Games Are Math

10 Core Mechanics that Drive Compelling Gameplay

Raph Koster

President, Metaplace

Sure, let's start with a very simple one.

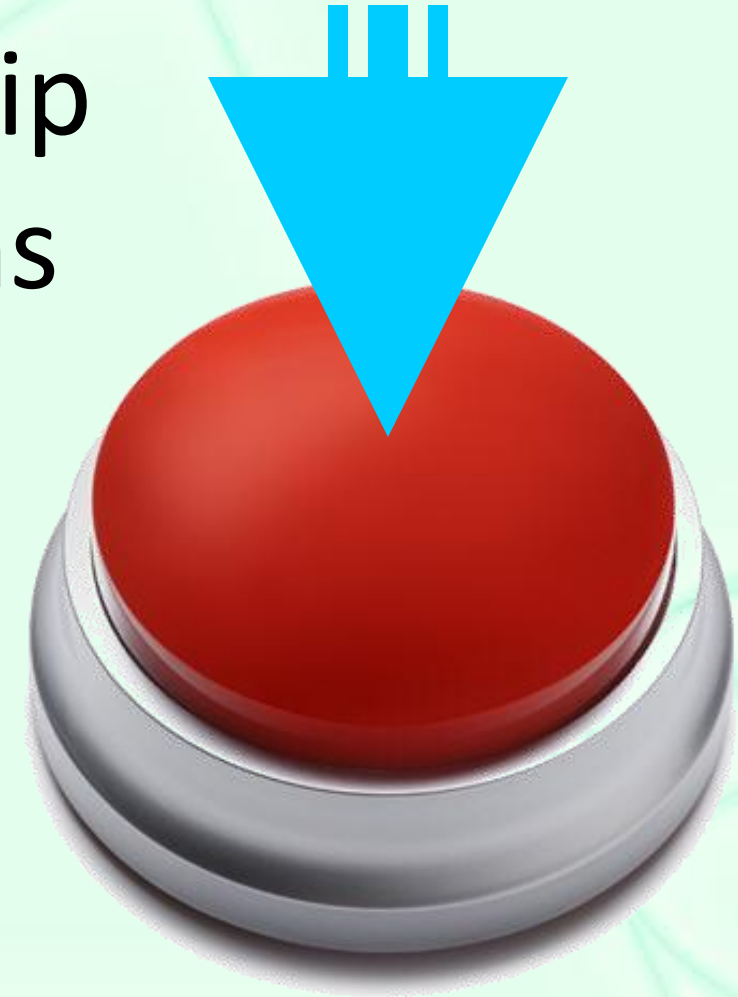
Tossing a coin.

Got one?

Good.

Heads or tails?

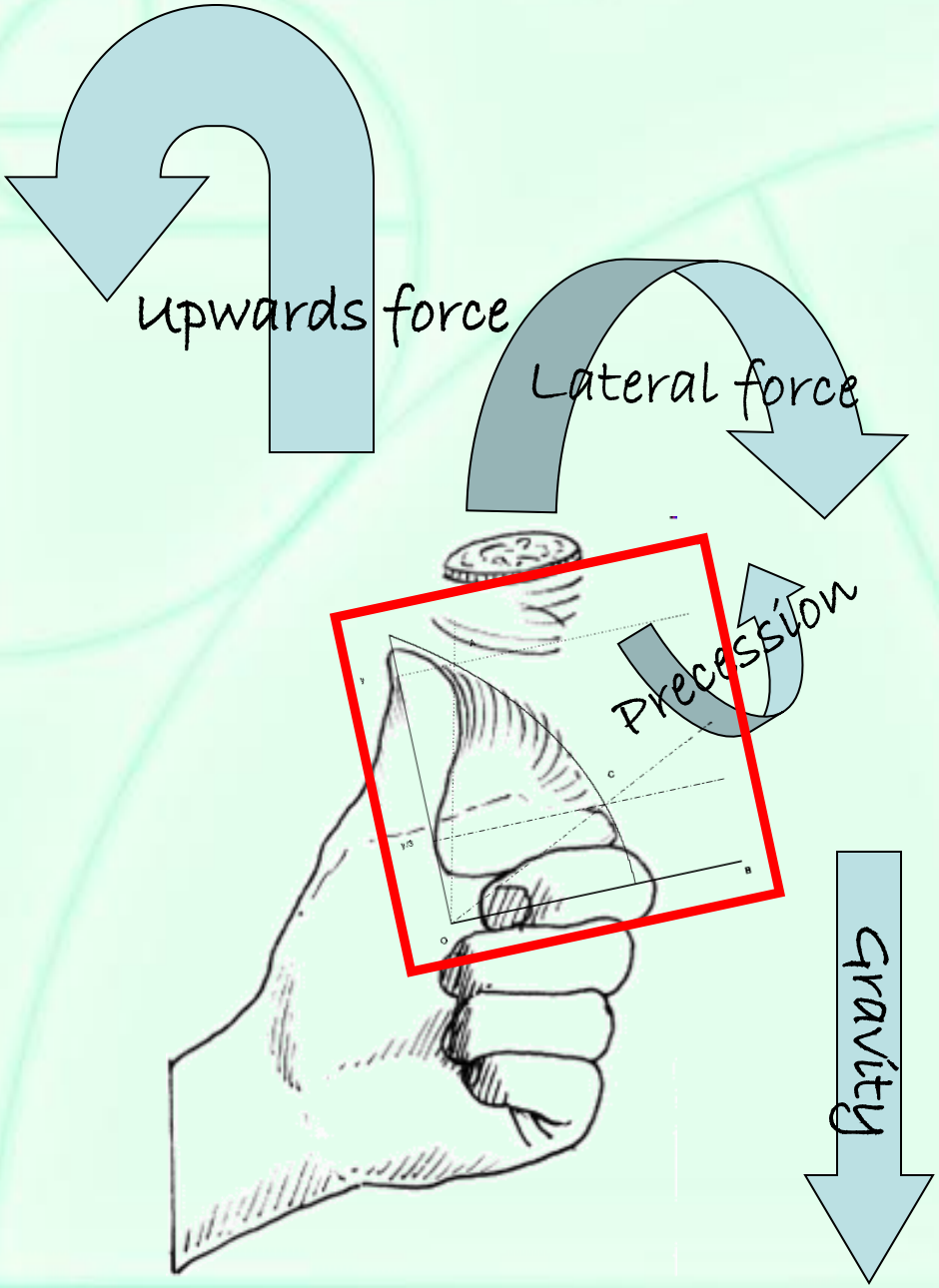
Let's all flip our coins now



Feedback


IN GOD WE TRUST
LIBERTY
2005
D
UNITED STATES OF AMERICA
E·PLURIBUS
·UNUM·
ONE CENT

The core model

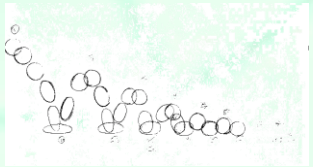


Slight weight bias towards tails

Distance to surface

Ignored factors:

- surface elasticity
- coin elasticity
- bounces
- surface irregularity
- etc

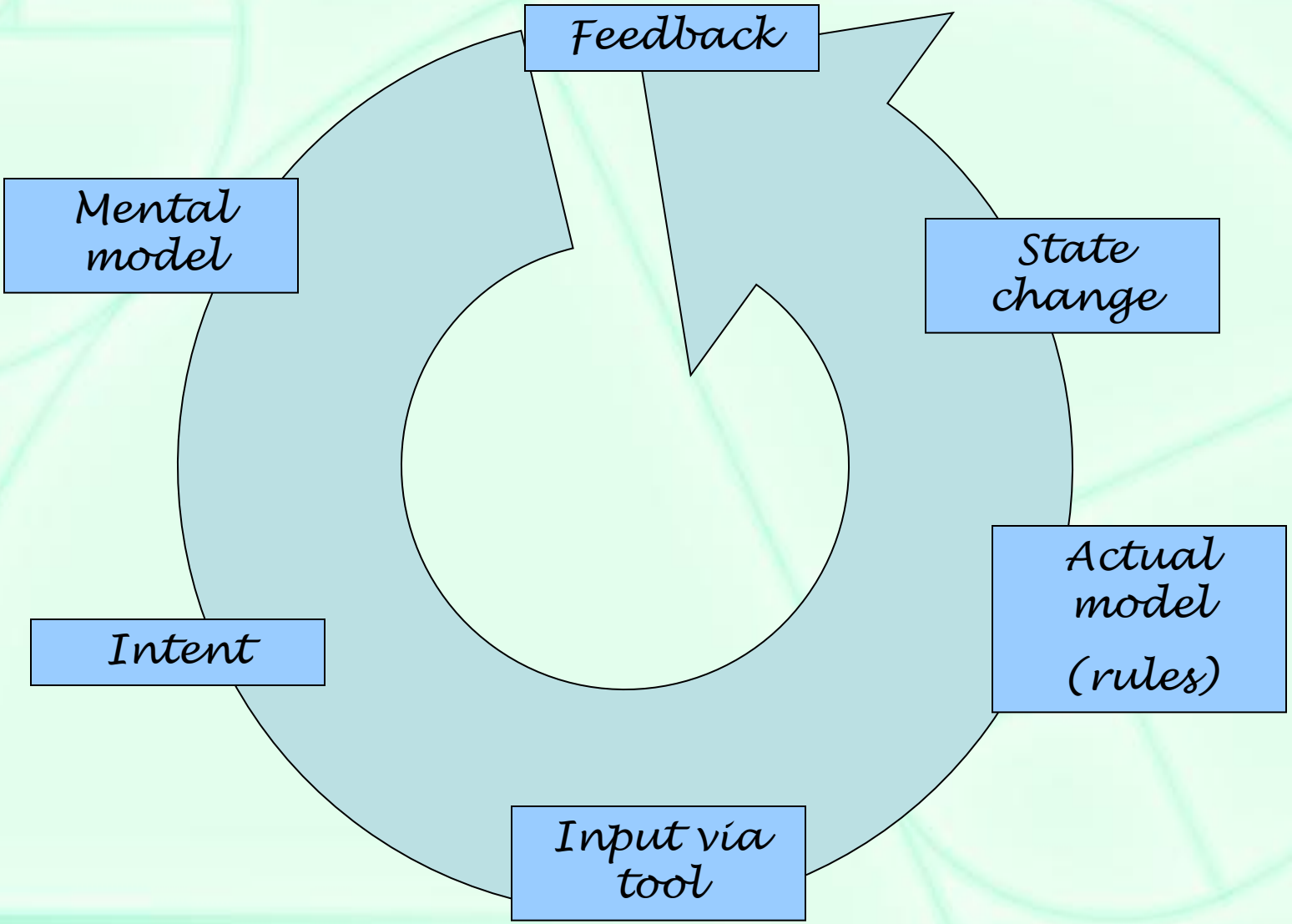


This is a game grammar approach

- A coin is actually a very small model
- We learn how games work via
  - Observation
  - Hypothesis
  - Experiment
  - Conclusion

