The Third Annual ICFP Programming Contest

John Reppy
Bell Labs

Greg Morrisett
Cornell University

Huge thanks

- *Sponsers:*
  - *ICFP and ICFP sponsers*
    - *Compaq, EAPLS, Lucent, Sun, Microsoft Research, University of Montreal*
  - *Cambridge University Press.*
  - *O'Reilly and Associates.*
- *Running things:*
  - *Stephanie Weirich, Fred Smith, Dan Grossman, Dan Wang, Dean Eckstrom, Bill Holmes*

Overview

- *Aug. 26: we publish the task*
- *Teams have 72 hours to finish the task.*
- *Any team composition.*
- *Any programming language (that runs on our Linux boxes).*
- *Correctness counts.*
- *Features count.*
- *Efficiency counts.*

A Timeline

- *Feb. 2: I refuse to do the contest.*
- *Feb. 4: John signs on if I will. Sigh.*
- *Jun. 6: Martin asks if we've thought about the contest. We haven't.*
- *Jun. 14: John and I meet in Cannes.*
- *Aug: John & Fred put together sample code*
- *Aug 23: Web site up -- 70 teams registered*
- *Aug 24: Slashdot advertises the contest.*
- *Aug 26: 800 teams! The contest begins…*
- *Aug 27: Cornell web site crashes.*

The Task: A Ray Tracer

Implement a simple functional language called GML used to describe and render scenes.

- *dynamically-typed, postscript-style language*
- *geometric primitives (sphere, cube, etc.)*
  - *parameterized by a surface function*
- *constructive solid geometry operations*
  - *union, difference, intersection, etc.*
- *affine transforms (scale, rotate, translate)*
- *lights: directional, point, & spot*
- *render operation: spits out a .ppm file*

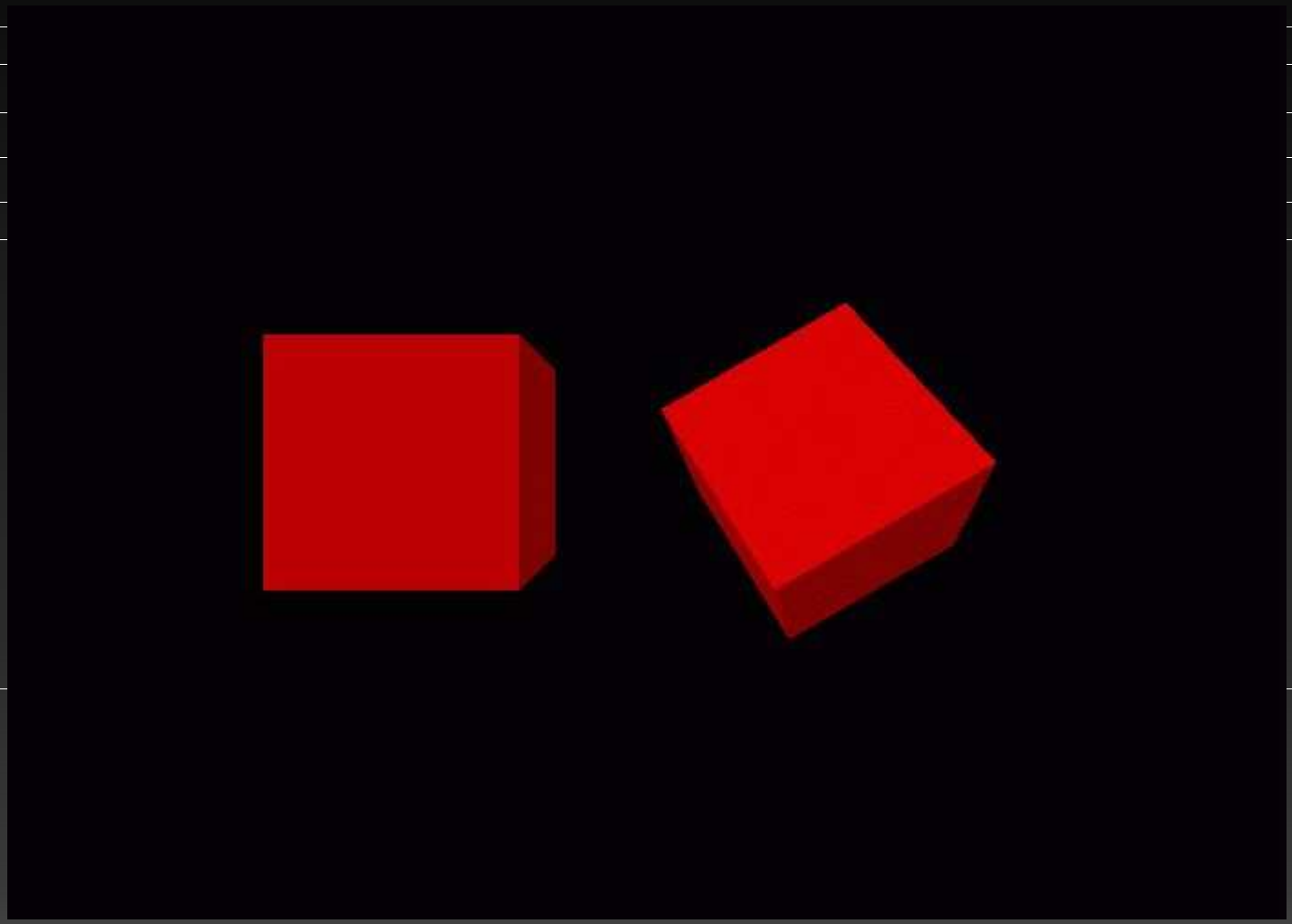
Example GML code

```
{ /v /u /face red 1.0 0.0 1.0} cube /cube1

%% the scene consists of two cubes
cube1 -1.5 -0.5 1.5 translate
cube1 30.0 rotatex 30.0 rotatez
0.5 -0.5 1.5 translate
union /scene

0.5 0.5 0.5 point %% Ambient
[ sun ]           %% Lights (not shown here)
scene
2                 %% Depth
90.0              %% fov
320               %% width (pixels)
200               %% height (pixels)
"cube.ppm" render
```

