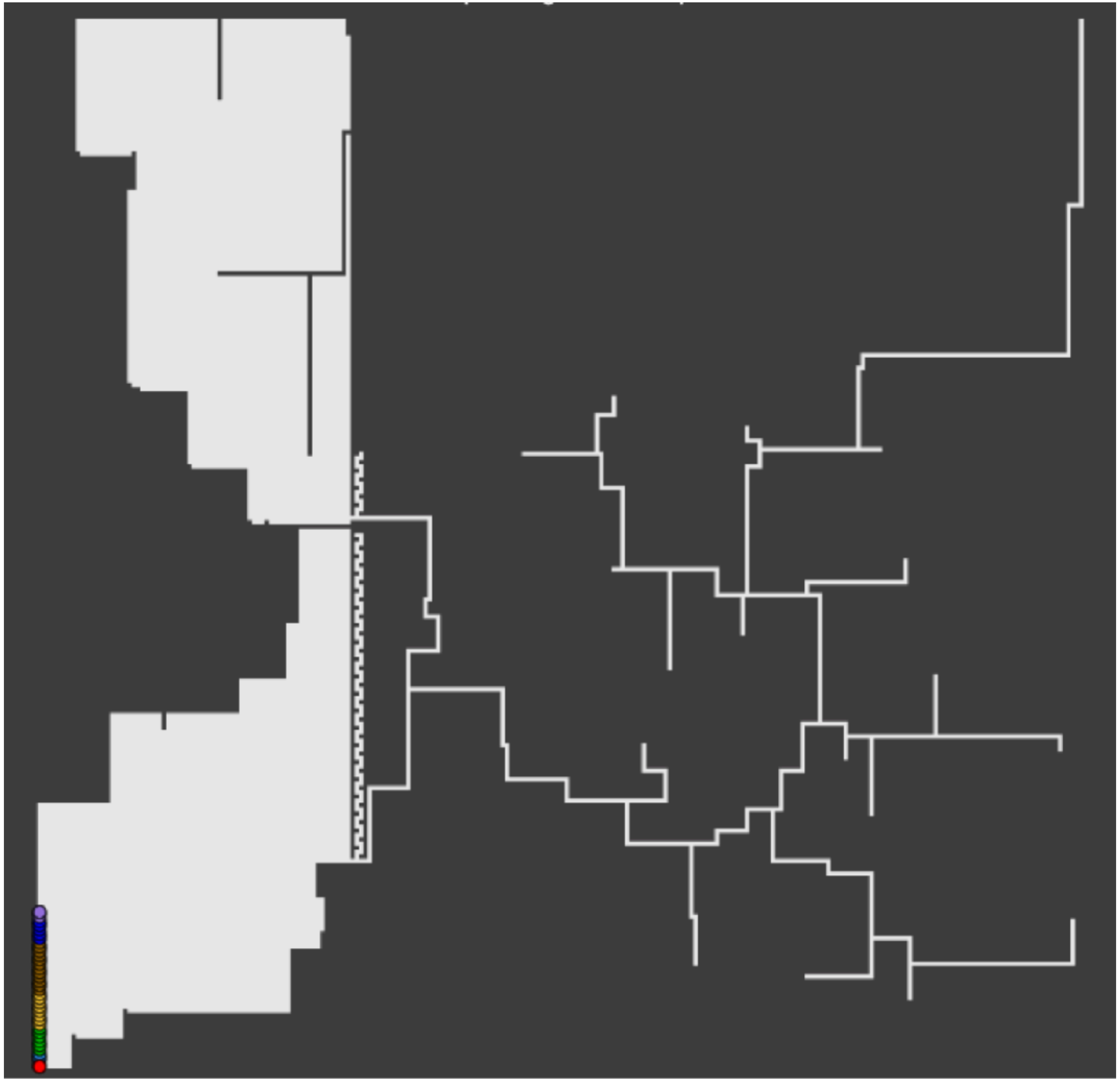
50
40
30
20
10
0
-10
0
20
40
60
80
100
120
total_subs
good_subs

Contributed Task Maps:

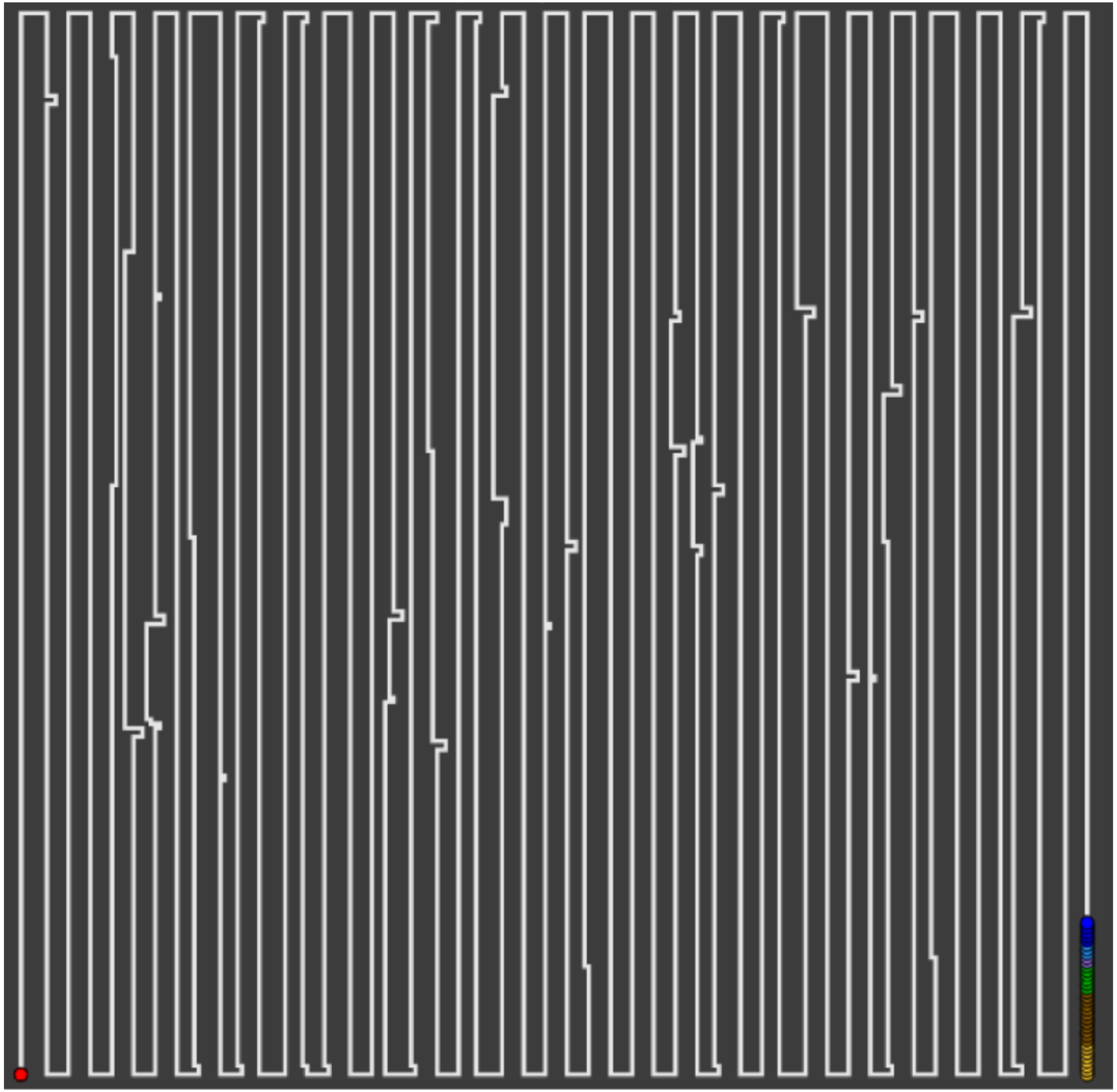
Expectations vs Reality



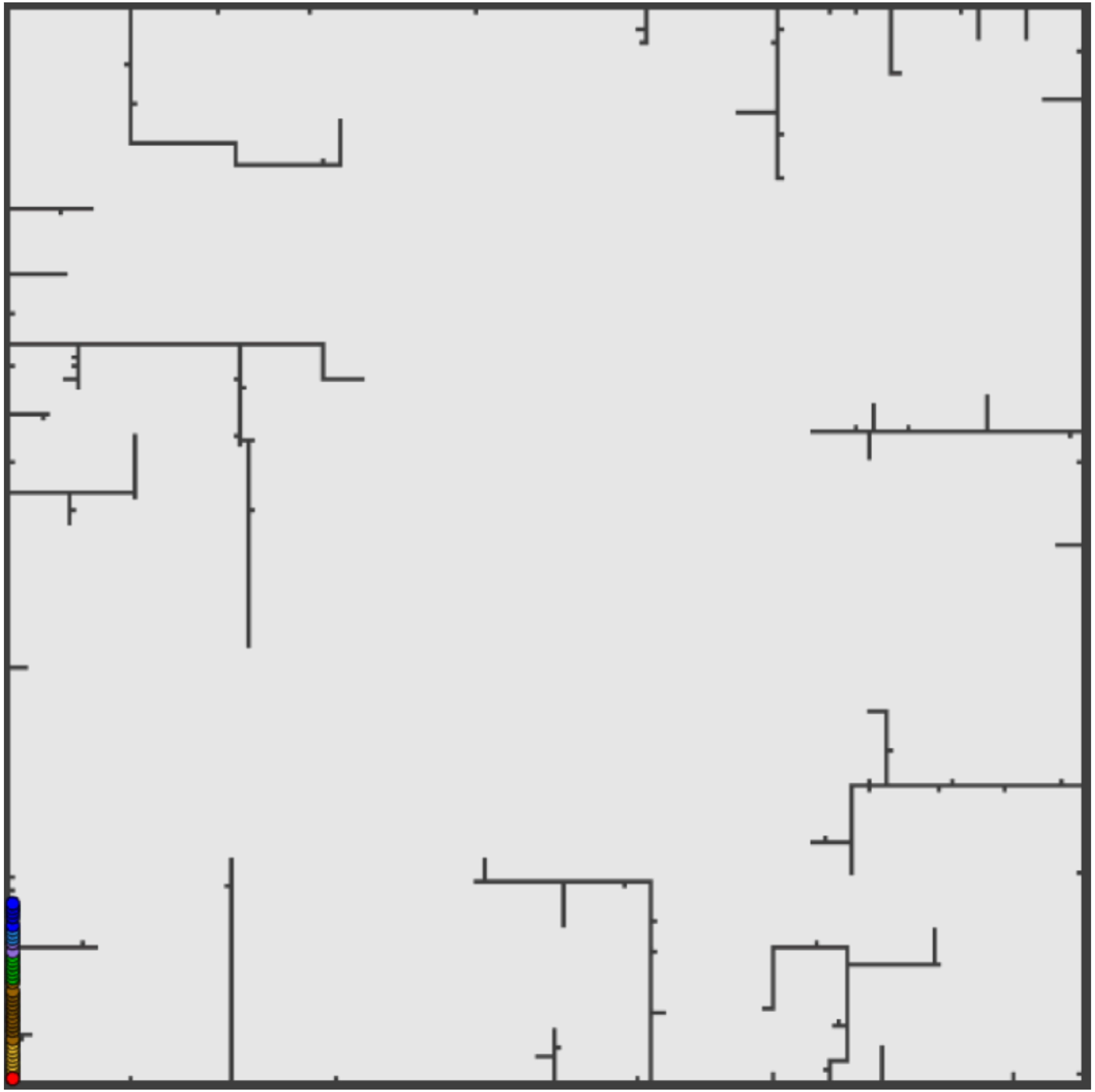
A possible proposal
for Block #40 task



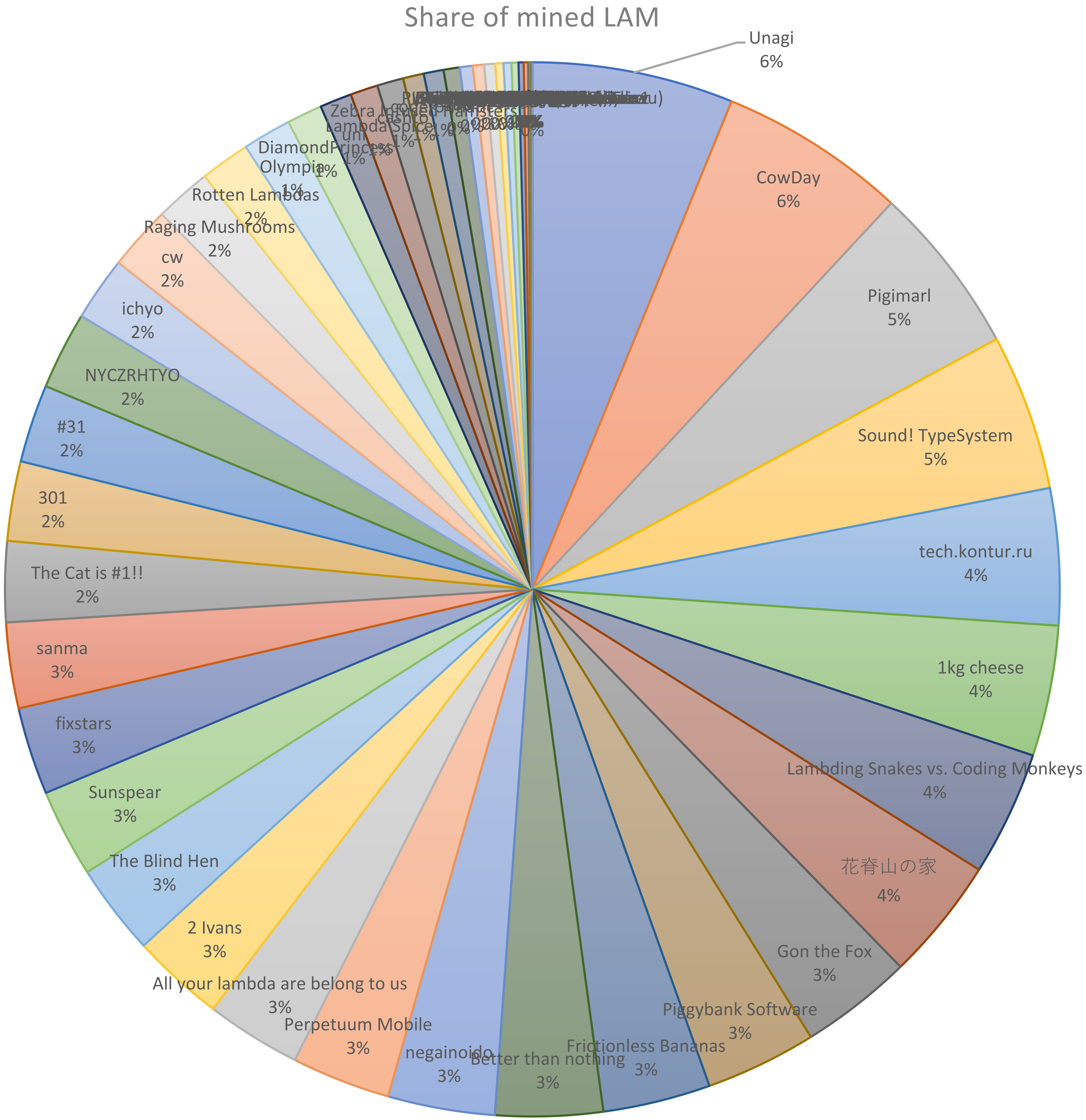
An actual proposal for Block #40 task



An actual proposal
for Block #40 task



Accepted proposal
for Block #40 task


Share of mined LAM
Unagi 6%
CowDay 6%
Pigimarl 5%
Sound! TypeSystem 5%
tech.kontur.ru 4%
1kg cheese 4%
Lambding Snakes vs. Coding Monkeys 4%
花脊山の家 4%
Gon the Fox 3%
Piggybank Software 3%
Frictionless Bananas 3%
Better than nothing 3%
negainoido 3%
Perpetuum Mobile 3%
All your lambda are belong to us 3%
2 Ivans 3%
The Blind Hen 3%
Sunspear 3%
fixstars 3%
sanma 3%
The Cat is #1!! 2%
301 2%
#31 2%
NYCZRHTYO 2%
ichyo 2%
cw 2%
Raging Mushrooms 2%
Rotten Lambdas 2%
Olympia 1%
DiamondPrincess 1%
Zebra Infused 1%
Lambda Spice 1%
uni 1%
cashto 1%
codeo 0%
Hamsters 0%
0%

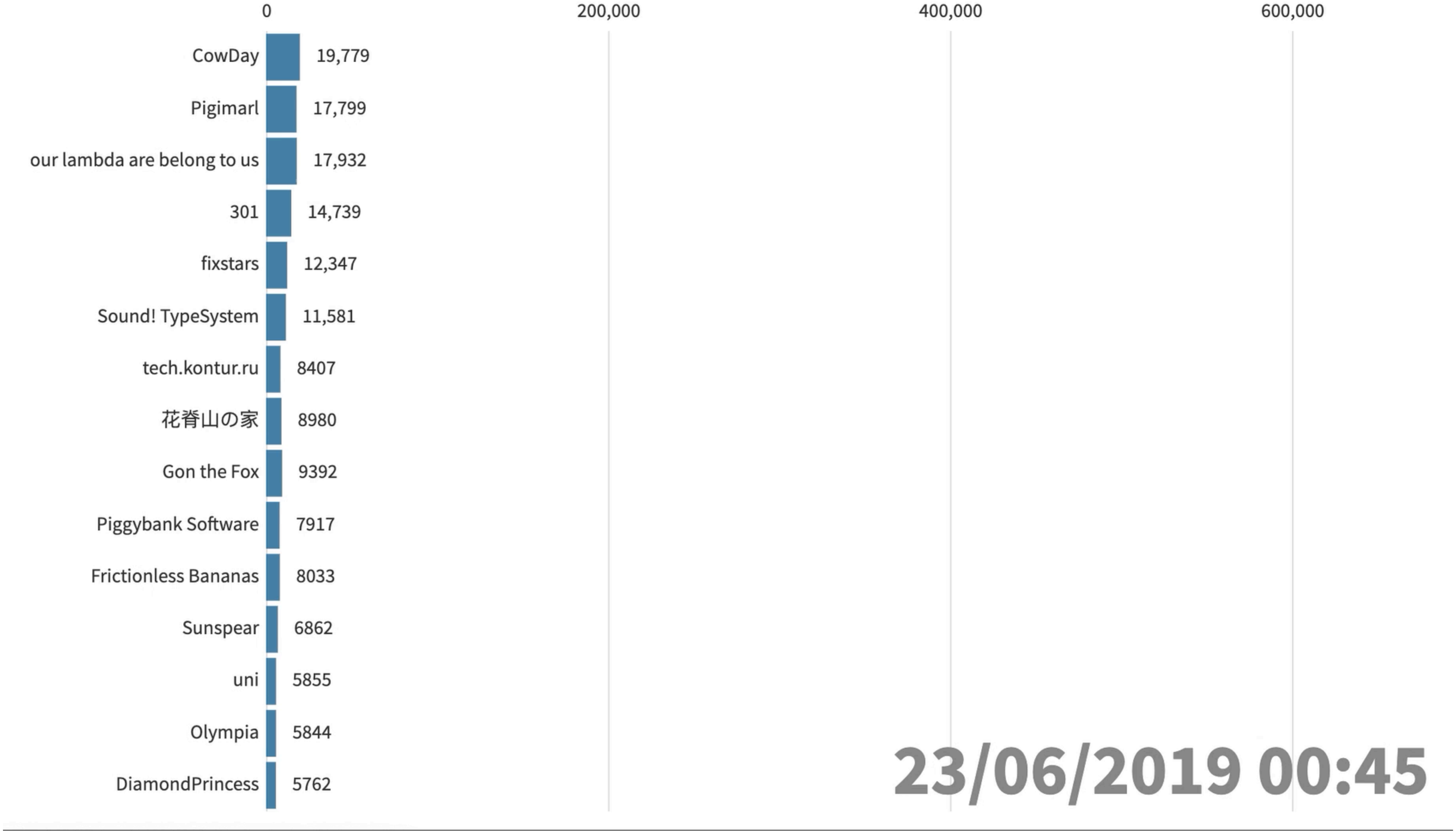
HODLing or Spending?

