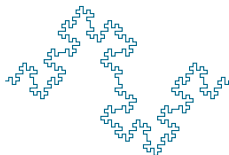


# Bend or Feed Back

Gábor Melis

@GaborMelis | <http://quotenil.com> | [mega@retes.hu](mailto:mega@retes.hu)

*Franz Inc., Fixnum Services*



2014 December 12

# THE PROBLEM

There are three parties that each want something different.

- ▶ Organizers: crowdsourcing of data science, visibility.
- ▶ Contestants: winning, learning, having fun.
- ▶ Platform Providers: sustainable business.

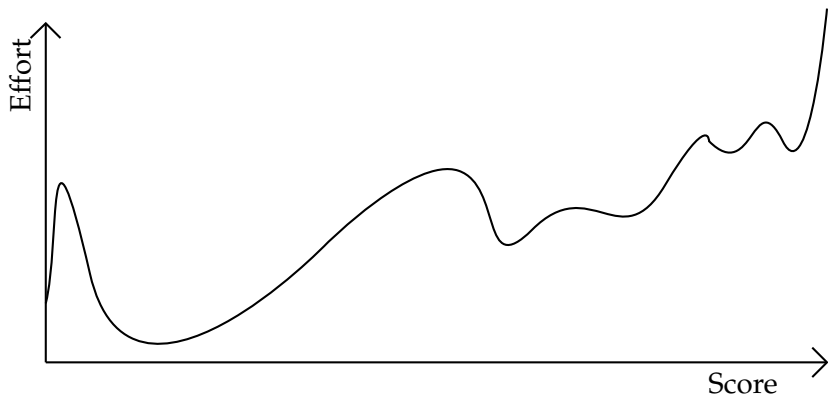
# BACKGROUND

My experience comes from various competitions:

- ▶ Computer Games Olympiads (Hex)
- ▶ Google AI Challenges (Tron, Planet Wars)
- ▶ Kaggle contests

# MODELLING CONTESTS

Think of the contest as a solution space where one can move by taking small steps.



# MODELLING CONTESTANTS

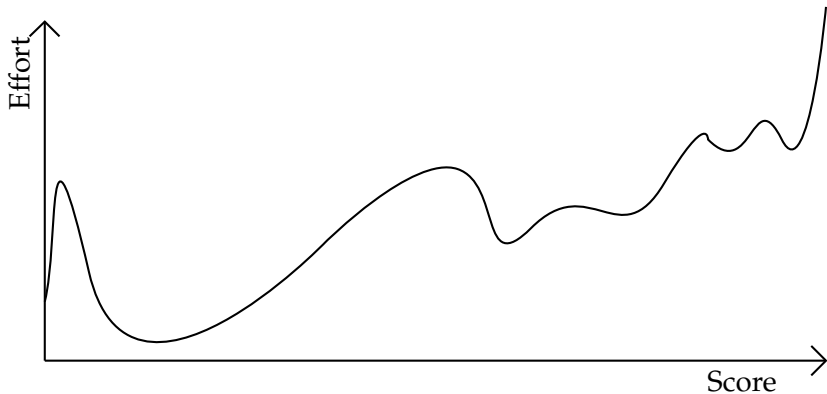
Agents (contestants) move around in this solution space.  
Agents have different intrinsic **strength** that determines how high they can climb, and they have an **energy** level that goes down as they tire. Agents are reenergized by rewards.

# SUFFICIENTLY QUICK, UNIQUELY INSIGHTFUL AND ROBUST REINFORCEMENT LEARNERS



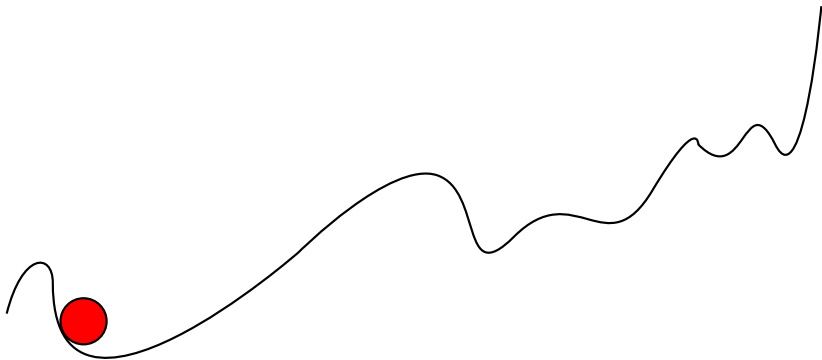
## SETTING THE CONTEST UP

Define the rules precisely. Map solutions to scores.



