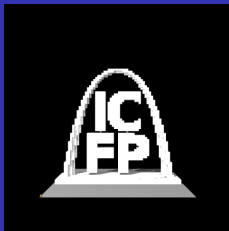


ICFP'18 Programming Contest Report



Matthew Fluet

Rochester Institute of Technology

September 25, 2018

Sponsors:  MANNING

ICFP Programming Contest

- Annual contest since 1998 sponsored by ICFP
- Held for 72 consecutive hours (usually with a 24 hour lightning division)
- Modest cash prizes, but unlimited bragging rights:
 - “??? is the programming language of choice for discriminating hackers”
 - “??? is a fine tool for many applications”
 - “??? is very suitable for rapid prototyping”
 - “??? is an extremely cool bunch of hackers”
- Each programming contest organized by a different group and each contest task is totally different:
 - play games, draw images, optimize structures, control robots, ...

Timeline (The Usual Story)

Timeline (The Usual Story)

- April/May 2017:
 - Get contacted by ICFP GC about organizing Programming Contest.
 - Get a small number of local faculty interested.
 - Agree to organize the contest.
- ... promptly forget about the contest for 7 months ...
- January – May 2018:
 - Brainstorm task ideas, starting with impossibly impractical ones.
 - Throw together some prototypes and narrow the scope.
 - Begin building the contest infrastructure for the finalized task idea.

Timeline (The Usual Story)

- April/May 2017:
 - Get contacted by ICFP GC about organizing Programming Contest.
 - Get a small number of local faculty interested.
 - Agree to organize the contest.
- ... promptly forget about the contest for 7 months ...
- January – May 2018:
 - Brainstorm task ideas, starting with impossibly impractical ones.
 - Throw together some prototypes and narrow the scope.
 - Begin building the contest infrastructure for the finalized task idea.
- June 2018:
 - **PANIC!**
 - Make significant changes to task and scramble to pull everything together.
- July 2018:
 - **REALLY SCRAMBLE** to pull everything together with a day to spare.

Timeline (The Usual Story)

- April/May 2017:
 - Get contacted by ICFP GC about organizing Programming Contest.
 - Get a small number of local faculty interested.
 - Agree to organize the contest.
- ... promptly forget about the contest for 7 months ...
- January – May 2018:
 - Brainstorm task ideas, starting with impossibly impractical ones.
 - Throw together some prototypes and narrow the scope.
 - Begin building the contest infrastructure for the finalized task idea.
- June 2018:
 - **PANIC!**
 - Make significant changes to task and scramble to pull everything together.
- July 2018:
 - **REALLY SCRAMBLE** to pull everything together with a day to spare.
- July 20 – July 23, 2018:
 - Run the contest with virtually no issues.

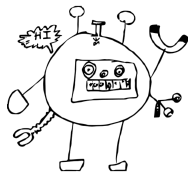
Contest Task

Contest Task Desiderata

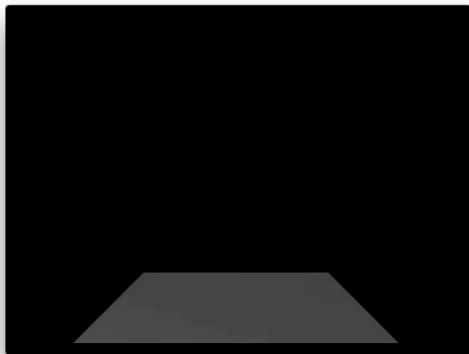
- Contestants should submit some kind of “checkable artifact”, not a “runnable executable”.
- Task should have a “low barrier to entry”; it should be easy to get something that works and can then be incrementally improved.
- Task should emphasize “programming as problem solving”; there should be “big-picture” solutions that are obvious to a human, but challenging to translate into “low-level” actions for a computer.
- Task should not be “deep learnable”.

Contest Task: Lightning Division

Generate nanobot traces
to construct target 3D objects
while minimizing energy used.



The initial nanobot is affectionately known as the Build-a-Tron 4000.



