Depth and Design

Contrasting AI and Human Understandings

Raph Koster, independent game designer

AAAI-18 Workshop on Knowledge Extraction from Games

The problem

- I come at this as a designer of games.

- I want to make good games.
  - Maybe computers can tell me how.
  - I don't want them to make them for me though, because I enjoy it!

- I am a lot less interested in how computers can beat games.
  - The knowledge to extract: "is this a good game?"

- I am also not interested in what a computer thinks is a good game.
  - My final platform is humans, not silicon.

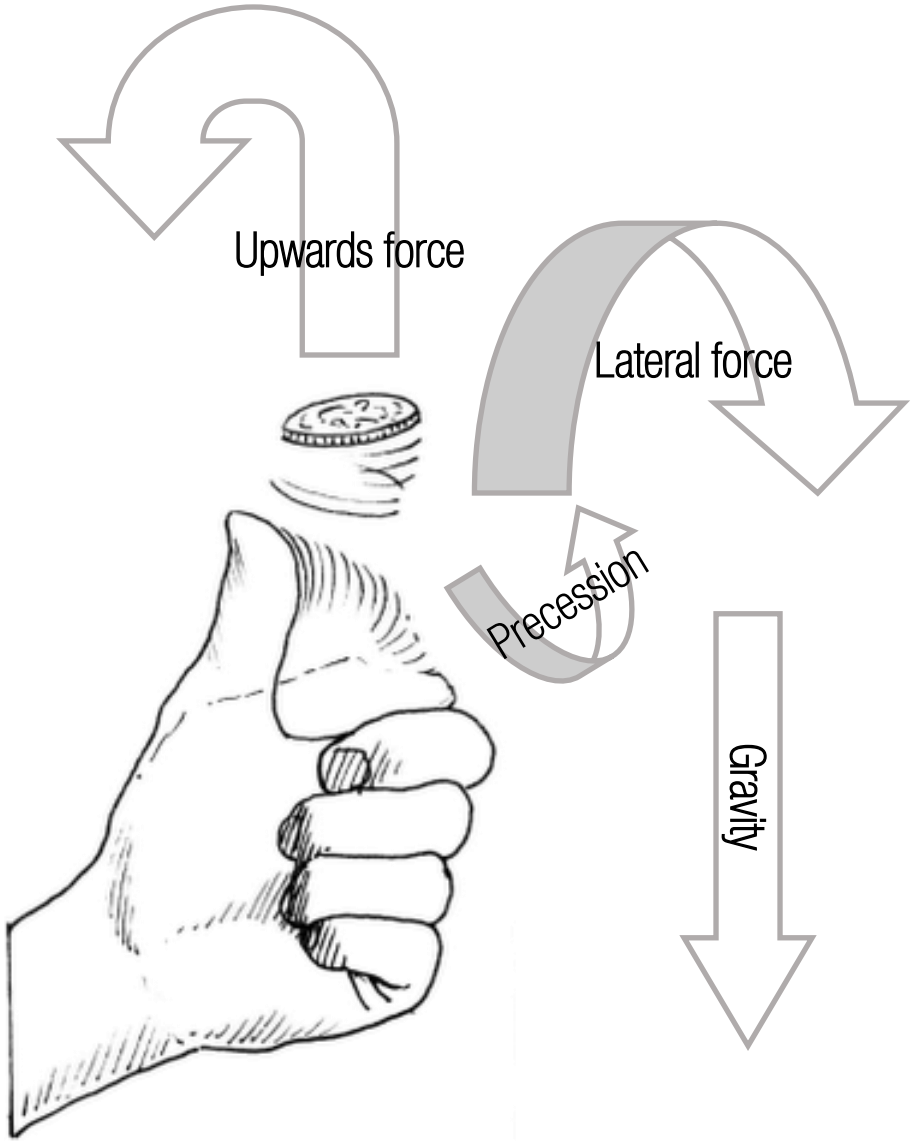
Defining “good”

- A first approximation: “fun.”
  - Best understood as a psychological **response** when we want a **quality**.
- A second approximation: “complex.”
  - There is a strong association between PSPACE-Complete (and higher) complexity class and games that audiences have termed “fun.”
  - But complex games are often not fun.
- A third approximation: “deep.”
  - There is no formal definition of “deep.”
  - It cannot mean simply large possible state space, as there are large boring games.
  - It cannot mean simply branching, as there are games with many meaningless choices.

Game grammar overview

- Games as a whole do not yet have a BNF grammar, but subdomains do.
  - Mostly used for procedural level design, for specific games.
- There is an ongoing informal project to at least identify grammatical elements.
  - Decent success here which has seeded design tooling and academic work, particularly in PCG.
- Problem: we might derive a full context-free grammar without addressing depth.
  - Fully descriptive of form, without being descriptive of scope of statements in the grammar.

A step back: the coin flip

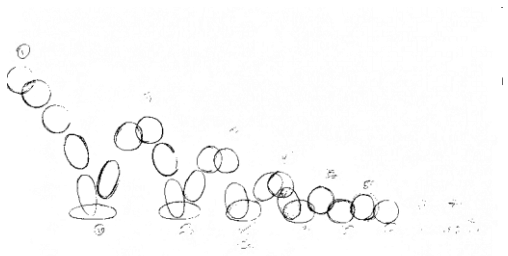


Distance to surface

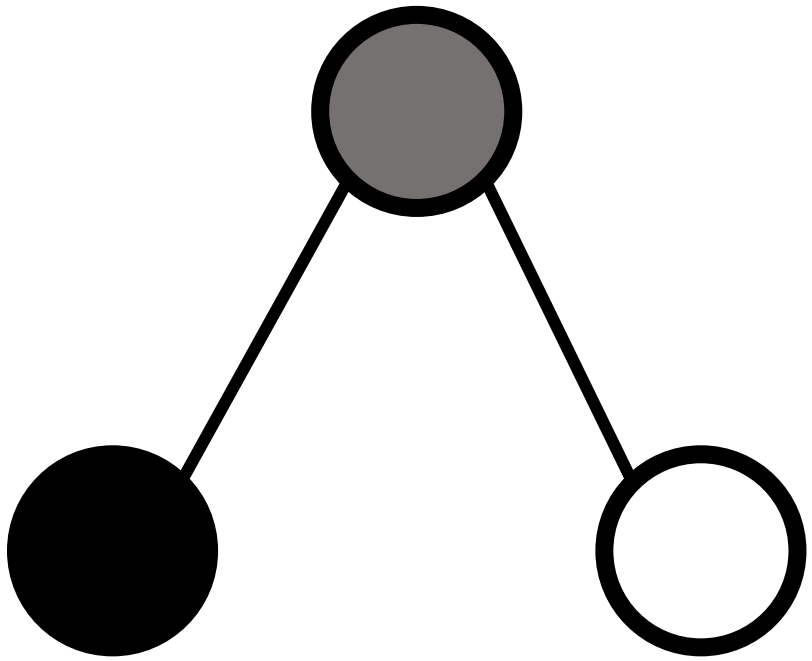
Slight weight bias towards tails

Ignored factors:

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



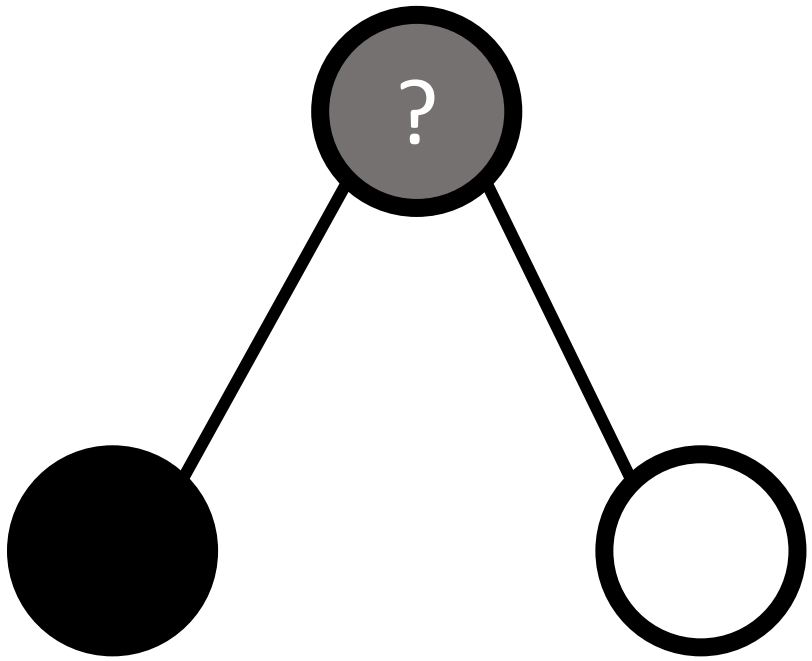
A step back: the coin flip



- Starts out indeterminate.
- Simple binary output.

- This is actually the case for most games.