Black box

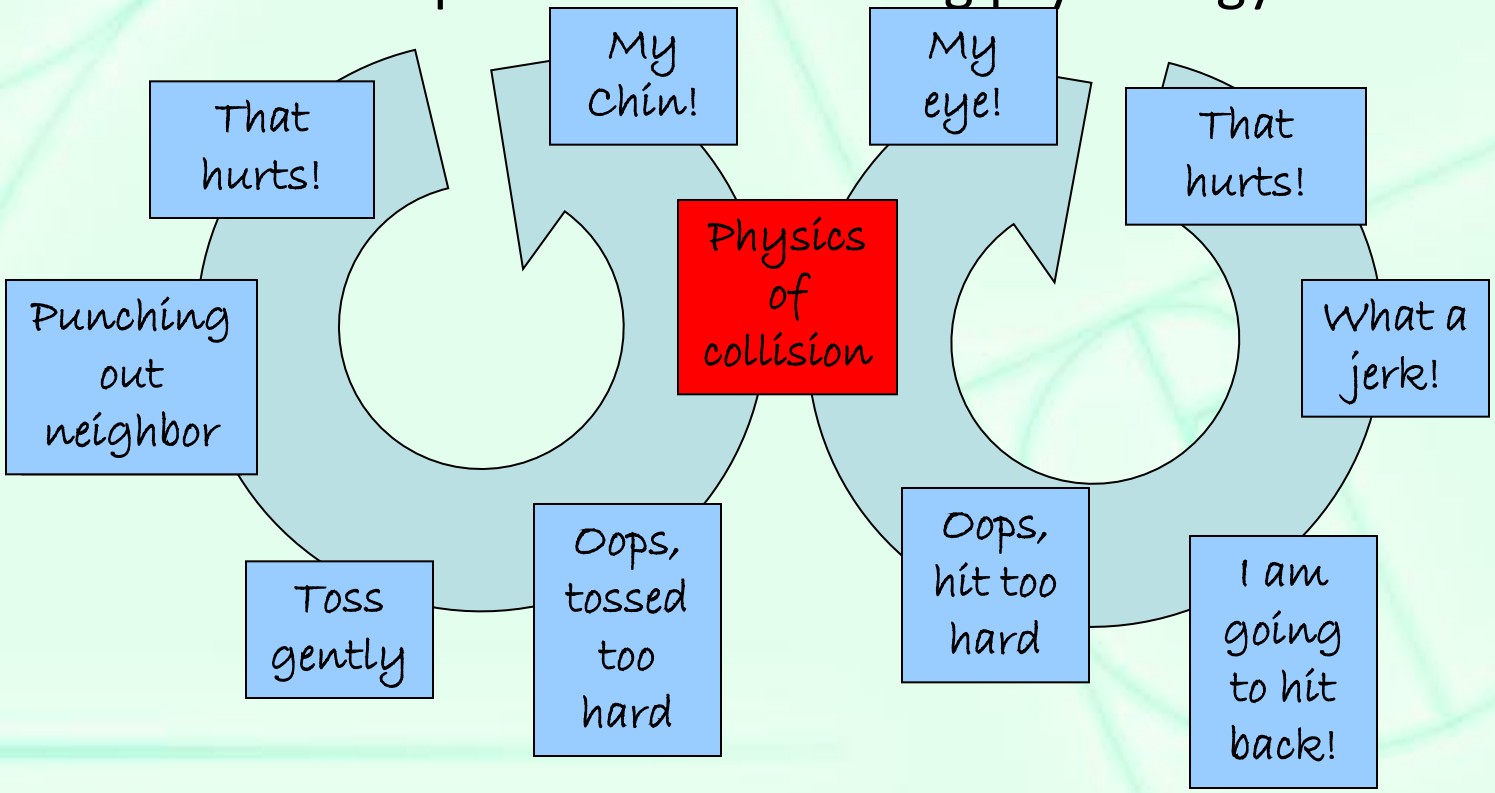
The basic game grammar model



Throw the coin at your neighbor!

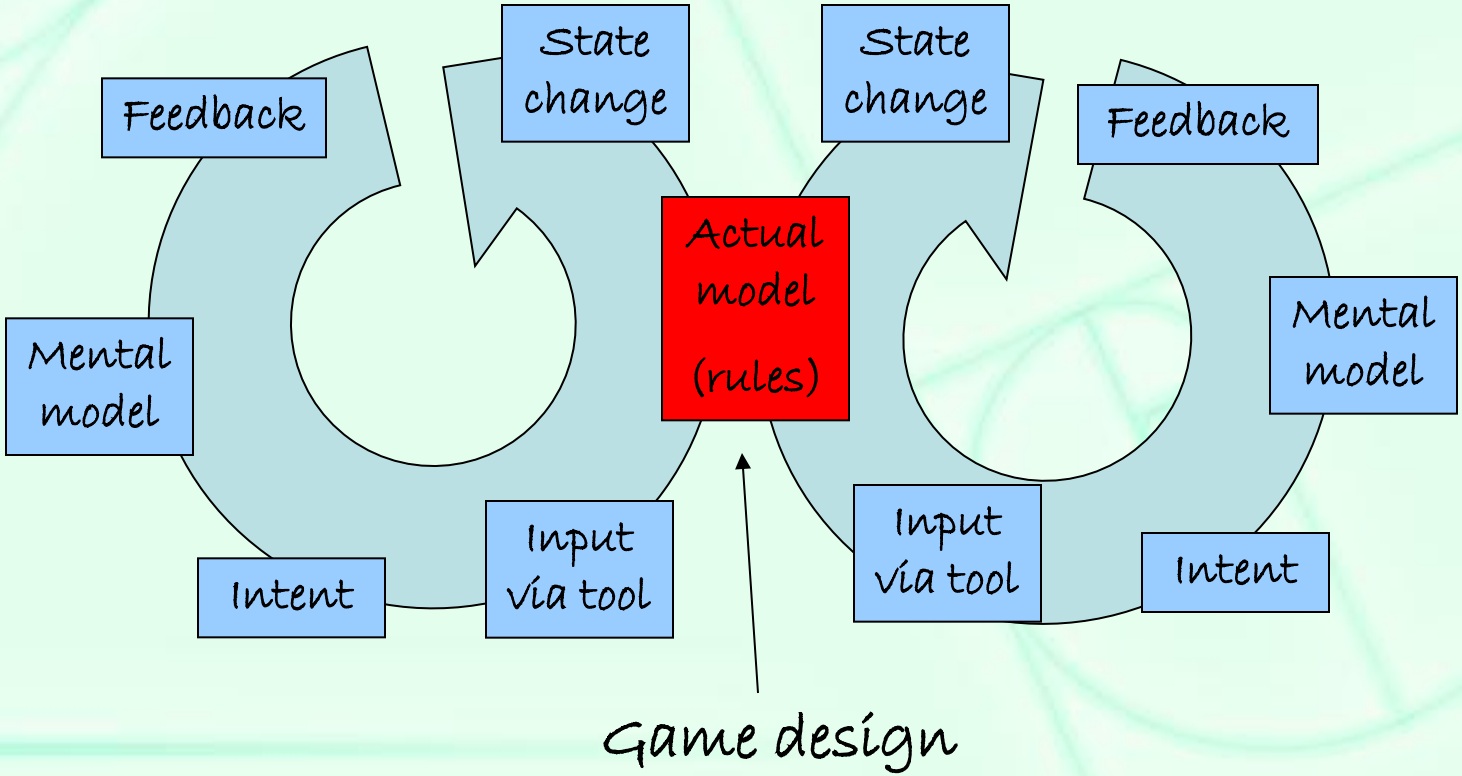
Game grammar assumes an opponent

- The opponent can be a very simple model
  - A coin, gravity, subtle variations in different sorts of coins
- Or a complex one
  - A coin and a person with differing psychology



This opponent can be described in math

- We don't have the math for people except in a general sense
- But we do for quite a lot of other things
- And in fact, usually game rules sit in between people, constraining their possible actions



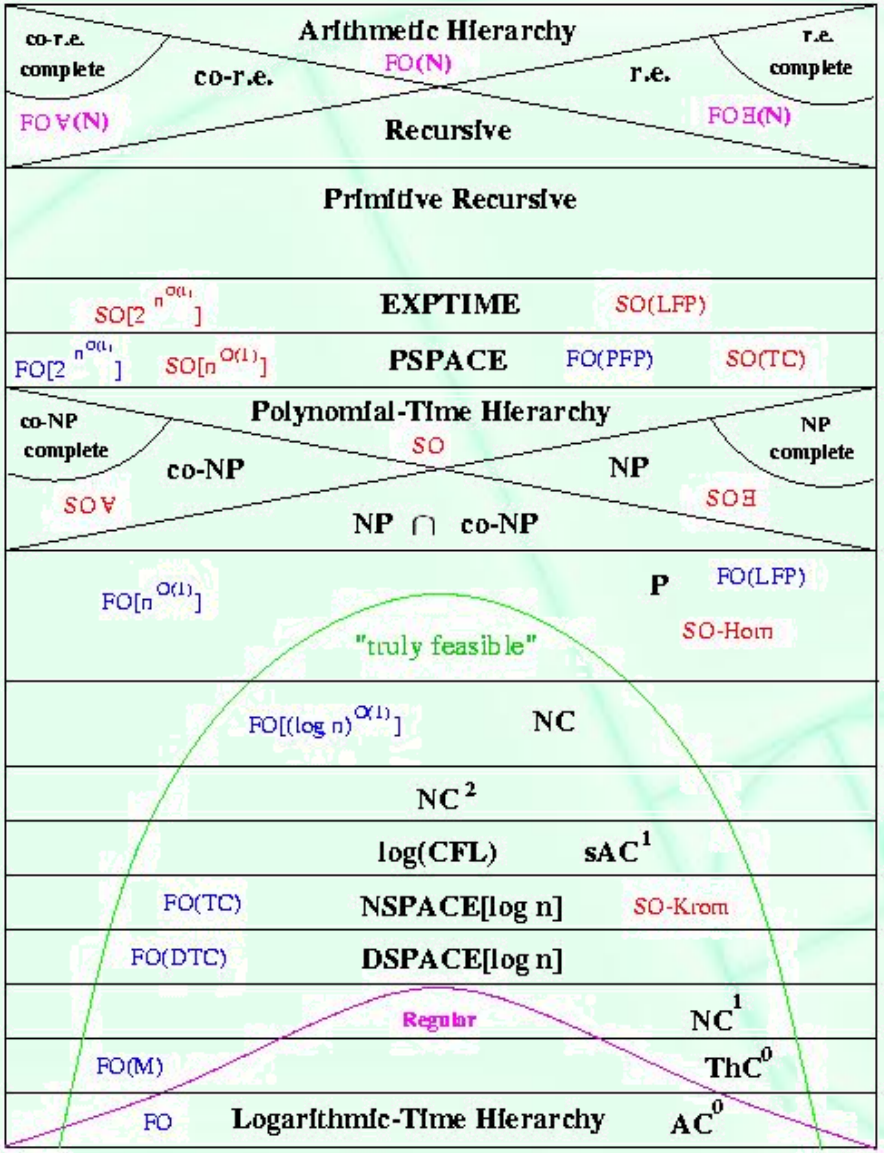
Classic example

Take turns picking numbers from 1 to 9.

Can't pick a number your opponent already picked.

If your set equals 15, you win!

Lots of kinds of math

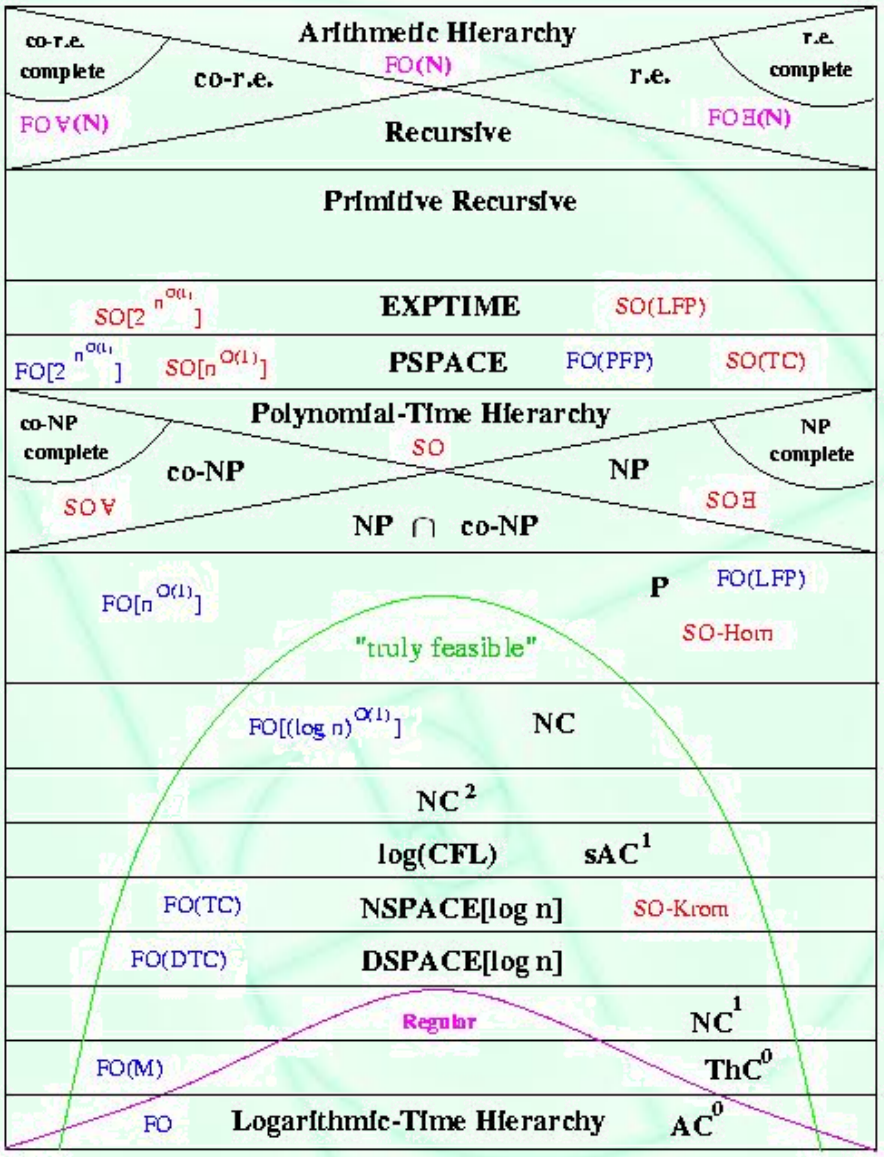


Complexity class

A set of problems of related complexity

Complexity is defined as "how long it would take to computer given a certain amount of resources."