With additional boosters
it’s easier to become the best!

$$score_{team,T} \triangleq \left\lceil 1000 \times \log_2 \left(X_T \times Y_T \right) \times \frac{\boxed{t_{best}}}{t_{team}} \right\rceil$$

- Unspent LAM are *added* to the final scores (1 to 1)
  - With an absolute score maximum **4,219,436** this is a substantial addition
- However, LAM spent wisely can *reduce* the scores of the competition!

https://public.flourish.studio/visualisation/602197/

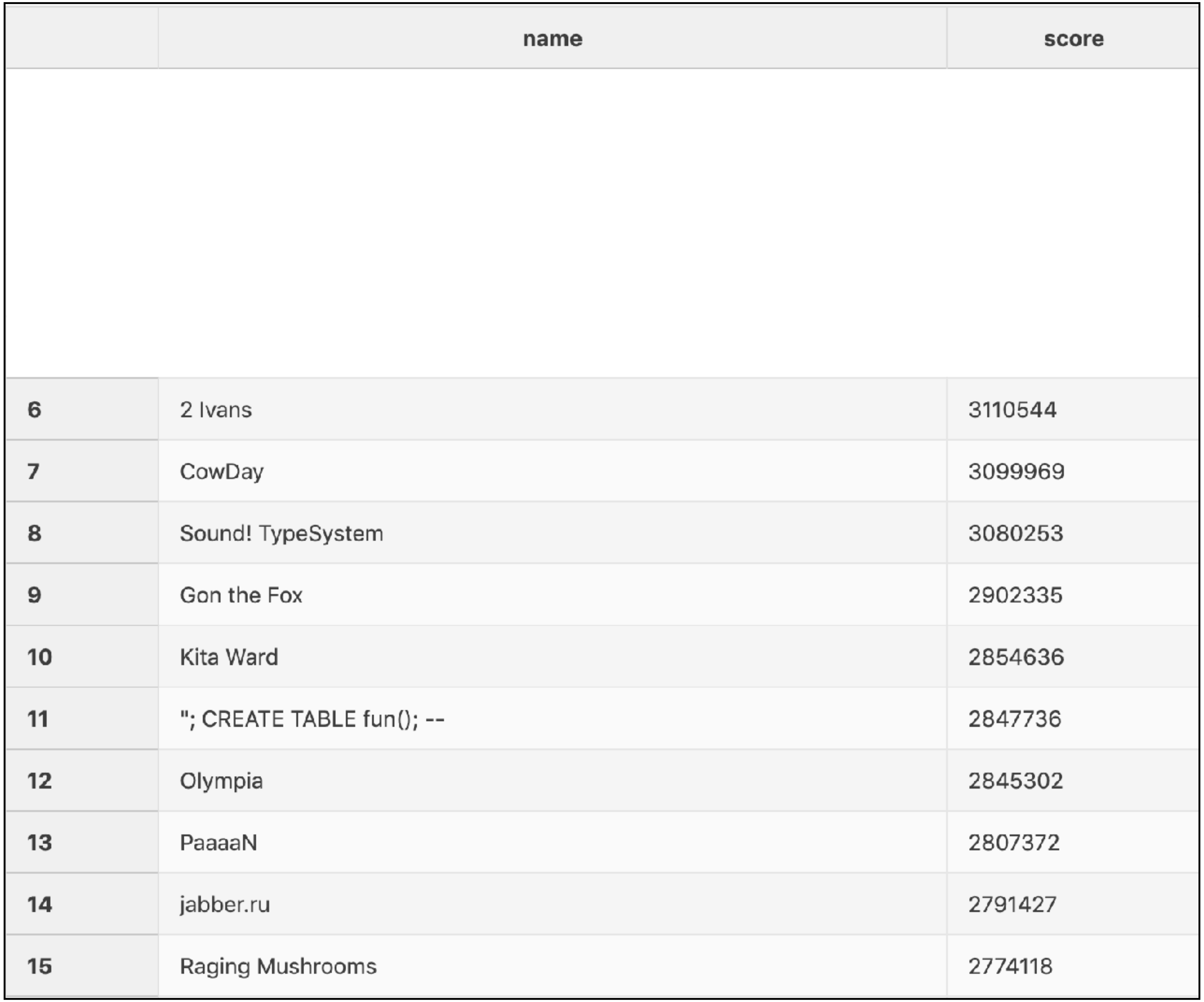


Spending Lambda-coins

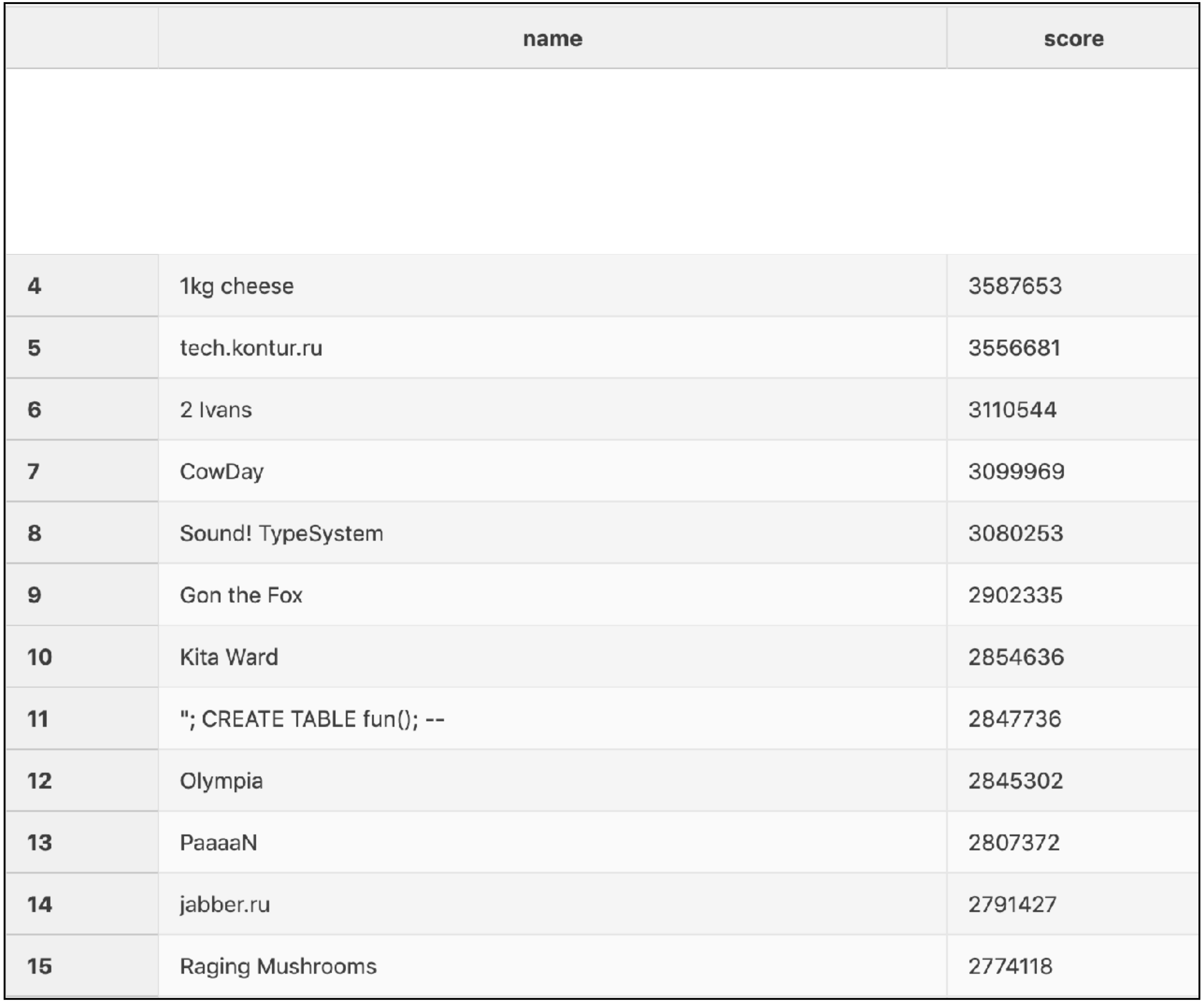
- Of **12,136,480** LAM mined…
- **10,047,700** LAM have been spent
- **99.15%** of them were spent on *clones*

Results

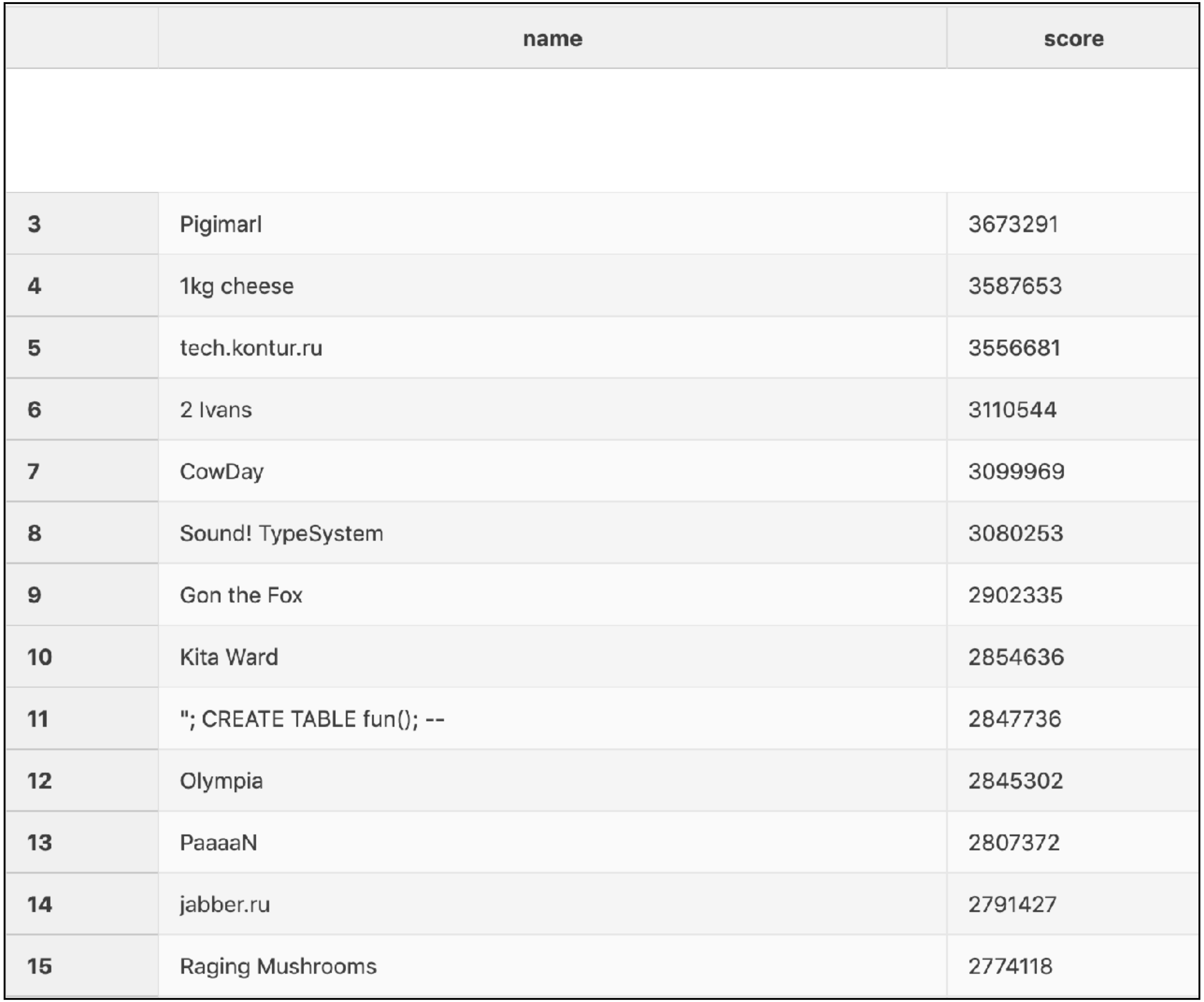
Lightning Division



	name	score
6	2 Ivans	3110544
7	CowDay	3099969
8	Sound! TypeSystem	3080253
9	Gon the Fox	2902335
10	Kita Ward	2854636
11	"; CREATE TABLE fun(); --	2847736
12	Olympia	2845302
13	PaaaaN	2807372
14	jabber.ru	2791427
15	Raging Mushrooms	2774118

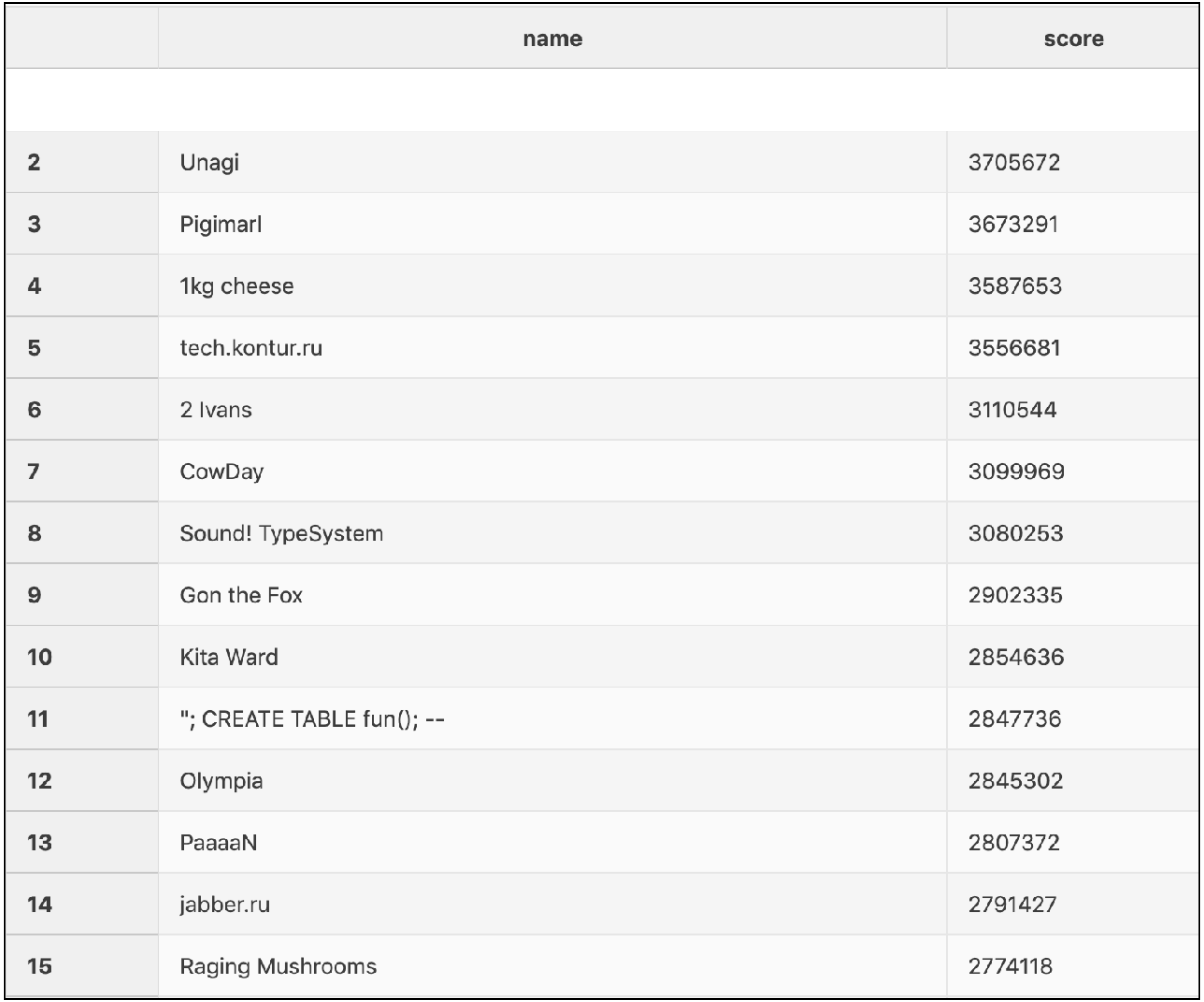


	name	score
4	1kg cheese	3587653
5	tech.kontur.ru	3556681
6	2 Ivans	3110544
7	CowDay	3099969
8	Sound! TypeSystem	3080253
9	Gon the Fox	2902335
10	Kita Ward	2854636
11	"; CREATE TABLE fun(); --	2847736
12	Olympia	2845302
13	PaaaaN	2807372
14	jabber.ru	2791427
15	Raging Mushrooms	2774118

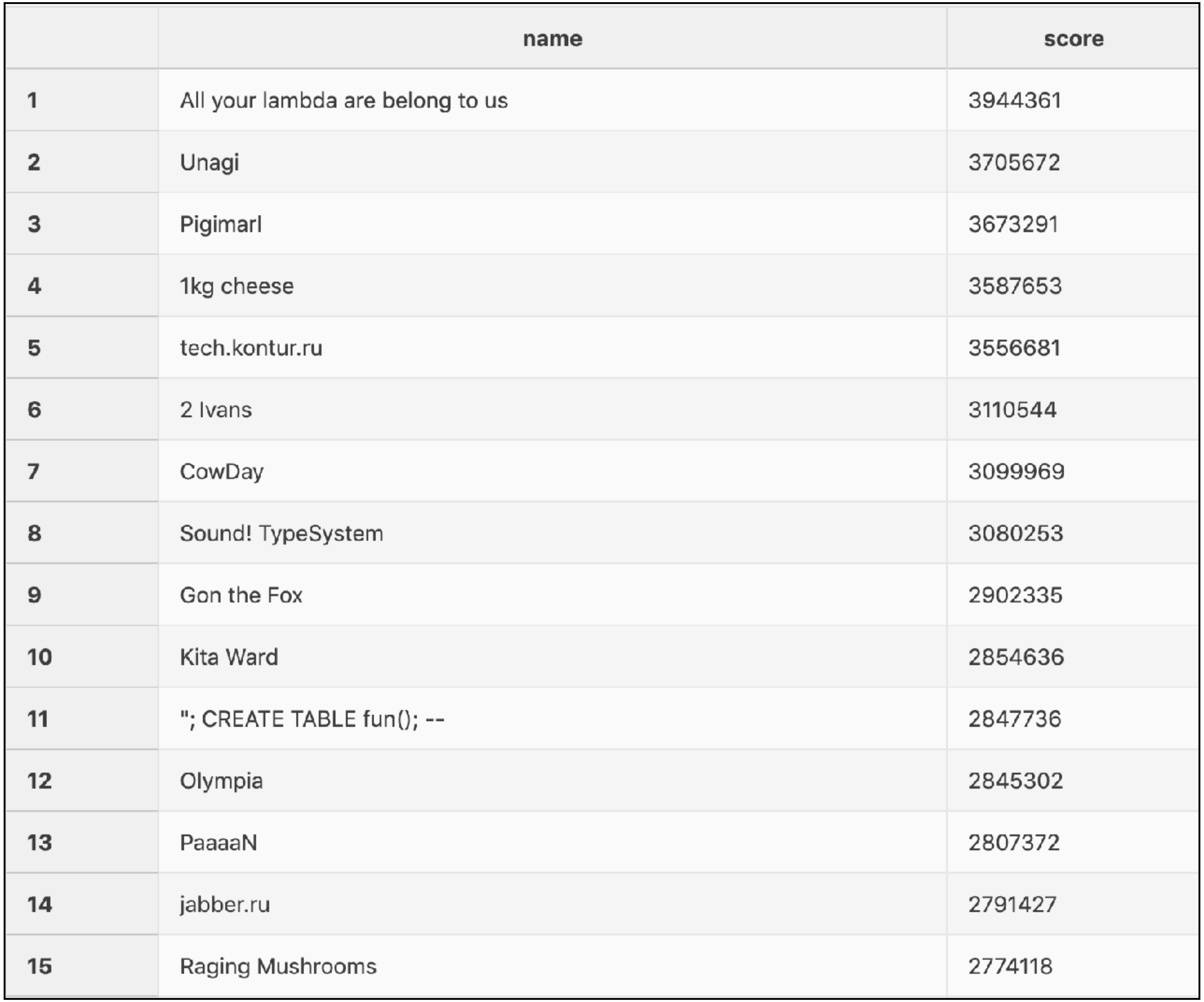


	name	score
3	Pigimarl	3673291
4	1kg cheese	3587653
5	tech.kontur.ru	3556681
6	2 Ivans	3110544
7	CowDay	3099969
8	Sound! TypeSystem	3080253
9	Gon the Fox	2902335
10	Kita Ward	2854636
11	"; CREATE TABLE fun(); --	2847736
12	Olympia	2845302
13	PaaaaN	2807372
14	jabber.ru	2791427
15	Raging Mushrooms	2774118

Winner of the Lightning Division



	name	score
2	Unagi	3705672
3	Pigimarl	3673291
4	1kg cheese	3587653
5	tech.kontur.ru	3556681
6	2 Ivans	3110544
7	CowDay	3099969
8	Sound! TypeSystem	3080253
9	Gon the Fox	2902335
10	Kita Ward	2854636
11	"; CREATE TABLE fun(); --	2847736
12	Olympia	2845302
13	PaaaaN	2807372
14	jabber.ru	2791427
15	Raging Mushrooms	2774118



	name	score
1	All your lambda are belong to us	3944361
2	Unagi	3705672
3	Pigimarl	3673291
4	1kg cheese	3587653
5	tech.kontur.ru	3556681
6	2 Ivans	3110544
7	CowDay	3099969
8	Sound! TypeSystem	3080253
9	Gon the Fox	2902335
10	Kita Ward	2854636
11	"; CREATE TABLE fun(); --	2847736
12	Olympia	2845302
13	PaaaaN	2807372
14	jabber.ru	2791427
15	Raging Mushrooms	2774118

Team

All your lambda are belong to us

Rafaël Bocquet

***C++** and **Haskell** are very suitable for rapid prototyping.*

$500 cash prize

Team “All your lambda are belong to us”

Rafaël Bocquet

Two programming languages:

- C++ (Solver)
- Haskell (Scripts)

Three external solvers:

- aspino (Maximum satisfiability problem)
- LKH (Travelling salesman problem)
- Gurobi (Mixed-integer linear programming)

Solution outline

Main idea: Without boosters and manipulators, the problem would be a variant of the Hamiltonian path problem.