Example Image

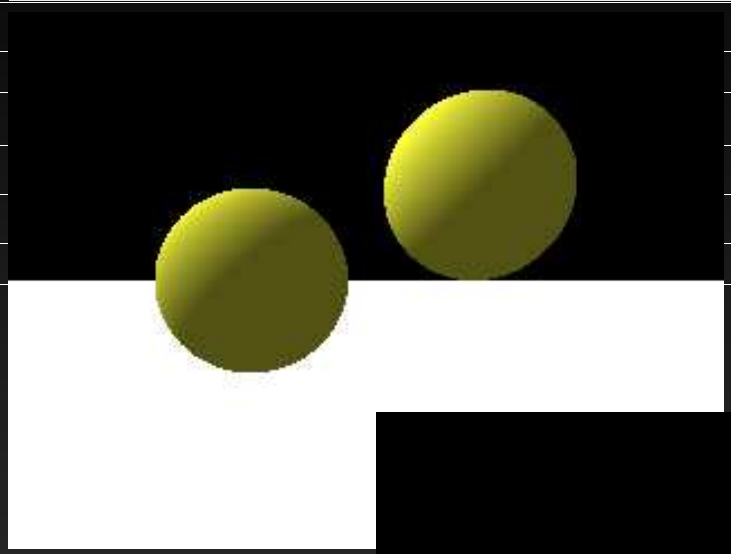


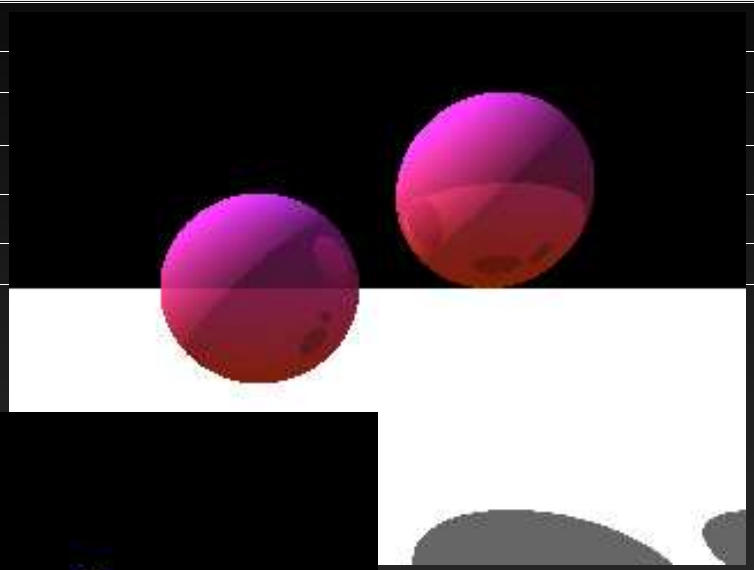
Three Tiers

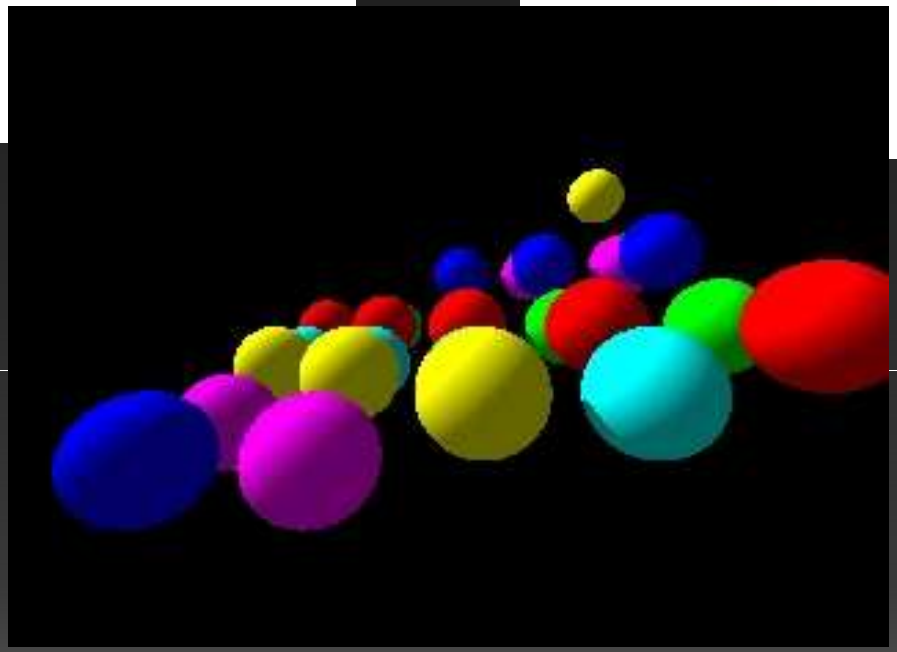
- *Tier 1: GML ops, planes, spheres, unions, & directional lights*
- *Tier 2: cones, cubes, cylinders, & point lights.*
- *Tier 3: difference, intersect, & spotlight.*

A Tier-n implementation would have to be at least as correct and significantly faster to beat a Tier-n+1 implementation.