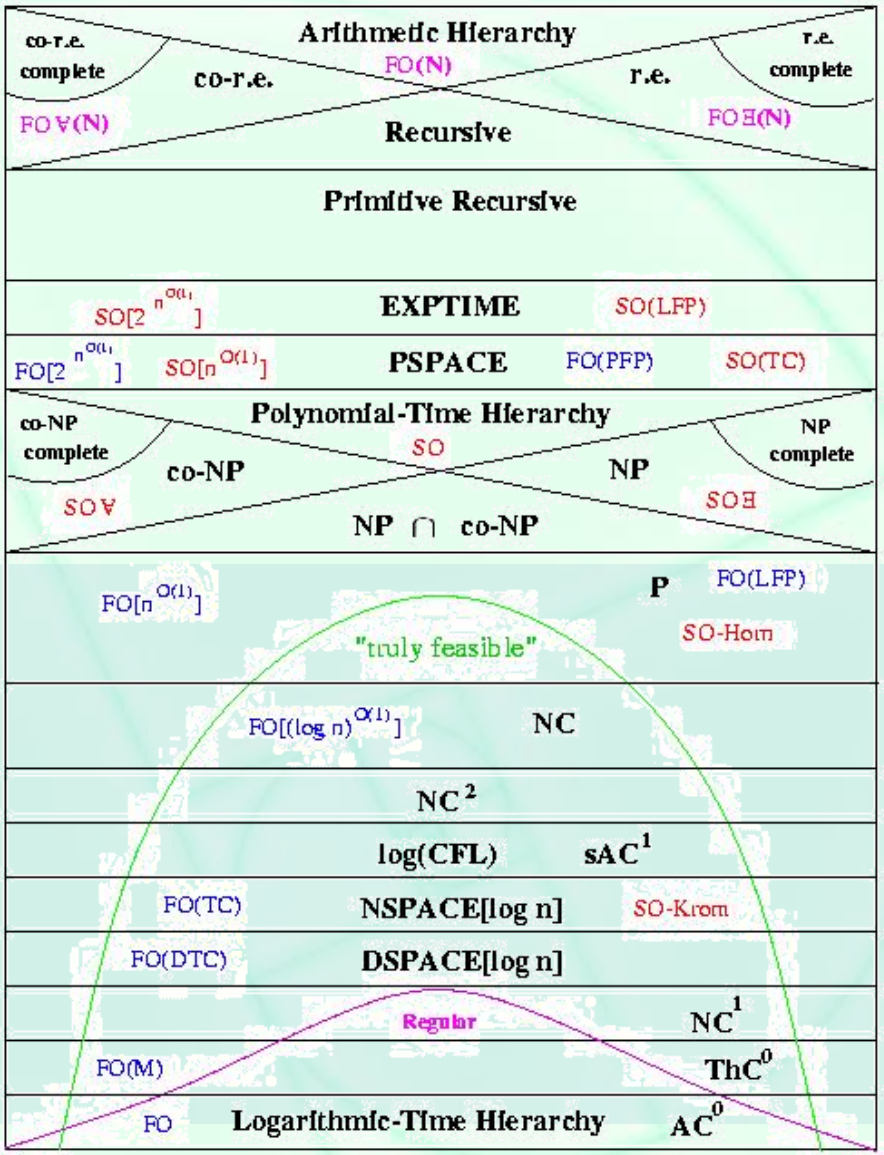
488 extremely geeky definitions of classes



Easy vs hard problems

- A problem is hard if instead it needs x^2 or x^x steps.
  - Because suddenly, we don't know when you will finish.
  - These are NP problems.
  - NP problems often have easy solutions. They just aren't guaranteed to work for **every** set.

- A problem is "easy" if you can ALWAYS solve it in a guaranteed X steps. ANY number of steps, for any version of the problem.
  - (This is called polynomial time, x steps, x^50 steps, whatever)
  - These problems are "P" problems.



The brain is a computer

- But it isn't like the computers we use every day, which function on one level

- For one thing, we have different measurable sorts of intelligences, whereas computers only have one: computation speed

- By any measure when compared to that, we suck next to a solar powered calculator
  - Even though our processing power is estimated at 100 teraFLOPS

(Latest announced Radeon card can do 2.5 teraFLOPS btw)

- And yet, a huge amount of what we do is theoretically computable
  - And computers are nowhere near catching up

How our brain works (Gf-Gc model)

- The simple form:
  - Autonomic
  - Intuitive
  - Logical
- We actually solve most problems with the intuitive brain
  - Logic is hard and most of us are “asleep” all day long
- Another word we could use for this brain function is “integrative”

Gf

- “Fluid intelligence includes such abilities as problem-solving, learning, and pattern recognition.”

Gc

- “Crystallized intelligence… relies on specific, acquired knowledge.”

So, back to math

- The brain is good at some sorts of math and bad at others!
- Games are training you in specific sorts of math problems

$$\mathrm{L} \subseteq \mathrm{AL} = \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXPTIME}.$$

NP-complete problems

- You can verify the solution quickly
- But you can't FIND it quickly
  - (In fact, it might take trillions of years)
- Instead, you use approximation to get there

Game design application

- Preface over! So why care?
- Because even though the mechanics I am about to list are going to be really familiar, this way of *looking* at them may not be
- And it may open new doors for you

Some of Karp's 21!

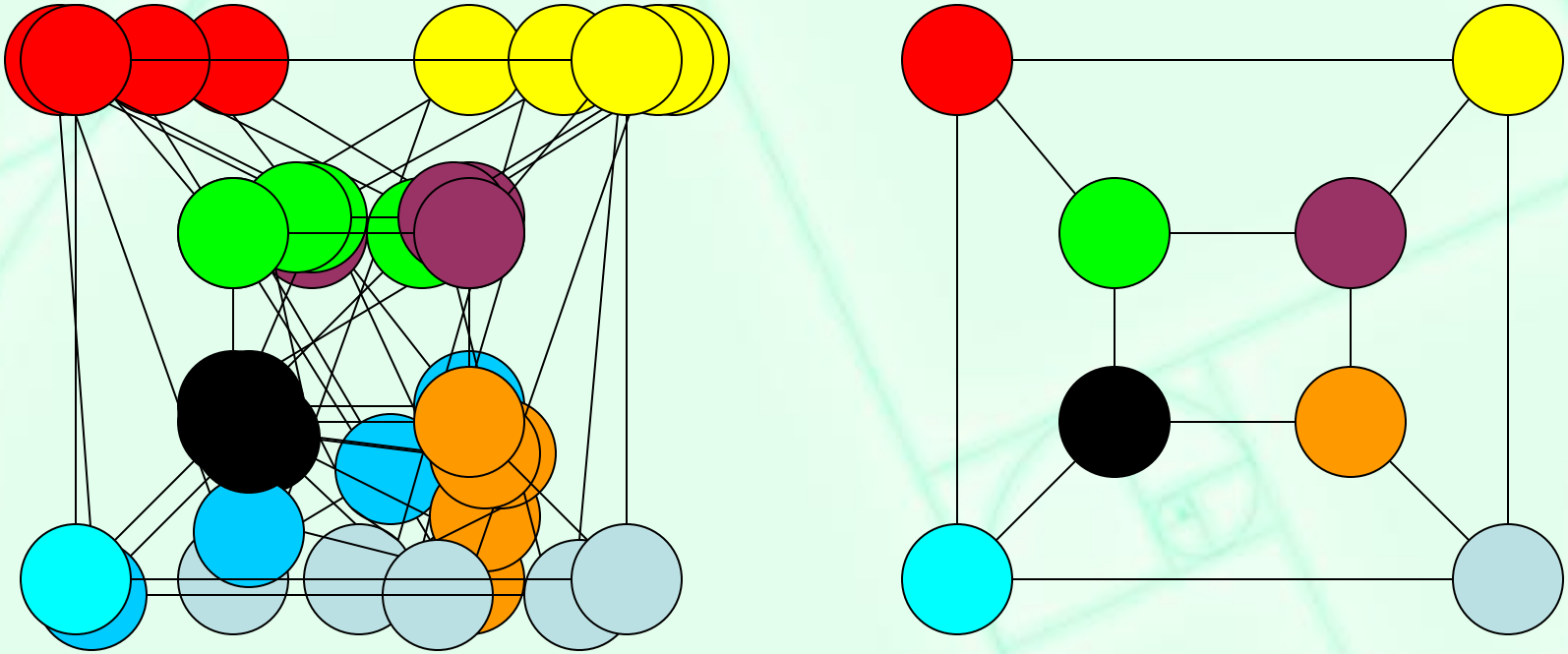
1. Graph isomorphism
2. Exact cover
3. Set packing
4. Vertex cover
5. Graph coloring (the four color map problem)
6. Knapsack problem
7. Three-dimensional matching
8. Steiner tree
9. Maximum cut
10. The partition problem

Bonus: Brain hacks

This guy still teaches computer science at Berkeley…

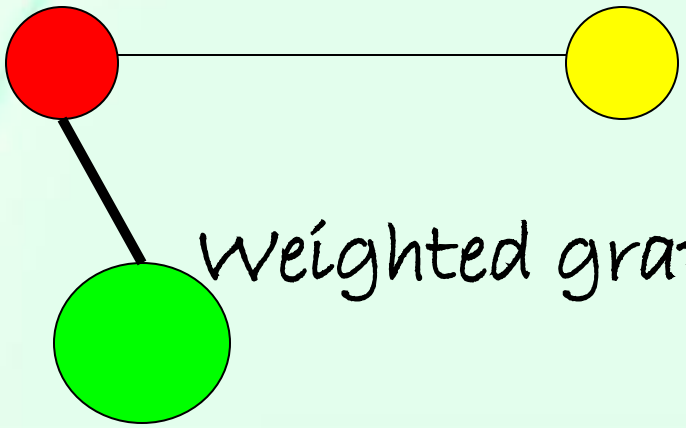
Graph isomorphism

- Are these two graphs the same?

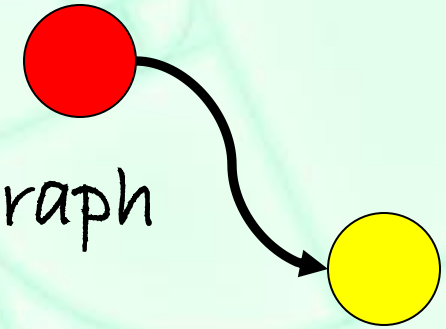


A brief excursion into graph theory

- Graph theory is the branch of math about relationships between objects
- But you should treat that as abstractly as possible!