Contest Task: Lightning Division

A new form of 3D printing:

- nanobots facilitate direct conversion of energy to matter
- a matrix of voxels in which matter can be created
 - low harmonics: all matter must be part of a connected component that rests on the floor (“grounded”; low energy cost)
 - high harmonics: matter is unconstrained (“floating”; high energy cost)
- nanobots move through empty voxels and are unconstrained (“flying”)
- nanobots can create matter in neighboring voxels
- nanobots can fission (to fork off another nanobot) and fusion (to join with another nanobot), with a maximum of 20 nanobots, but execution must start and end with a single nanobot at the origin
- each nanobot executes a single command each time step; error if commands of different nanobots interfere (“deterministic”)

Generate nanobot traces to construct, **deconstruct, and reconstruct** target 3D objects while minimizing energy used.

- nanobots can create **and destroy** matter in neighboring voxels
- nanobots can fission and fusion with with a maximum of **40** nanobots, but execution must start and end with a single nanobot at the origin
- **nanobots can form groups to create and destroy matter in lines, planes, and boxes**

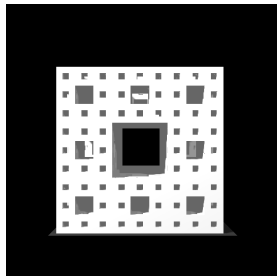
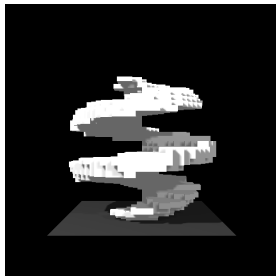
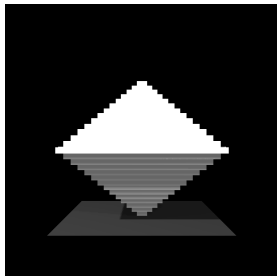
Some interesting aspects:

- Irregular parallelism:
fair division of space not necessarily a fair division of work
- Spatial locality:
non-zero cost (time and energy) to move a nanobot to new work space
- Indirect interference:
error if nanobots move through same voxel,
even when ending at different voxels
- High harmonics (“floating”) vs. low harmonics (“grounded”):
high harmonics expensive, but easier to parallelize and coordinate;
low harmonics cheap, but can restrict construction/destruction order
- Assymetry in group fill vs. group void:
pen tool vs. eraser tool

Contest Task Details

- Models

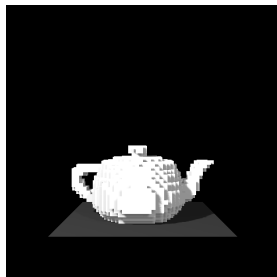
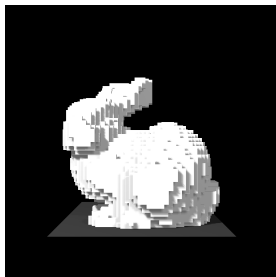
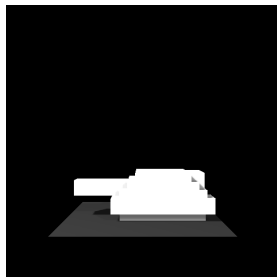
- Boolean matrix up to $250 \times 250 \times 250$ describing 3D object
- Guaranteed to be “grounded” and have free space along top and sides
- Binary encoding: resolution followed by bitmap
- A few procedurally generated geometric objects; most objects taken from Thingiverse



Contest Task Details

- Models

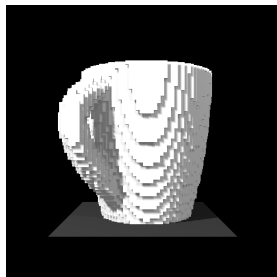
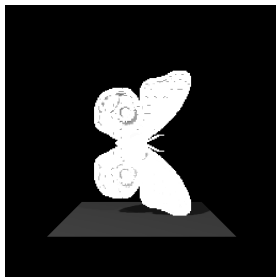
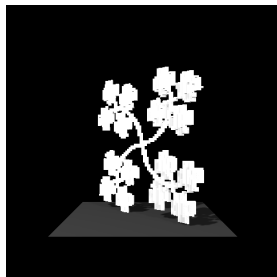
- Boolean matrix up to $250 \times 250 \times 250$ describing 3D object
- Guaranteed to be “grounded” and have free space along top and sides
- Binary encoding: resolution followed by bitmap
- A few procedurally generated geometric objects; most objects taken from Thingiverse