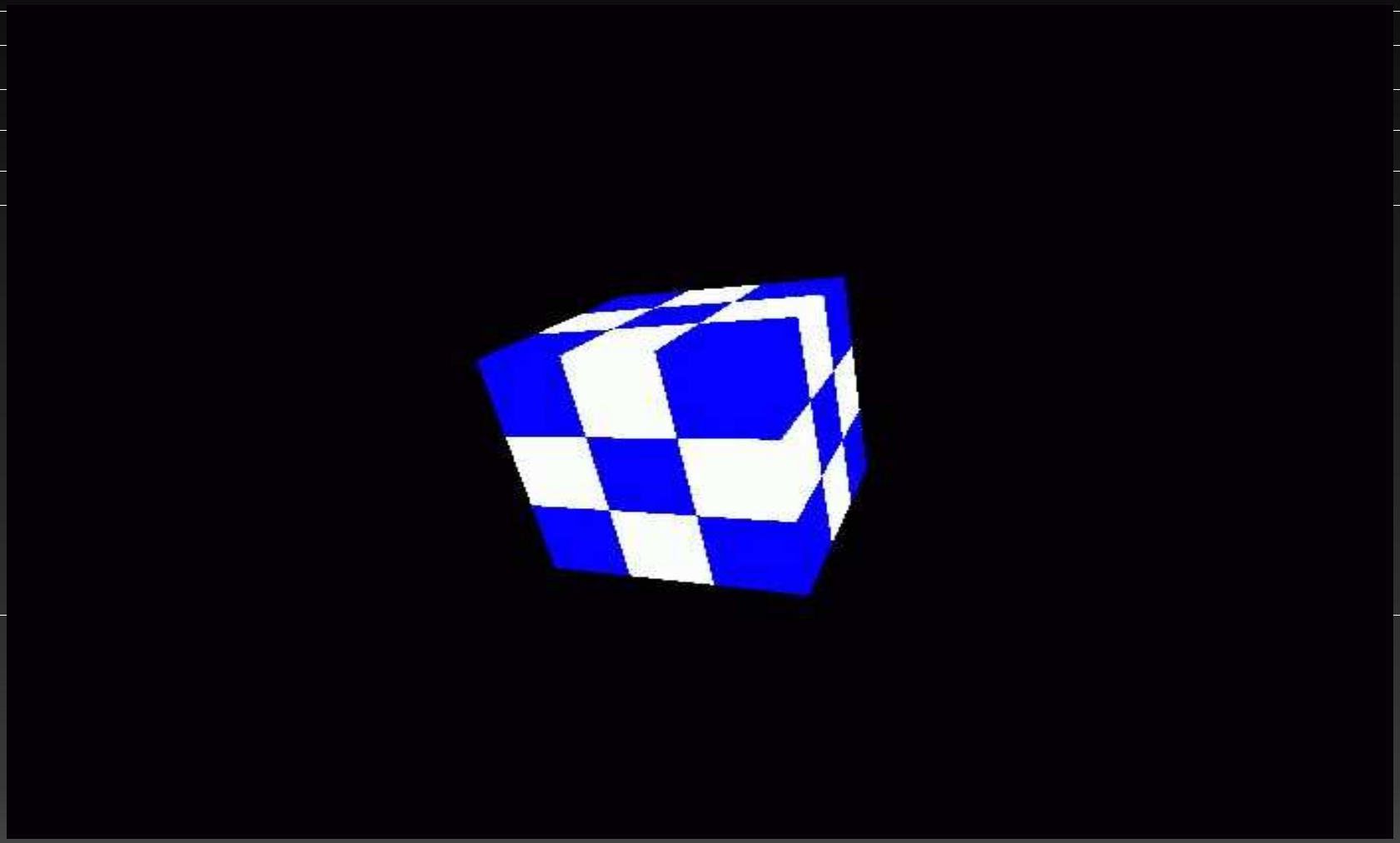
Some Tier-1 Images



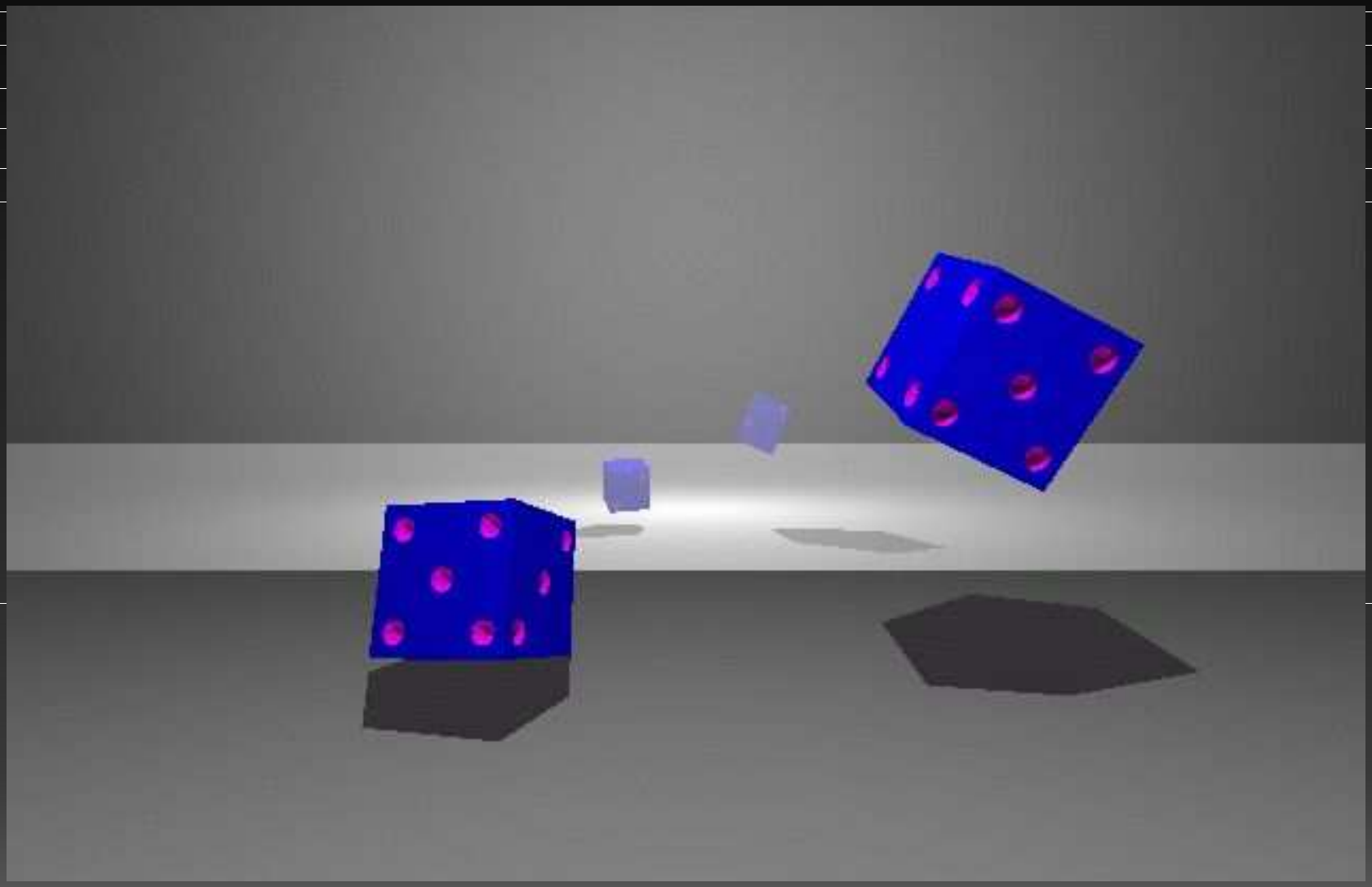