A step back: the coin flip

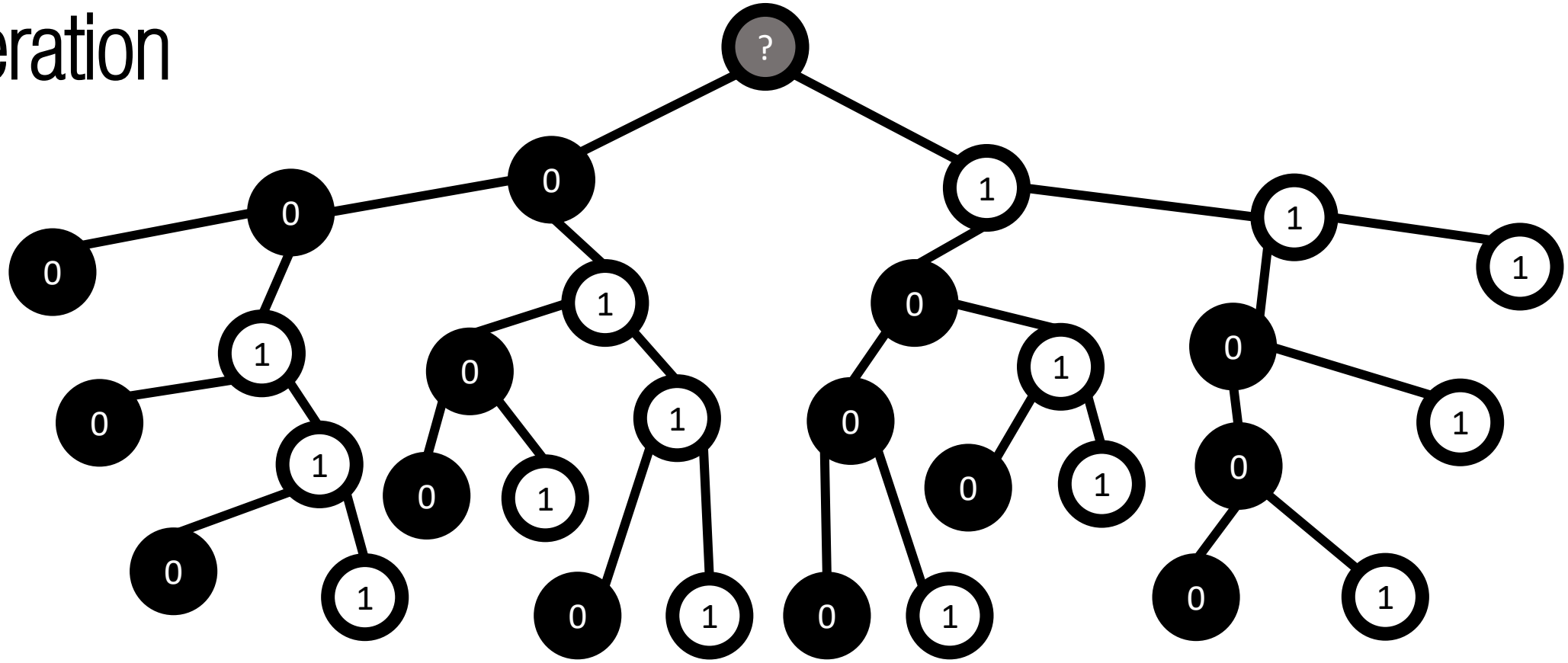


- Starts out indeterminate.
- Simple binary output.

- This is actually the case for most games.

- Some end indeterminate; we call this a tie or draw.
  - (Coin bounced down the sewer. . .)

Iteration



- In practice simple games are played iteratively.
  - “Best of five.”

- The outcome becomes determinate at a specific point down the tree.

Forms of bounds

- Goal bounded: Reach a fixed number of points or an objective.
  - “Best of five” has a point bound of three.
  - Connect Four has a point bound of one, a singular objective.

- Turn-bounded: Have a higher number of victory points after a fixed set of turns.
  - Reversi has a fixed number of spaces.
  - Turns may also be token-bounded; you may have a fixed number of tokens but a choice of a larger number of spaces.

Victory points

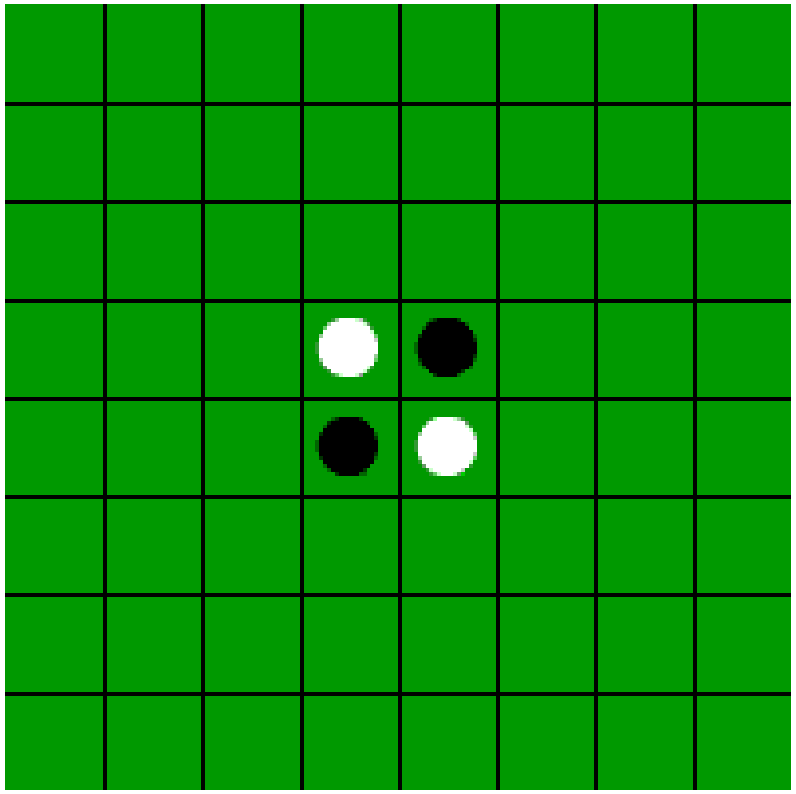
- Term from tabletop games.

- We can generalize all those bounds into victory points.
  - Victory points in a game are not always obvious. For example, a level is a victory point in Mario.
  - Some games have multiple kinds of victory points, like Pente.
  - Score isn't usually the formal victory point in a game, but it may be the victory point in a higher-order meta-game.

- In some games victory points move in tandem with turns; in others not.
  - The parity relationship between victory points and turn/space bounds determines whether the game is binary or ternary.

Binary and ternary outcomes

- $VP_a > bounds / 2 =$ win for black
- $VP_b > bounds / 2 =$ win for white
- $VP_a = VP_b = bounds / 2 =$ a draw
  - Possible because there's an even number of moves.



Useful game design output: three player games

- We can generalize this to a useful design tool: **vp > bound / numPlayers**.

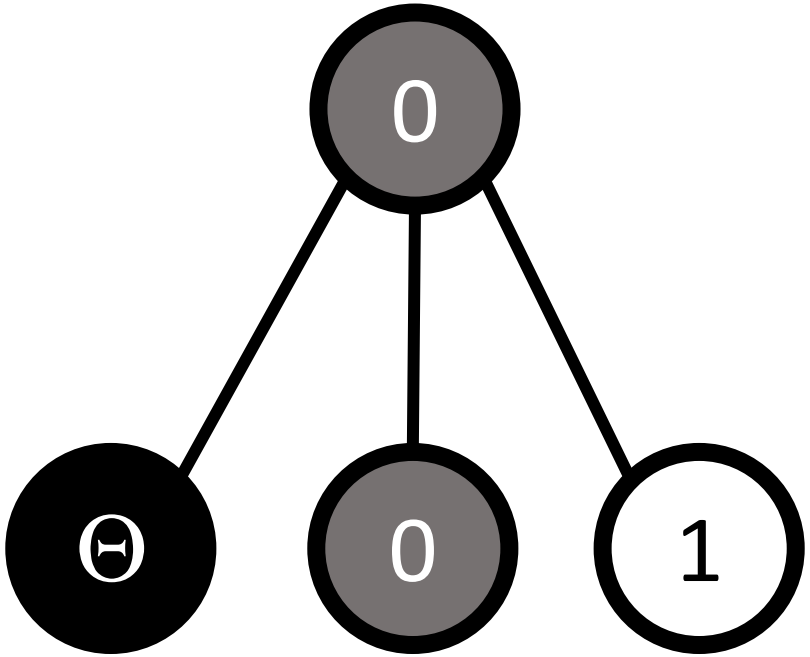
Number of players	Boundary	Even split		
2	64	32	ternary	*ties possible*
3	64	21.33333	binary	
3	69	23	ternary	*ties possible*
4	64	16	ternary	*ties possible*
2	65	32.5	binary	

Games are generators of outcomes, distillers of processes back down to

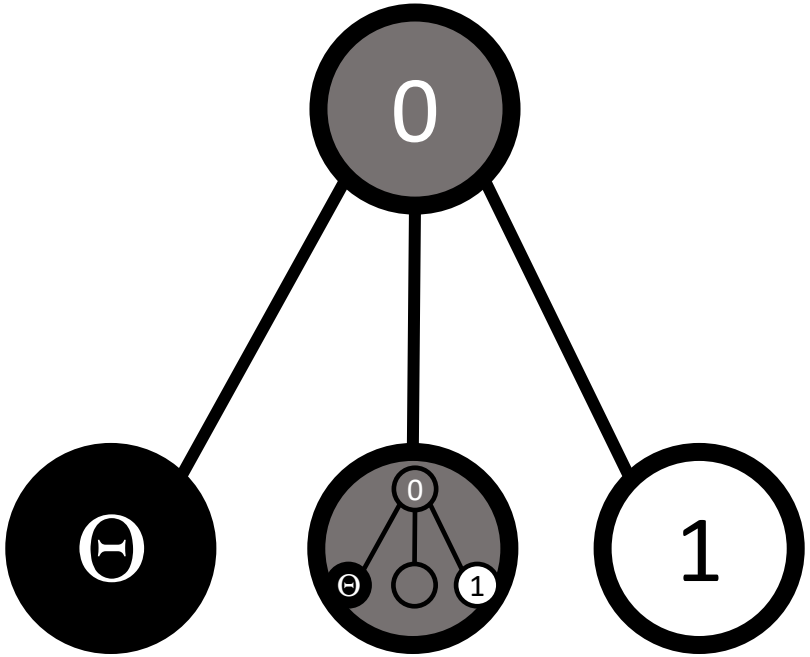
yes,

no,

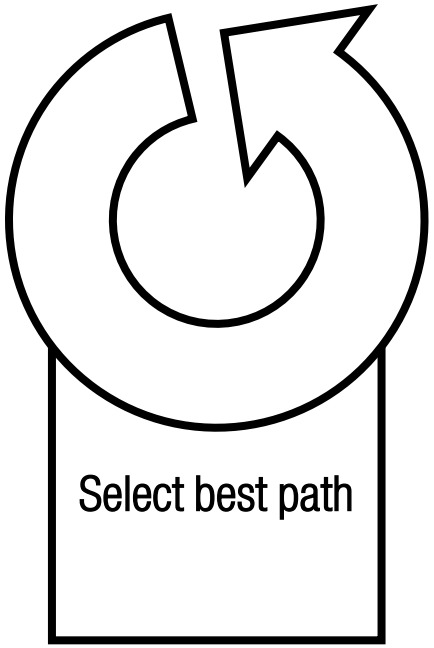
and indeterminate.