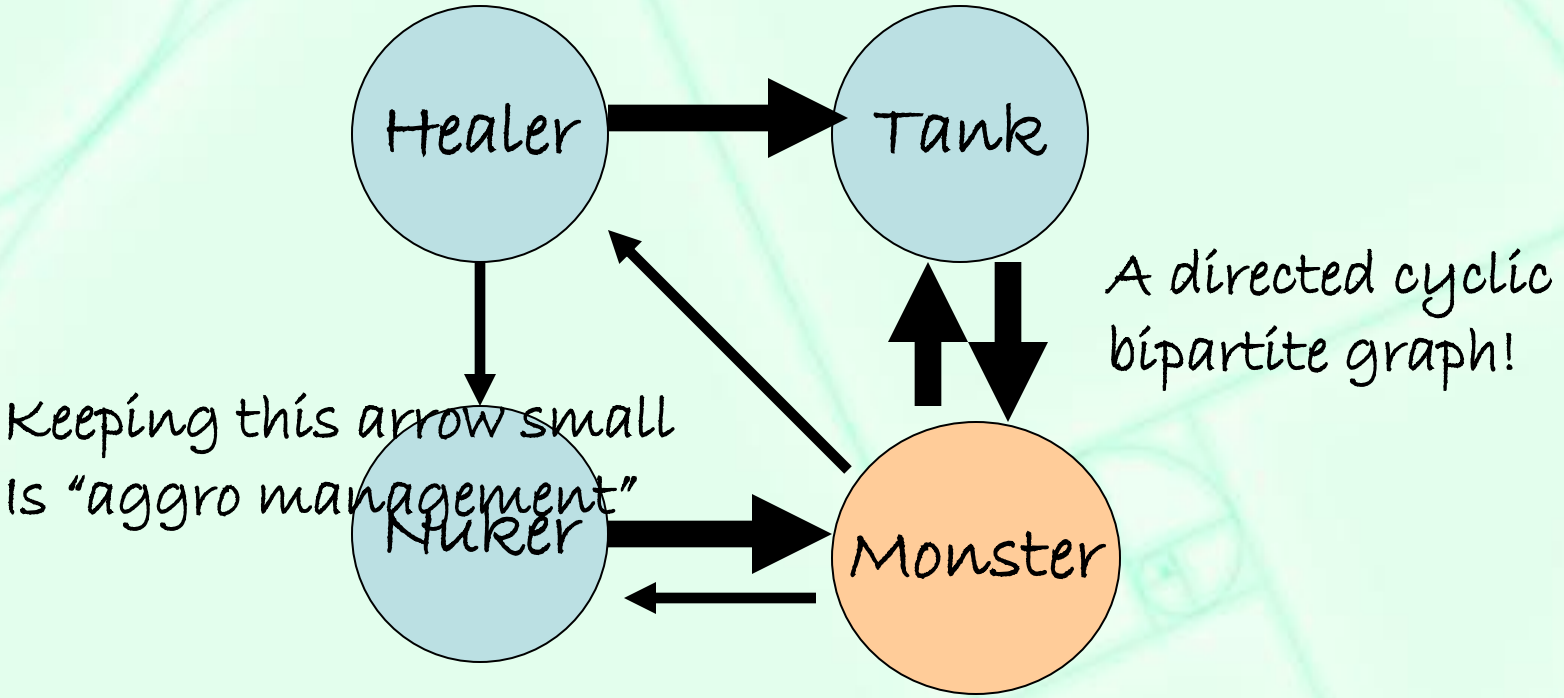


Weighted graph



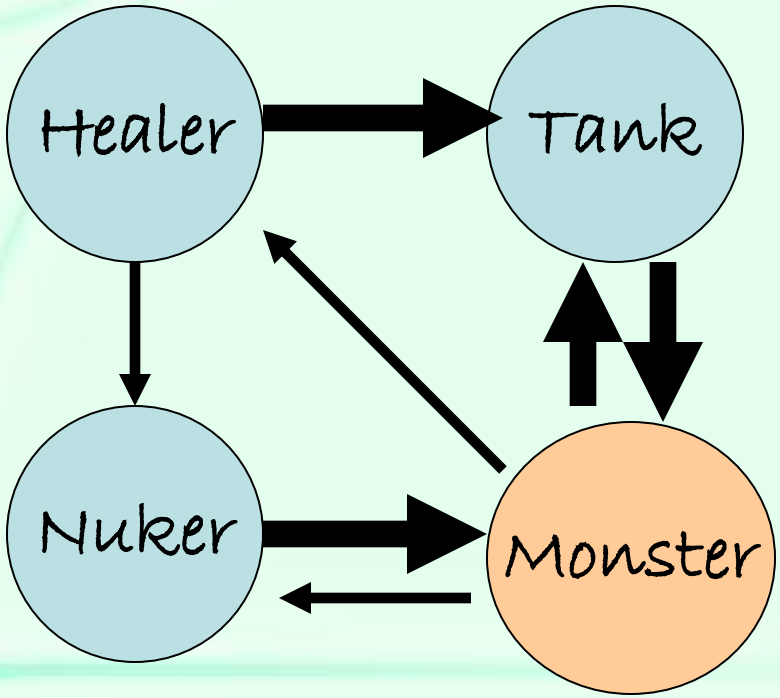
Directed graph

Graph theory & MMORPG combat



Game mechanic

- This mechanic is about determining similarity between two different webs of relationship.
- Picking the right mix of skills in a party to do a raid is an example



What's the isomorphic graph that solves this combat for me?

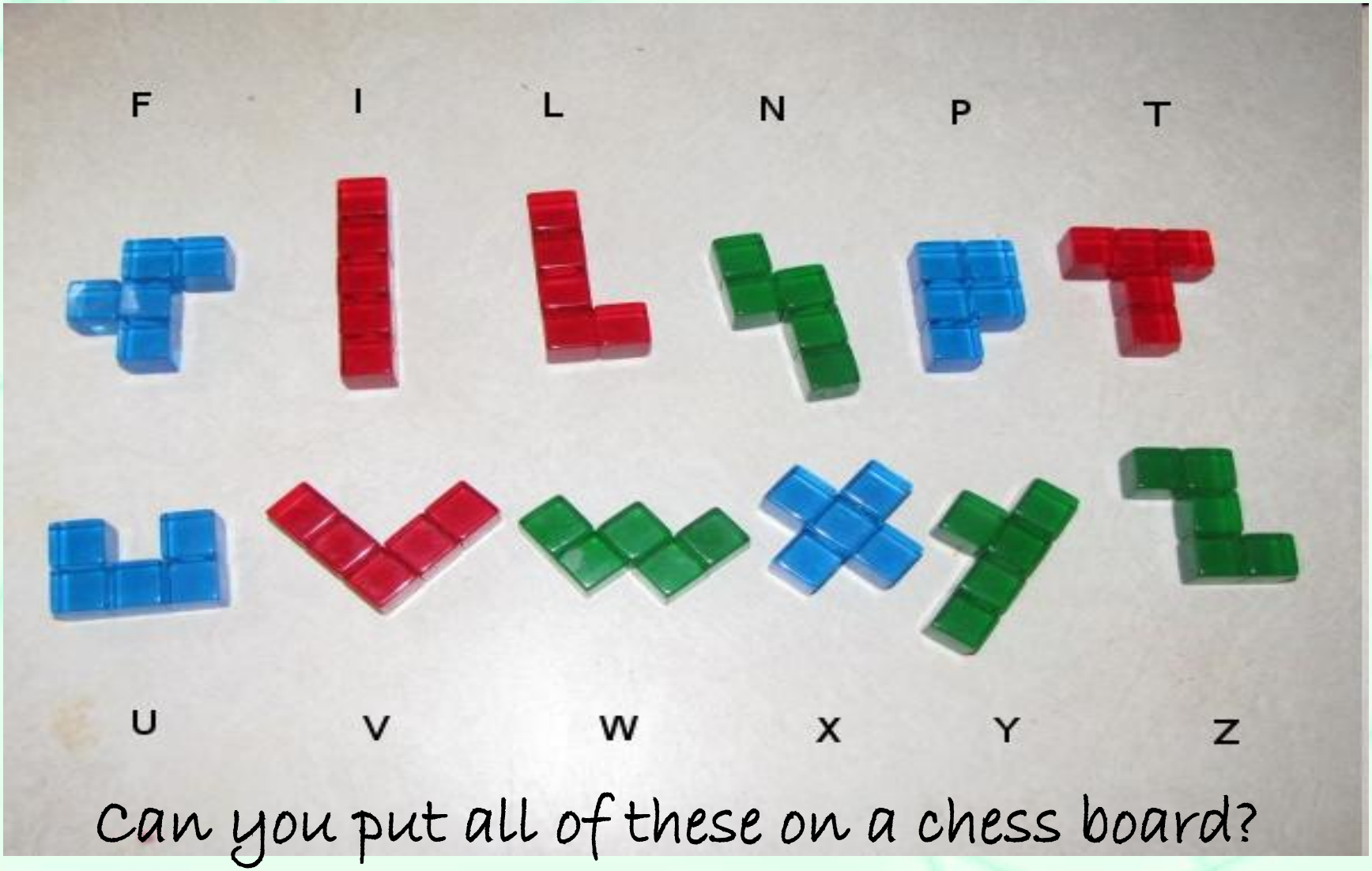
Are there other interesting graphs for players?

Exact cover

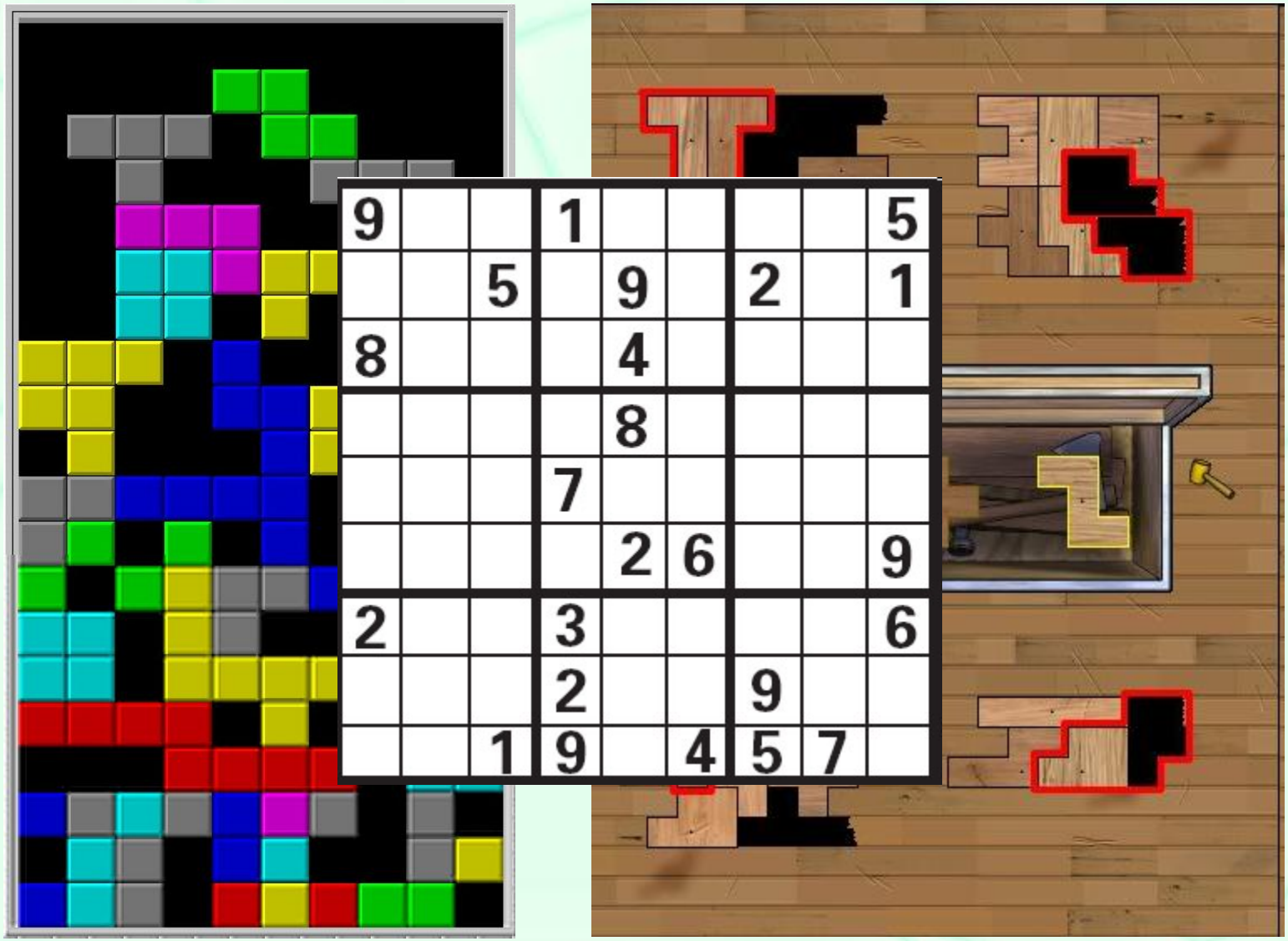
- The exact cover problem can be described as

"given this set of pieces, fit them all onto this map so that none overlap and you cover everything possible."