Contest Task Details

- Models

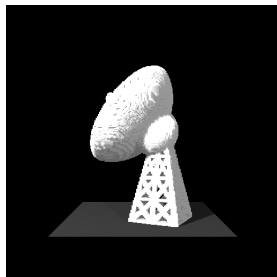
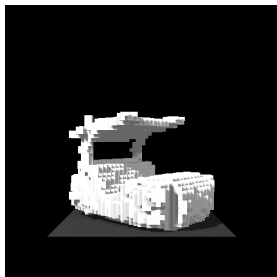
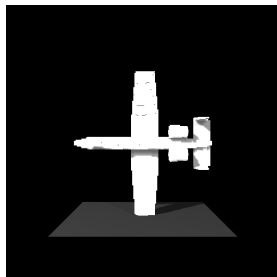
- Boolean matrix up to $250 \times 250 \times 250$ describing 3D object
- Guaranteed to be “grounded” and have free space along top and sides
- Binary encoding: resolution followed by bitmap
- A few procedurally generated geometric objects; most objects taken from Thingiverse



Contest Task Details

- Models

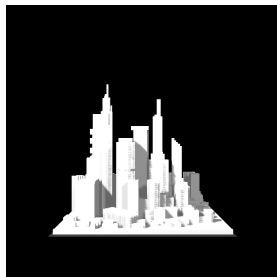
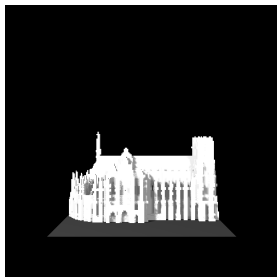
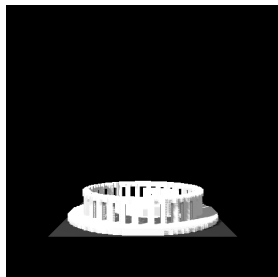
- Boolean matrix up to $250 \times 250 \times 250$ describing 3D object
- Guaranteed to be “grounded” and have free space along top and sides
- Binary encoding: resolution followed by bitmap
- A few procedurally generated geometric objects; most objects taken from Thingiverse



Contest Task Details

- Models

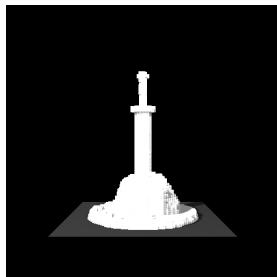
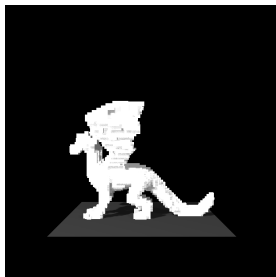
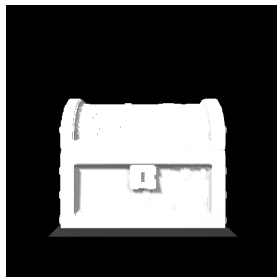
- Boolean matrix up to $250 \times 250 \times 250$ describing 3D object
- Guaranteed to be “grounded” and have free space along top and sides
- Binary encoding: resolution followed by bitmap
- A few procedurally generated geometric objects; most objects taken from Thingiverse



Contest Task Details

- Models

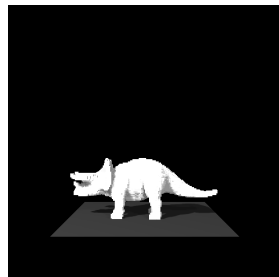
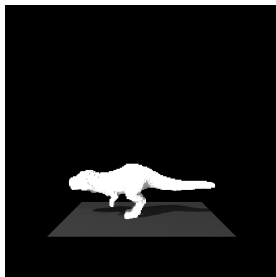
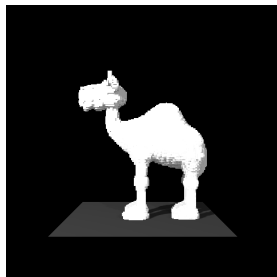
- Boolean matrix up to $250 \times 250 \times 250$ describing 3D object
- Guaranteed to be “grounded” and have free space along top and sides
- Binary encoding: resolution followed by bitmap
- A few procedurally generated geometric objects; most objects taken from Thingiverse