- Step 1 Collect boosters.
- Step 2 Find a covering of the map by the robot shape.
- Step 3 Find a Hamiltonian path in the induced graph.
- Step 4 Insert rotations in the Hamiltonian path.
- Step 5 Split the solution between clones.
- Step 6 Local optimization ($\sim$ 10% improvement).

Solution outline

Main idea: Without boosters and manipulators, the problem would be a variant of the Hamiltonian path problem.

Step 1 Collect boosters. **(LKH)**

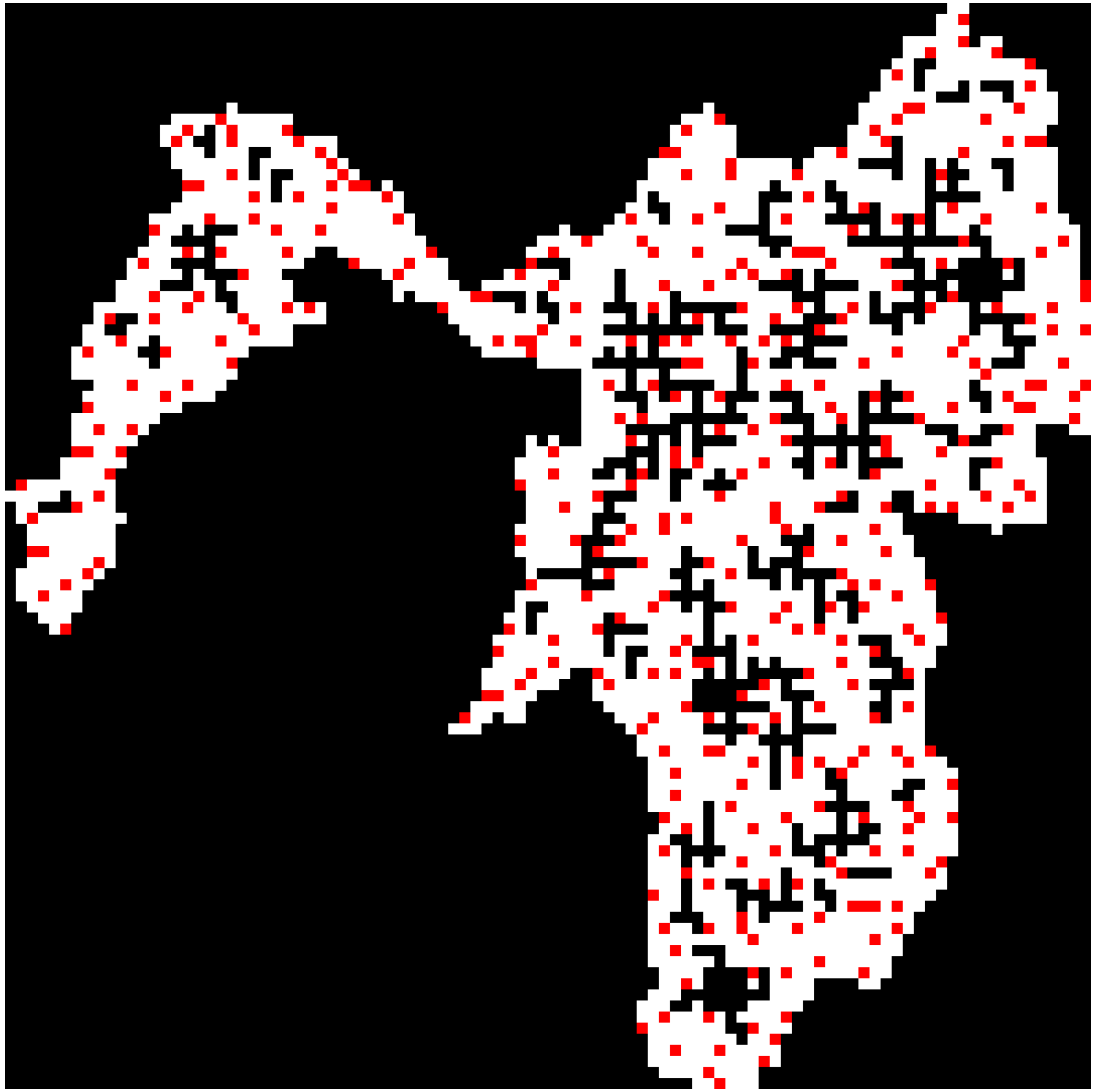
Step 2 Find a covering of the map by the robot shape. **(aspino)**

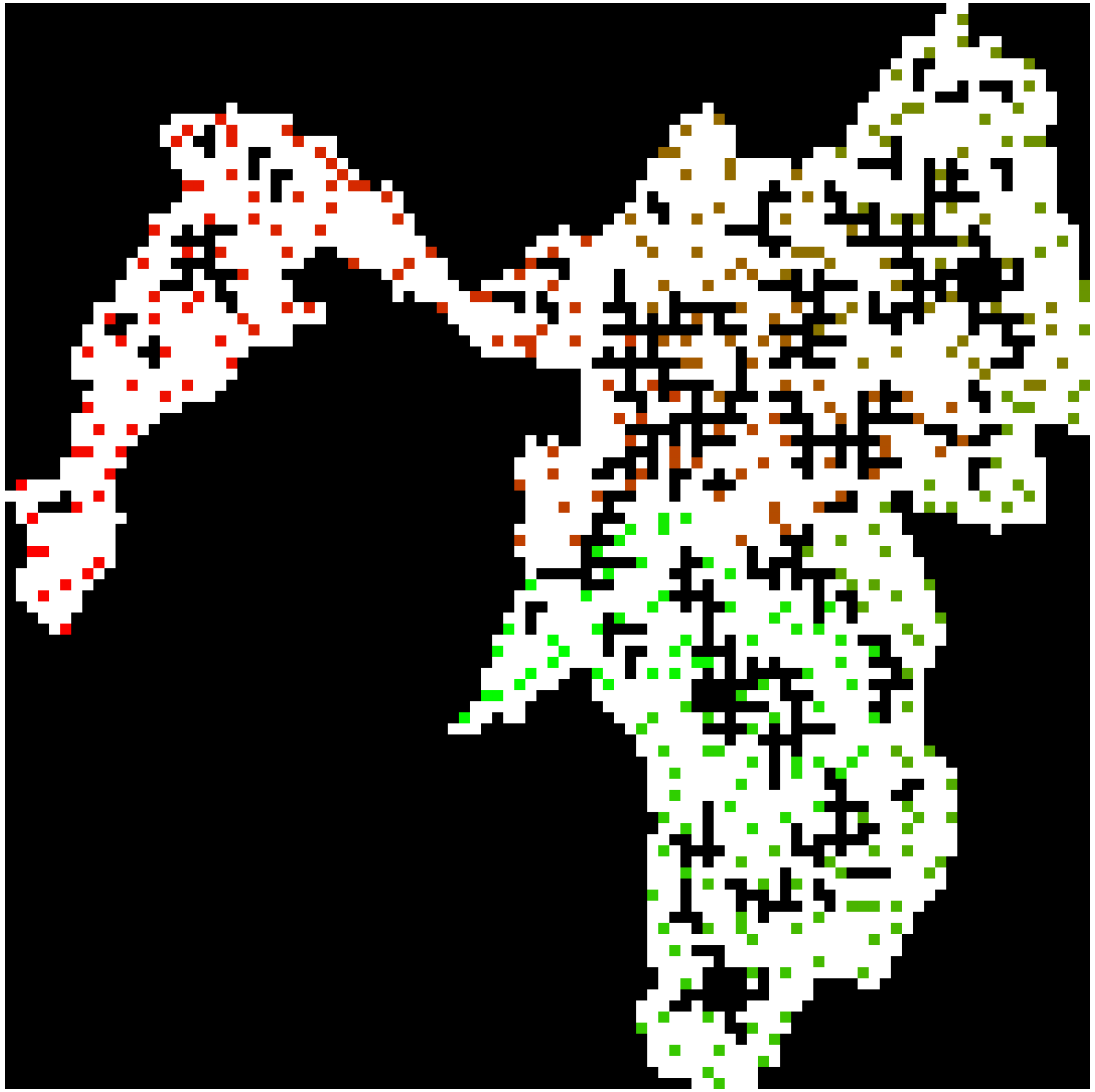
Step 3 Find a Hamiltonian path in the induced graph. **(LKH)**

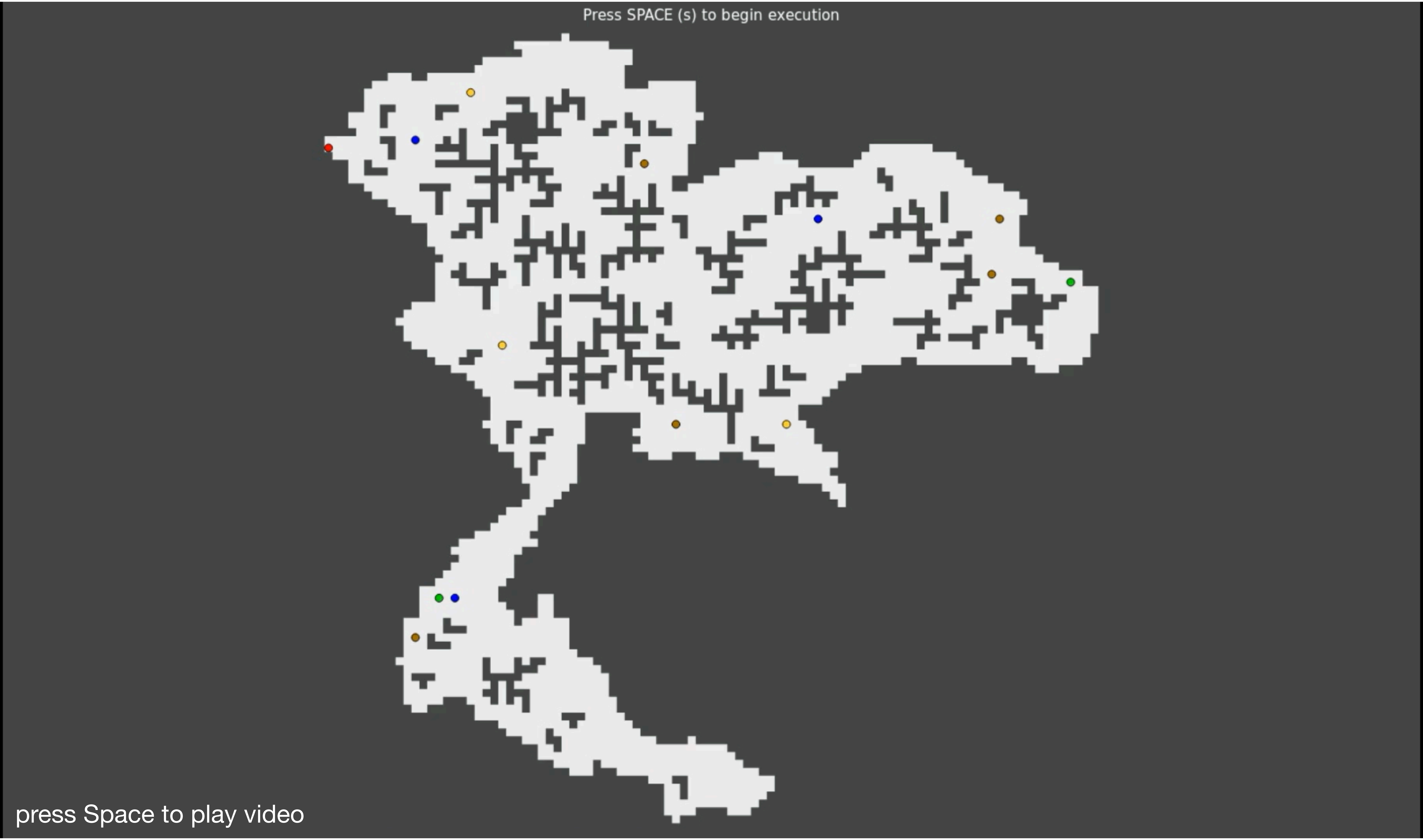
Step 4 Insert rotations in the Hamiltonian path. **(aspino)**

Step 5 Split the solution between clones. **(Gurobi)**

Step 6 Local optimization ($\sim$ 10% improvement).



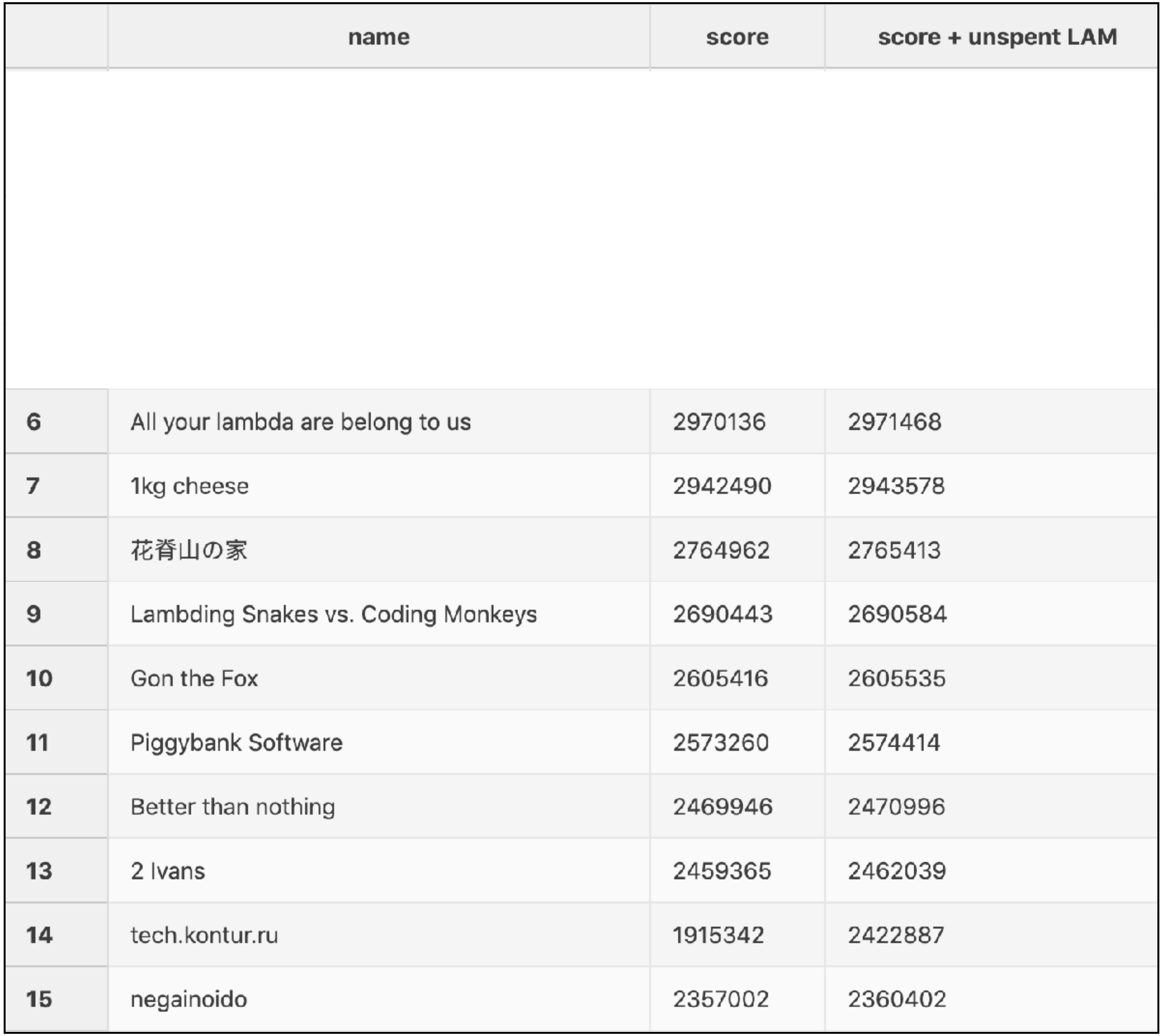




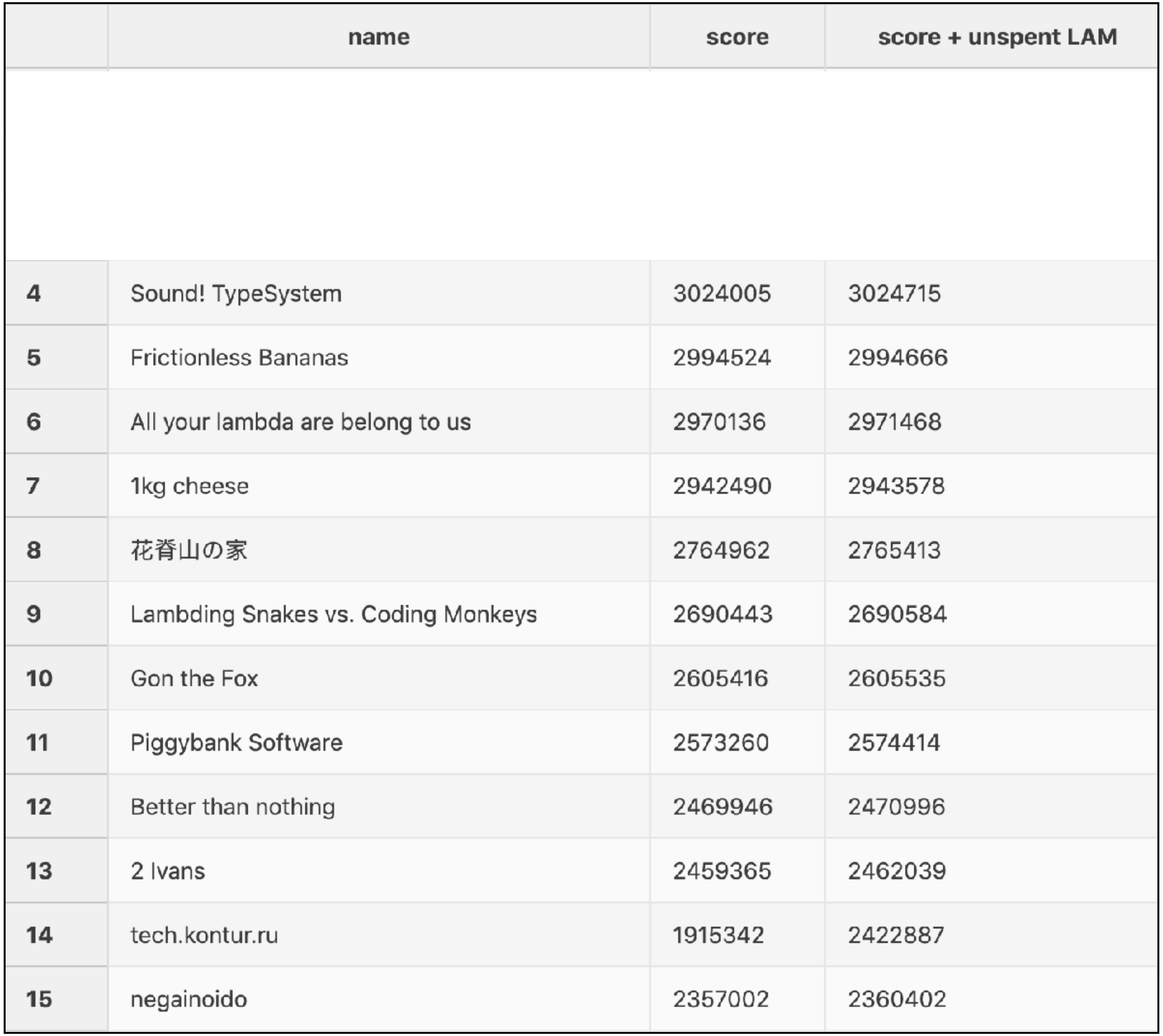
Press SPACE (s) to begin execution

press Space to play video

Main Division



	name	score	score + unspent LAM
6	All your lambda are belong to us	2970136	2971468
7	1kg cheese	2942490	2943578
8	花脊山の家	2764962	2765413
9	Lambding Snakes vs. Coding Monkeys	2690443	2690584
10	Gon the Fox	2605416	2605535
11	Piggybank Software	2573260	2574414
12	Better than nothing	2469946	2470996
13	2 Ivans	2459365	2462039
14	tech.kontur.ru	1915342	2422887
15	negainoido	2357002	2360402



	name	score	score + unspent LAM
4	Sound! TypeSystem	3024005	3024715
5	Frictionless Bananas	2994524	2994666
6	All your lambda are belong to us	2970136	2971468
7	1kg cheese	2942490	2943578
8	花脊山の家	2764962	2765413
9	Lambding Snakes vs. Coding Monkeys	2690443	2690584
10	Gon the Fox	2605416	2605535
11	Piggybank Software	2573260	2574414
12	Better than nothing	2469946	2470996
13	2 Ivans	2459365	2462039
14	tech.kontur.ru	1915342	2422887
15	negainoido	2357002	2360402

Judges’ Prize

for the most elegant use of the entire set of boosters

Team

Sound! TypeSystem

bakaming, chiro, fuqinho, gusmachine, nya3jp,
phoenixstarhiro, shunsakuraba, tanakh, yuizumi

*Team **Sound! Type System***
*(using **Rust, C++, Python, JavaScript, and Go**)*
is an extremely cool bunch of hackers.