## SETTING THE CONTEST UP

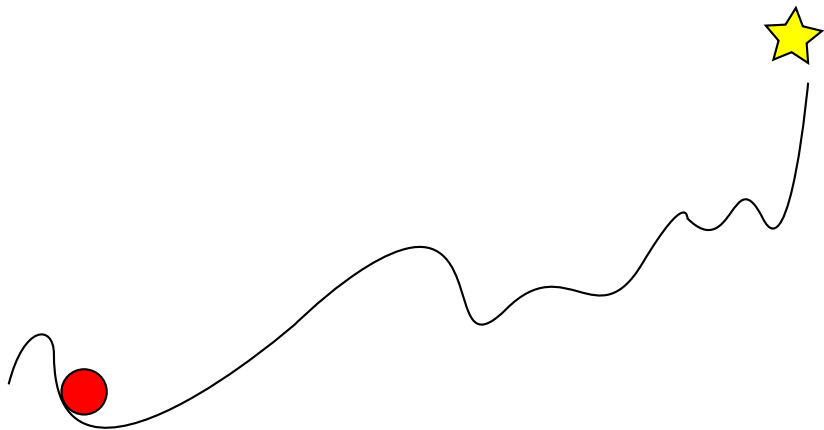
- ▶ Make it easy to start by cleaning data.
- ▶ Oil the wheels: don't require use of a certain framework or language.
- ▶ But do provide starter kits in the most popular languages.





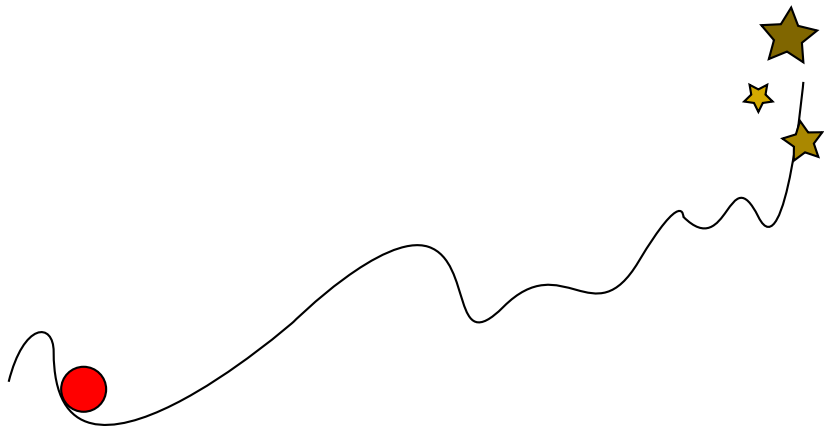
## SETTING THE CONTEST UP

Put a very desirable reward at the end.



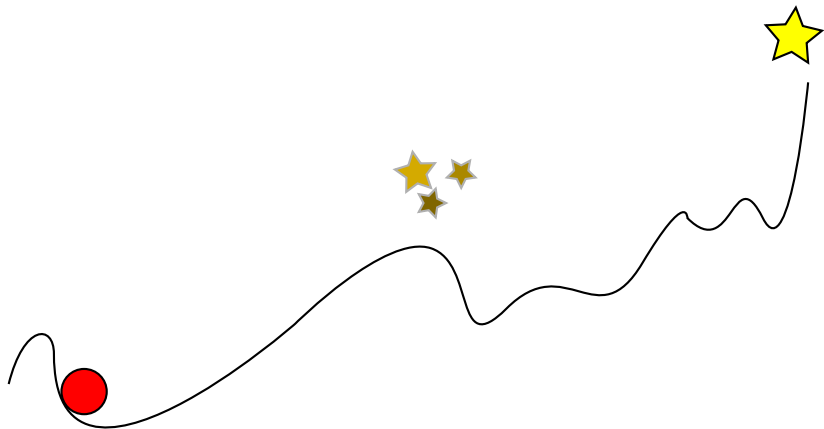
## SETTING THE CONTEST UP

Maybe make the outcome a tiny bit uncertain. Scares very skilled people off, but may attract more contestants overall.