Games are also fractal,
self-similar.



An example



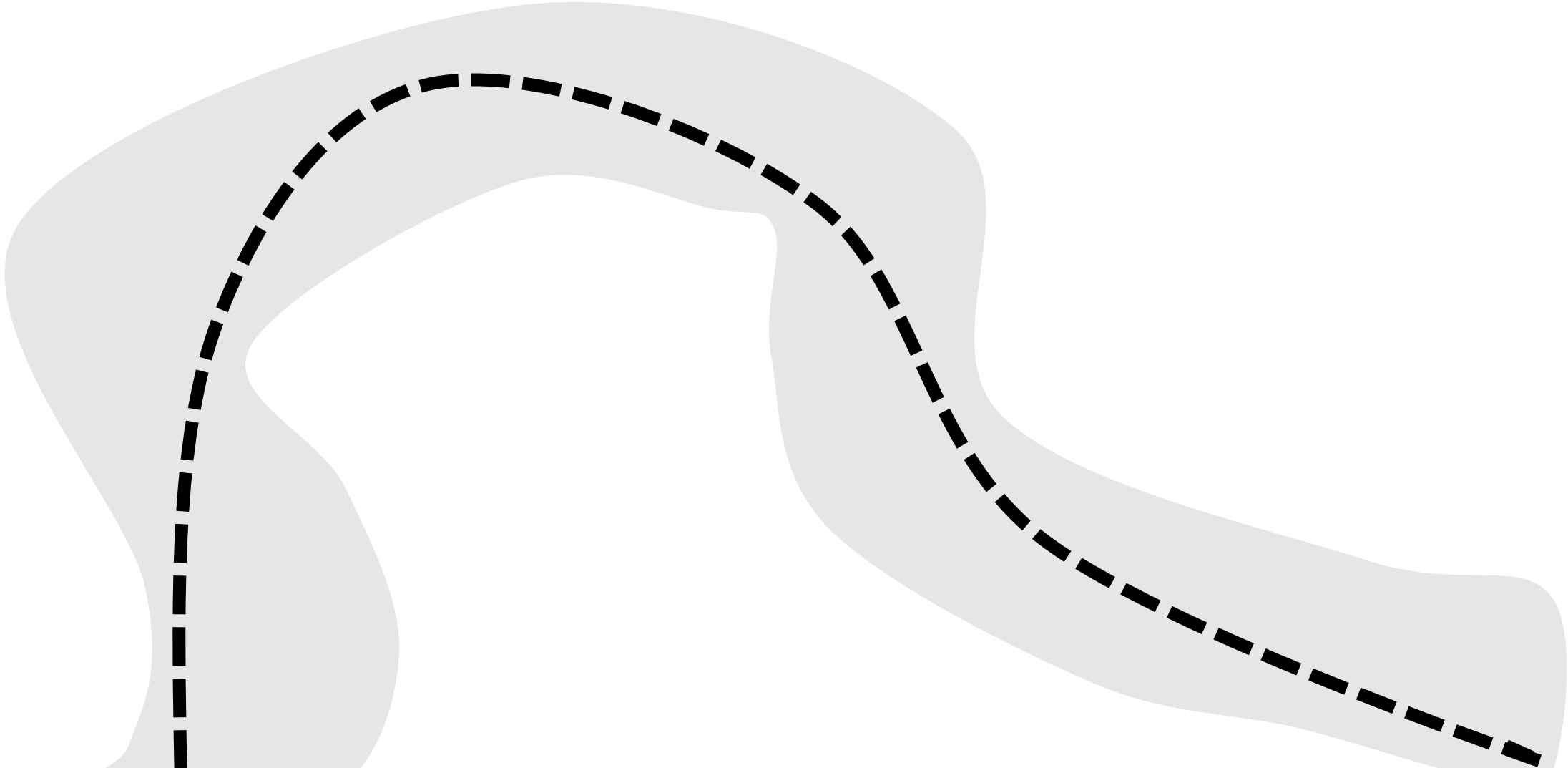
Cognitive problem

Modify for avoidance

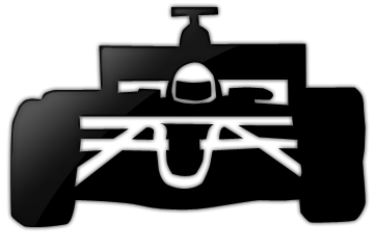
Steer on path

An example

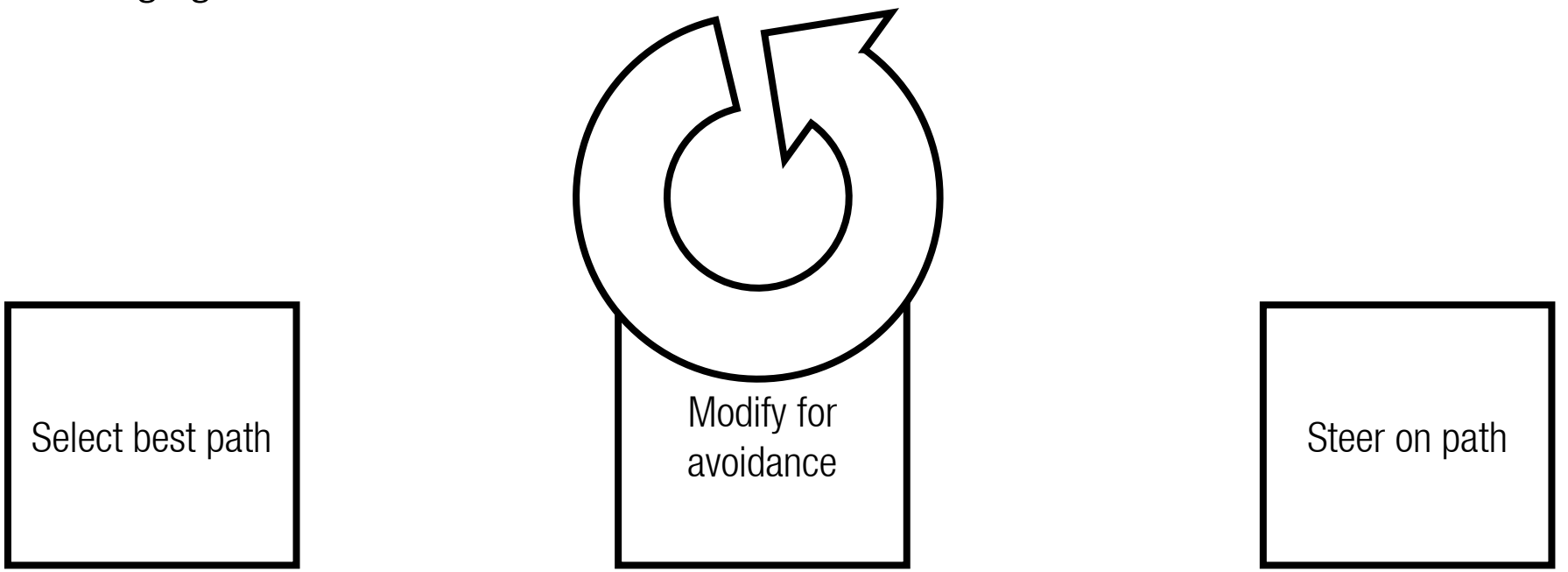
An example



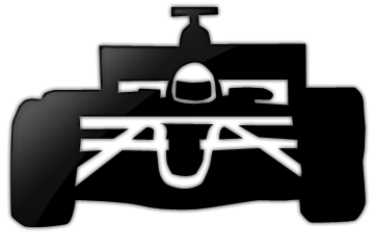
An example



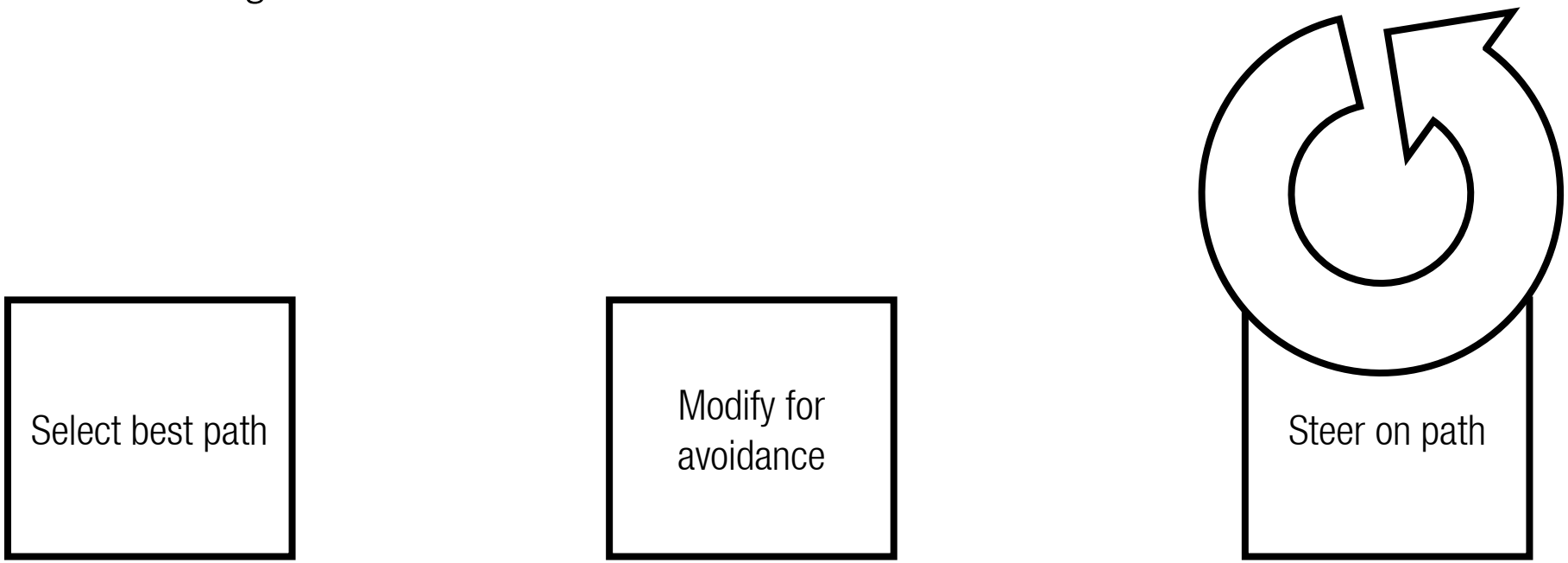
- Best possible curve path is of course constantly blocked. So at a tighter cognitive loop, you engage in avoidance.