Contest Task Details

- Models

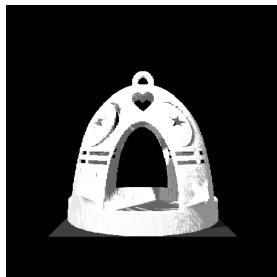
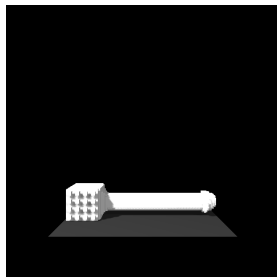
- Boolean matrix up to $250 \times 250 \times 250$ describing 3D object
- Guaranteed to be “grounded” and have free space along top and sides
- Binary encoding: resolution followed by bitmap
- A few procedurally generated geometric objects; most objects taken from Thingiverse



Contest Task Details

- Models

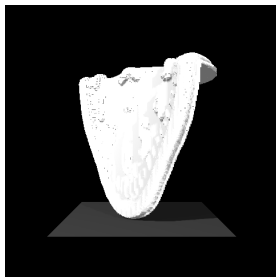
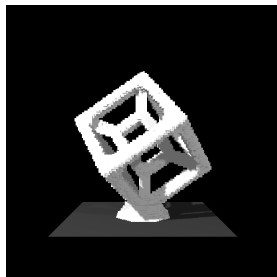
- Boolean matrix up to $250 \times 250 \times 250$ describing 3D object
- Guaranteed to be “grounded” and have free space along top and sides
- Binary encoding: resolution followed by bitmap
- A few procedurally generated geometric objects; most objects taken from Thingiverse



Contest Task Details

- Models

- Boolean matrix up to $250 \times 250 \times 250$ describing 3D object
- Guaranteed to be “grounded” and have free space along top and sides
- Binary encoding: resolution followed by bitmap
- A few procedurally generated geometric objects; most objects taken from Thingiverse



Contest Task Details

- Traces

- Sequence of commands; ordered by time, then by nanobot identifier
- Binary encoding: each command encoded as 1-, 2-, or 4-bytes
- Commands:
 - **Halt** – halt execution w/ one nanobot at origin
 - **Wait** – no effect
 - **Flip** – flip harmonics
 - **SMove ld** – straight move, up to 15 voxels
 - **LMove $sld_1 sld_2$** – L move, up to 5 voxels each direction
 - **Fill nd ; Void nd** – create/destroy matter in one of 18 neighboring voxels
 - **GFill $nd fd$; GVoid $nd fd$** – identify a cuboid by near corner and far corner; group nanobots by region, with a different nanobot at each “corner”; 2 fill/void a “line”; 4 fill/void a “plane”; 8 fill/void a “box”
 - **Fission $nd m$** – fork nanobot into neighboring voxel with m seeds
 - **FusionP nd ; FusionS nd** – secondary joins with primary in neighboring voxel

Contest Task Details

- Tools

- Javascript and WebGL-based model viewer
- Javascript trace checker
- Javascript and WebGL-based trace executor
- Javascript trace executor (w/o visualization; marginally faster)

Received a number of compliments about the online tools.

It is exactly the same problem without the visualization of trace execution, but seeing things in action makes it much more exciting.

Contest Task Details

- Problems
 - Lightning Division: 186 construction
 - Full Contest: 186 construction, 186 destruction, 115 reconstruction
- Default traces
 - Establish an upper-bound on energy for each problem
 - Construction (classic 3D-printing): high harmonics and one “print head” nanobot
 - Destruction (classic 3D-printing): high harmonics and one “eraser head” nanobot
 - Reconstruction: high harmonics and one nanobot void-ahead/fill-behind
- Scoring
 - Problem score based on default trace energy and best team’s energy; linearly interpolate from 0 (if team’s energy no better than default) to $\lfloor \log_2 R \rfloor \times 1000$ (if team’s energy is best, weighted by model size)
 - Contest score is sum of problem scores

Statistics

- 187 registered teams;
95 completed profiles (team member(s), team country(ies), team programming language(s))
- Lightning Division
 - 485 submissions (zip of traces for all problems)
 - 54 teams with last submission evaluated for frozen live standings (6hrs);
42 with non-zero scores
 - 100 teams with last submission evaluated for final standings;
91 with non-zero scores
- Full Contest
 - 1099 submissions (zip of traces for all problems)
 - 95 teams with last submission evaluated for frozen live standings (6hrs);
95 with non-zero scores
 - 108 teams with last submission evaluated for final standings;
107 with non-zero scores
- 20Gb total submissions
- 731M largest submission