$500 cash prize

Sound! TypeSystem

Sound! TypeSystem: Members

kaming

ro

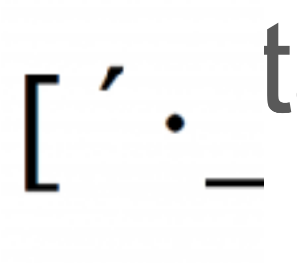
jinho

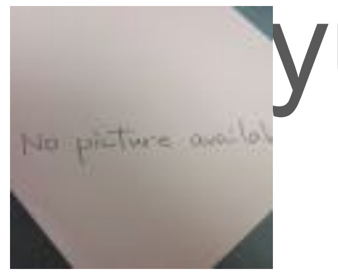
smachine

nya3jp

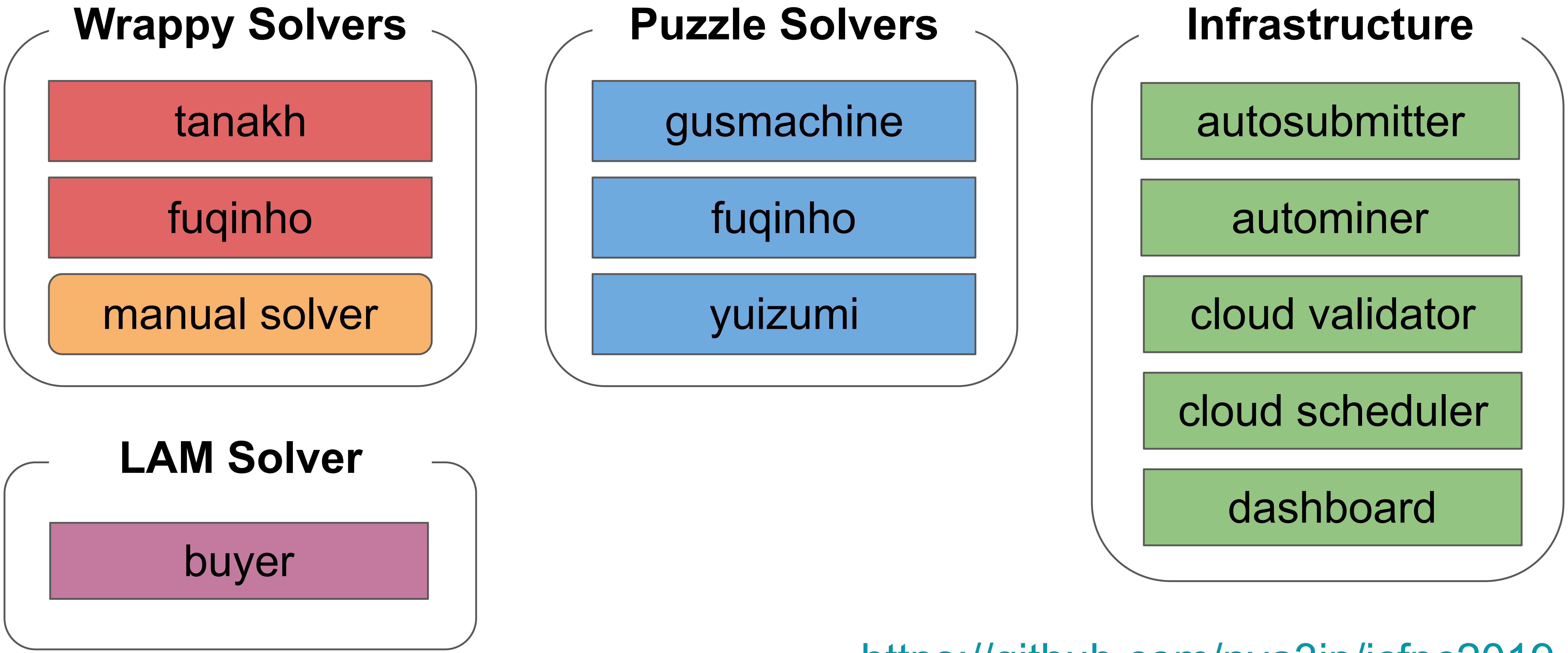
phoenixstarhiro

chunjp


tanakh


yuizumi

Sound! TypeSystem: Overview



Wrappy Solvers

- tanakh
- fuqinho
- manual solver

Puzzle Solvers

- gusmachine
- fuqinho
- yuizumi

Infrastructure

- autosubmitter
- autominer
- cloud validator
- cloud scheduler
- dashboard

LAM Solver

- buyer

https://github.com/nya3jp/icfpc2019

Sound! TypeSystem: Main Solutions

Wrappy solver by tanakh

- Written in Rust
- Based on BFS with plenty of hacky heuristics:
  - Give the first priority to cloning.
  - Give priority to filling small isolated areas to paint.
  - Get clones well spread out.
  - Manipulators are expanded to the left and right.
  - Use randomness to "improve" the score.
  - Some parameters were determined using simulated annealing.

Sound! TypeSystem: Main Solutions

Wrappy solver by fuqinho

- Written in C++
- Considers all cells within 5 steps not to be painted by another wrappy.
- Calculates the size of "islands" (isolated areas to paint) for each cell; the smallest island takes the priority.
- Uses the beam search to determine the steps to cover the island and uses the number of painted cells, with heuristical weights (e.g. to give higher score to borders).

Sound! TypeSystem: Main Solutions

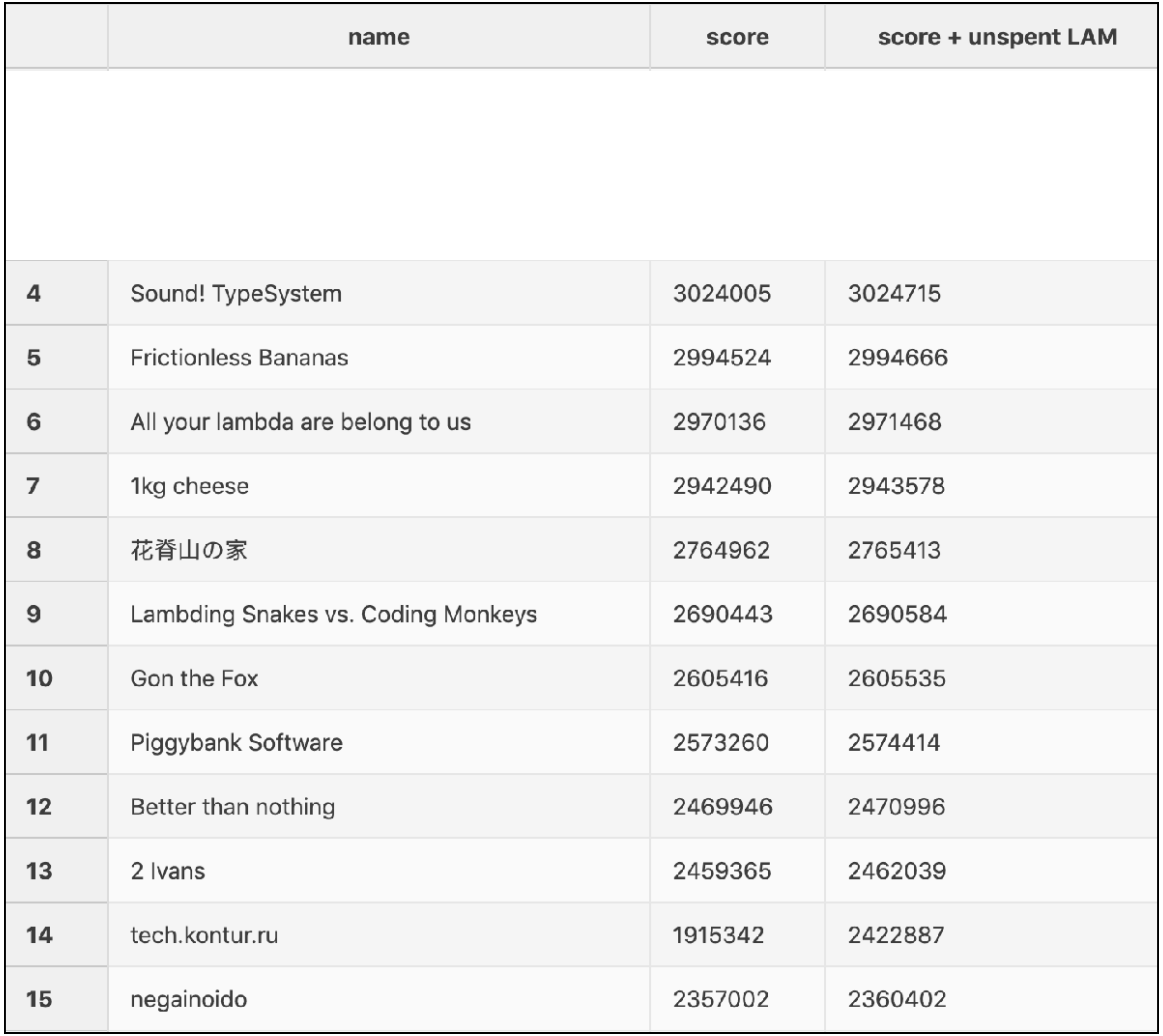
LAM solver by bakaming

- Written in C++
- Main solvers make solutions with several purchase patterns. LAM solver joins these solutions with earned LAM.
- We can solve this problem with dynamic programming as the number of LAM and solution is very small.

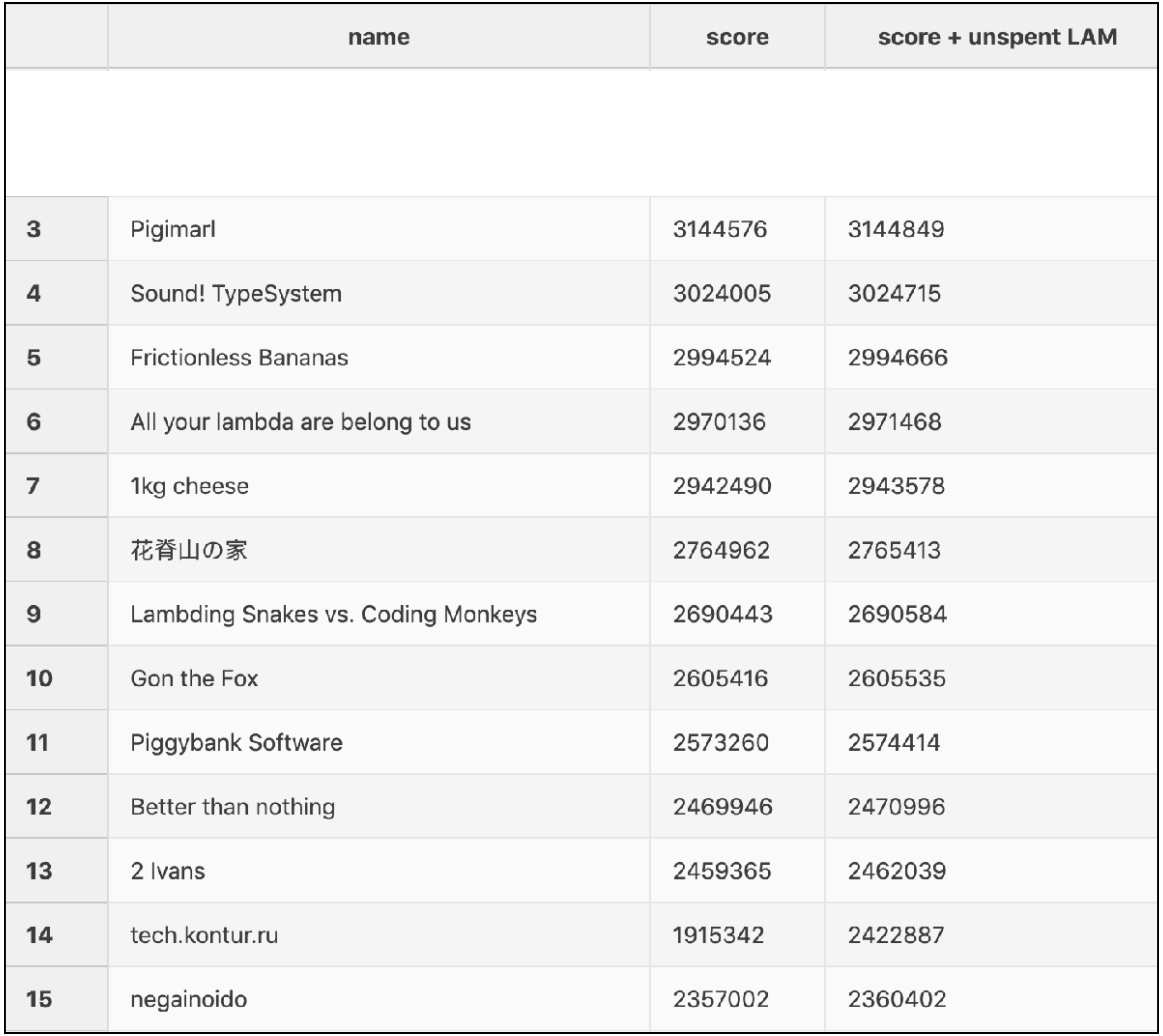
Sound! TypeSystem: Puzzle Solvers

Puzzle solvers by yuizumi / gusmachine

- Written in Python
- Kept the logic simple to make it work before the original first round starts in 4 hours.
  - Start from a vacant field,
  - build a wall from each *oSqs* until the wall reaches edges or other walls, and
  - put bumps on walls to satisfy *vMin* vertices condition.

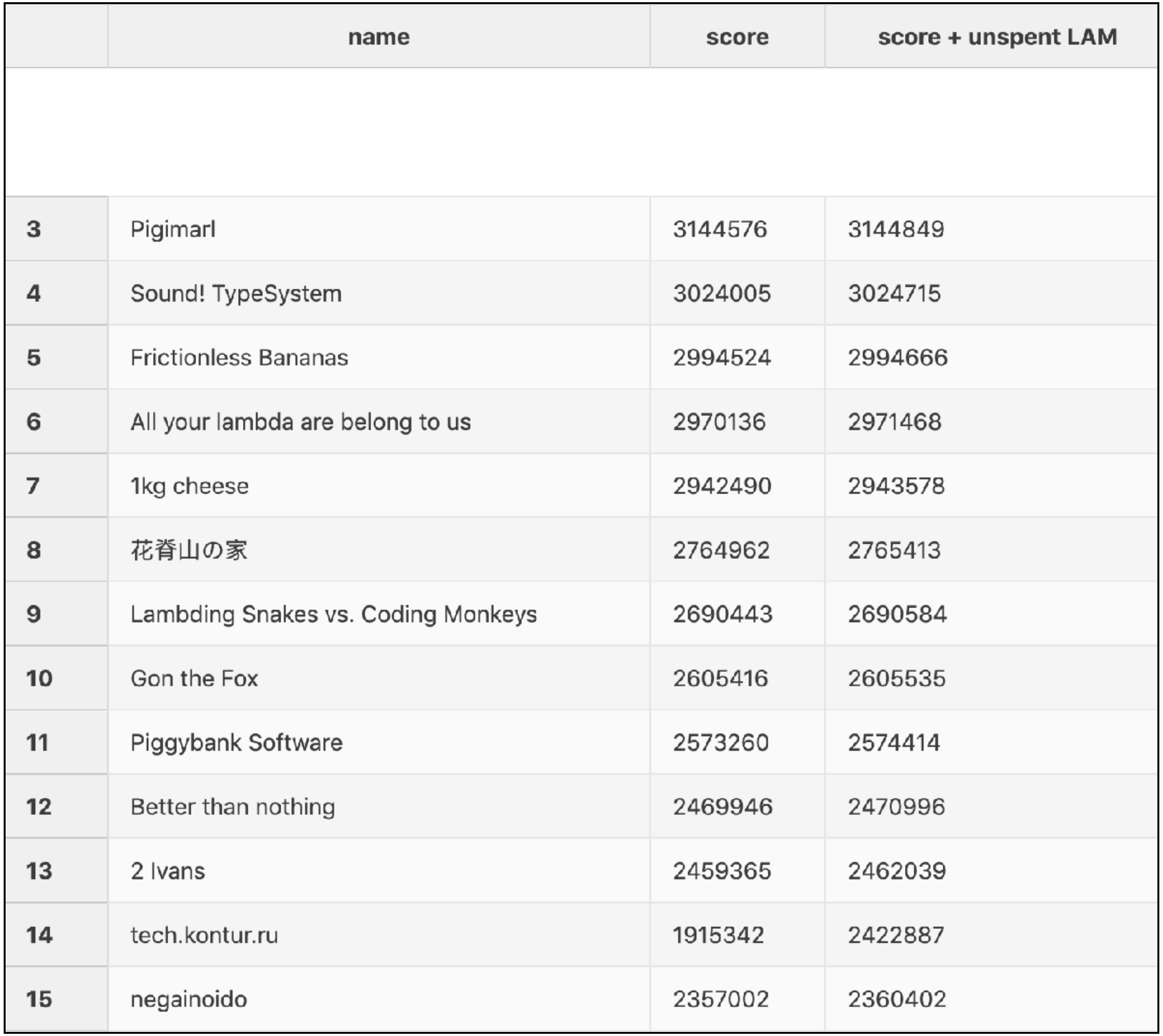


	name	score	score + unspent LAM
4	Sound! TypeSystem	3024005	3024715
5	Frictionless Bananas	2994524	2994666
6	All your lambda are belong to us	2970136	2971468
7	1kg cheese	2942490	2943578
8	花脊山の家	2764962	2765413
9	Lambding Snakes vs. Coding Monkeys	2690443	2690584
10	Gon the Fox	2605416	2605535
11	Piggybank Software	2573260	2574414
12	Better than nothing	2469946	2470996
13	2 Ivans	2459365	2462039
14	tech.kontur.ru	1915342	2422887
15	negainoido	2357002	2360402