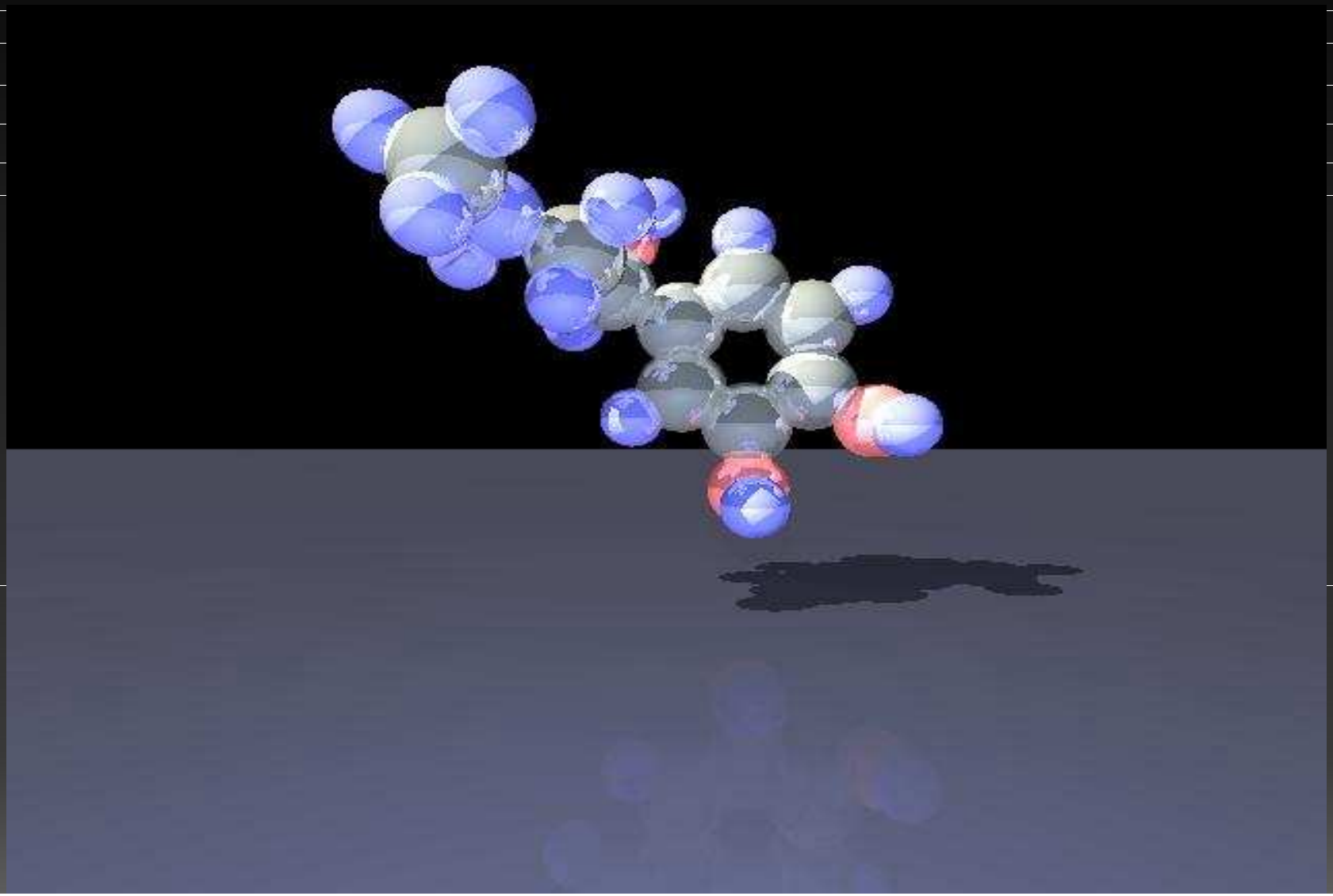
A Tier-2 Image



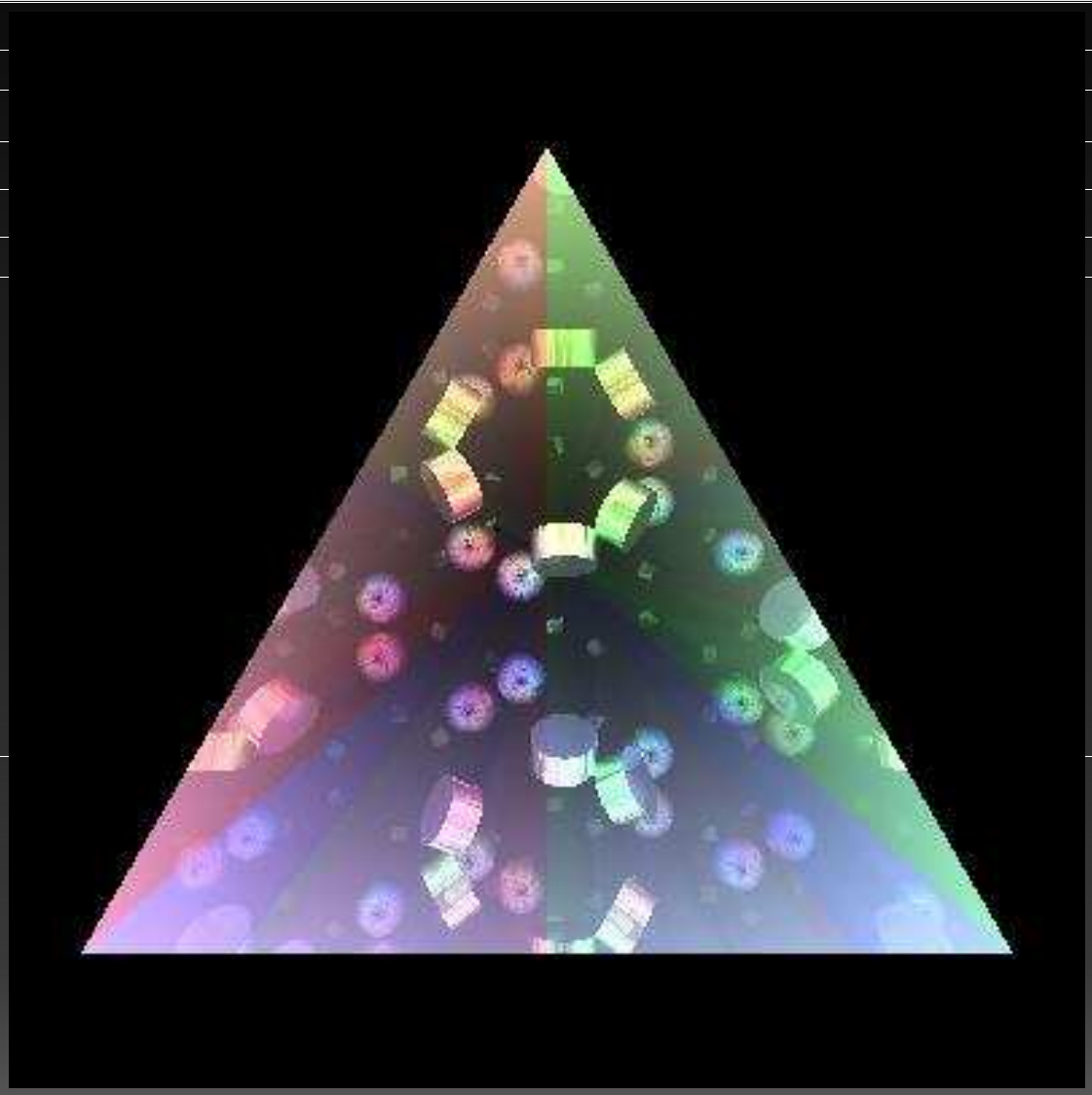
A Tier-3 Image



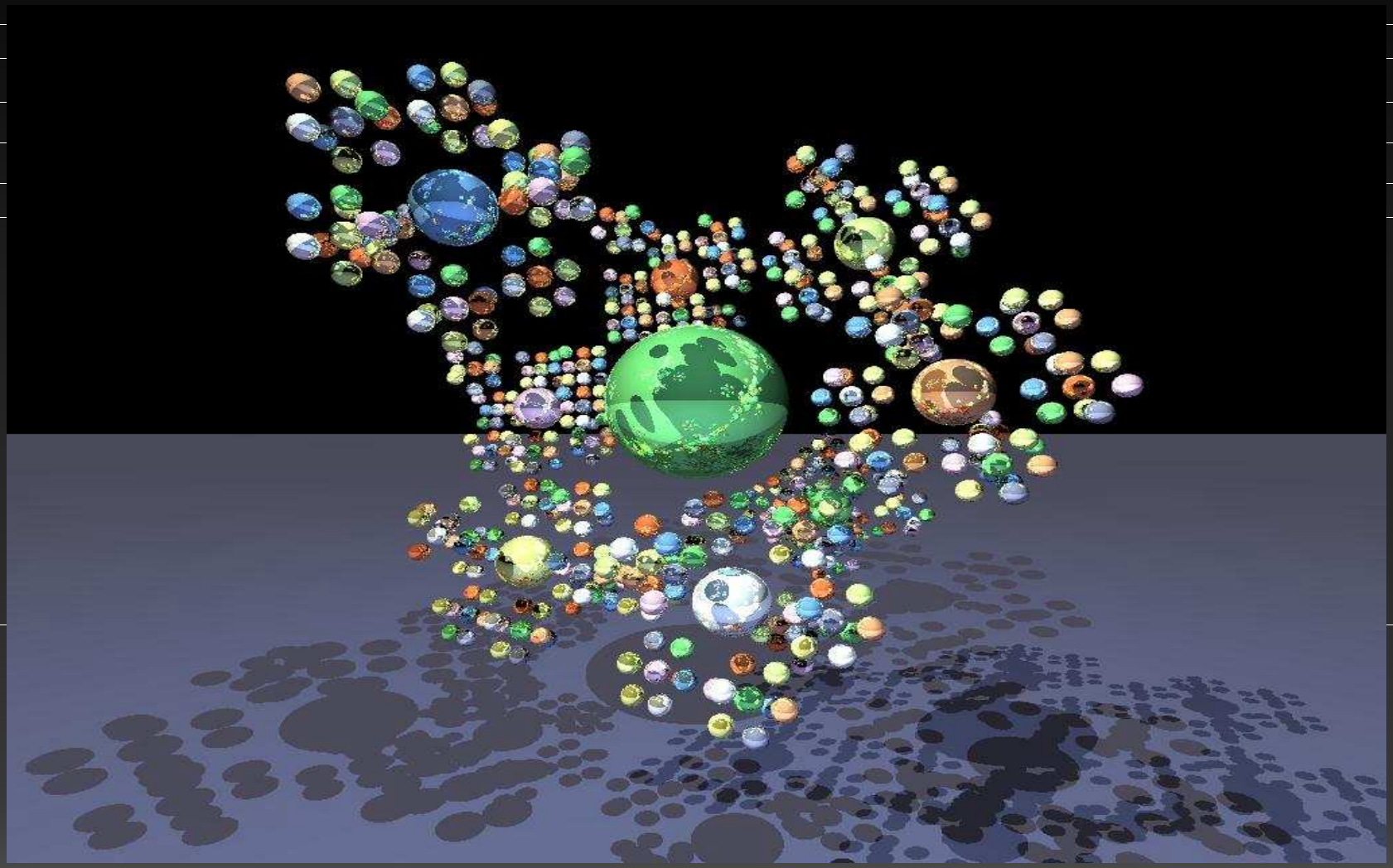