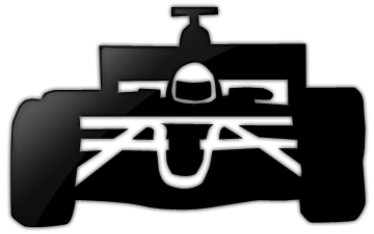
A haptic example



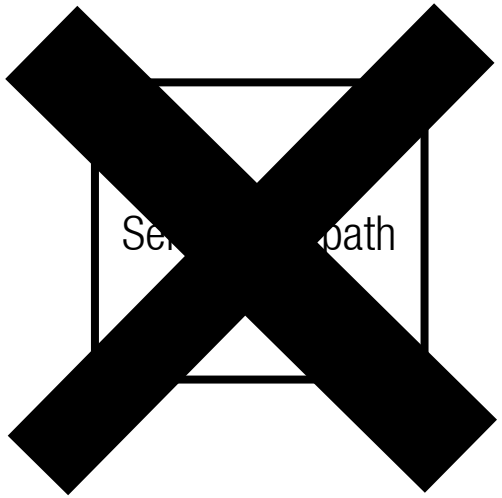
- But the tightest game loop is the loop of the controls, which is a massive autonomic and reflex challenge.

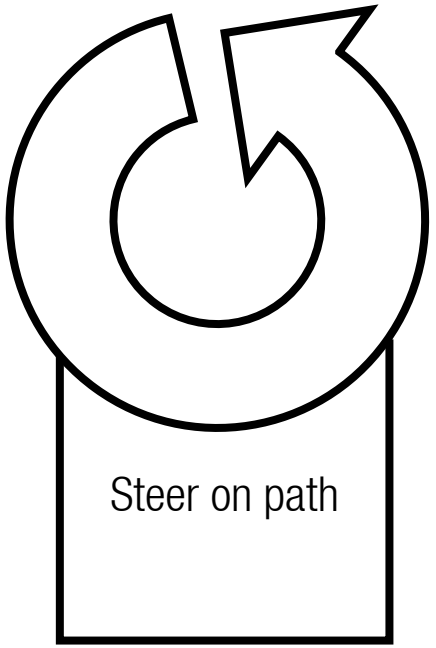


A haptic example



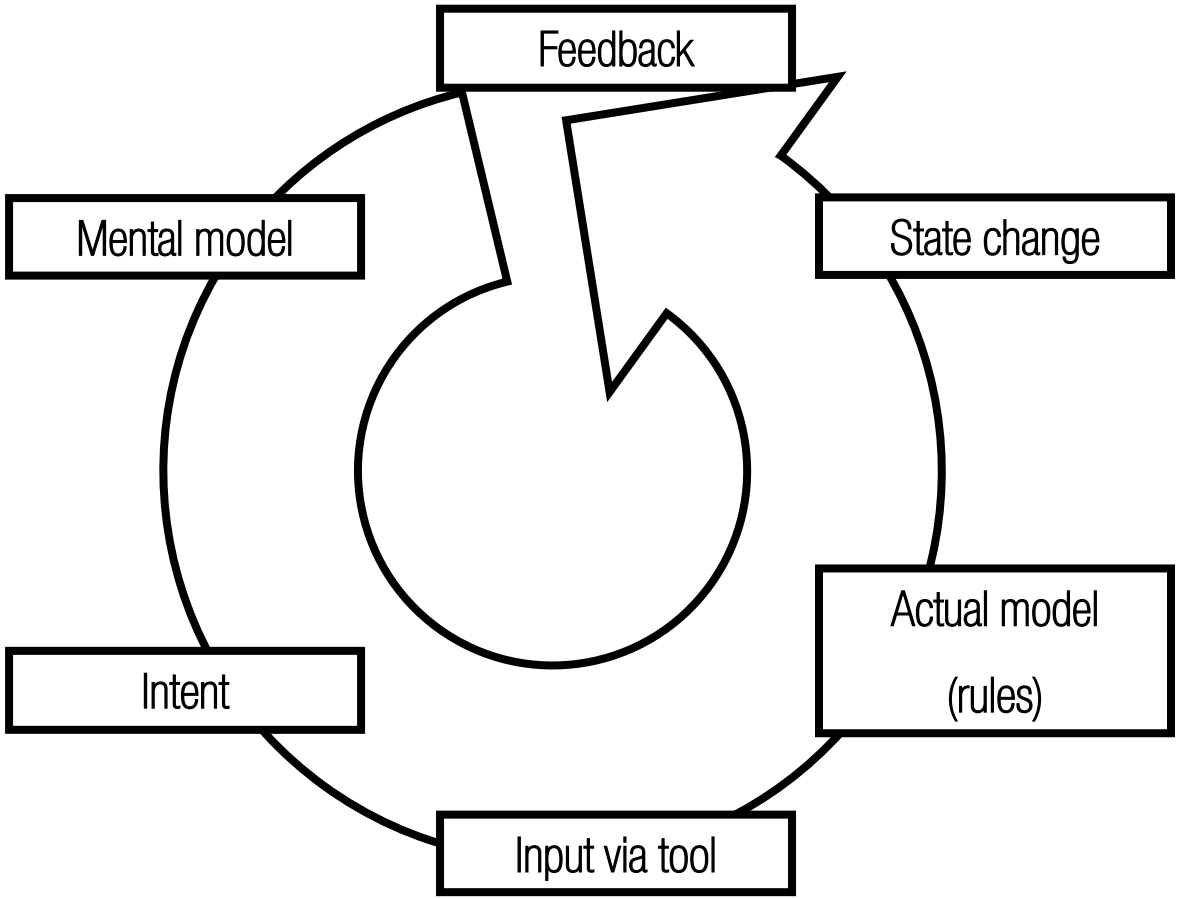
- So racing games these days **give you** the optimal path.