Statistics: Team Size, Country(ies), Programming Language(s)

<u>Size</u>	<u>Teams</u>
1	24
2	17
5	12
3	11
6	11
4	8
8	5
7	3
9	2
11	1
15	1

<u>Country(ies)</u>	<u>Teams</u>
1	80
2	12
3	1
4	1
7	1

<u>Language(s)</u>	<u>Teams</u>
1	66
2	14
3	7
4	3
6	3
5	2

Statistics: Team Country(ies)

Country	Teams	Scaled	Country	Teams	Scaled
JPN	29	26.64	AUT	1	1.00
USA	25	21.23	AZE	1	0.25
RUS	19	15.14	BLR	1	1.00
UKR	8	5.50	CHN	1	1.00
GBR	5	2.98	DNK	1	1.00
DEU	4	3.50	GRC	1	1.00
AUS	3	1.75	KAZ	1	0.50
POL	3	1.39	LVA	1	1.00
CAN	2	1.14	NZL	1	1.00
CHE	2	0.64	PRK	1	0.50
FRA	2	1.33	PRT	1	1.00
ITA	2	2.00	XXX	1	1.00
NLD	2	1.50			

Statistics: Team Programming Languages

Language	Teams	Scaled	Language	Teams	Scaled
C++	30	18.48	Cython	1	0.17
Python	28	15.57	Dart	1	0.33
Haskell	11	10.00	Elixir	1	0.50
bash/sh	10	3.15	Erlang	1	1.00
C#	8	5.28	F#	1	1.00
Go	7	4.75	Gradle	1	0.50
Java	7	5.67	Haxe	1	0.25
Javascript	7	1.98	Julia	1	0.50
OCaml	7	4.95	Jupyter	1	0.50
Ruby	7	4.42	Lisp	1	0.25
Rust	7	5.67	Netlogo	1	0.17
Kotlin	4	3.33	PHP	1	0.25
C	2	2.00	R	1	1.00
Scala	2	2.00	SQL	1	0.17
Clojure	1	1.00	Vimscript	1	0.17

Results

Lightning Division: Final Standings

Rank	Team	Total Energy	Total Score
6	Huloutte	204278898169317	1007644
7	Lambding Snakes vs. Coding Monkeys	216139553955014	1007419
8	uguu.org	313897920015327	995385
9	Side Effects May Include...	475134198213131	991838
10	no need for a type system	258092265876510	988608
11	A Storm of Minds	1408758074962804	986109
12	Drawing a Blank	376006355689028	984559
13	301	2752268618745760	982853
14	yowa	457446586997808	982164
15	pigimarudon	783688414203897	978685

Lightning Division: Final Standings

Rank	Team	Total Energy	Total Score
2	forM_	63814182182188	1023741
3	fixstars	290169090894495	1021405
4	AIM Tech	171361916186560	1016120
5	srush	199581157194348	1011345
6	Huloutte	204278898169317	1007644
7	Lambding Snakes vs. Coding Monkeys	216139553955014	1007419
8	uguu.org	313897920015327	995385
9	Side Effects May Include...	475134198213131	991838
10	no need for a type system	258092265876510	988608
11	A Storm of Minds	1408758074962804	986109
12	Drawing a Blank	376006355689028	984559
13	301	2752268618745760	982853
14	yowa	457446586997808	982164
15	pigimarudon	783688414203897	978685

Lightning Division: Final Standings

Rank	Team	Total Energy	Total Score
1	jabber.ru	44012042556870	1024331
2	forM_	63814182182188	1023741
3	fixstars	290169090894495	1021405
4	AIM Tech	171361916186560	1016120
5	srush	199581157194348	1011345
6	Huloutte	204278898169317	1007644
7	Lambding Snakes vs. Coding Monkeys	216139553955014	1007419
8	uguu.org	313897920015327	995385
9	Side Effects May Include...	475134198213131	991838
10	no need for a type system	258092265876510	988608
11	A Storm of Minds	1408758074962804	986109
12	Drawing a Blank	376006355689028	984559
13	301	2752268618745760	982853
14	yowa	457446586997808	982164
15	pigimarudon	783688414203897	978685

Team jabber.ru

(RUS)

Alexey Shchepin

OCaml
is very suitable for rapid prototyping.