Adrenaline [Harley]



Kaleidoscope [Schmitt]

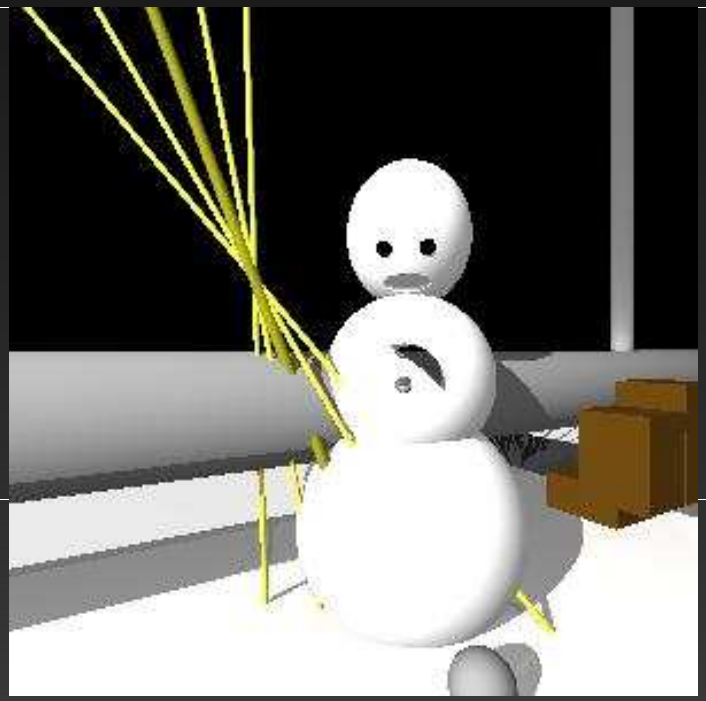


Fractal [Doligez & Harley]