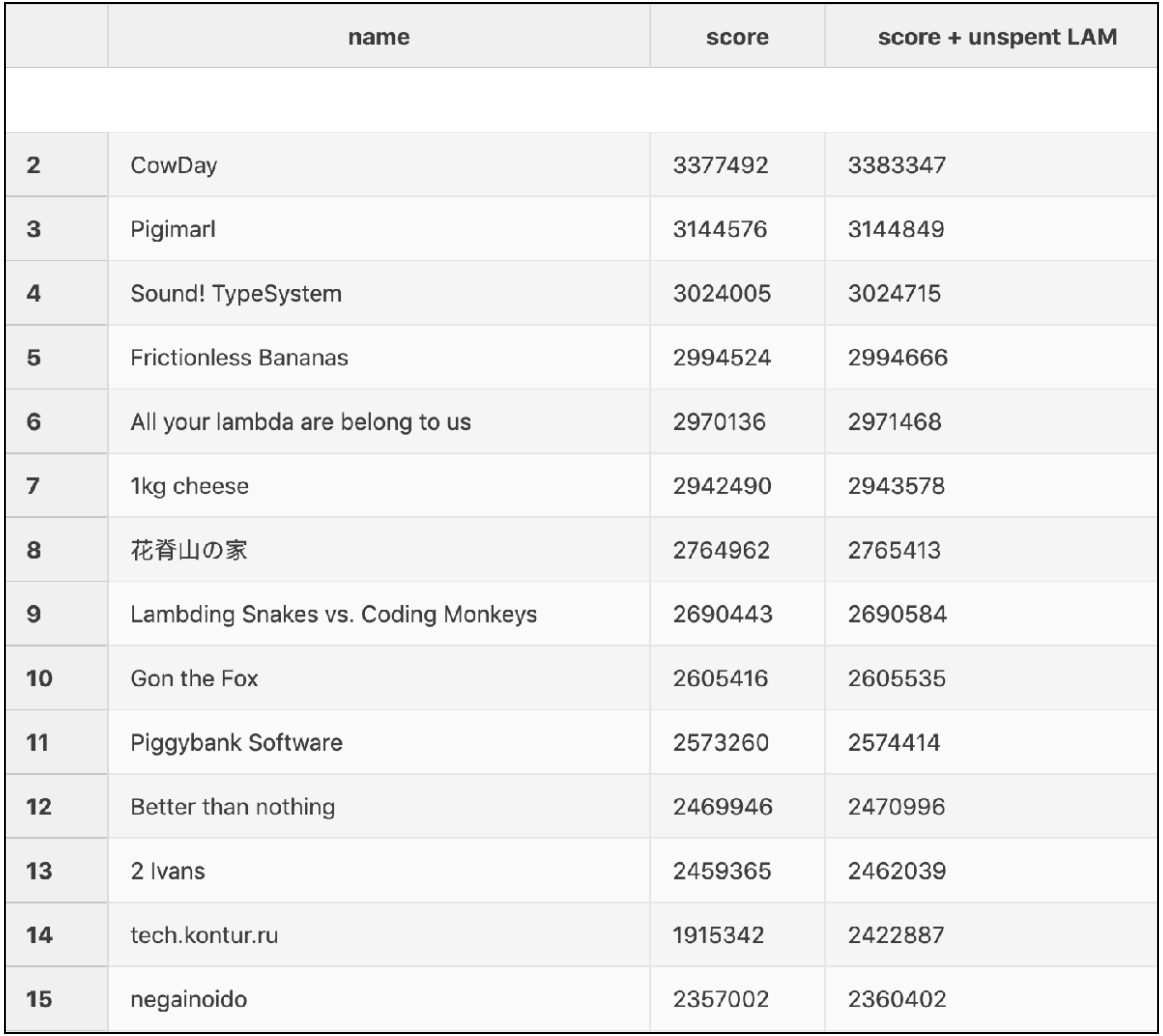


	name	score	score + unspent LAM
3	Pigimarl	3144576	3144849
4	Sound! TypeSystem	3024005	3024715
5	Frictionless Bananas	2994524	2994666
6	All your lambda are belong to us	2970136	2971468
7	1kg cheese	2942490	2943578
8	花脊山の家	2764962	2765413
9	Lambding Snakes vs. Coding Monkeys	2690443	2690584
10	Gon the Fox	2605416	2605535
11	Piggybank Software	2573260	2574414
12	Better than nothing	2469946	2470996
13	2 Ivans	2459365	2462039
14	tech.kontur.ru	1915342	2422887
15	negainoido	2357002	2360402

Second Prize



	name	score	score + unspent LAM
3	Pigimarl	3144576	3144849
4	Sound! TypeSystem	3024005	3024715
5	Frictionless Bananas	2994524	2994666
6	All your lambda are belong to us	2970136	2971468
7	1kg cheese	2942490	2943578
8	花脊山の家	2764962	2765413
9	Lambding Snakes vs. Coding Monkeys	2690443	2690584
10	Gon the Fox	2605416	2605535
11	Piggybank Software	2573260	2574414
12	Better than nothing	2469946	2470996
13	2 Ivans	2459365	2462039
14	tech.kontur.ru	1915342	2422887
15	negainoido	2357002	2360402



	name	score	score + unspent LAM
2	CowDay	3377492	3383347
3	Pigimarl	3144576	3144849
4	Sound! TypeSystem	3024005	3024715
5	Frictionless Bananas	2994524	2994666
6	All your lambda are belong to us	2970136	2971468
7	1kg cheese	2942490	2943578
8	花脊山の家	2764962	2765413
9	Lambding Snakes vs. Coding Monkeys	2690443	2690584
10	Gon the Fox	2605416	2605535
11	Piggybank Software	2573260	2574414
12	Better than nothing	2469946	2470996
13	2 Ivans	2459365	2462039
14	tech.kontur.ru	1915342	2422887
15	negainoido	2357002	2360402

Team
CowDay

Akifumi Imanishi, Kohei Morita, Kohji Liu, Nozomu Nakajima,
Riku Kawasaki, Takuto Shigemura, Seiya Kamiya

C++ *is a fine programming tool for many applications.*

$500 cash prize

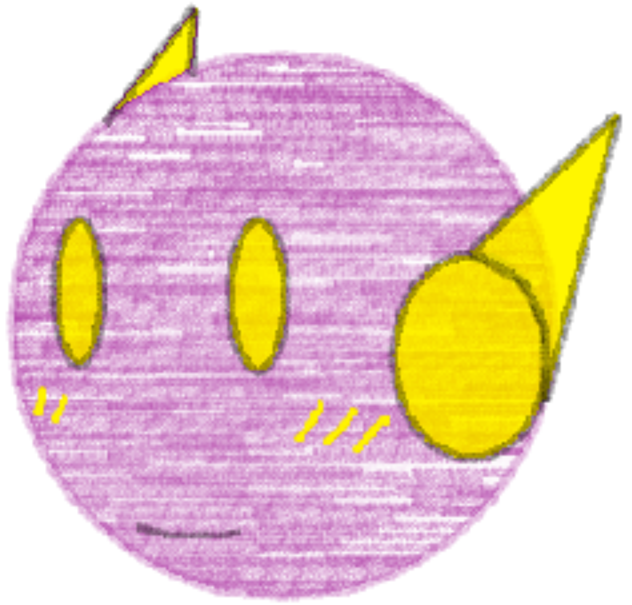
ICFP'19 Contest Cowday Solution

Cowday Team

Members

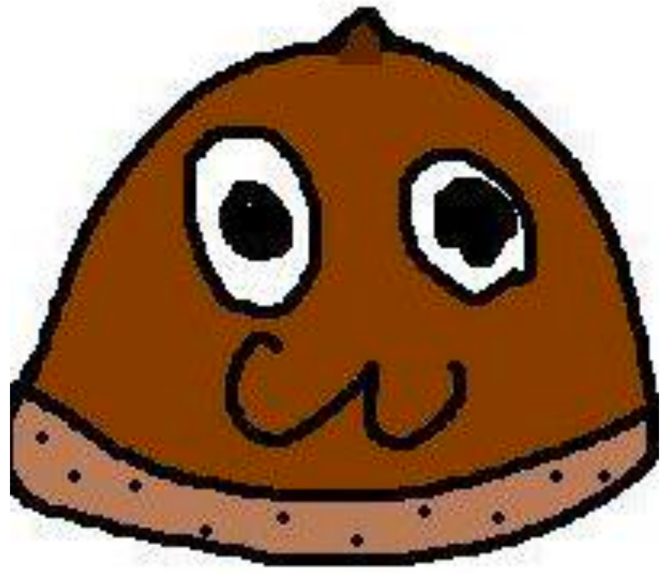
Infrastructure

Akifumi Imanishi



Seiya Kamiya

Algorithm



Riku Kawasaki

Nozomu Nakajima

Puzzle

Kohei Morita

Takuto Shigemura

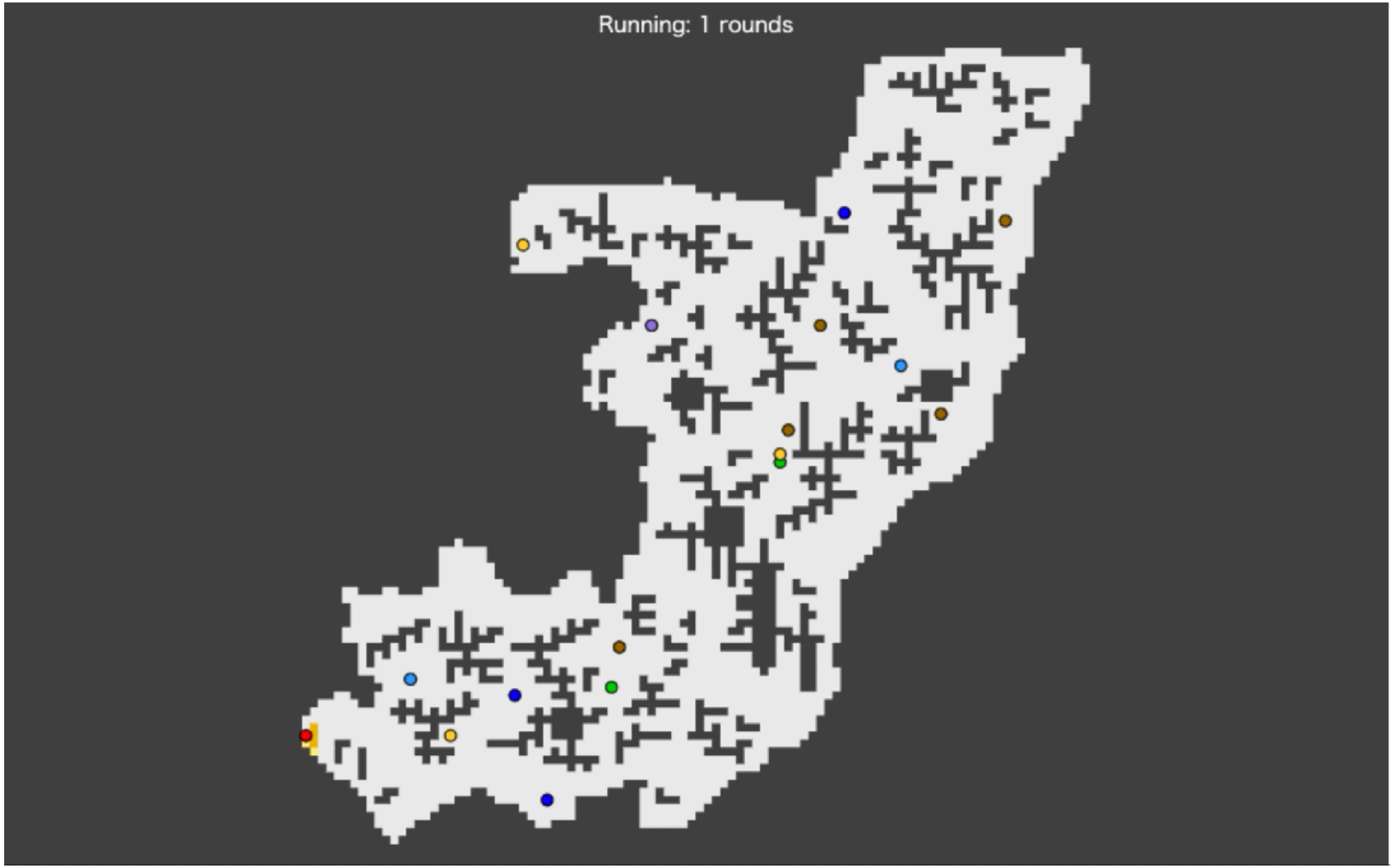
Coordinator(???)

Kohji Liu

3 Students, 4 employees

Algorithm-competition Fanatics

Visualization

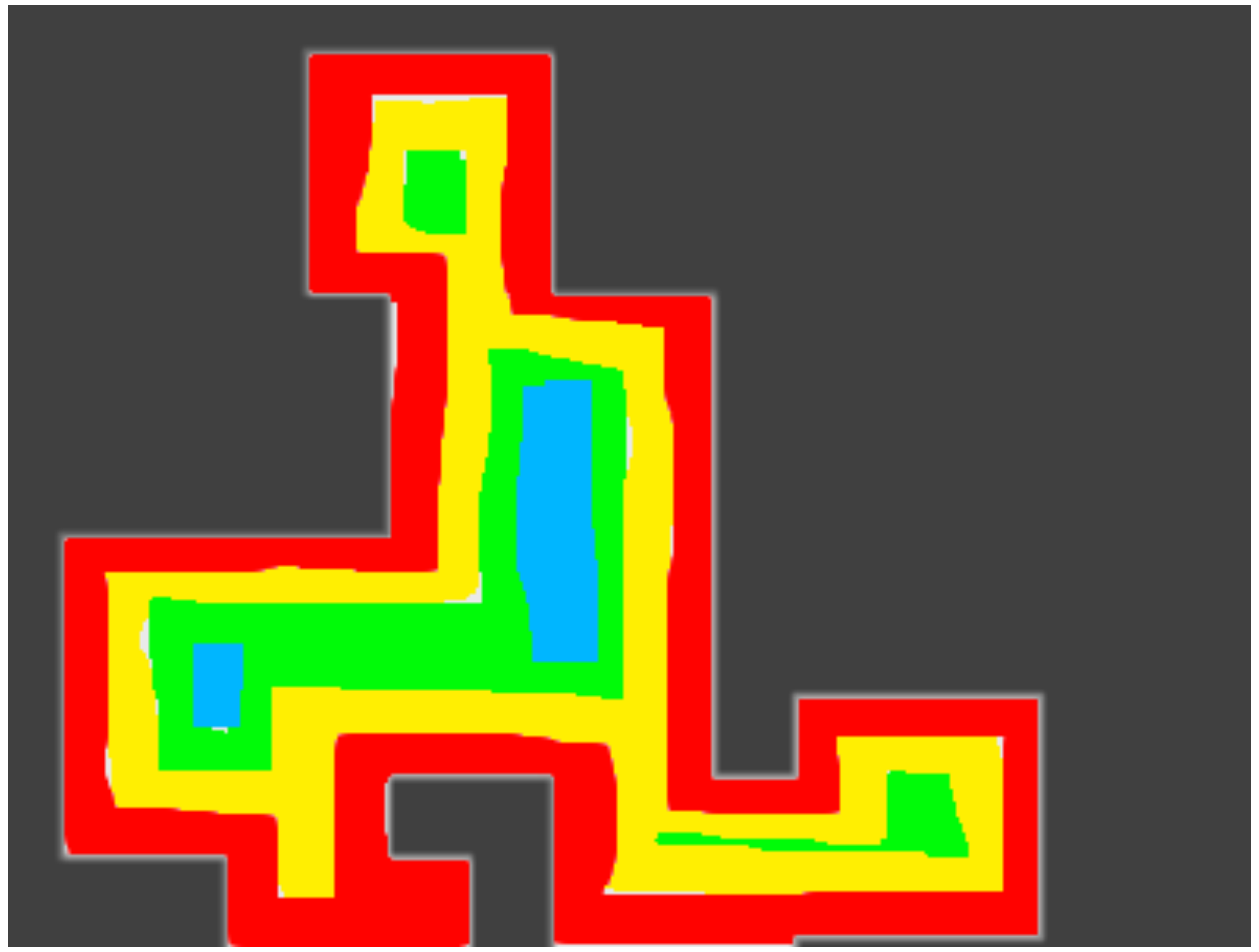


press Space to play

Solution

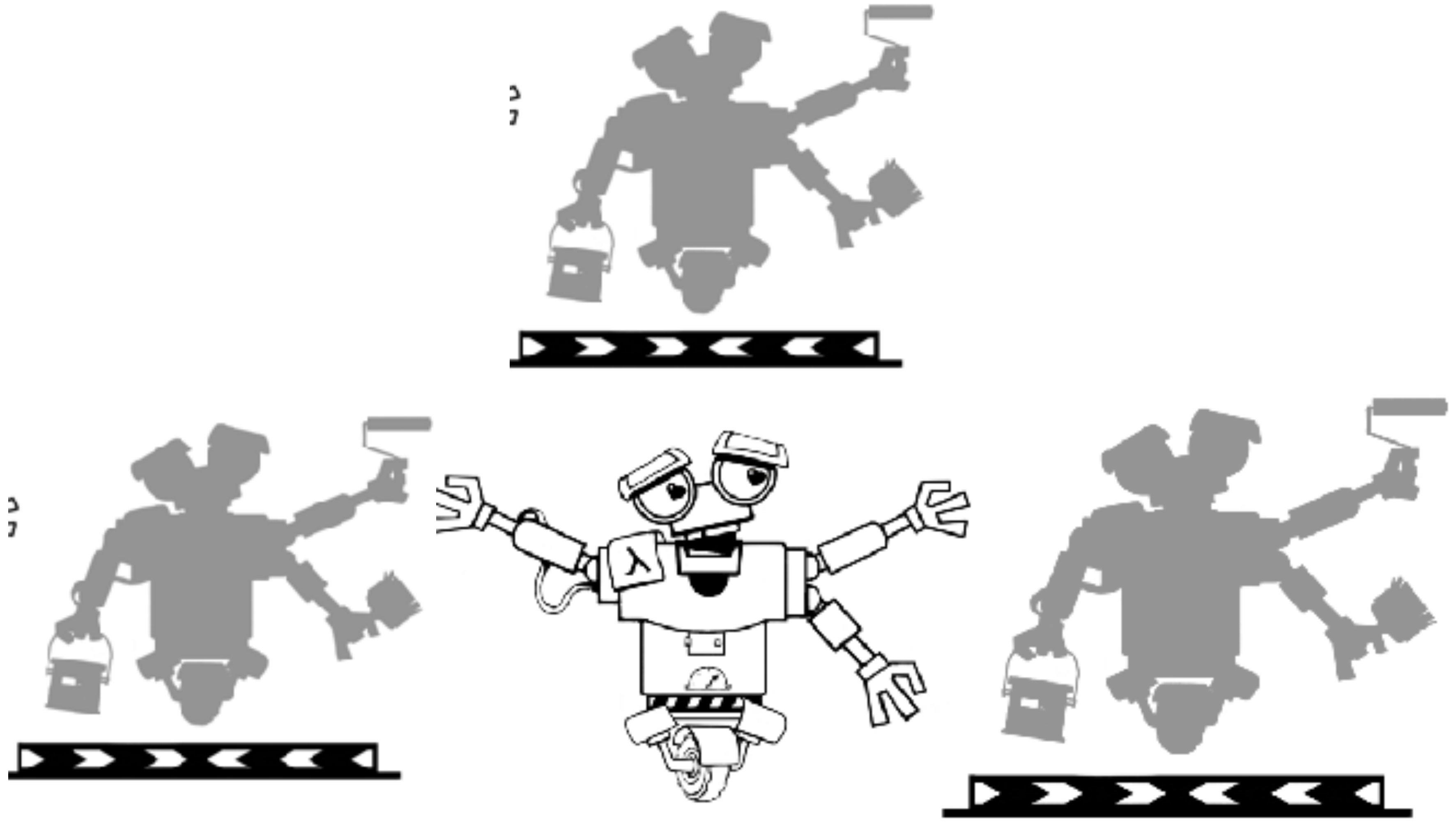
Preprocess

- Assign weight to each cell, according to the distance to an obstacle or a booster