\$500 cash prize

Print and electronic copies of *Get Programming in Haskell*
donated by Manning Publications

Team: jabber.ru

Members: Alexey Shchepin

Language: OCaml

Full Contest: Final Standings

Rank	Team	Total Energy	Total Score
6	perpetuum mobile	550338521287339	2664576
7	Line Graph	48920723499491	2664495
8	negainoido	294442762485236	2660993
9	銀閣寺GOLD	511134370817665	2659284
10	sanma	171895756884244	2658788
11	Play Race for the Galaxy on your...	245232444154538	2658435
12	The Blind Hen	153228923157128	2656600
13	Whippet Bobsled	190651538471586	2654936
14	All your energy are belong to us	9932428727922403	2654067
15	chirimenjako	285891478626169	2648656

Team perpetuum mobile

(AUS)

Andrew Conway

John Fitzgerald

Team perpetuum mobile (using Scala)
are an extremely cool bunch of hackers.

\$500 cash prize

Print and electronic copies of *Get Programming in Haskell*
donated by Manning Publications

¹For most visually interesting constructions.

Judges' Prize Winner: Team perpetuum mobile

Team Perpetuum Mobile



Full Contest: Final Standings

Rank	Team	Total Energy	Total Score
6	perpetuum mobile	550338521287339	2664576
7	Line Graph	48920723499491	2664495
8	negainoido	294442762485236	2660993
9	銀閣寺GOLD	511134370817665	2659284
10	sanma	171895756884244	2658788
11	Play Race for the Galaxy on your...	245232444154538	2658435
12	The Blind Hen	153228923157128	2656600
13	Whippet Bobsled	190651538471586	2654936
14	All your energy are belong to us	9932428727922403	2654067
15	chirimenjako	285891478626169	2648656

Full Contest: Final Standings

Rank	Team	Total Energy	Total Score
3	Frictionless Bananas	30179212951271	2667310
4	shinh	89429372348781	2667121
5	CowDay	99779058011264	2666478
6	perpetuum mobile	550338521287339	2664576
7	Line Graph	48920723499491	2664495
8	negainoido	294442762485236	2660993
9	銀閣寺GOLD	511134370817665	2659284
10	sanma	171895756884244	2658788
11	Play Race for the Galaxy on your...	245232444154538	2658435
12	The Blind Hen	153228923157128	2656600
13	Whippet Bobsled	190651538471586	2654936
14	All your energy are belong to us	9932428727922403	2654067
15	chirimenjako	285891478626169	2648656

Full Contest: Final Standings

Rank	Team	Total Energy	Total Score
2	manarimo	27863838353220	2668481
3	Frictionless Bananas	30179212951271	2667310
4	shinh	89429372348781	2667121
5	CowDay	99779058011264	2666478
6	perpetuum mobile	550338521287339	2664576
7	Line Graph	48920723499491	2664495
8	negainoido	294442762485236	2660993
9	銀閣寺GOLD	511134370817665	2659284
10	sanma	171895756884244	2658788
11	Play Race for the Galaxy on your...	245232444154538	2658435
12	The Blind Hen	153228923157128	2656600
13	Whippet Bobsled	190651538471586	2654936
14	All your energy are belong to us	9932428727922403	2654067
15	chirimenjako	285891478626169	2648656

Team manarimo

(JPN)

Osamu Koga

Yuki Kawata

Yosuke Yano

mkut

Yu Fujikake

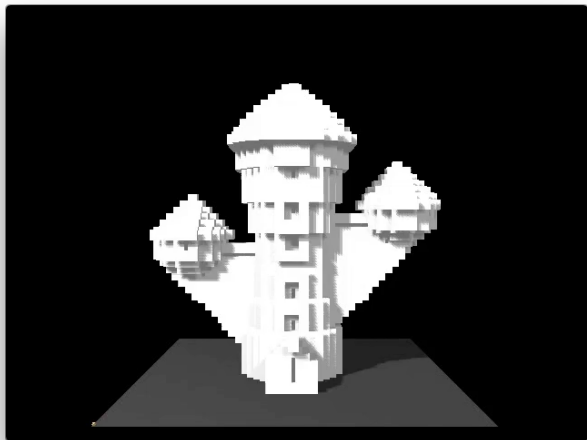
Shunsuke Ohashi

C++, Python, Ruby, JavaScript, bash, SQL
are fine tools for many applications.