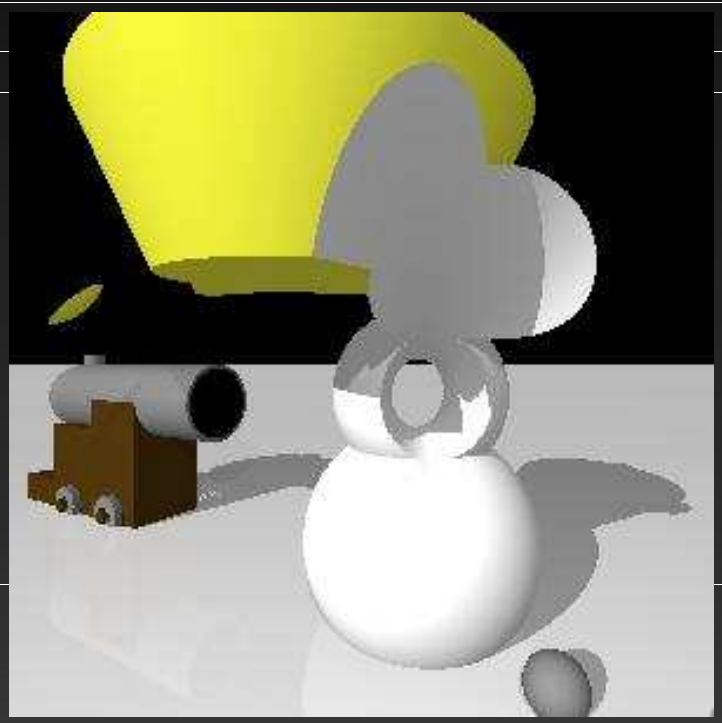


Example Bugs

Snowgoon [Smith & Yang]



Hmmm...



Subtle...

Some Statistics

- *Number of teams registered: > 800*
- *Number of entries submitted: 39*
- *Countries represented:*
  - *USA(18), Germany(4), Canada(3), Australia(3), New Zealand(3), France(2), UK(2), Poland, Switzerland, Belgium, Sweden, Netherlands*
- *Team size: 1-13*
- *# of versions of task description: 18!*

Languages Used

C/C++	7	Mercury	1
Clean	1	ML	9
Dylan	1	Perl	2
Eiffel	1	Python	1
Haskell	6	Scheme	2
Java	6	Smalltalk	2

How We Judged

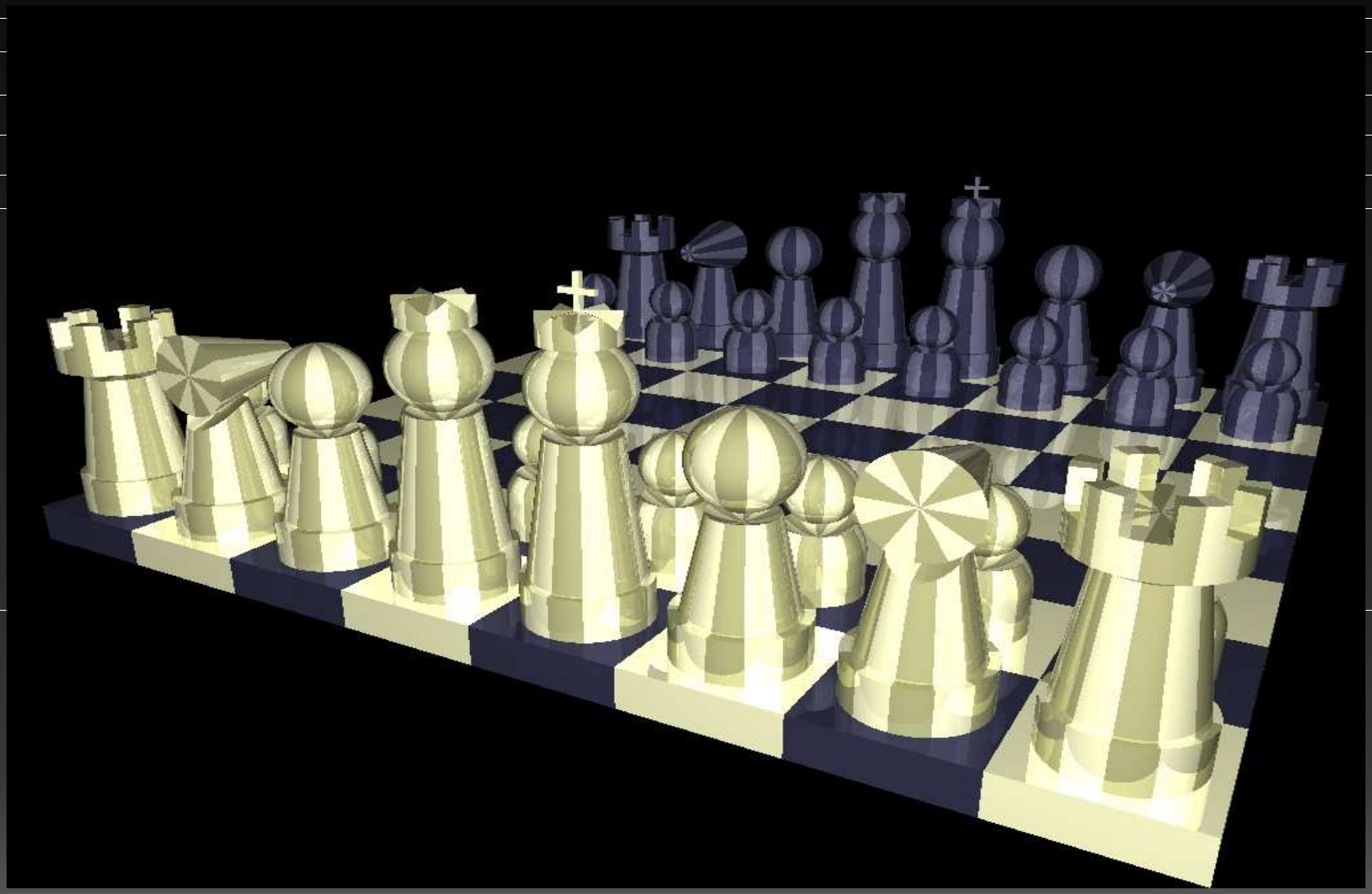
Several rounds to weed out entries:

1. *test GML primitive operations*
2. *test "correctness" of tier-1 images*
   - *avg. distance in color space from ref.*
   - *eyeball double-check*
3. *test correctness & speed of tier-2*
4. *test correctness & speed of tier-3*