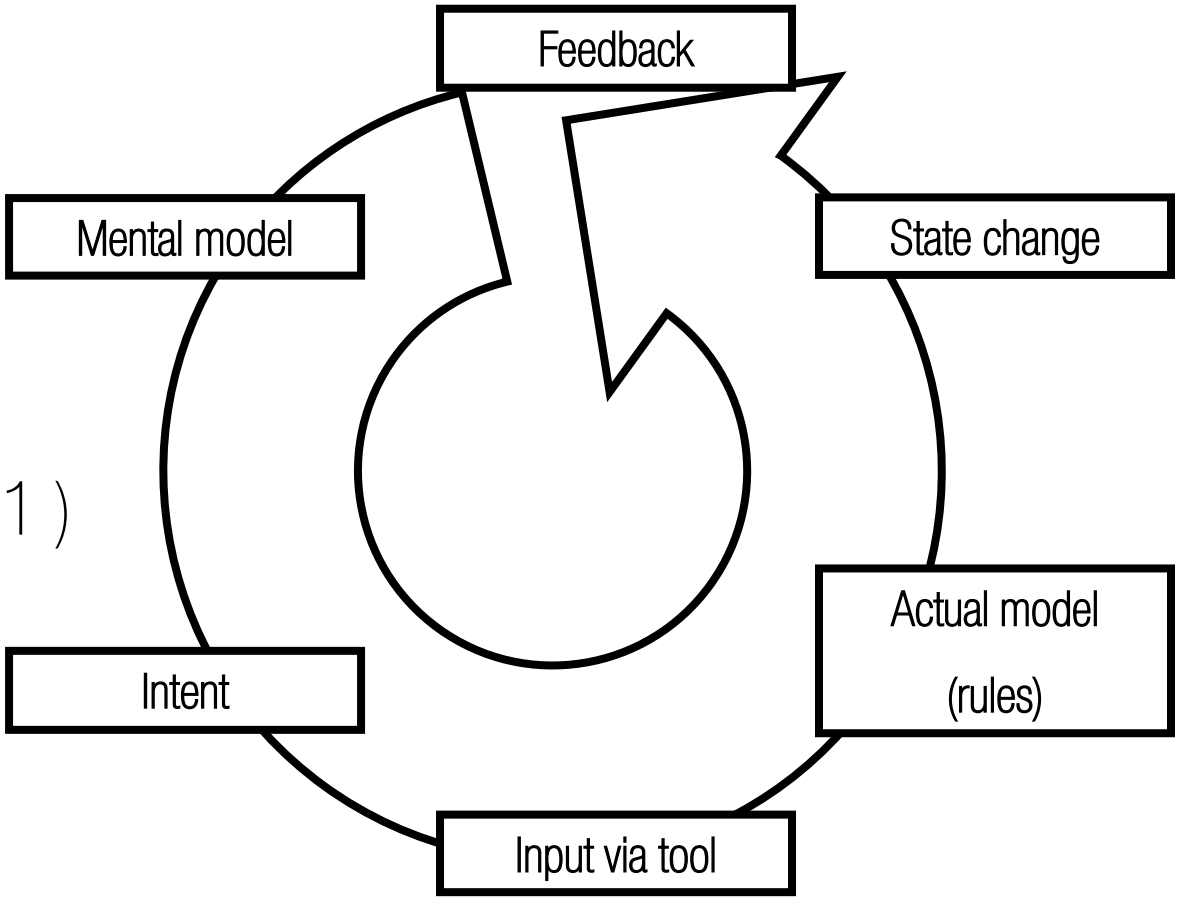
Haptic
problem

Gestures towards a BNF grammar



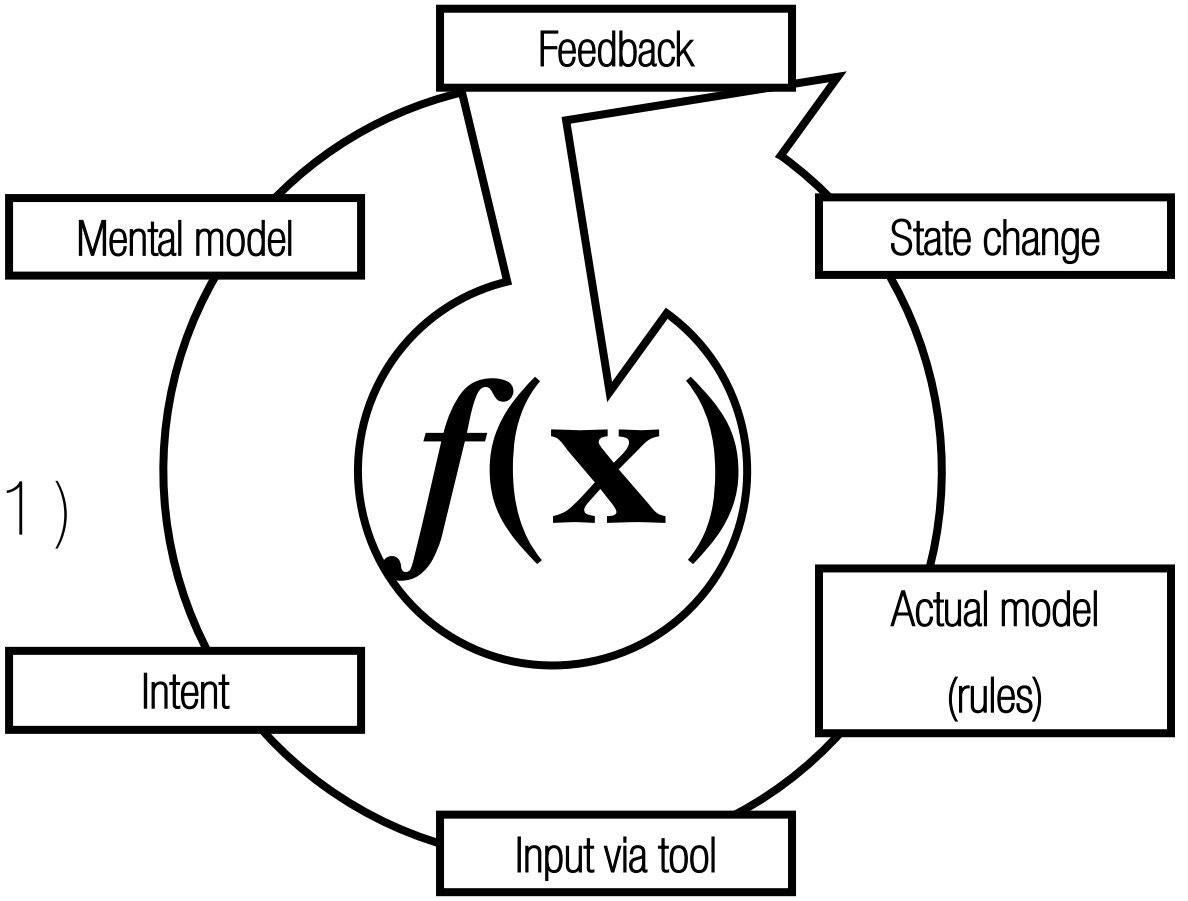
Gestures towards a BNF grammar

- game $\Rightarrow$ game
- game $\Rightarrow$ game + result
- result $\Rightarrow$ system(action) = (Θ , [0 ,] 1)
- result $\Rightarrow$ system(action) = 1



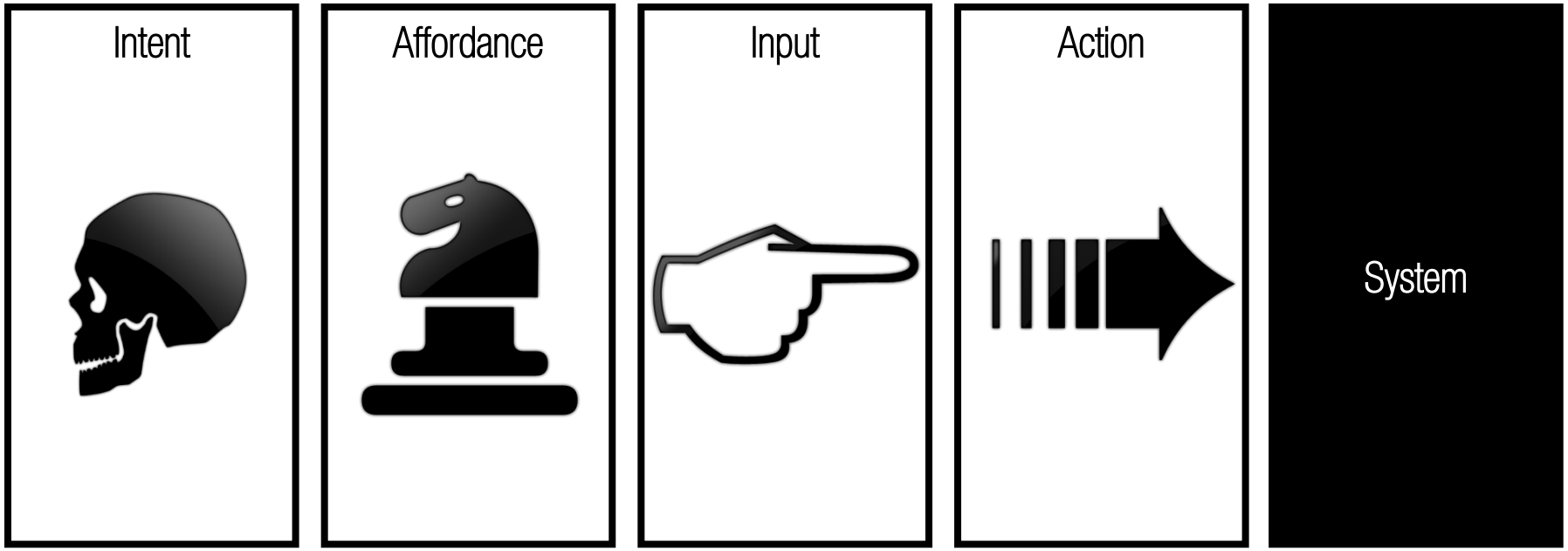
Gestures towards a BNF grammar

- game $\Rightarrow$ game
- game $\Rightarrow$ game + result
- result $\Rightarrow$ system(action) = (Θ , [0 ,] 1)
- result $\Rightarrow$ system(action) = 1



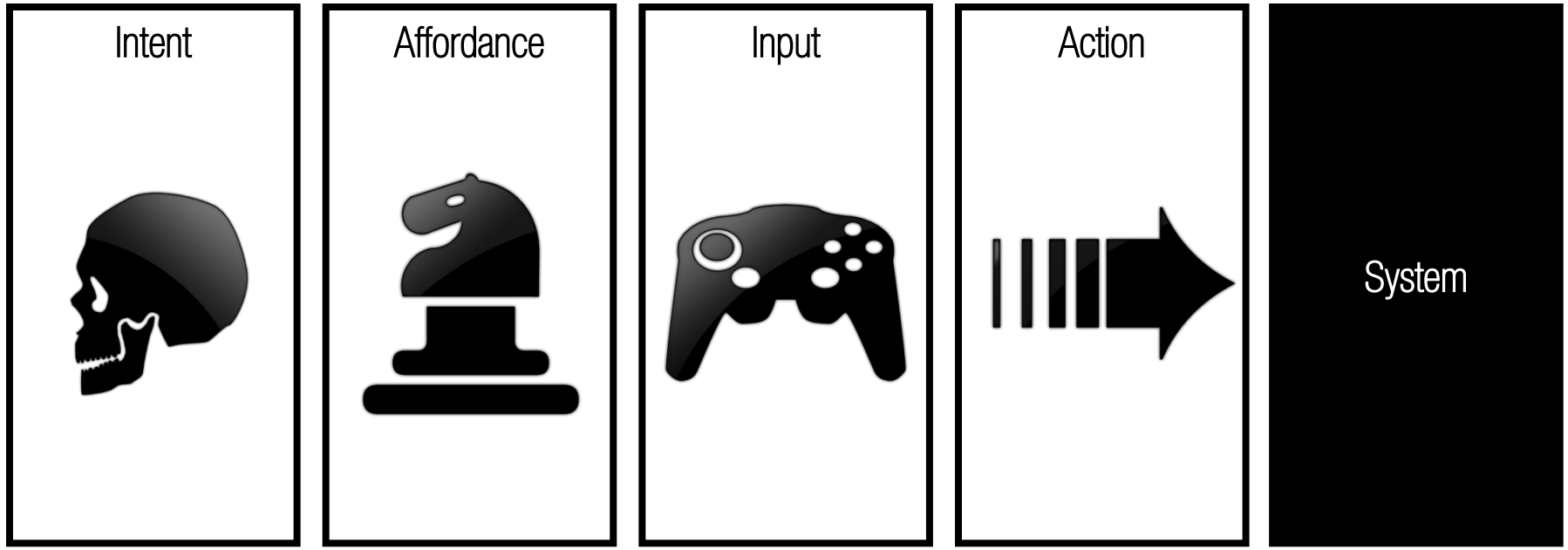
Gestures towards a BNF grammar

- action ⟹ verb [{ + action }] (data)
- verb ⟹ intent + input
- input ⟹ affordance + action