Pentominoes



Exact cover



Again, think out of the box

- Given X resources
- Each capable of covering Y tasks
- And Z tasks to cover
- With no ability to have 2 X's on one Z

Hmm, that sounds like allocating my team of friends onto enemies in a social RPG

Or working through a game of Strategic War…

Set packing

- What if things **can** overlap?

Tetris gets a lot easier.

Wikipedia example

- Let's say you have suppose you're at a convention of foreign ambassadors…

Morena Baccarin

- each of which speaks English and also various other languages *and want to eat you**.

*This part not on Wikipedia

- You want to make an announcement to a group of them

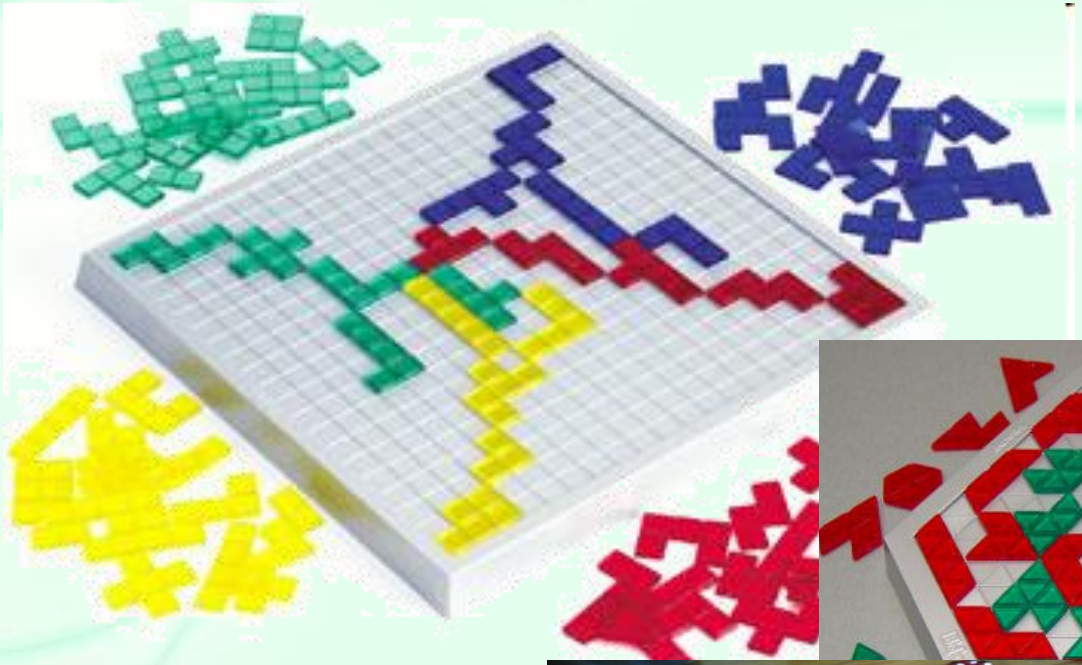
- but because you don't trust them, you don't want them to be able to speak among themselves without you being able to understand them

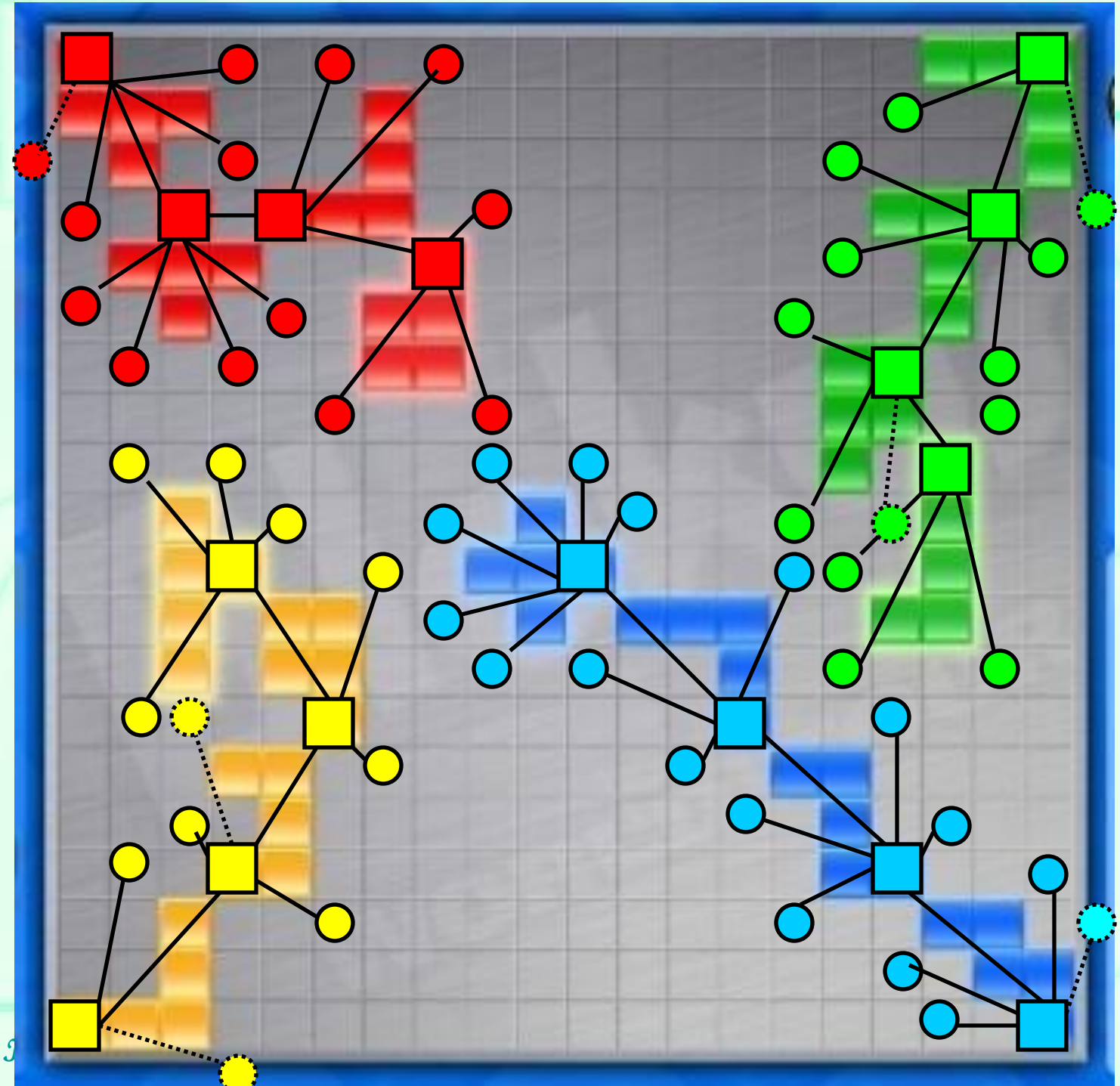
Otherwise this happens

- To ensure this, you will choose a group such that no two ambassadors speak the same language, other than English (or lizard-alien)
- On the other hand you also want to give your announcement to as many ambassadors as possible.

Suddenly, this sounds like many card games... including Texas Hold'Em, and deck-building in most CCGs...

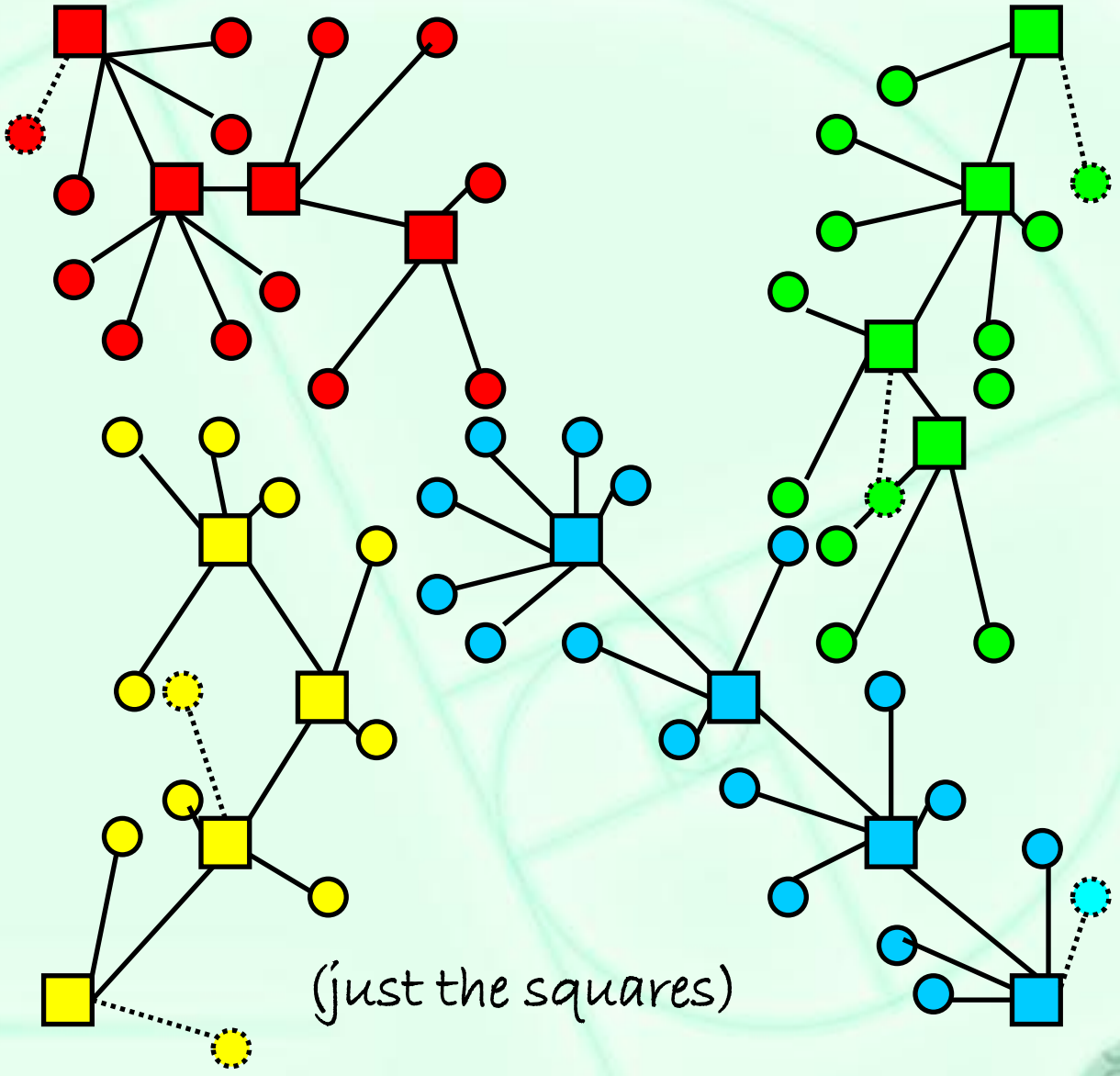
Vertex cover





The formal definition of vertex cover

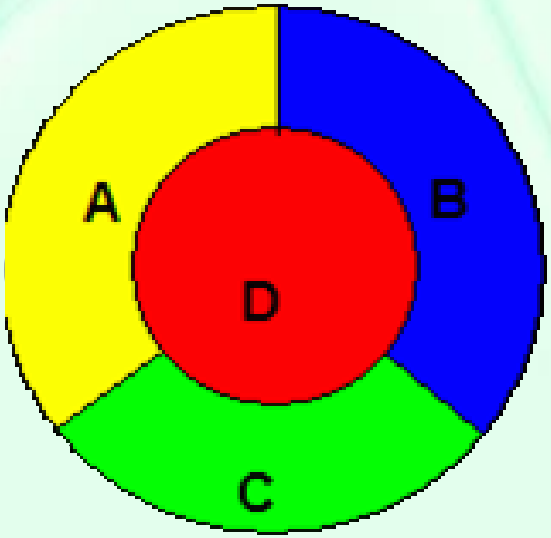
- A set of nodes in a graph that is the minimum needed to touch the whole graph with lines.