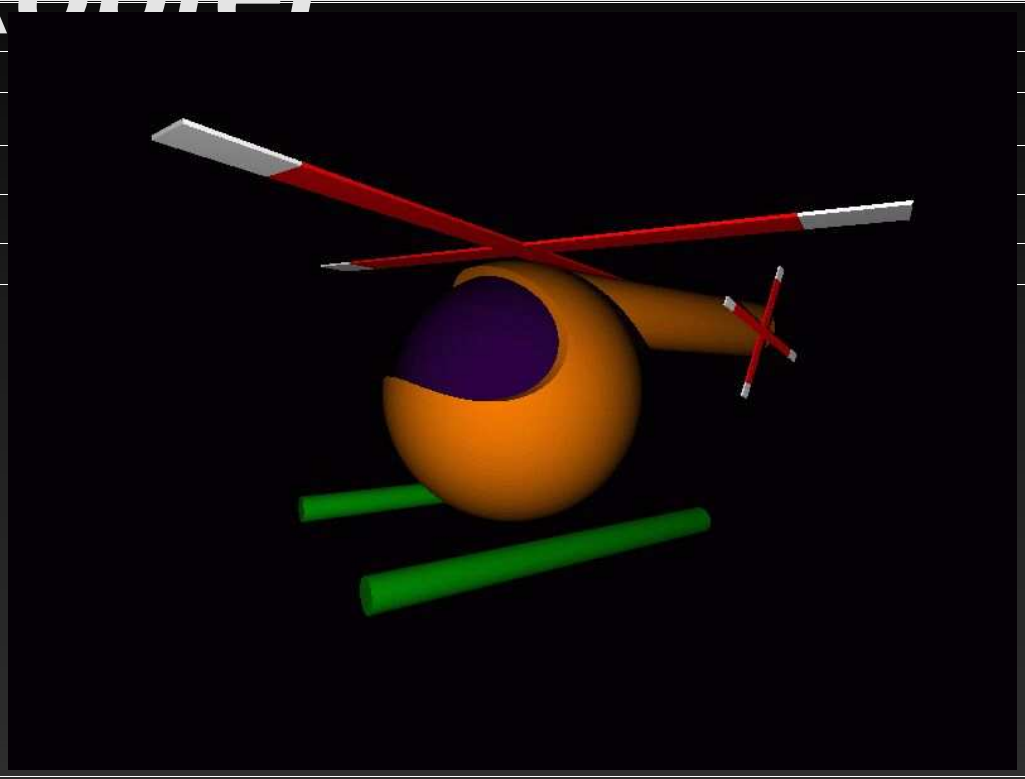
The Rankings

- *Judges Prize (best image)*
  - *($100+ books, journals, registration)*
- *4th place*
- *3rd place*
- *2nd place*
  - *($400+ books, journals, registration)*
- *1st place*
  - *($1000+ books, journals, registration)*

Judges' Prize Image



Judges' Prize: Team Helikopter



Congratulationss to
Andreas Rossberg & Leif Kornstaedt!

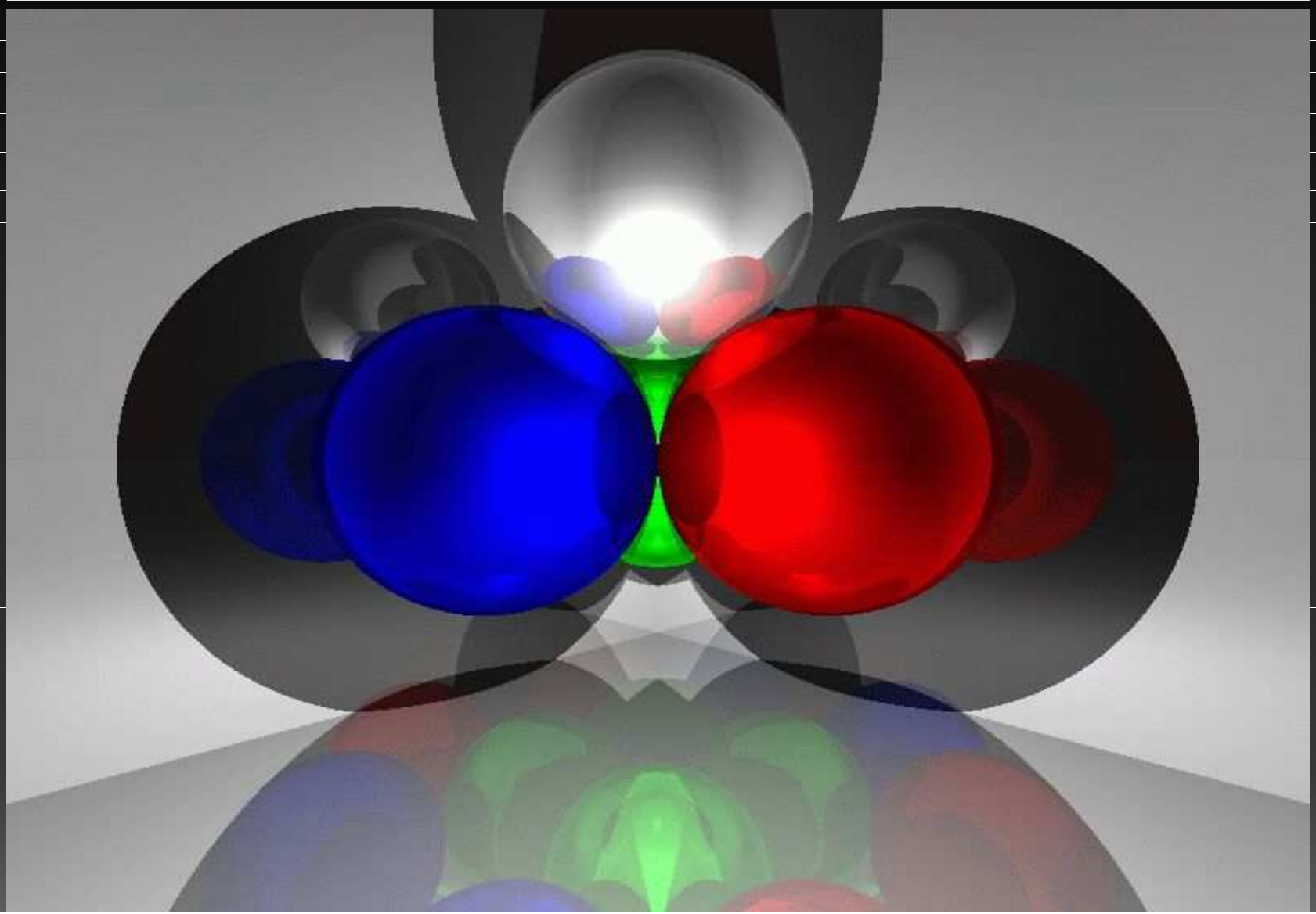
The Judges Proclaim:

Team Helikopter is an extremely cool pair of hackers.