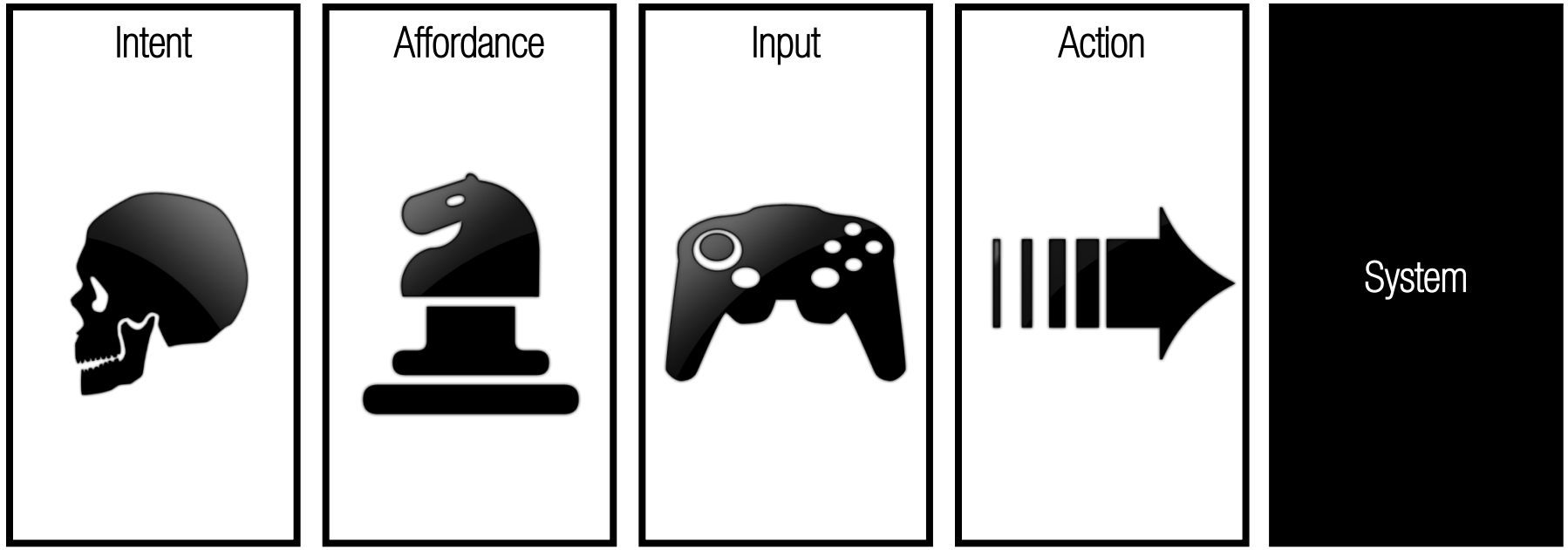
Gestures towards a BNF grammar

- action ⟹ verb [{ + action }] (data)
- verb ⟹ intent + input
- input ⟹ affordance + action

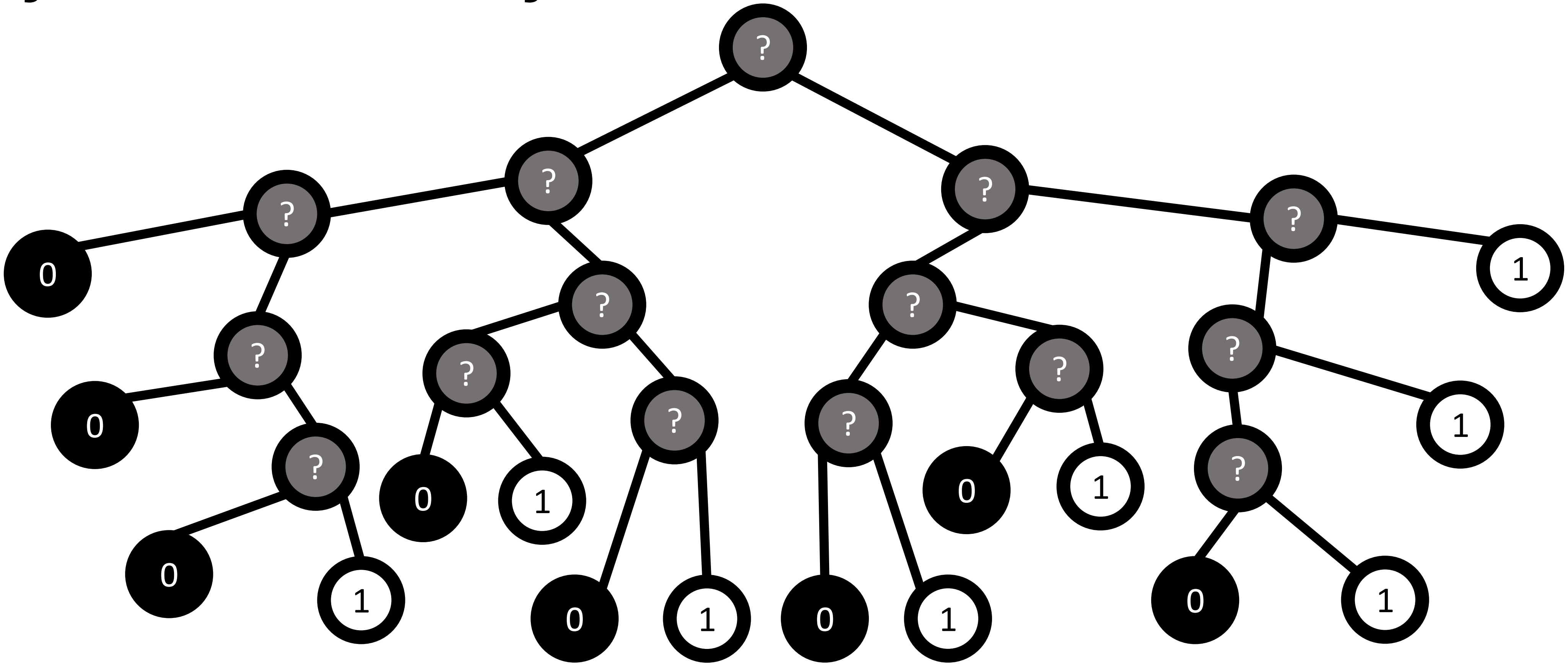


Gestures towards a BNF grammar

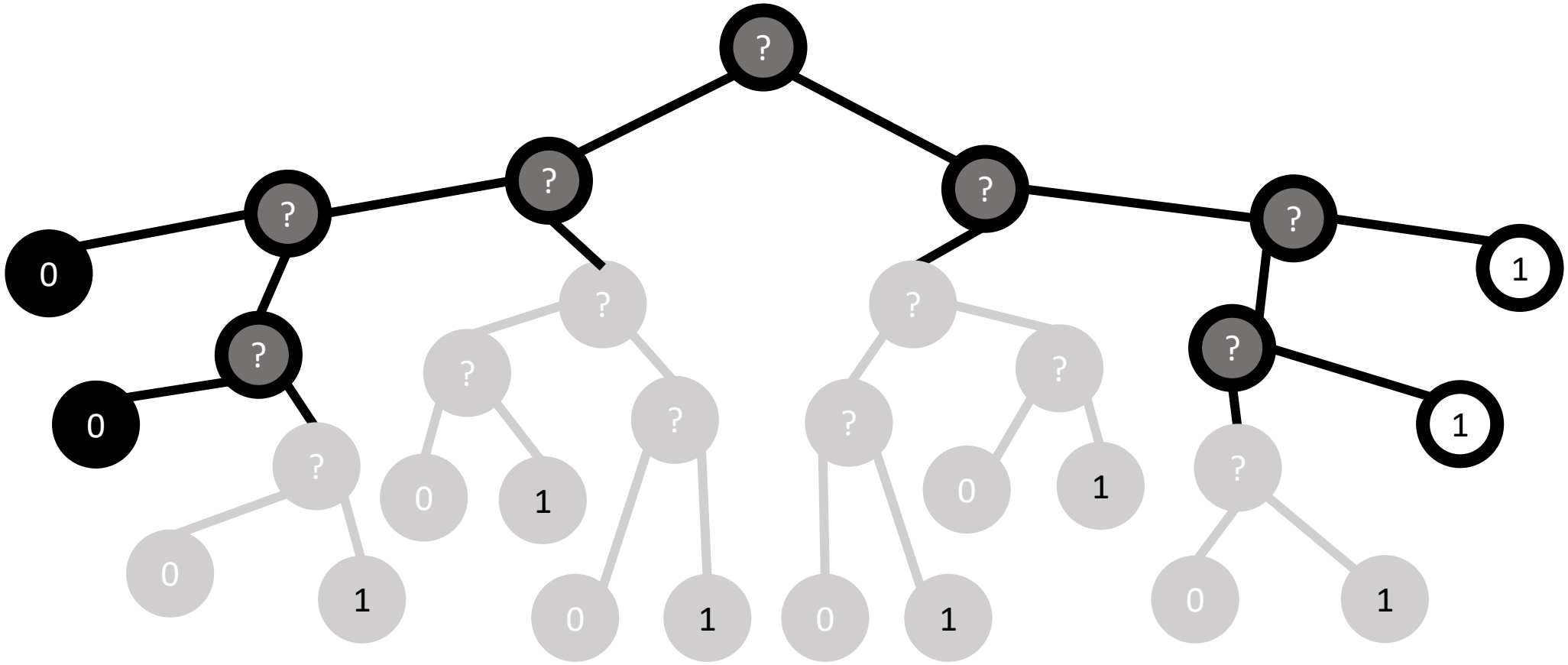
- action ⟹ verb [{ + action }] (data)
- verb ⟹ intent + input
- input ⟹ affordance + action **[+ game]**