Before filling area,
Collect Clone Items and Spawn

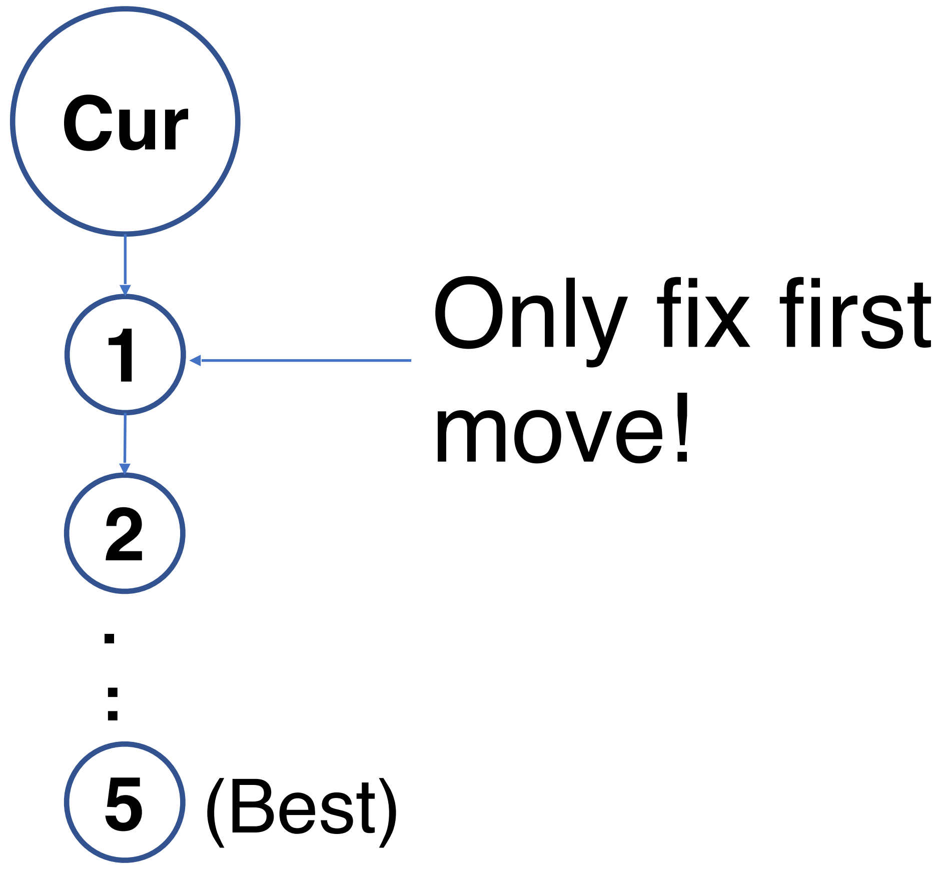
- Visit nearest cloner/spawn point greedily



Solution

Basis:
greedy-ish approach

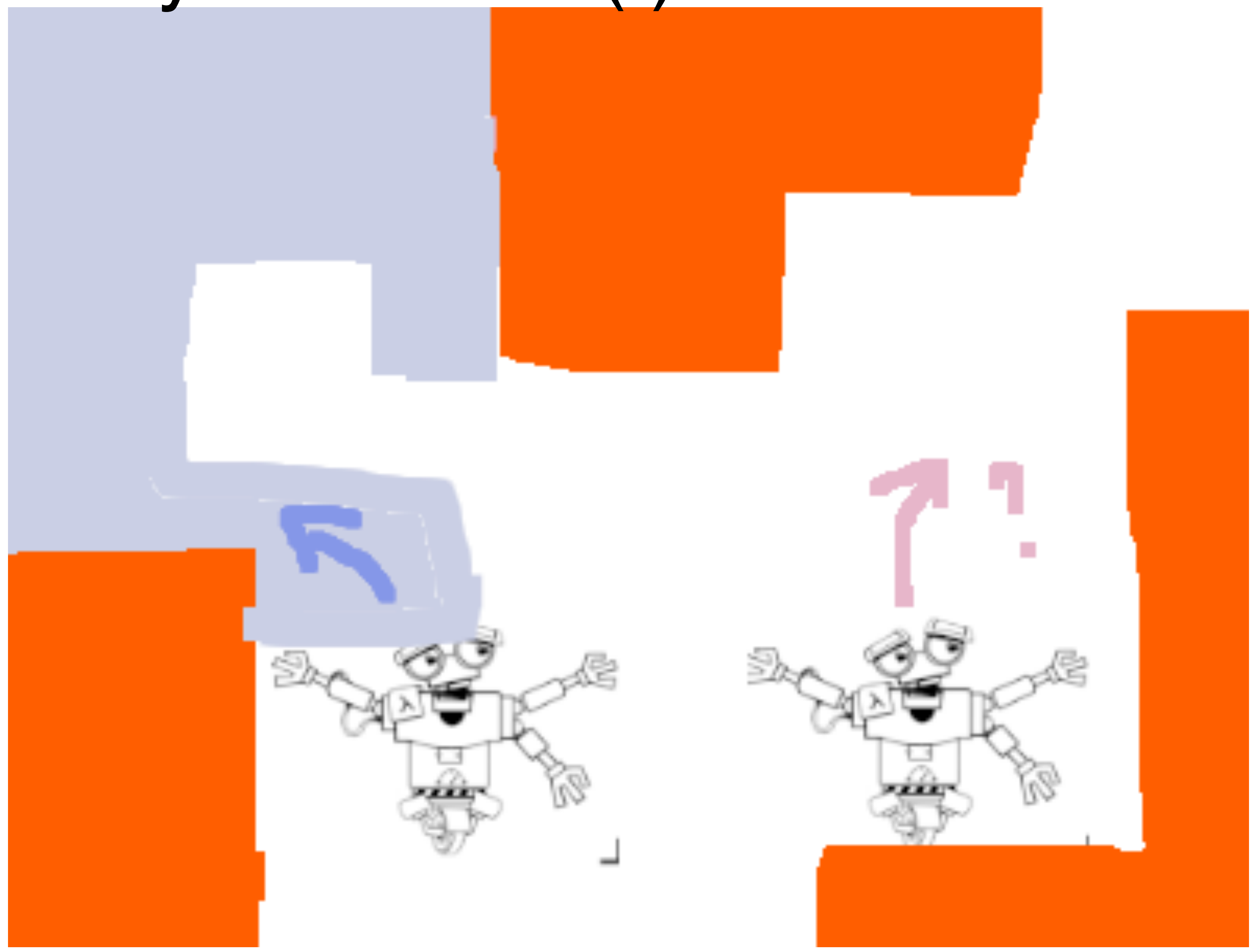
- Enumerate all possible next 5 steps and fix one step which leads to the largest weight in 5 steps, iteratively



To use multiple robots efficiently:

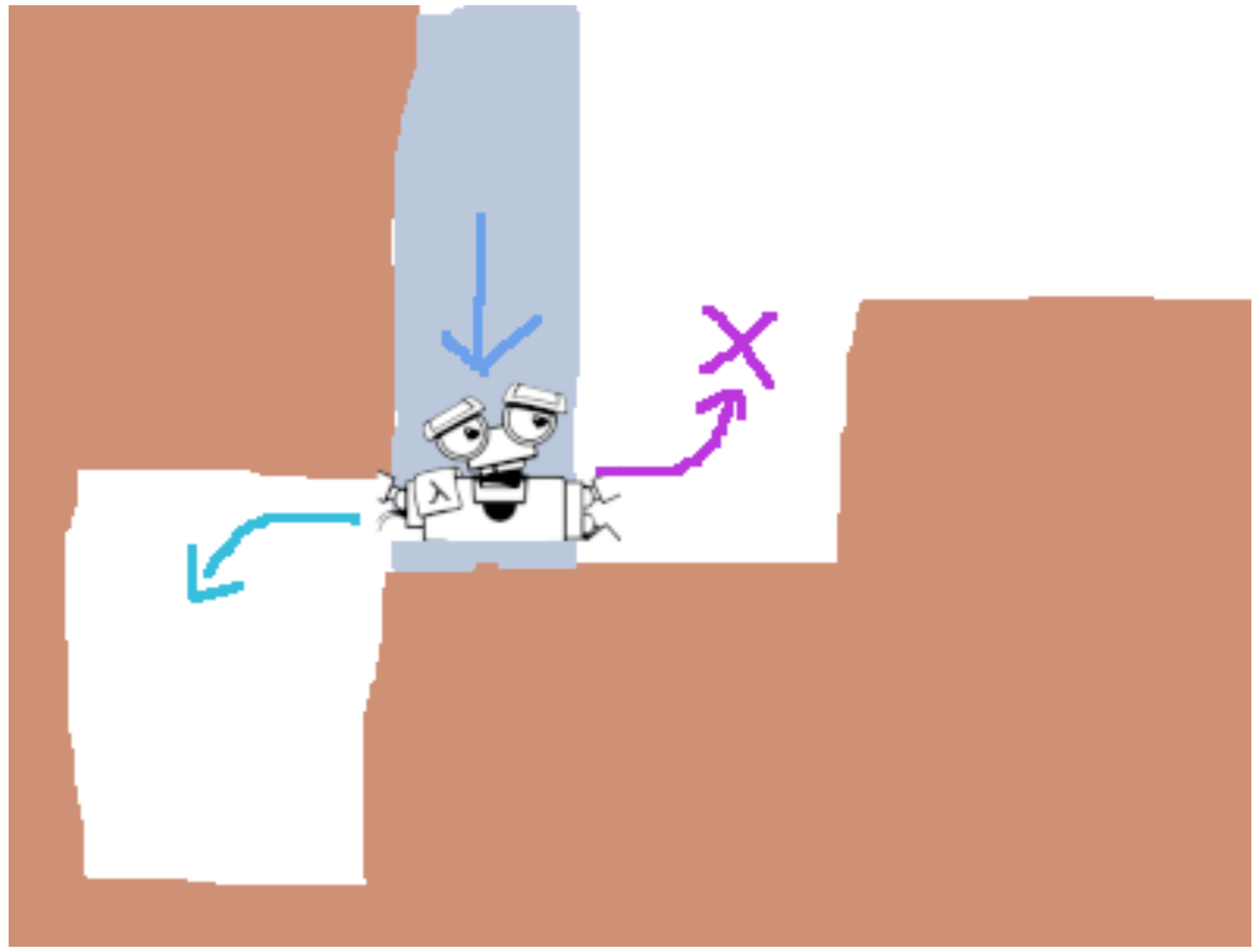
- Each robot moves 100 turns at one time

This way, our robots are asynchronous(!)



To avoid leaving small areas unfilled:

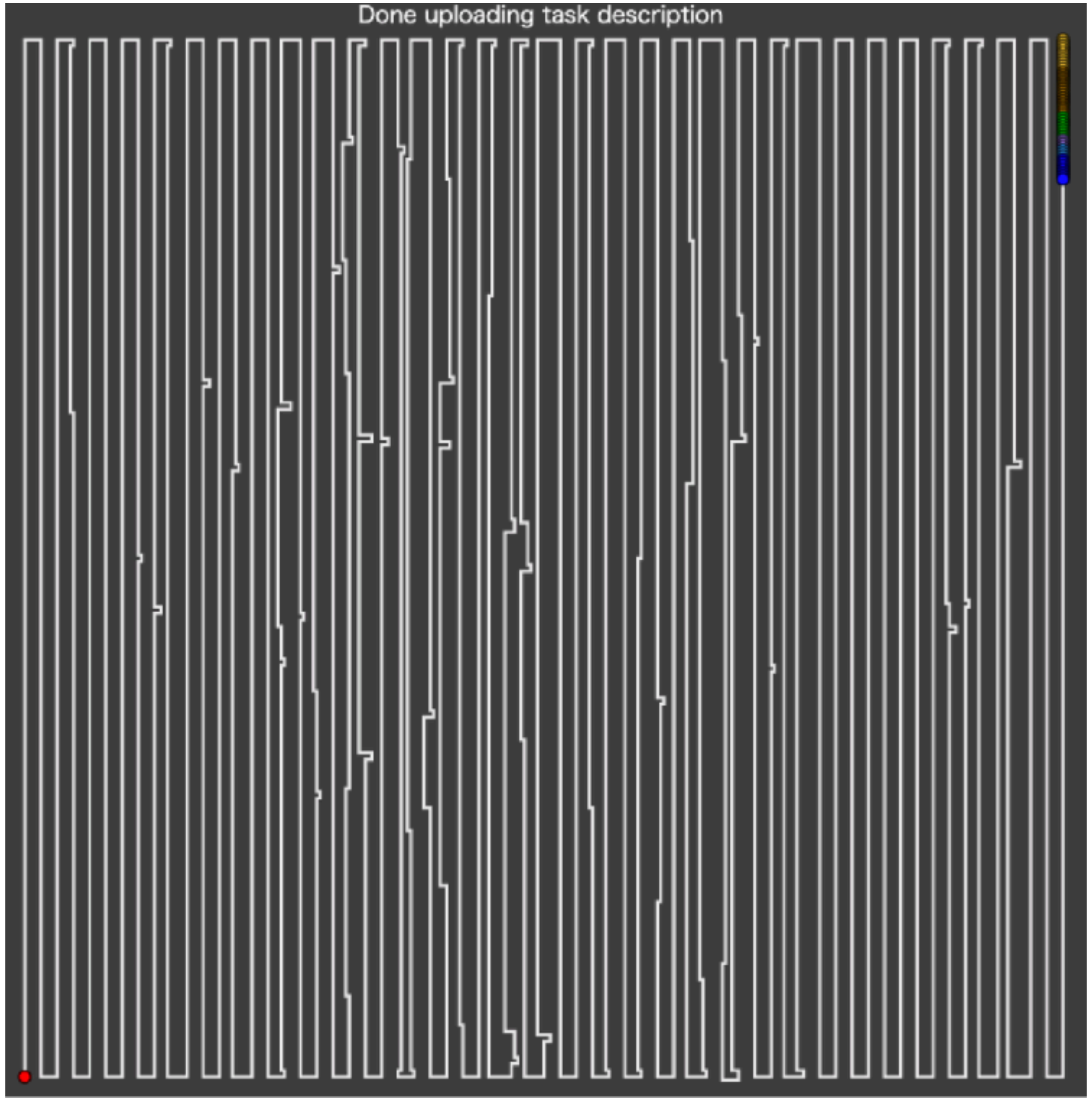
- If a move divides unfilled area into separate component, it tries filling smaller one first.



Misc

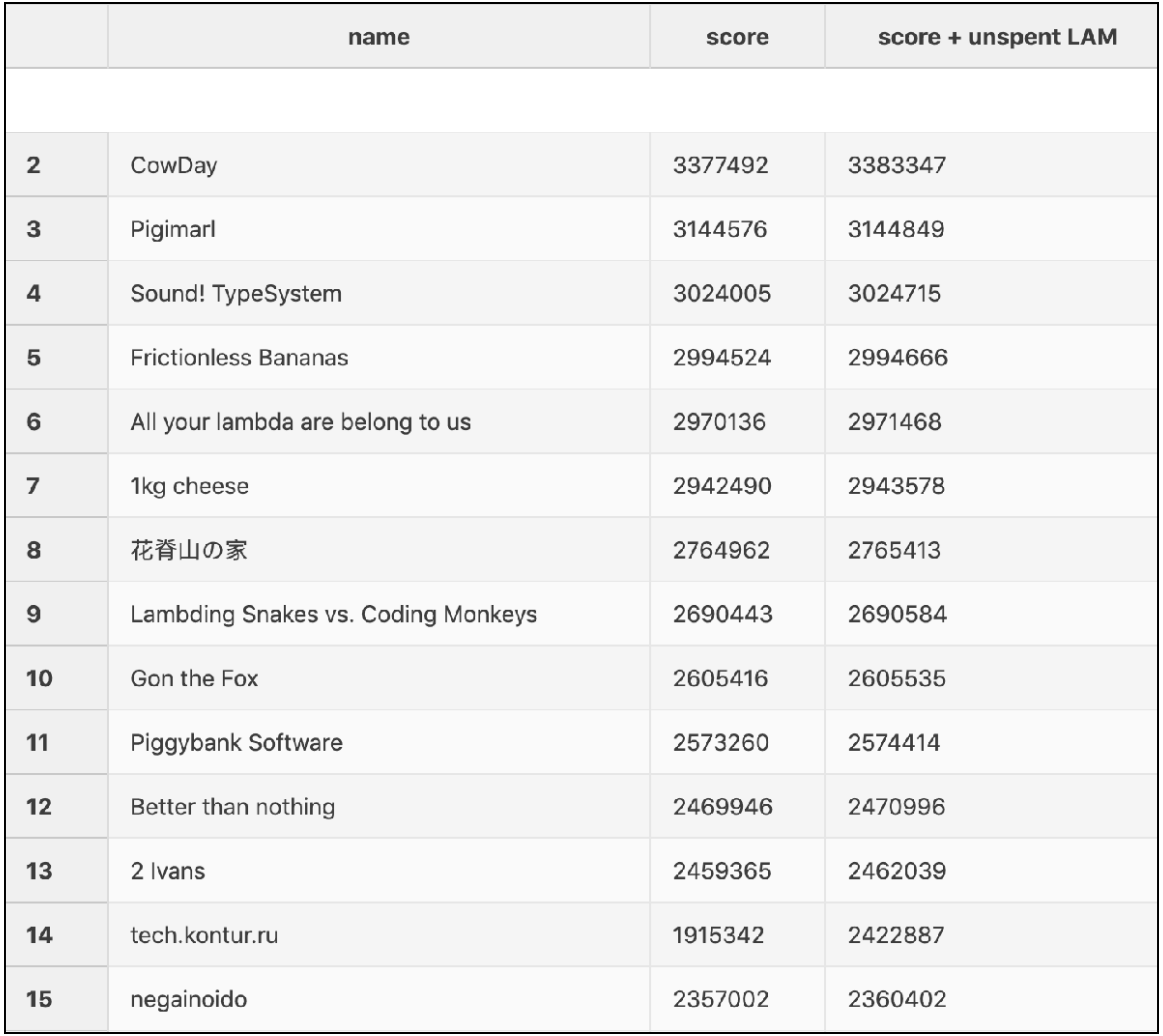
- To spend LAM, we always buy Clones booster.
  - Knapsack-like Dynamic Programming algorithm in our internal judge
- Do not use Drill or Teleporter
- Try both using/not using Fast

And our “equal” puzzle!

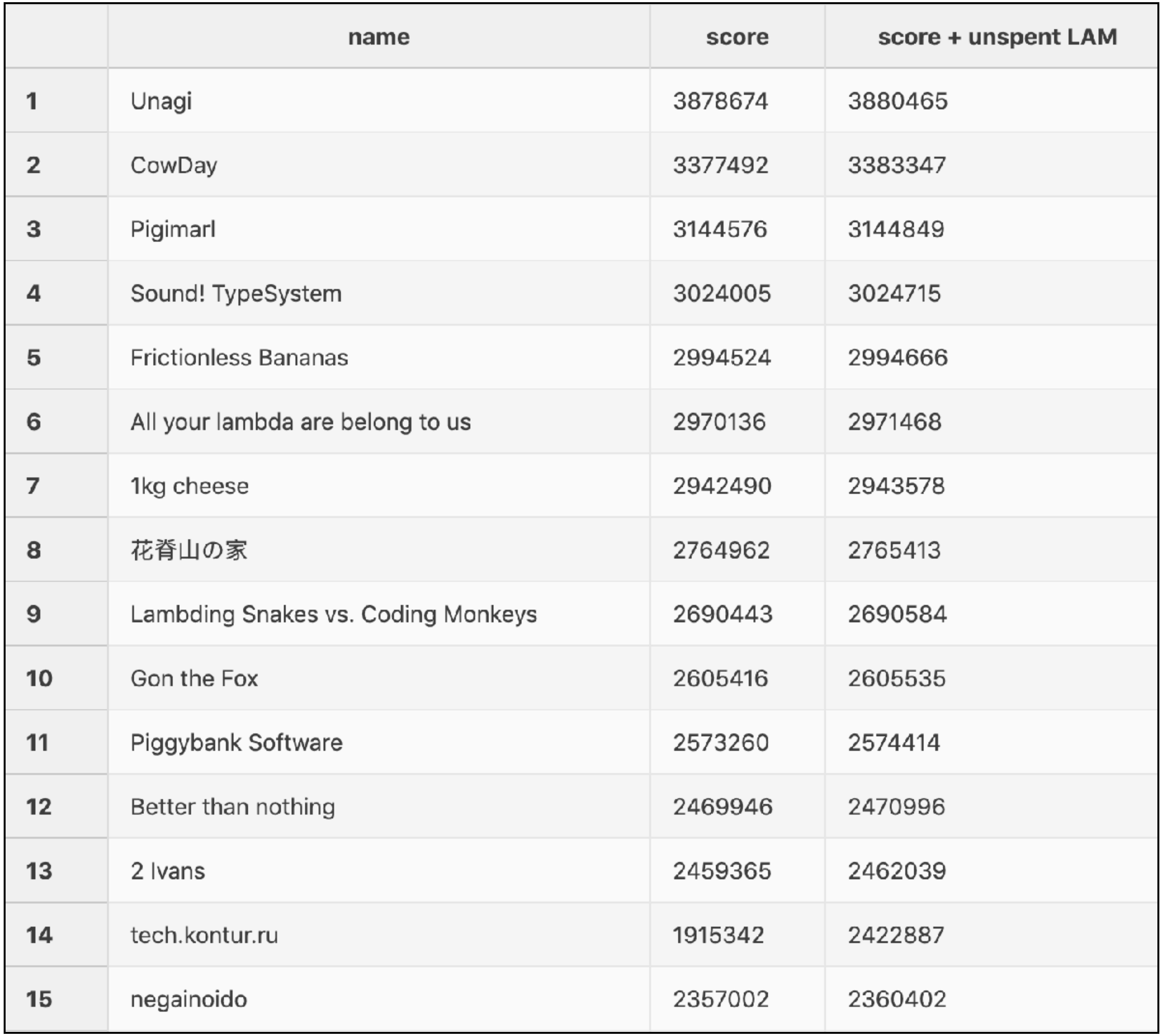


Thank you organizers!