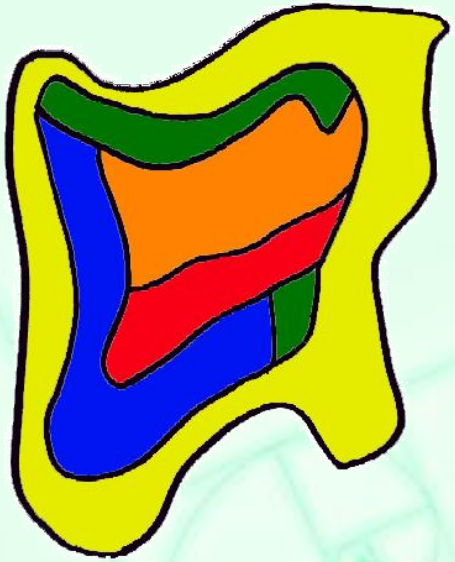


A game design application

Graph coloring

- Sudoku again, but let's ignore that for a moment



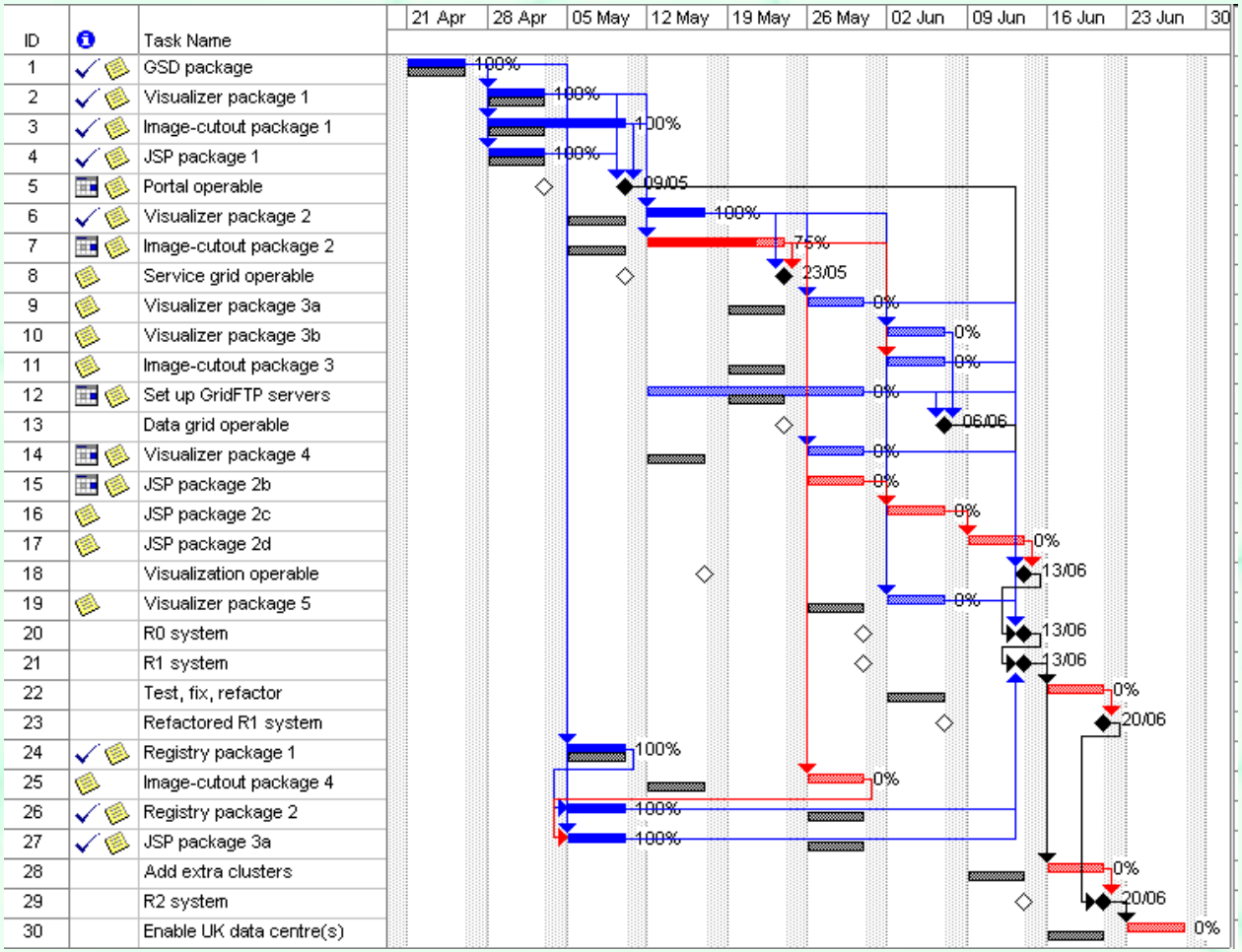


If you found that hard, it's because it's

$$O(n(\log n) - {}^{3}(\log\log n)^{2})$$

More importantly…

- This is why MS Project can't fix your Gantt chart



Job = node

Conflict = edge

# of colors, or "chromatic valence," = "makespan" or total time!

Every scheduling game

- Restaurant City
- Sims
- etc!

Knapsack problem

- Given a set of items, each with
  - a weight
  - a value
- Decide how many to keep
  - so that the total weight is less than a given limit
  - and the total value is as large as possible.

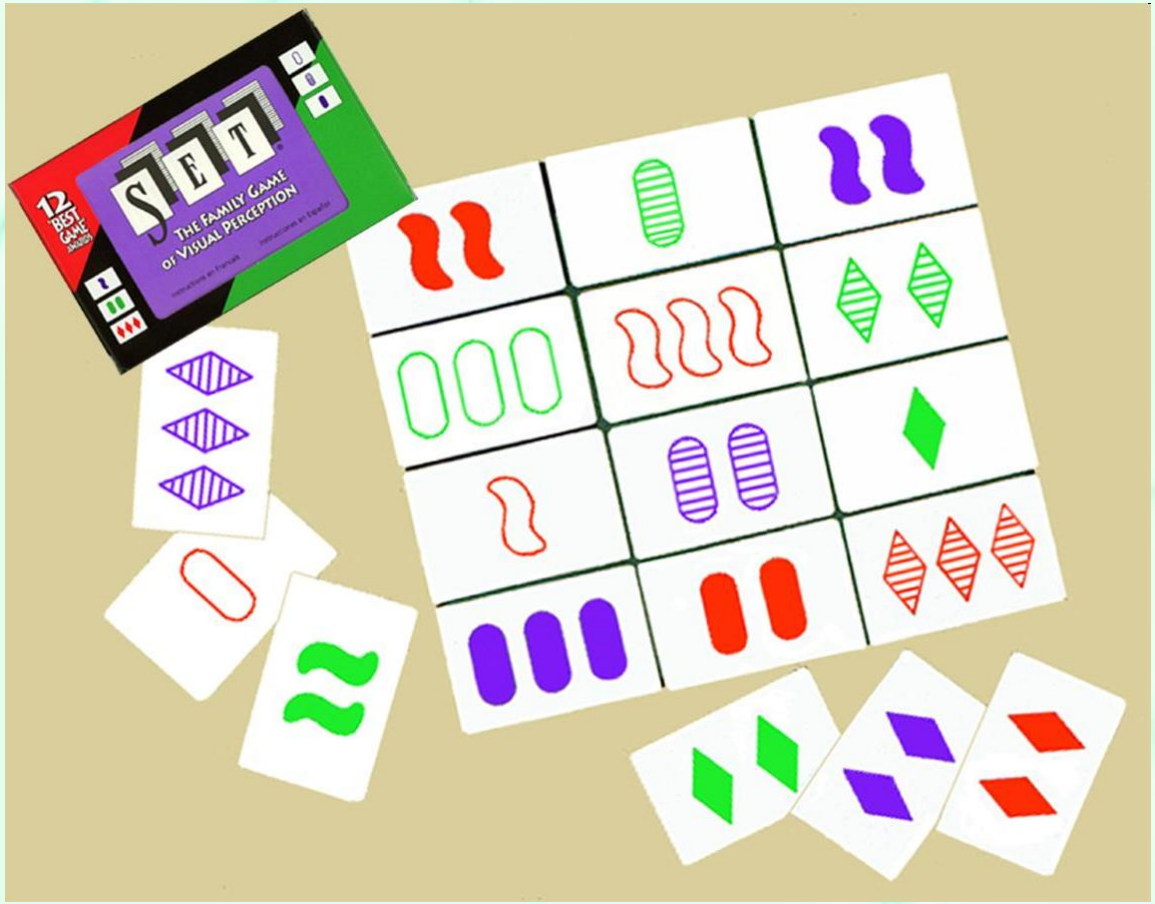
In game design we call this

The KNAPSACK PROBLEM!

Actually, I always called it "inventory tetris"

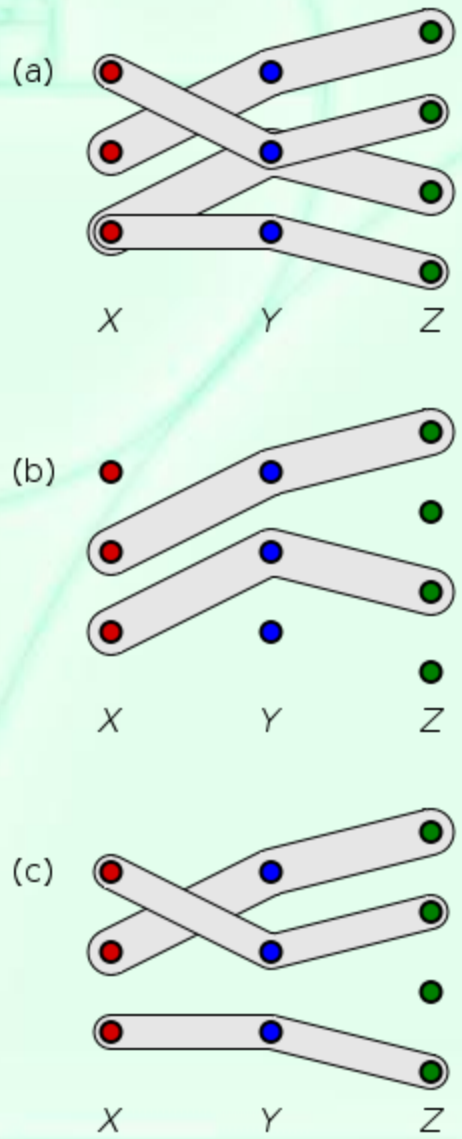
Three-dimensional matching

- Technically, a subset of 3-satisfiability



E = (x1 or ¬x2 or ¬x3) and (x1 or x2 or x4)

A non-visual application



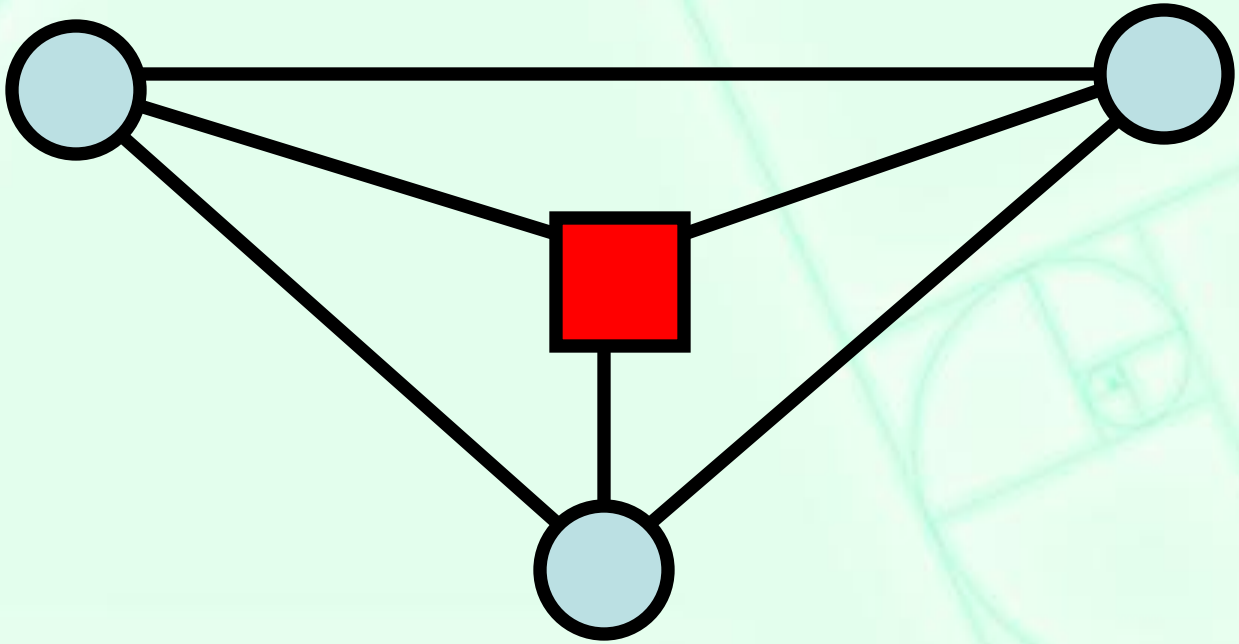
- Think of red, green, and blue as three different stat mods
- And each gray ribbon as an equippable item

  My slots are kinda full…
- What's the minimum items to get best benefit?