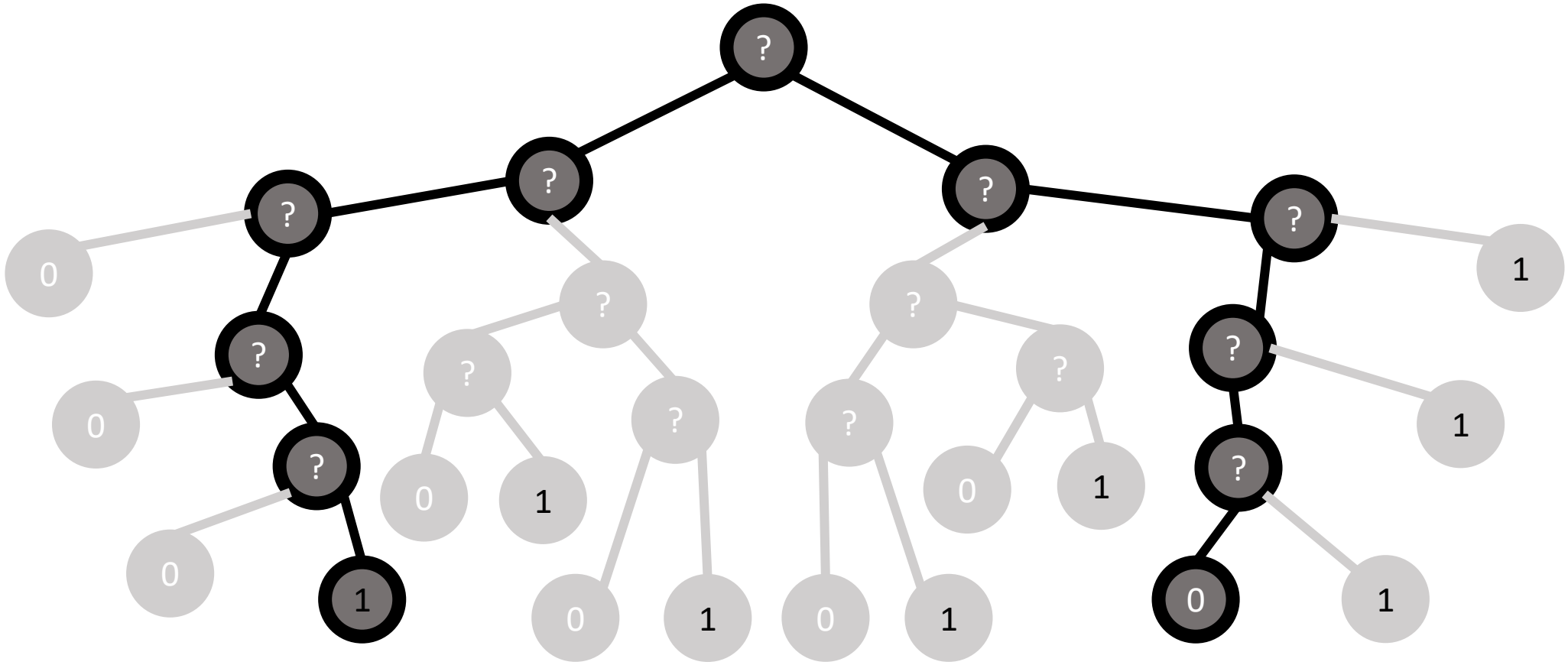
Play is indeterminacy



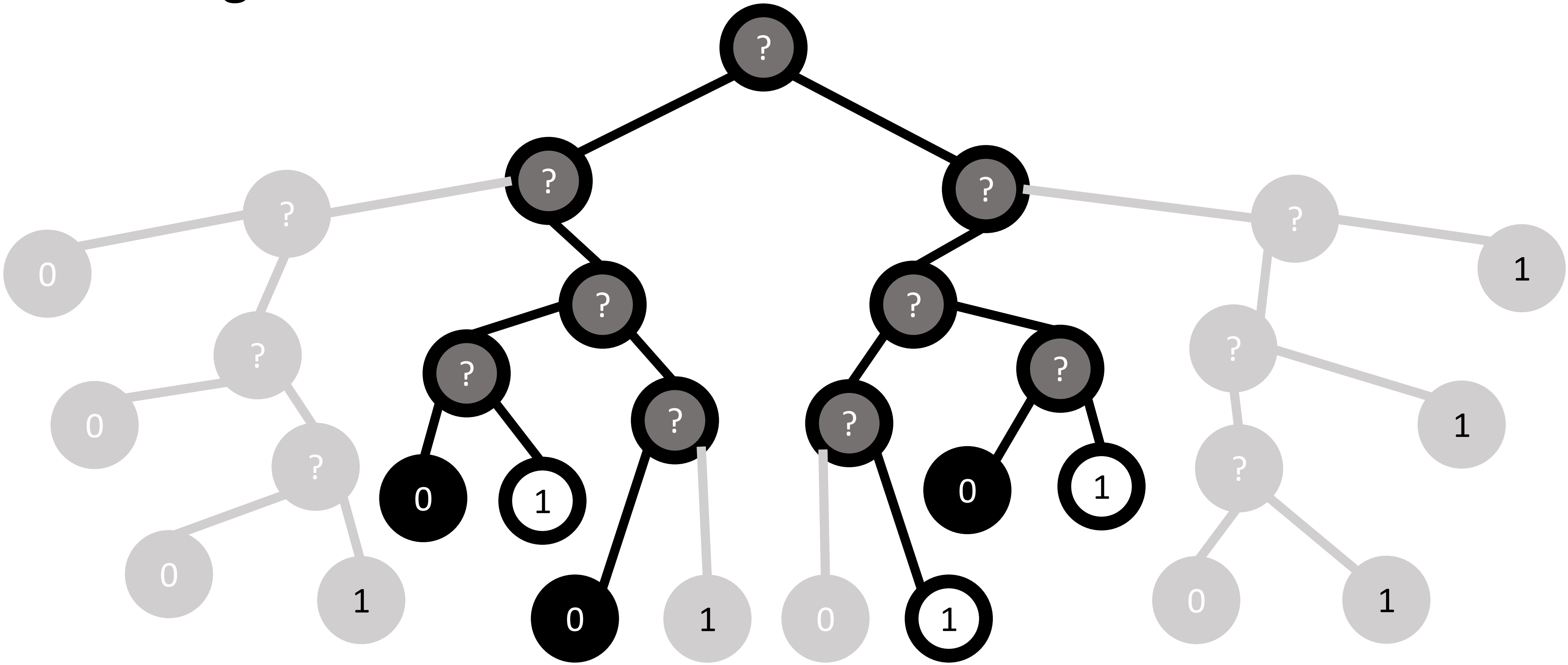
“Blow outs”



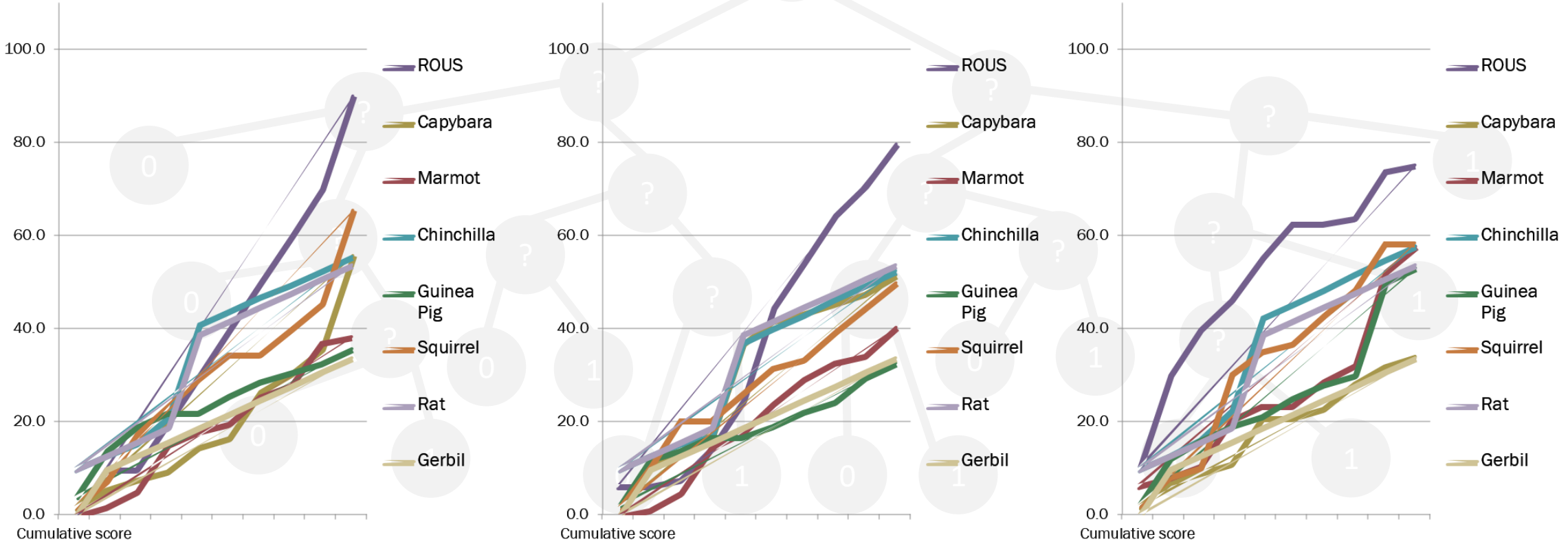
“Comebacks”