First Prize



	name	score	score + unspent LAM
2	CowDay	3377492	3383347
3	Pigimarl	3144576	3144849
4	Sound! TypeSystem	3024005	3024715
5	Frictionless Bananas	2994524	2994666
6	All your lambda are belong to us	2970136	2971468
7	1kg cheese	2942490	2943578
8	花脊山の家	2764962	2765413
9	Lambding Snakes vs. Coding Monkeys	2690443	2690584
10	Gon the Fox	2605416	2605535
11	Piggybank Software	2573260	2574414
12	Better than nothing	2469946	2470996
13	2 Ivans	2459365	2462039
14	tech.kontur.ru	1915342	2422887
15	negainoido	2357002	2360402



	name	score	score + unspent LAM
1	Unagi	3878674	3880465
2	CowDay	3377492	3383347
3	Pigimarl	3144576	3144849
4	Sound! TypeSystem	3024005	3024715
5	Frictionless Bananas	2994524	2994666
6	All your lambda are belong to us	2970136	2971468
7	1kg cheese	2942490	2943578
8	花脊山の家	2764962	2765413
9	Lambding Snakes vs. Coding Monkeys	2690443	2690584
10	Gon the Fox	2605416	2605535
11	Piggybank Software	2573260	2574414
12	Better than nothing	2469946	2470996
13	2 Ivans	2459365	2462039
14	tech.kontur.ru	1915342	2422887
15	negainoido	2357002	2360402

Team

Unagi

Takuya Akiba, Kentaro Imajo, Hiroaki Iwami,
Yoichi Iwata, Toshiki Kataoka, Naohiro Takahashi

Rust *is the programming tool of choice for discriminating hackers.*

$1000 cash prize

Team Unagi

Takuya Akiba
Preferred Networks

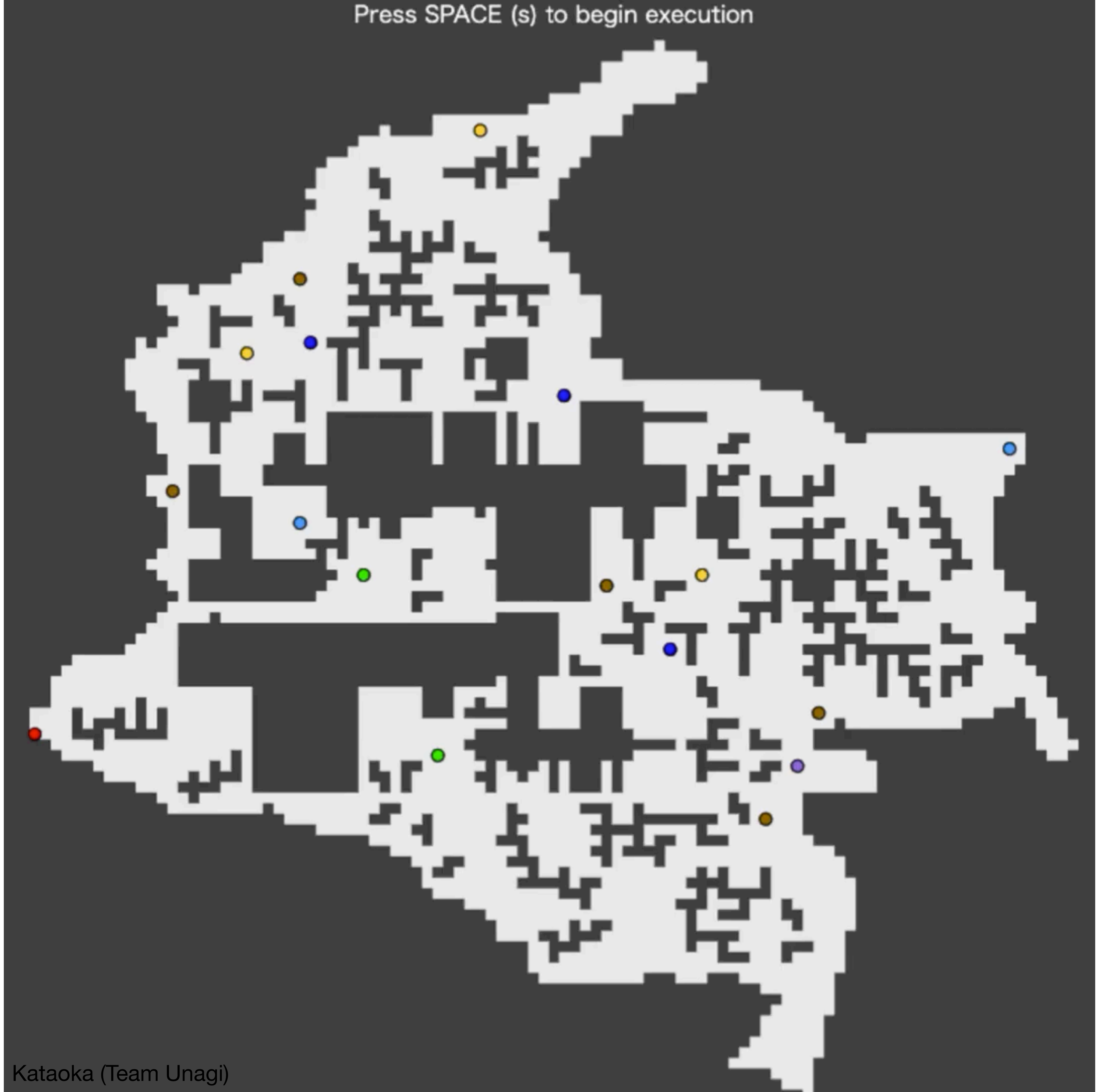
Kentaro Imajo
Preferred Networks

Hiroaki Iwami
FLYWHEEL

Yoichi Iwata
National Institute of Informatics

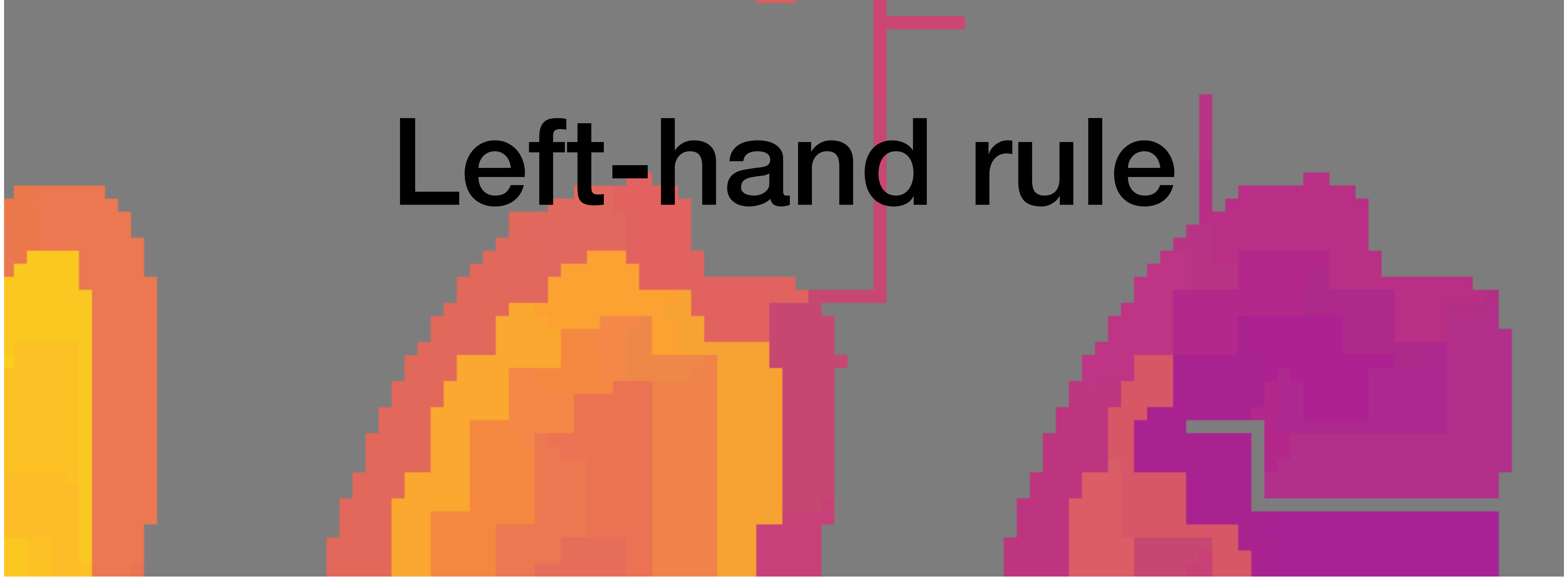
Toshiki Kataoka
Preferred Networks

Naohiro Takahashi
AtCoder



Kataoka (Team Unagi)

- Left-hand rule
- Split single-worker solution



Left-hand rule

- Wrap squares near walls
- Repeat

Clone

- Single-worker solution → multi-worker solution
  - Load balancing
- Bootstrap: visit C, X, C, C, C, C, ...
  - 1st replica stays at X
    and uses boosters immediately

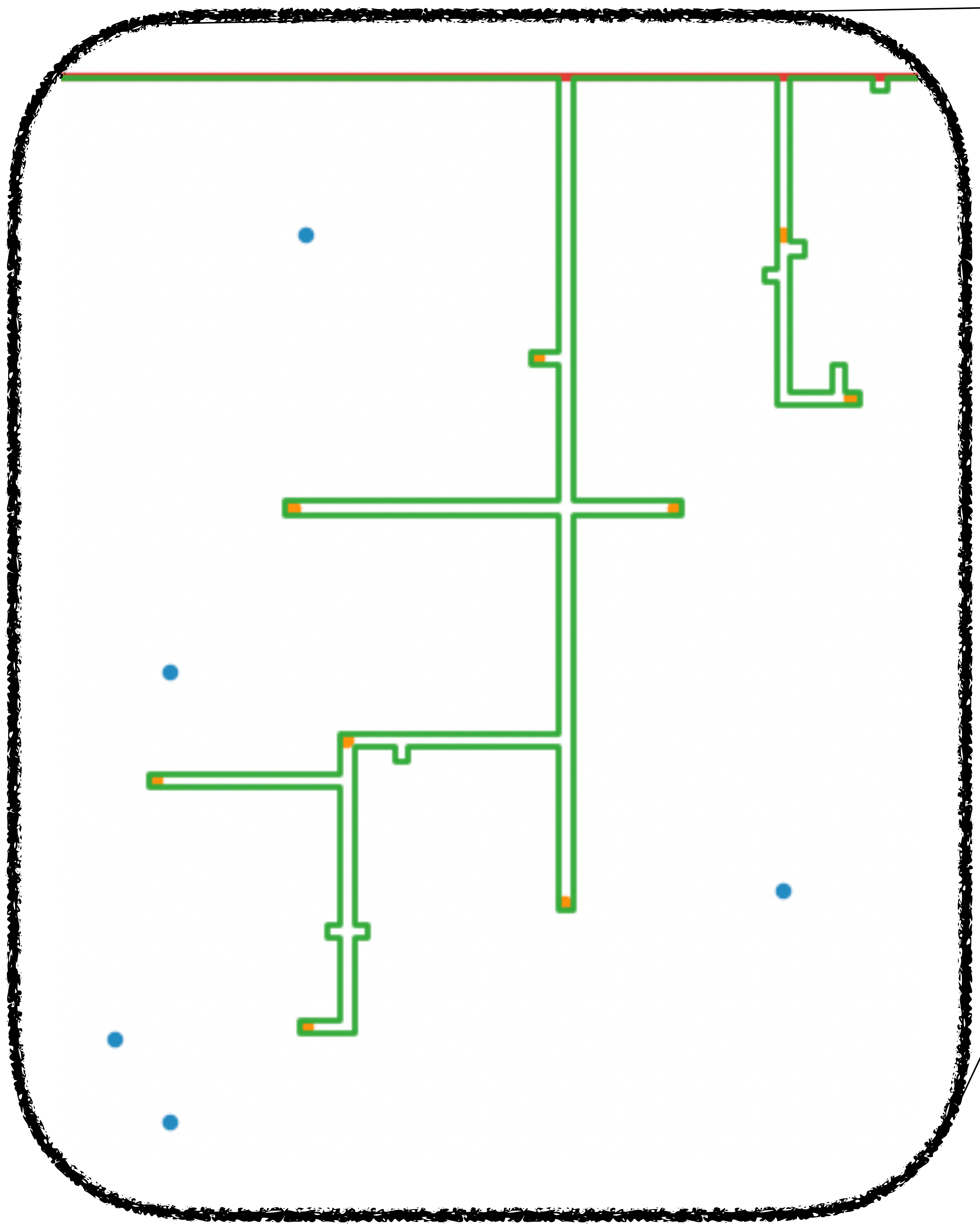
● **C (booster)**

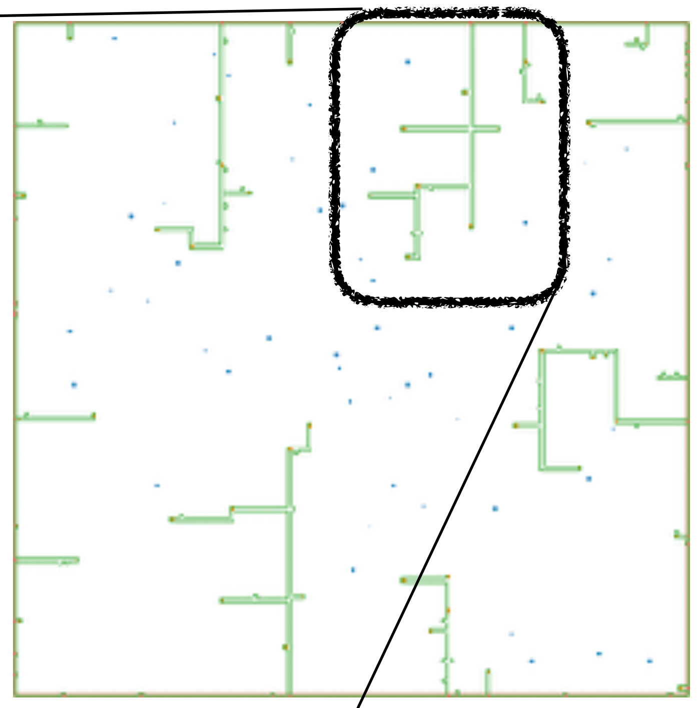
● **X (spawn point)**

Local refinement

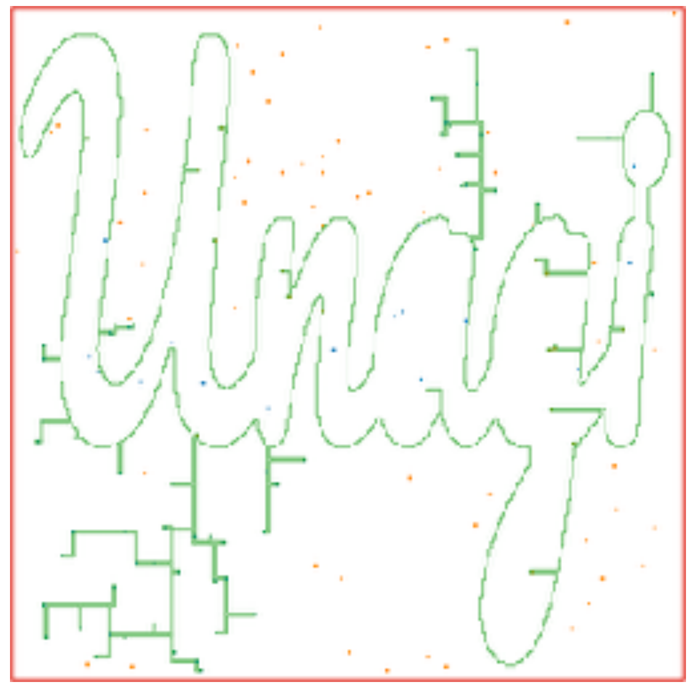
- Replace subsequence with shorter one
  - Use depth-first and breadth-first searches

Earn coins





Round 2



Round 82

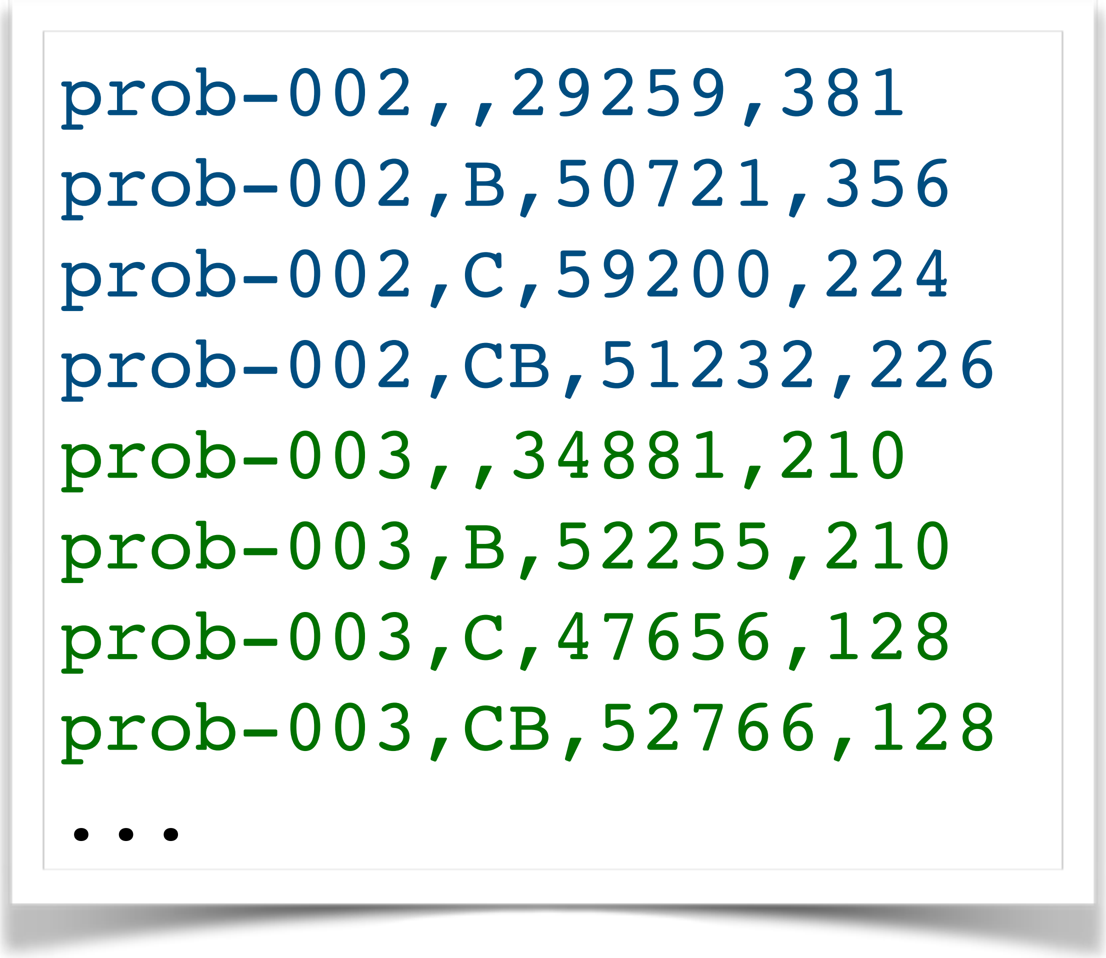
- Connect points
- Add corners

Kataoka (Team Unagi)