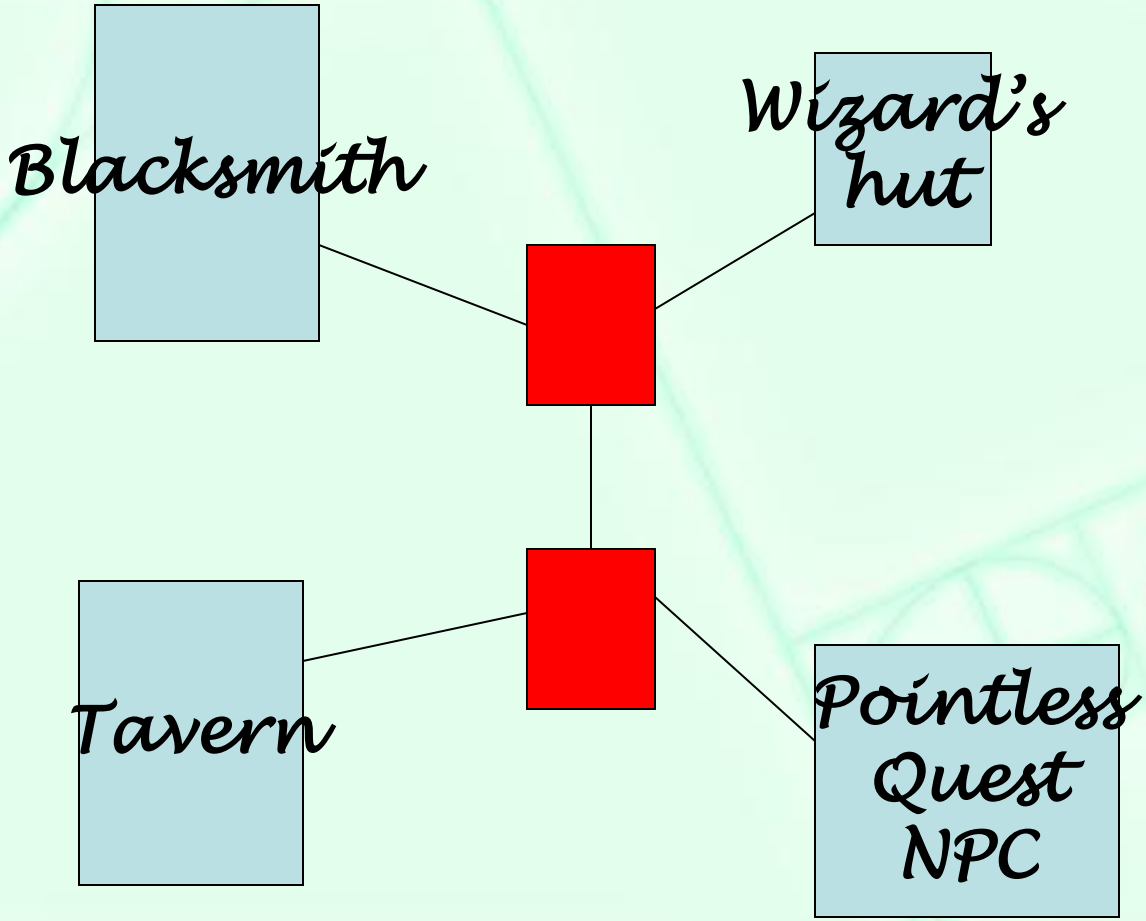
  What if I ditch my older gear?
- And what's the maximum usable items?

Steiner tree

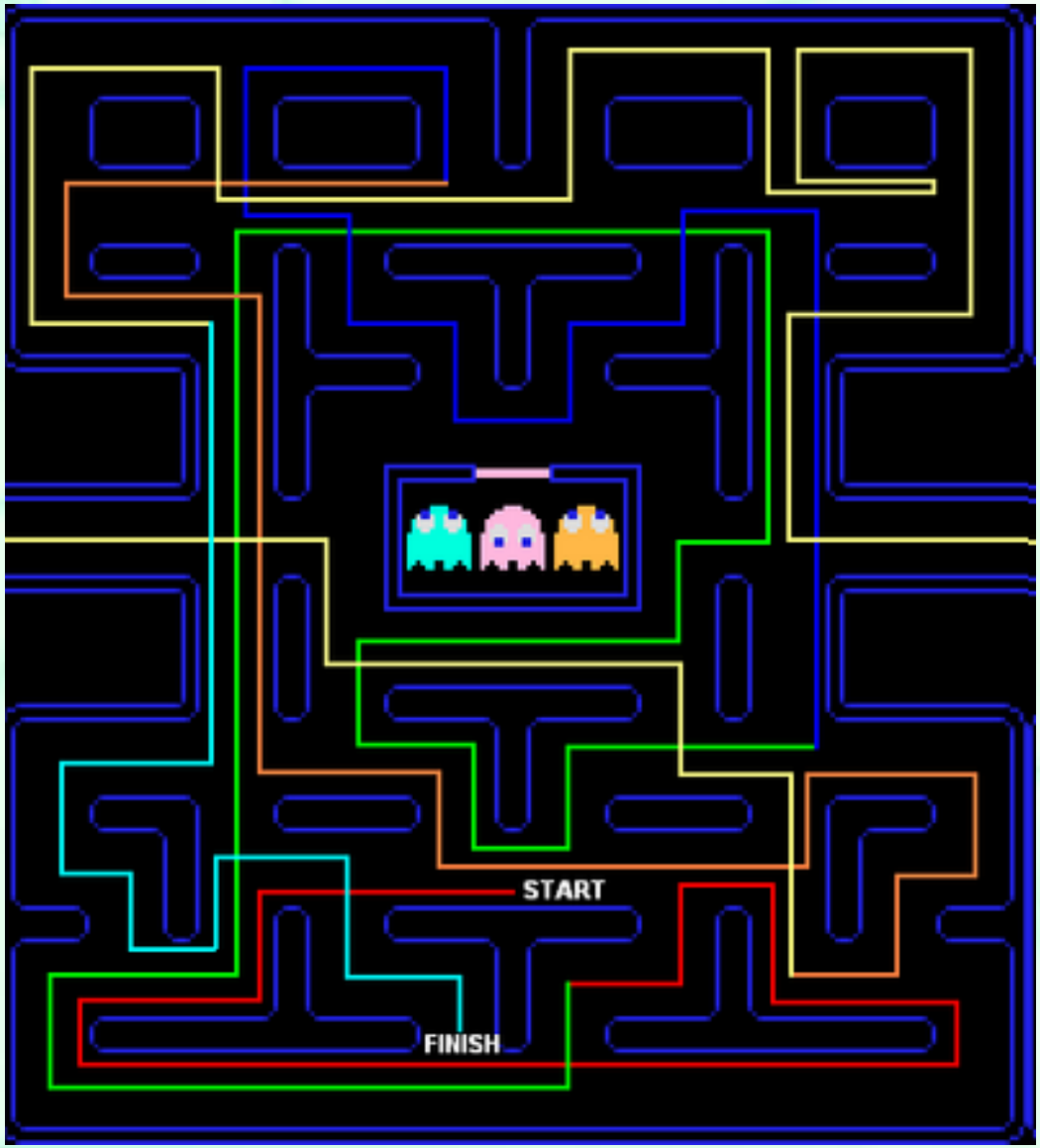
- AKA
  - Decision path traversal
  - Travelling salesman problem



Used for level design



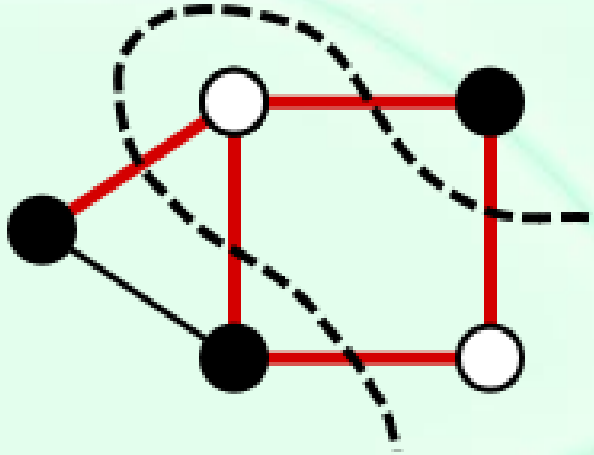
A classic Steiner tree

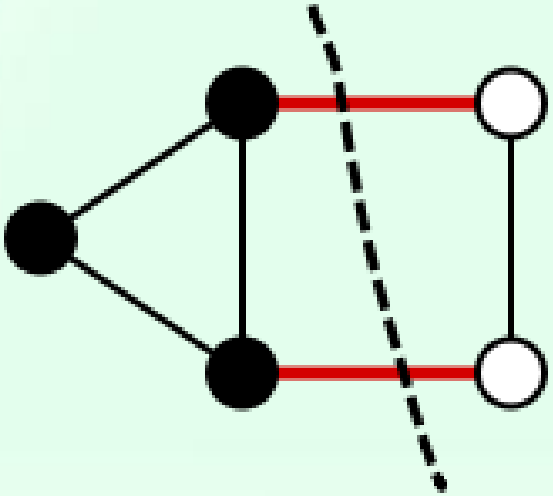


The higher the number of Steiner points, the less fun the game will be; the challenge is finding the tree of minimal weight, the fewest points

Maximum cut

Try to go through as many of the edges as you can without hitting any twice

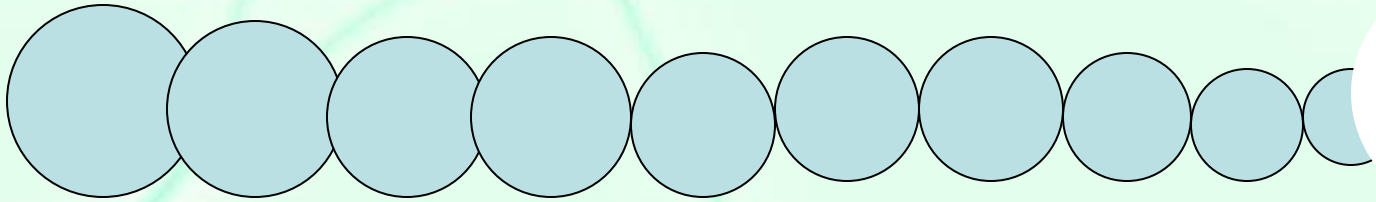




Minimum cut is also cool – cut the graph in at least two, and have the smallest number of edges hit

Partition problem

- Setting up two teams on the playground



- Classic solution is a greedy algorithm
  - There's actually better ways
  - But greedy solves this 99% of the time