\$500 cash prize

Print and electronic copies of *Get Programming in Haskell*
donated by Manning Publications

Full Contest 2nd Place Winner: Team manarimo



Full Contest: Final Standings

Rank	Team	Total Energy	Total Score
2	manarimo	27863838353220	2668481
3	Frictionless Bananas	30179212951271	2667310
4	shinh	89429372348781	2667121
5	CowDay	99779058011264	2666478
6	perpetuum mobile	550338521287339	2664576
7	Line Graph	48920723499491	2664495
8	negainoido	294442762485236	2660993
9	銀閣寺GOLD	511134370817665	2659284
10	sanma	171895756884244	2658788
11	Play Race for the Galaxy on your...	245232444154538	2658435
12	The Blind Hen	153228923157128	2656600
13	Whippet Bobsled	190651538471586	2654936
14	All your energy are belong to us	9932428727922403	2654067
15	chirimenjako	285891478626169	2648656

Full Contest: Final Standings

Rank	Team	Total Energy	Total Score
1	Unagi	6886885323098	2669818
2	manarimo	27863838353220	2668481
3	Frictionless Bananas	30179212951271	2667310
4	shinh	89429372348781	2667121
5	CowDay	99779058011264	2666478
6	perpetuum mobile	550338521287339	2664576
7	Line Graph	48920723499491	2664495
8	negainoido	294442762485236	2660993
9	銀閣寺GOLD	511134370817665	2659284
10	sanma	171895756884244	2658788
11	Play Race for the Galaxy on your...	245232444154538	2658435
12	The Blind Hen	153228923157128	2656600
13	Whippet Bobsled	190651538471586	2654936
14	All your energy are belong to us	9932428727922403	2654067
15	chirimenjako	285891478626169	2648656

Team Unagi

(JPN)

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

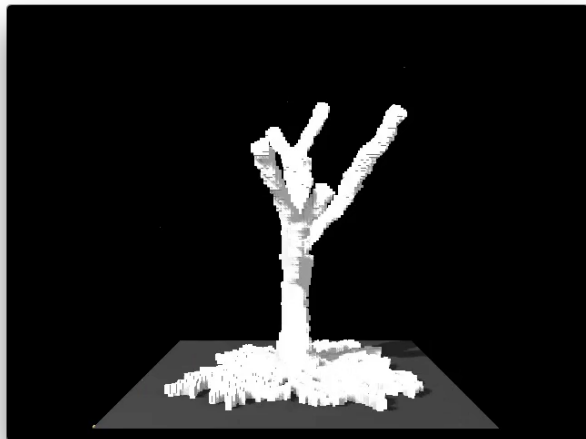
Rust

is the programming language of choice
for discriminating hackers.

\$1000 cash prize

Print and electronic copies of *Get Programming in Haskell*
donated by Manning Publications

Full Contest 1st Place Winner: Team Unagi



Conclusion

Rochester Institute of Technology; Department of Computer Science

- Matthew Fluet (Chair)
 - contest design and specification
 - back-end infrastructure (`exec-trace`; `gen-model`; `voxelizer-json-to-model`; `gen-assembler-trace`; `gen-disassembler-trace`; `gen-reassembler-1pass-trace`; submission evaluation and scoring)
 - target models
 - front-end infrastructure (Javascript/WebGL tools; registration and submission system; live standings)
 - website and social media
- Carlos Rivero
 - contest design discussions and review of specification
 - target models
- Hossein Hojjat, Arthur Nunes-Harwitt
 - contest design discussions and review of specification

Acknowledgements

Software and platforms used in the execution of the contest:

- GitHub Pages (<https://pages.github.com>)
- Google Apps Script (<https://script.google.com/home>)
- MLton (<http://mlton.org>)
- SMLtoJS (<http://www.smlserver.org/smltojs>)
- Thingiverse (<https://www.thingiverse.com>)
- Voxelizer (<http://drububu.com/miscellaneous/voxelizer>)
- three.js (<https://threejs.org>)

Languages and tools used in the execution of the contest:

- Standard ML
- Javascript
- Markdown, Jekyll
- make, bash, curl, bc, zip

Thank You!!

Thank you to all of the contestants for participating in the contest!!



Get ready for the ICFP'19 Programming Contest!!