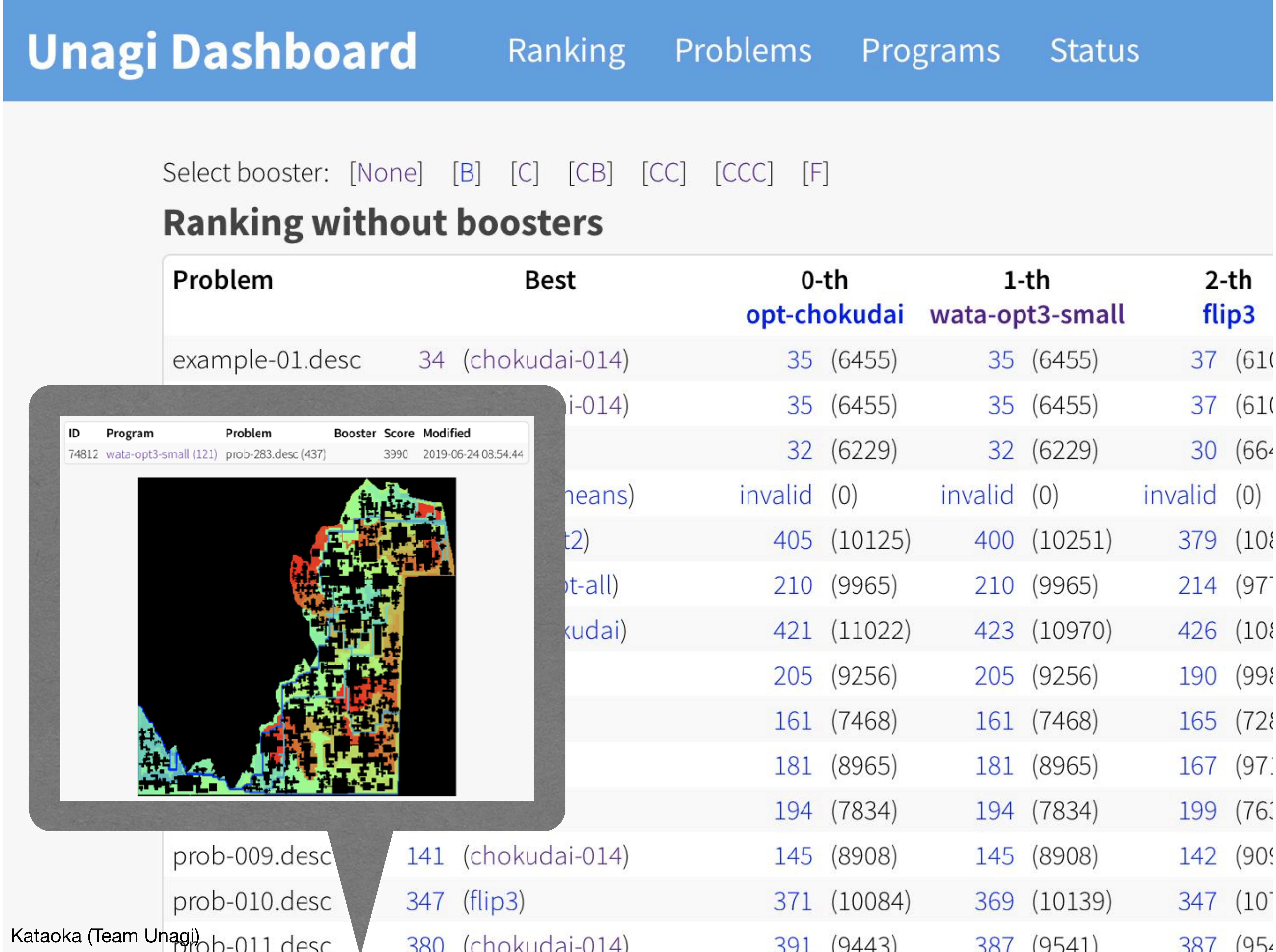
Spend coins

- Prepare solutions for buy settings:
  [ ],
  [B],
  [C],
  [C, B],
  [C, C],
  [C, C, C]

- Solve a knapsack problem



```
prob-002,,29259,381
prob-002,B,50721,356
prob-002,C,59200,224
prob-002,CB,51232,226
prob-003,,34881,210
prob-003,B,52255,210
prob-003,C,47656,128
prob-003,CB,52766,128
...
```



Select booster: [None] [B] [C] [CB] [CC] [CCC] [F]

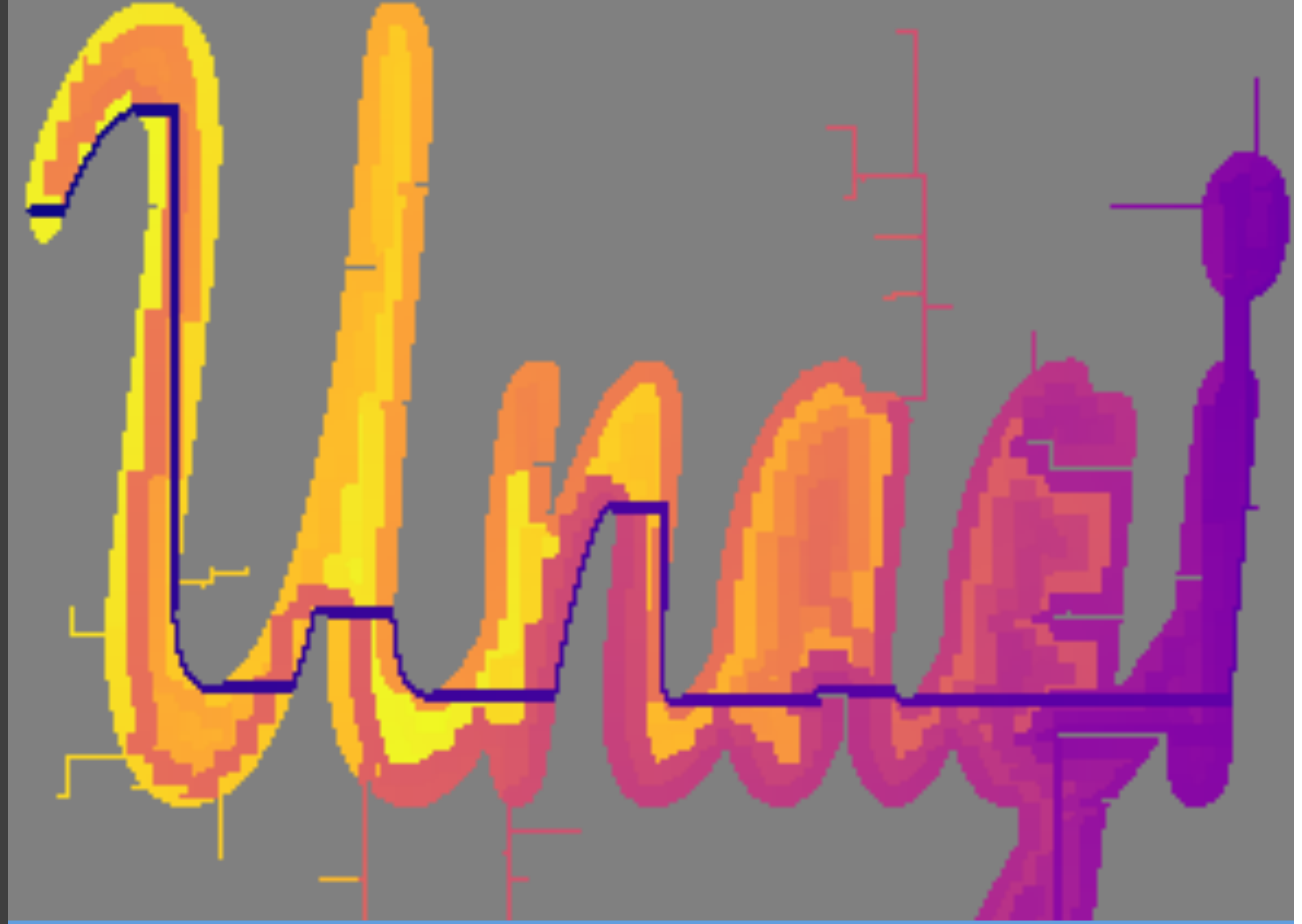
Ranking without boosters

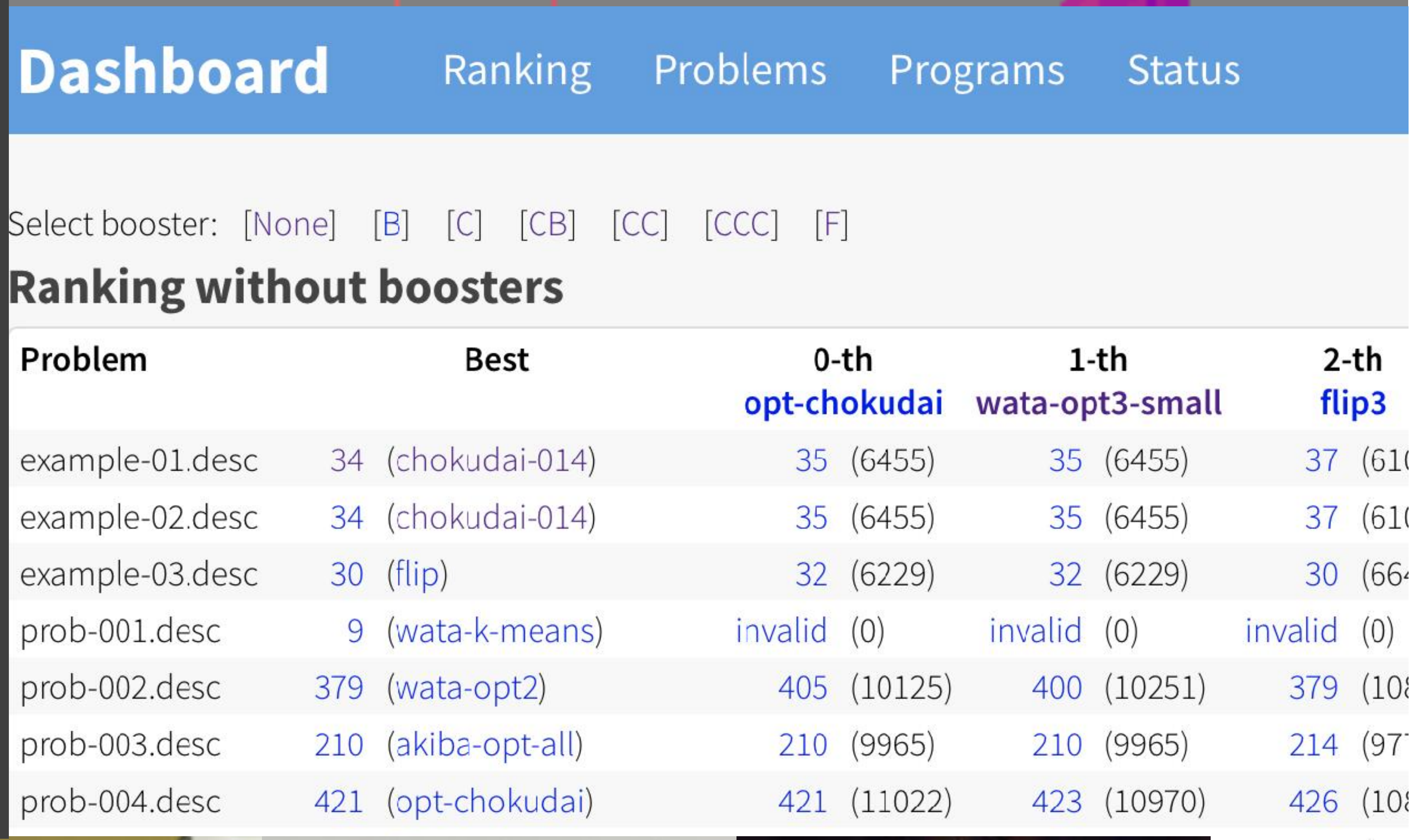
Problem	Best	0-th opt-chokudai	1-th wata-opt3-small	2-th flip3
example-01.desc	34 (chokudai-014)	35 (6455)	35 (6455)	37 (61
	i-014)	35 (6455)	35 (6455)	37 (61
		32 (6229)	32 (6229)	30 (66
	neans)	invalid (0)	invalid (0)	invalid (0)
	t2)	405 (10125)	400 (10251)	379 (10
	t-all)	210 (9965)	210 (9965)	214 (97
	udai)	421 (11022)	423 (10970)	426 (10
		205 (9256)	205 (9256)	190 (99
		161 (7468)	161 (7468)	165 (72
		181 (8965)	181 (8965)	167 (97
		194 (7834)	194 (7834)	199 (76
prob-009.desc	141 (chokudai-014)	145 (8908)	145 (8908)	142 (90
prob-010.desc	347 (flip3)	371 (10084)	369 (10139)	347 (10
prob-011.desc	380 (chokudai-014)	391 (9443)	387 (9541)	387 (95

ID	Program	Problem	Booster	Score	Modified
74812	wata-opt3-small (121)	prob-283.desc (437)		3990	2019-06-24 03:54:44

Language

- Solvers (task, puzzle, buy) are written in **Rust**
- Dashboard: **Go**
- Few lines of **Shell script**, **JavaScript**, **Python**



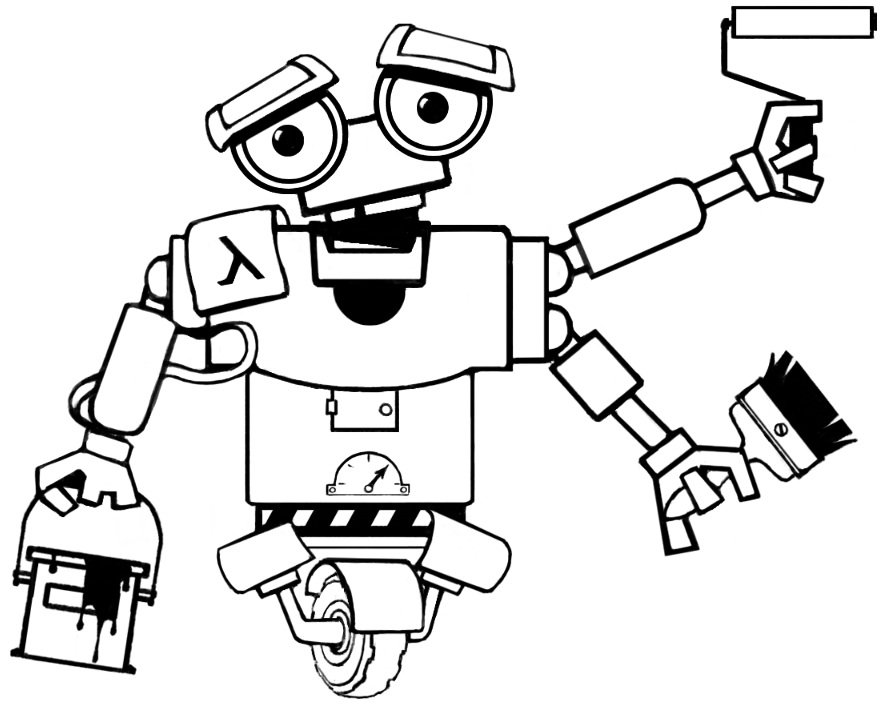


Select booster: [None] [B] [C] [CB] [CC] [CCC] [F]

Ranking without boosters

Problem	Best		0-th opt-chokudai		1-th wata-opt3-small		2-th flip3	
example-01.desc	34	(chokudai-014)	35	(6455)	35	(6455)	37	(61
example-02.desc	34	(chokudai-014)	35	(6455)	35	(6455)	37	(61
example-03.desc	30	(flip)	32	(6229)	32	(6229)	30	(66
prob-001.desc	9	(wata-k-means)	invalid	(0)	invalid	(0)	invalid	(0)
prob-002.desc	379	(wata-opt2)	405	(10125)	400	(10251)	379	(10
prob-003.desc	210	(akiba-opt-all)	210	(9965)	210	(9965)	214	(97
prob-004.desc	421	(opt-chokudai)	421	(11022)	423	(10970)	426	(10
							190	(99
							165	(72
							167	(97
							199	(76
							142	(90
							347	(10
							387	(95
							403	(95
prob-013.desc	377	(flip)	384	(9539)	388	(9441)	377	(97
prob-014.desc	334	(wata-opt3-small)	338	(9601)	334	(9716)	350	(92
prob-015.desc	353	(wata-opt3-small)	355	(9662)	353	(9716)	366	(93

To Wrap Up



Contest Organisers

Ilya Sergey

- Contest idea, design, specification
- Problem instances
- Back-end grader
- Solution visualiser
- Social media and mailing list

Paramdeep Singh Raina

- Back-up server support

Yale
NUS
College

NUS
National University
of Singapore

Contest Organisers

George Pîrlea

- Server-side programming
- Grading automation
- Rankings generation
- Data analysis
- Web-page

Contest Organisers

Lilia Anisimova

- Wrappy art

Past Contest Chairs

Sam Lindley
ICFP Contest 2017

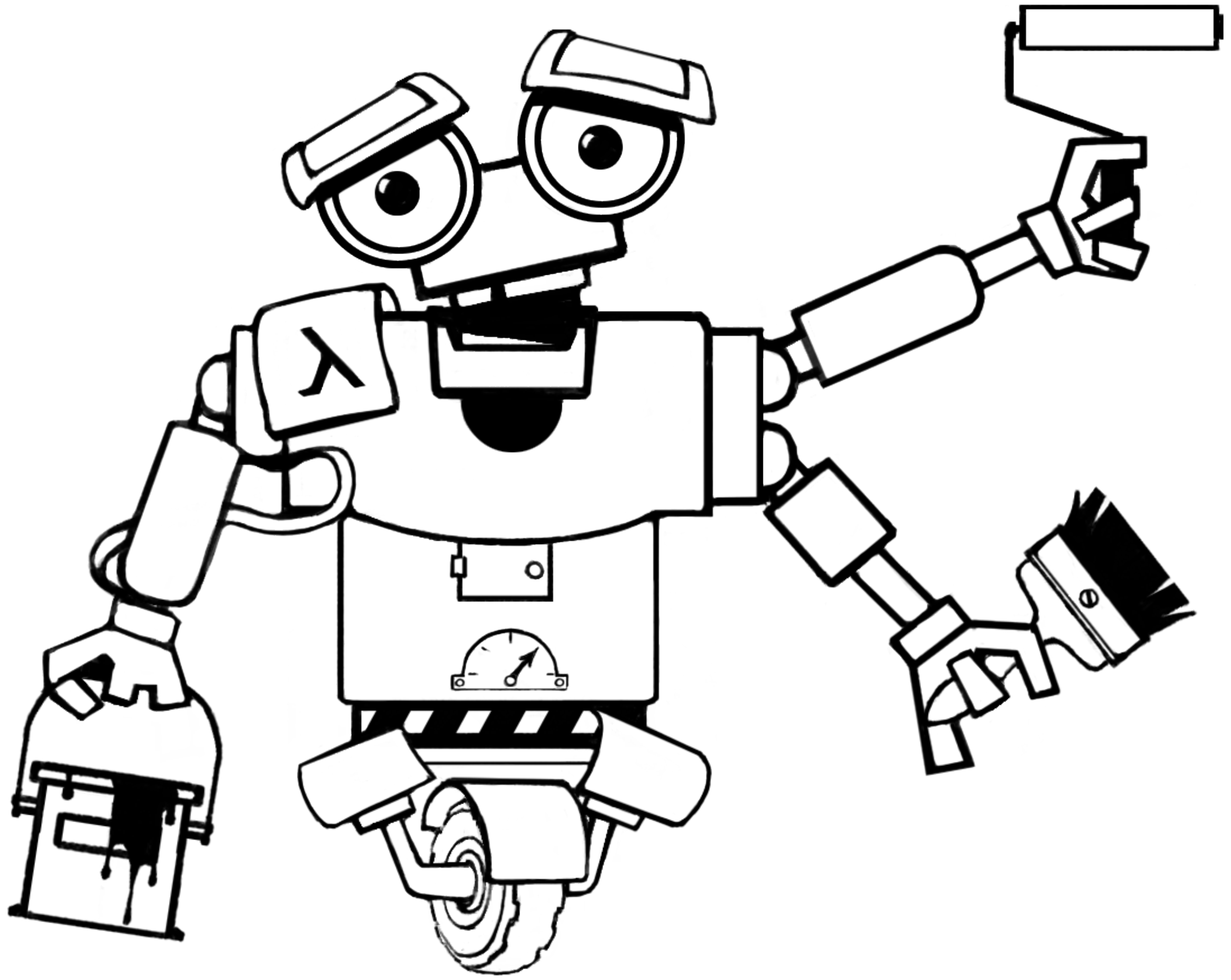
Matthew Fluet
ICFP Contest 2018

Software and Tools

- Problem generation and back-end grader:
  - Scala
- Visualisation:
  - Scala.js
- Server-side scripting and data management:
  - Python 3
  - Flask
  - SQLite
  - Celery
  - RabbitMQ
  - AWS Elastic File System
  - AWS EC2
- Web page:
  - GitHub pages & Jekyll markdown

Sponsors

Thanks to all contest participants!



Good luck for
ICFP Contest 2020!