4th Place

- *13 team members!*
- *Australia, Belgium, and UK members*
- *Tier-3 entry*
- *Bounding boxes*
- *coded in….Mercury!*
- *Congrats to the Merry Mercurians! (R. Becket, T. Dowd, F. Henderson, S. Taylor, P. Ross, D.G. Jeffery, T. Leeuwenburg, T. Conway, R. Jeschofnik, M. Brown, M. Liddell, D. Overton, P. Eckersley)*

A Mercurial Image



3rd Place

- *5 team members*
- *Tier-3 entry*
- *Bounding boxes*
- *100x faster on Chess than 4th place*
- *coded in...*

  Haskell!

- *Congrats to Team Galois Connections!*
  (J.Launchbury, A.Gill, J. Lewis, T. Nordin, T.Sauerwein)

Cubist Galois

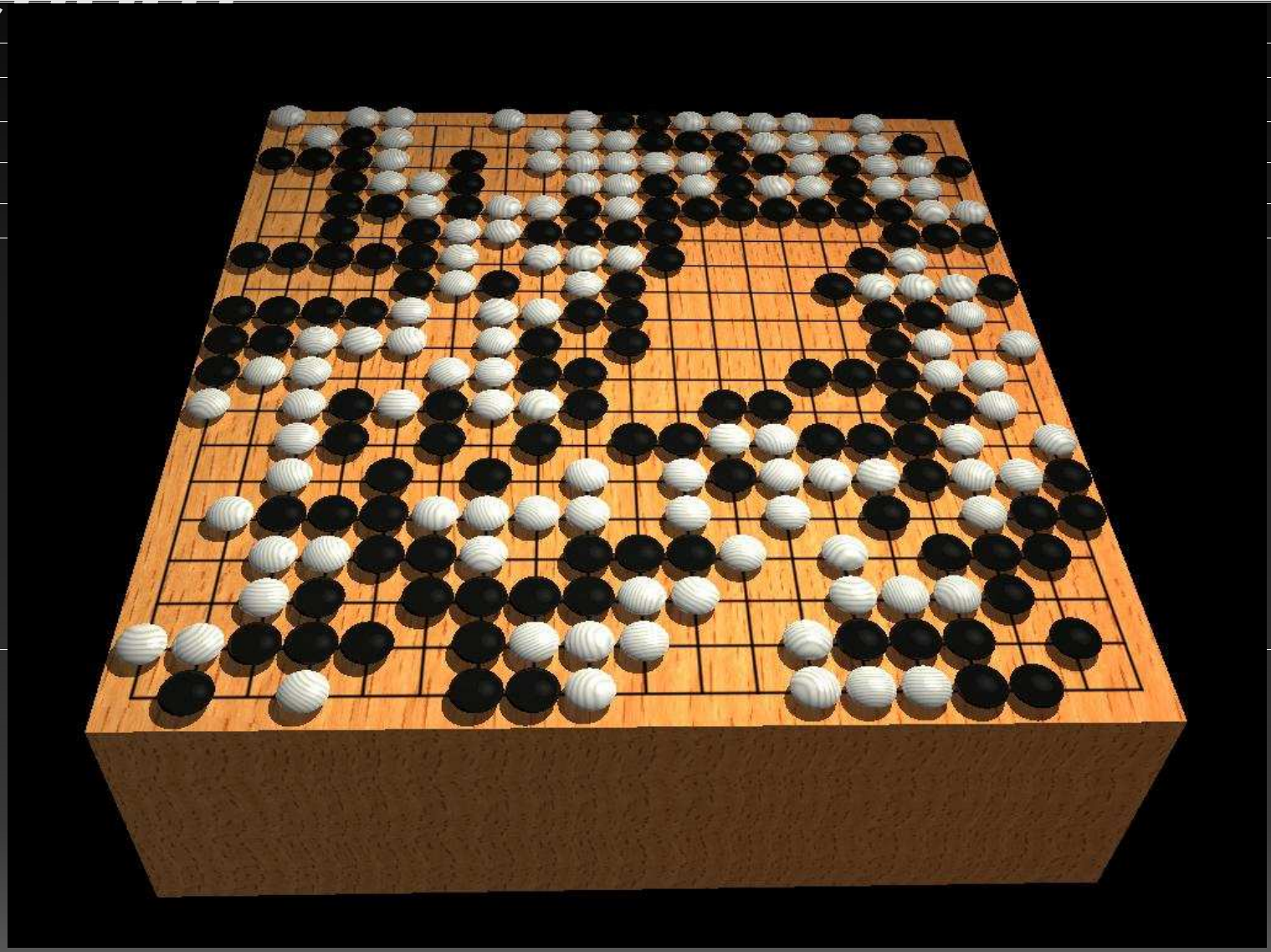
2nd Place (Prize Winner)

- *7 team members*
- *Tier-3 entry*
- *Bounding spheres, BSP-trees*
- *5.5x faster on Chess than 3rd place*
- *Coded in…*

  O'Caml!

- *Congrats to Team Camls 'R Us*
  (D.Doligez, S.Ailleret, P.Cuoq, R.Harley, F.Le Fessant, X.Leroy, A.Schmitt)

A Go game [Doligez & Cucq]



The Judges Proclaim:

O'Caml is a fine *programming tool for many applications.*

(and the INRIA team can run the contest next year...)

1st Prize

- *4 team members*
- *Tier-3 entry*
- *Bounding spheres, GML optimization*
- *4 different implementations!!!*
- *8x faster on Snowgoon than 2nd place*
- *Coded in…*

O'Caml!

- *Congrats to Team PLClub!!! (J.Vouillon, V.Gapeyev, H.Hosoya, & E.Sumii)*

The Judges Exclaim:

O'Caml is the *programming tool of choice for the discriminating hacker.*

Words of Wisdom

"An interesting observation is that every bug that took more than three minutes to find was a problem with a sign in the code."

-Lennart Augustsson

"A correct graphics program has an even number of sign errors..."

-Jim Blinn