## KEEP MOVING

Squirrels run out of energy, put recharges along the way (leaderboard, milestones, interviews, visualization of games).



## KEEP MOVING

Make leaderboard feedback noisy enough to make people overestimate their chances, but not too noisy to make it obviously useless.



## KEEP MOVING

Spread rewards out to keep agents hooked. Easier with short contests.



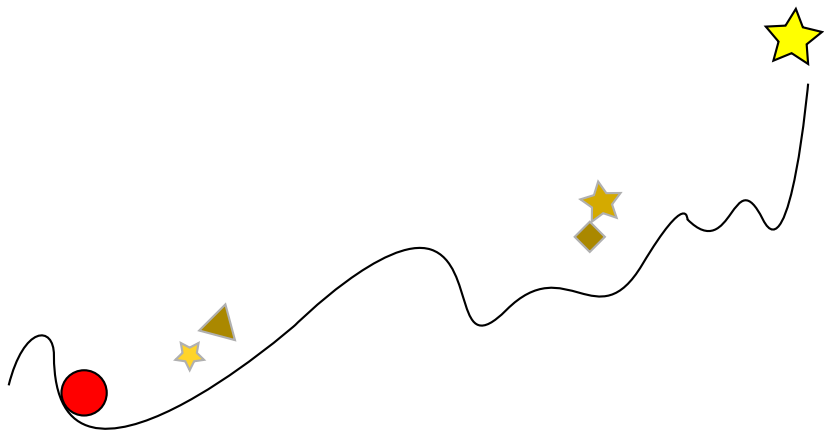
# REWARDS

There are different kinds of recharges.

- ▶ The Prize itself.
- ▶ Advance in rank on the leaderboard.
- ▶ Knowledge.
- ▶ Fame.
- ▶ Bragging rights.
- ▶ Fun.
- ▶ Scientific progress.
- ▶ Reusable software components.

# REWARDS

Combinations of different kinds of rewards are more powerful.



# THE MIDDLEMAN

What does the platform provider do?

- ▶ *Completes* the first phase of the project (data munging, customer interaction).
- ▶ Makes sure the contest is specified and run well.
- ▶ Tries to keep the contestants happy.
- ▶ Most importantly, it puts its reputation on the line.



# SHARING

- ▶ Information sharing allows cross-pollination of ideas.
- ▶ Sharing in the late stages increases luck factor.
- ▶ Sharing too early leads to loss of diversity.

## REWARDS DRYING UP

What's the effect of doing multiple contests?

- ▶ In the beginning, the main effect of prizes is indirect: higher level of competition, more visibility, more bragging rights.
- ▶ But rewards tend to diminish. Knowledge acquisition slows down. Nobody listens to your bragging anymore.
- ▶ For experienced contestants, these intangible rewards get harder to come by over time, but the prize alone rarely makes up for that.

## ISSUES WITH PRIZEFIGHTING

- ▶ So due to drying up rewards, top contestants are lost over time.
- ▶ Prizes could counter that but they are often small and have too high variance to warrant the effort.
- ▶ It's all or nothing: there is no way to contribute a small effort and get compensated.
- ▶ Organizers get suboptimal solutions (low diversity, missed opportunities to ensemble).

# SUSTAINABILITY

To attract and retain talent, contests need high prizes with lower variance.

- ▶ Find problems with high intrinsic value.
- ▶ Award prizes proportionally to the value produced.
- ▶ Ensemble solutions automatically as part of the evaluation.
- ▶ Make the ensembling method itself part of the contest.
- ▶ Divide the prize between multiple people proportionally to the contribution to the ensemble.

# 50 WAYS TO SPOIL YOUR CONTEST

- ▶ Insufficient data preparation (anonymization failures, corrupted data).
- ▶ Misspecified evaluation rule (mismatch with customer utility).
- ▶ Too little or too much data.
- ▶ Unstable rules or data.
- ▶ Sharing of ideas, code, data is allowed too late or too early.

# SUMMARY

- ▶ Gamify aggressively.
- ▶ Automatically ensemble solutions and split prizes.
- ▶ Quantify historical contestant behavior (leadeboard position vs submission data, ensemble analysis).