Game design implications

- It's very easy to do as long as the range of skills/values/etc is similar

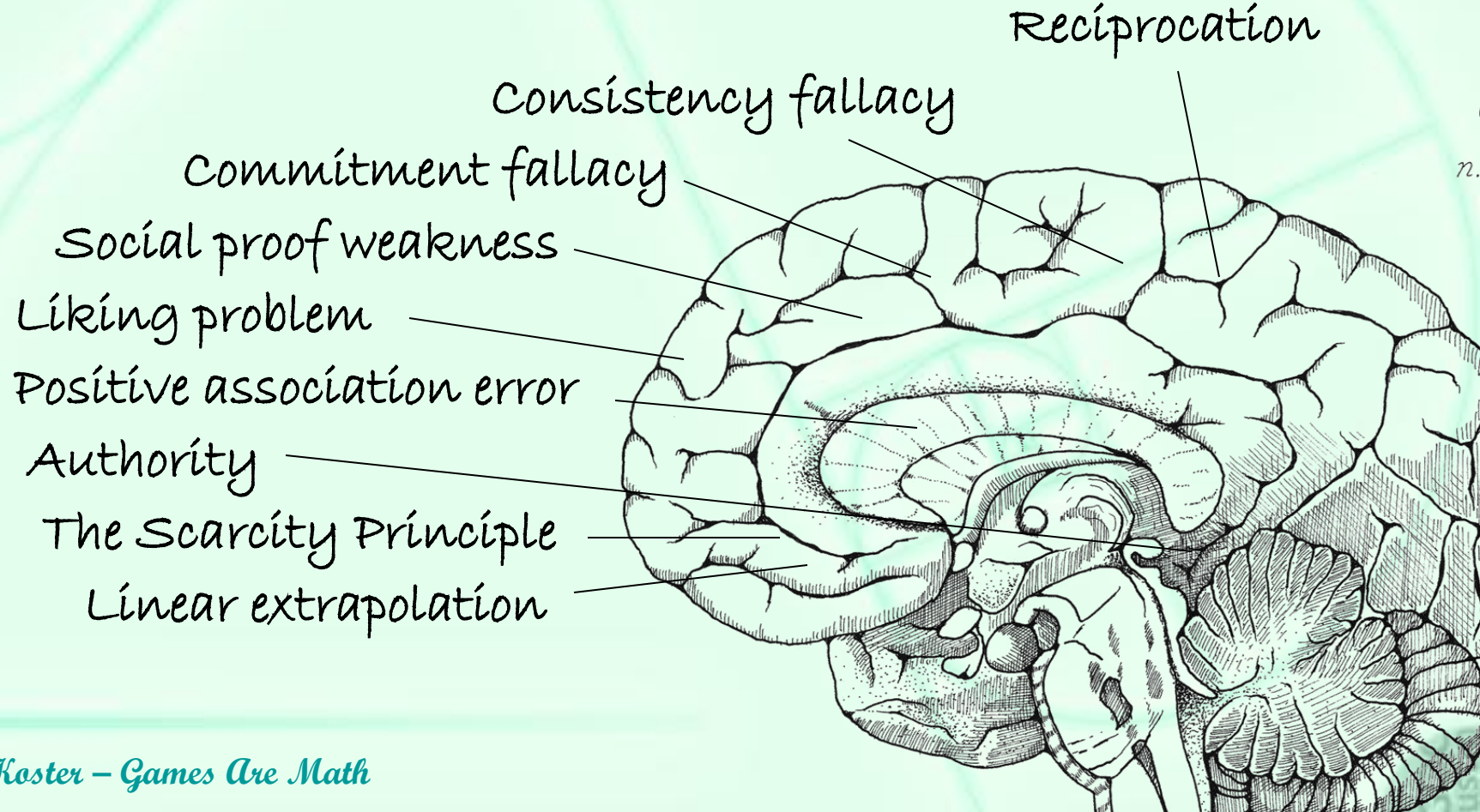
Based on the m/n ratio,
where *n* is the size of the set (total kids)
and *m* is the "bit depth" of the kids

- But when it's not… Suddenly we're in the realm of power laws and PvP monsters

This is a much better problem for players to have than for developers to have in their matchmaking system

Bonus #11: Brain hacks

- So the brain builds up algorithms to solve problems…
- But! The brain has a number of cognitive "bugs" in its software!



Reciprocation

- We will do things for people we think did us a favor, even if we don't like them
  - We'll disproportionately help them, even
  - And we'll do it even if the gift was forced on us

Typically used for extracting money from users.

Consistency fallacy

- We are far more likely to do things that we have made a promise to do, or go on record in some fashion
  - Even getting people to agree to a minor statement that is only partially on the way to a major promise will make them more likely to fulfill the larger promise

Can be used as a device for pulling users though open worlds, or for getting them to commit to codes of conduct

Commitment fallacy

- We value things we struggled through difficult painful times to get
  - More than stuff that comes easy, regardless of whether this is its true value
  - Hazing rituals, initiation ceremonies… free for all PvP?

Does your game offer challenging membership ceremonies? Does a retained (paying) user feel like a member of a secret club?

Social proof

- We do things that we see other people validating.
  - The crowd outside the popular nightclub, the most popular song this summer, canned laughter on a soundtrack, “most users reuse their towels”
  - At the scene of an accident, if lots of witnesses stand around, no one tends to do anything
  - The more like us the others are, the more likely we are to match behavior

Big implications for community management. Can you use claques?

The liking problem

- We instinctively like people like us
- We also try to please those who like us
  - Who signal it with flattery, team membership, favors
- Which means that if you can make someone feel like a liked member of a group, you can get them to do things
  - Sign up for Who's Who, buy Tupperware, join calling circles… or

"Join now and help your friend level up in Mafia Wars…" Automated flattery, team membership, and fear of snubbing a friend

Positive association error

- Put p
- Need

Authority

- Taller people with executive hair climb the ranks.
  - And halflings get screwed in MMORPGs.
- We go along with authority
  - And with sly sincerity

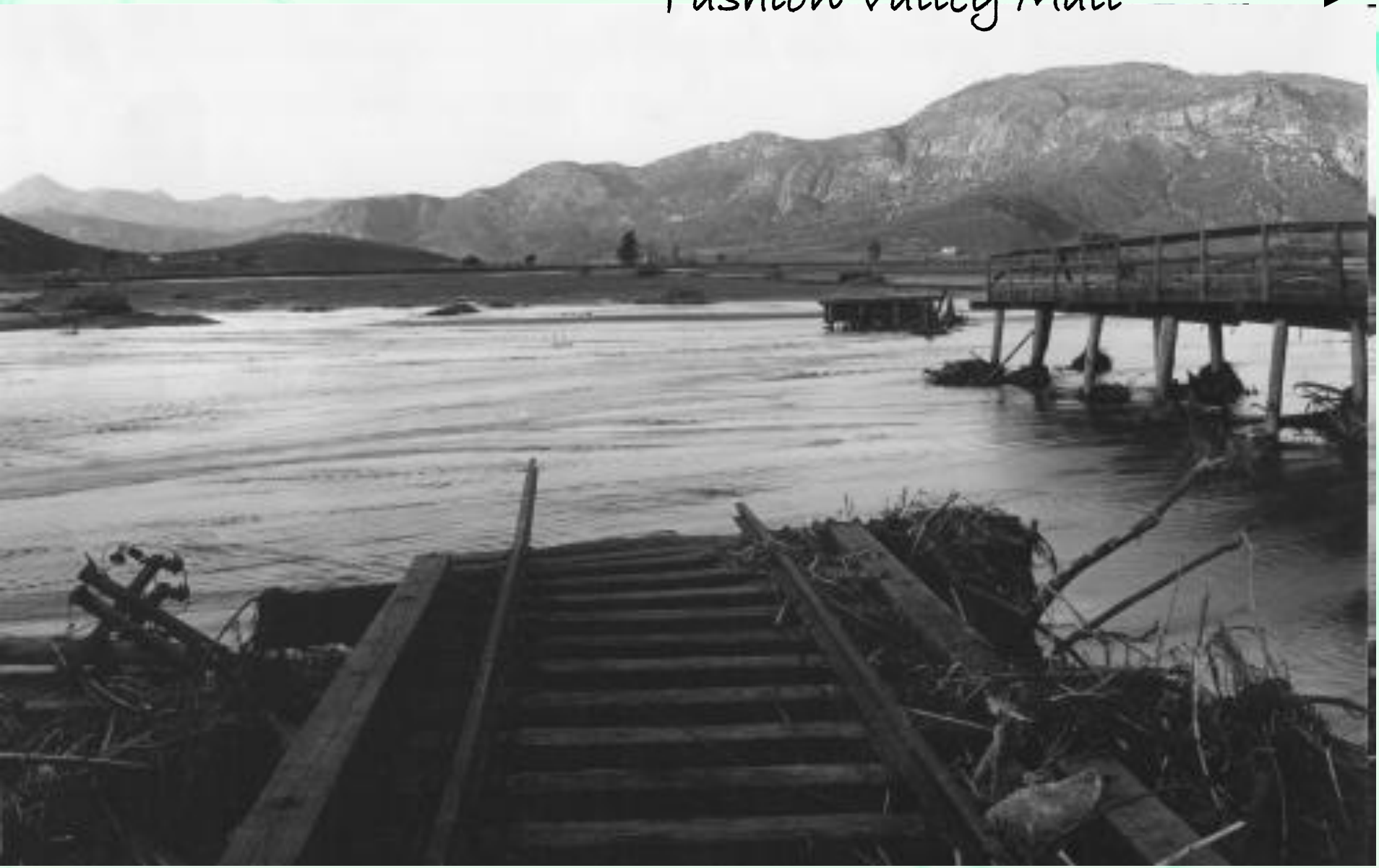
What is the mechanic that uses these consciously? We underthink our avatars, and we fail to provide the right NPC models.

The Scarcity Principle

- We overvalue what we think is scarce
  - We crave the censored
  - We click on the forbidden NSFW link
  - We desire what few have regardless of whether it is worth anything objectively
  - If there's competition for it, we value it even more

Velvet rope access, collection of "rares," the appeal of the banned, auctions not sales (we ended up leaving in the bug that cause UO rares...)

Fashion Valley Mall →



1916 flood

Linear extrapolation

- Our human brain tends to only see certain types of curves:
  - We extrapolate linearly really well
  - We fail at estimating exponential curves though we can project them
  - We utterly suck at non-linearity (cf. economic crisis)
- This manifests in all forms of odds calculation
  - Risk of terror attack vs risk of car crash, shark attack vs lightning, 500 year flood plains versus striking it big at a slot machine…

Black swan events, true random, and stochastic distributions will come as shocks –and you can use these for surprise, or set people to hunting for them.

"Easy" games

- Just because I said these were good systems doesn't mean they are the only good systems!
- P, or easy systems, are compelling until an adequate algorithm is developed.
  - You can gain some mileage from presenting more and more complex data sets for the algorithm to operate against

Hmm, I think we have now given you enough so that you can finally beat all of Professor Layton.

References

This stuff is here for further reading! Enjoy!

Typical algorithms to solve P games

- Sorting (many, many)
- Pairing (Gale-Shapley)
- Minimax algorithm (tic-tac-toe)
- Hungarian algorithm (optimal task assignment)
- Topological sort (task ordering)
- Cluster identification (Girvan-Newman)

List of NP games

- Battleship
- Making crossword puzzles
- FreeCell
- Mastermind
- Tetris
- Picross
- Instant Insanity
- Minesweeper
- Same Game aka SegaSwirl/Bubble Shot/Bubble Drop etc etc
- Pipes
- March Madness

Stuff harder than that!

- (EXPTIME or PSPACE complete)
  - Sliding blocks
  - Sokoban
  - Checkers
  - Chess
  - Go
  - Hex
  - Othello
  - Rush Hour/Parking Lot/etc

Further reading on the web

- My original presentation "Grammar of Gameplay"
  - http://www.theoryoffun.com/grammar/gdc2005.htm
- Dan Cook's "Chemistry of Game Design"
  - http://www.gamasutra.com/view/feature/1524/the_chemistry_of_game_design.php
- The Complexity Zoo: site all about complexity
  - http://qwiki.stanford.edu/wiki/Complexity_Zoo
- Karp's 21 NP-Complete problems:
  - http://en.wikipedia.org/wiki/Karp%27s_21_NP-complete_problems
- Fantastic puzzle page
  - http://home.comcast.net/~stegmann/assembly.htm
- The guinea pig getting eaten in the original V:
  - http://www.youtube.com/watch?v=VObQfWMgmIM
- Excellent walkthrough of map coloring:
  - http://www.ctl.ua.edu/math103/mapcolor/mapcolor.htm
- Great partition problem article
  - http://www.americanscientist.org/issues/pub/2002/3/the-easiest-hard-problem/1

Further reading in books

- On mental bugs:
  - *Influence: Science and Practice*, by Robert B. Cialdini
- On models of thinking and three-layer cognition:
  - *Sources of Power: How People Make Decisions*, by Gary Klein
- On graph theory:
  - *Small Worlds*, by Duncan Watts
- On linear extrapolation:
  - *The Black Swan: The Impact of the Highly Improbable*, by Nassim Nicholas Taleb

Raph Koster's Website

http://www.raphkoster.com

A Theory of Fun for Game Design

http://www.theoryoffun.com

Special thanks to J. C.Lawrence for saying "it's all about the vertices," to Laralyn for the last second review, and to all the game grammarians!