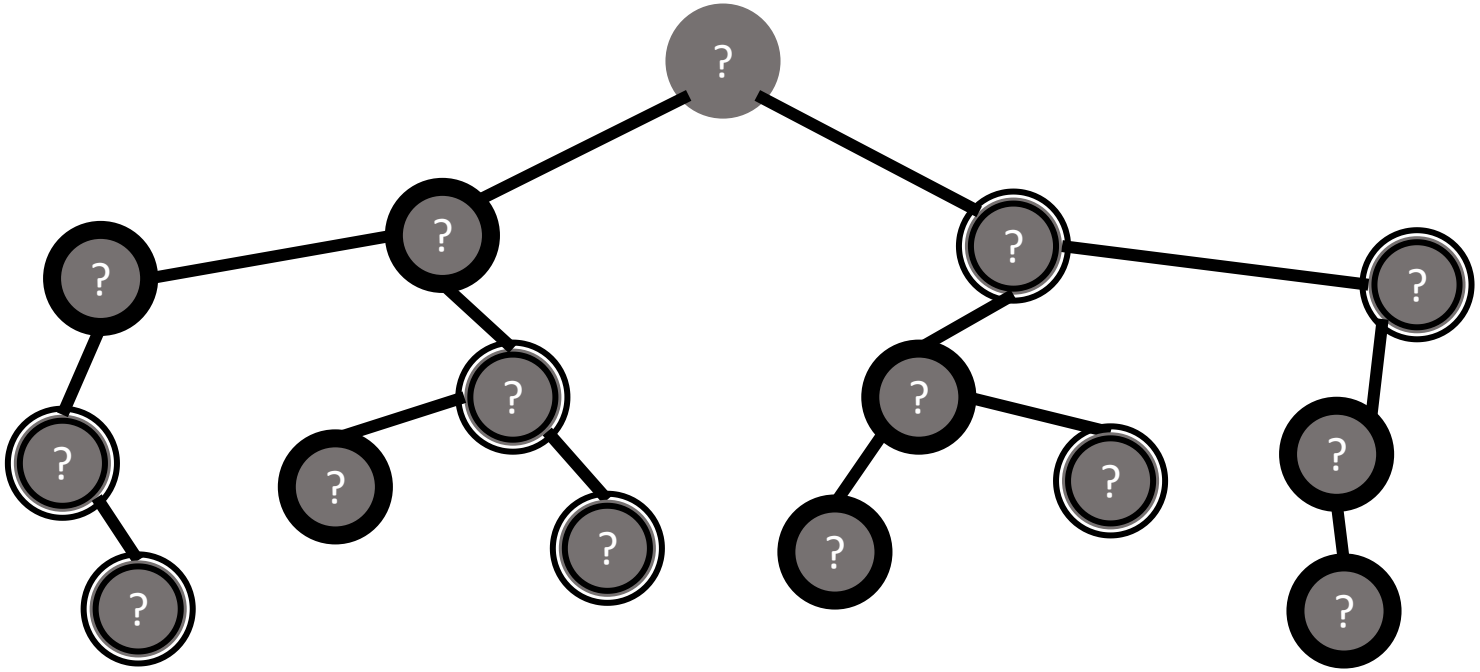
“Hard fought”



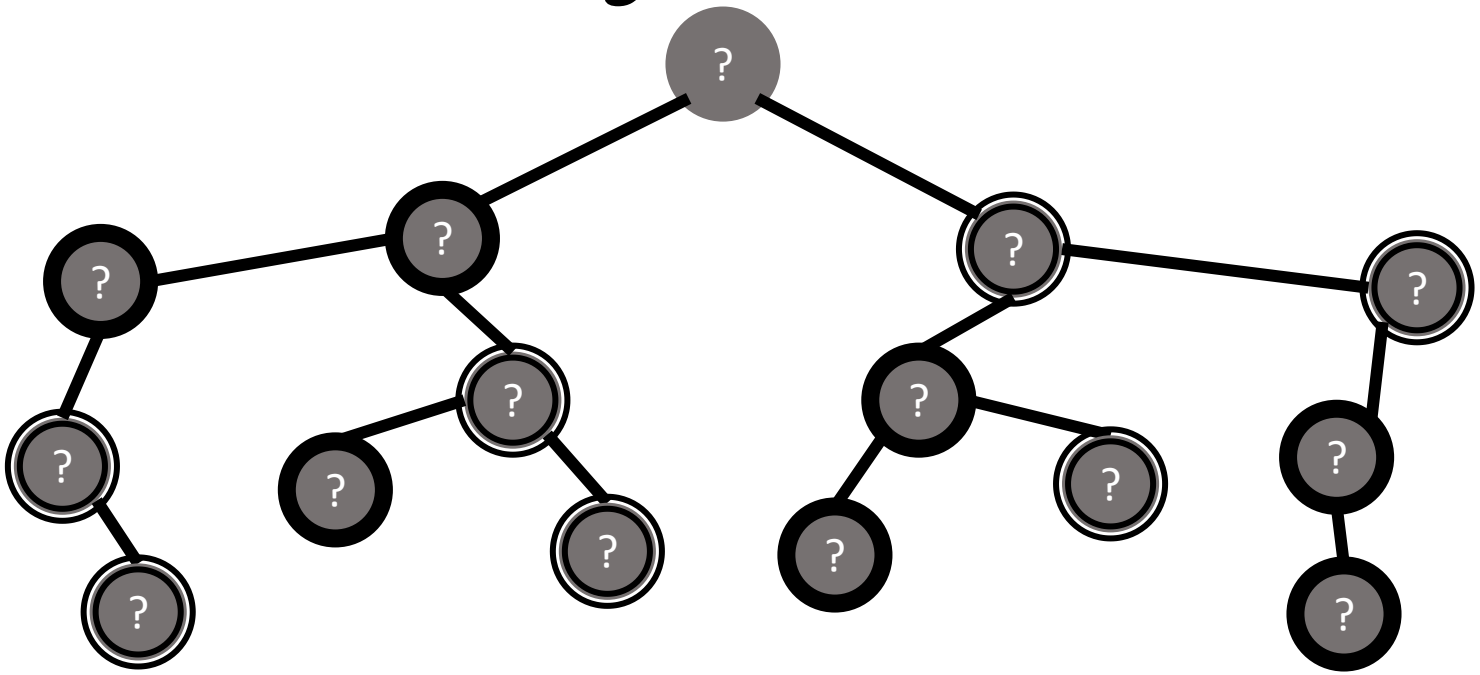
“Hard fought”


100.0
80.0
60.0
40.0
20.0
0.0
Cumulative score
ROUS
Capybara
Marmot
Chinchilla
Guinea Pig
Squirrel
Rat
Gerbil

Limited look-ahead



Consider nodes as *strategies*



- A human-centric rather than computer-centric view:
  - To a human not all affordances are visible.
  - And an affordance **is also a game**.

Parity problems

- We can think of gains and losses as we move down the tree as a parity problem.
  - In the complexity theory sense, or the permutation sense… not the sieve theory sense.
  - Indeed, there are games where passing (a shift-by-one) is an important strategy.

- More importantly, choice of a branch is effectively an evaluation of the total parity of the foreseeable tree.
  - Practical takeaway: orthogonal rules presenting two types of victory points instantly add depth for humans, because the two branches may favor different win conditions.

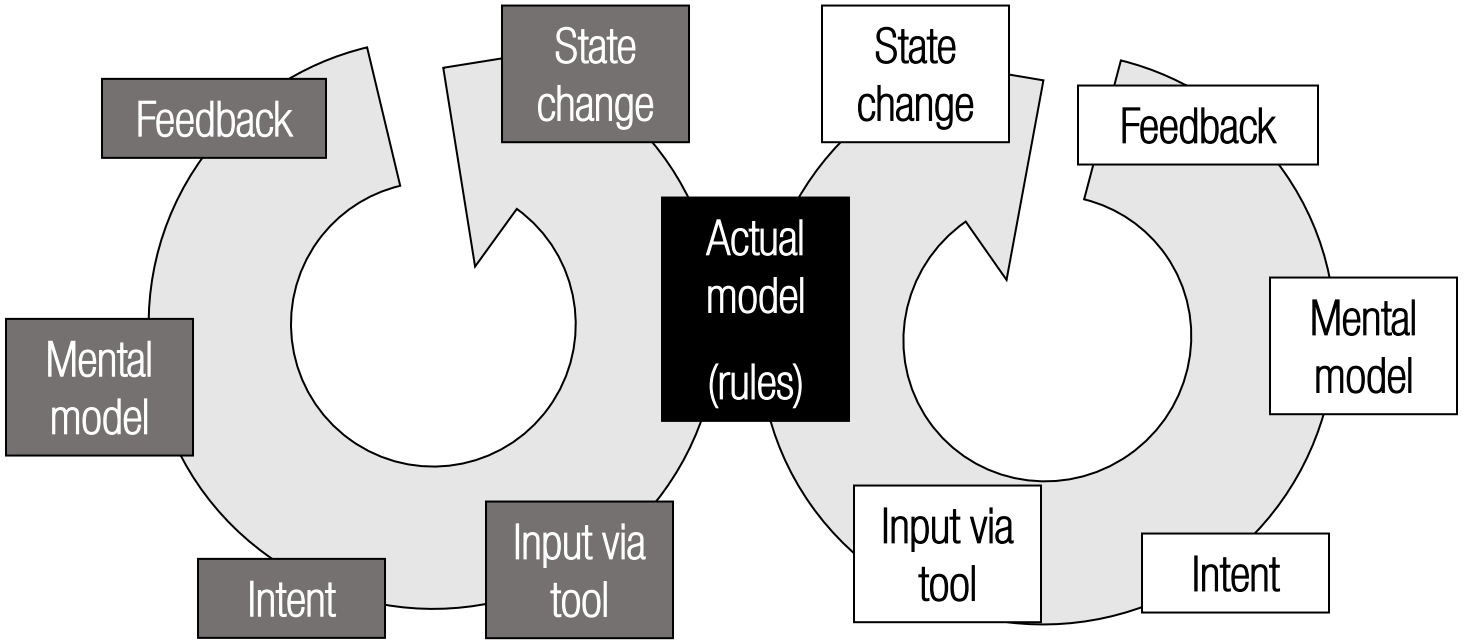
Unbounded or cyclical games

- Most games incorporate cycles in their tree.
  - Once you see the tree as loops that can run infinitely, trees start looking like an inadequate visualization or mental model.
  - Hence why systems theory tools are so popular in grammar-style approaches.

- Games where there isn't a unidirectional transfer of available resource.
  - Narratively centered games. Huge area!
  - State shifting, such as moving tokens.
  - Games where pieces can be added and removed, leading to cycles and loops.
  - Many games special case loops.
    - *Draw by threefold repetition in chess: a recursion limit.*
    - *The ko rule in go: an illegal branch.*

Gestures towards a BNF grammar

- game ⟹ game + result
- result ⟹ system(action) = (Θ , [0 ,] 1)

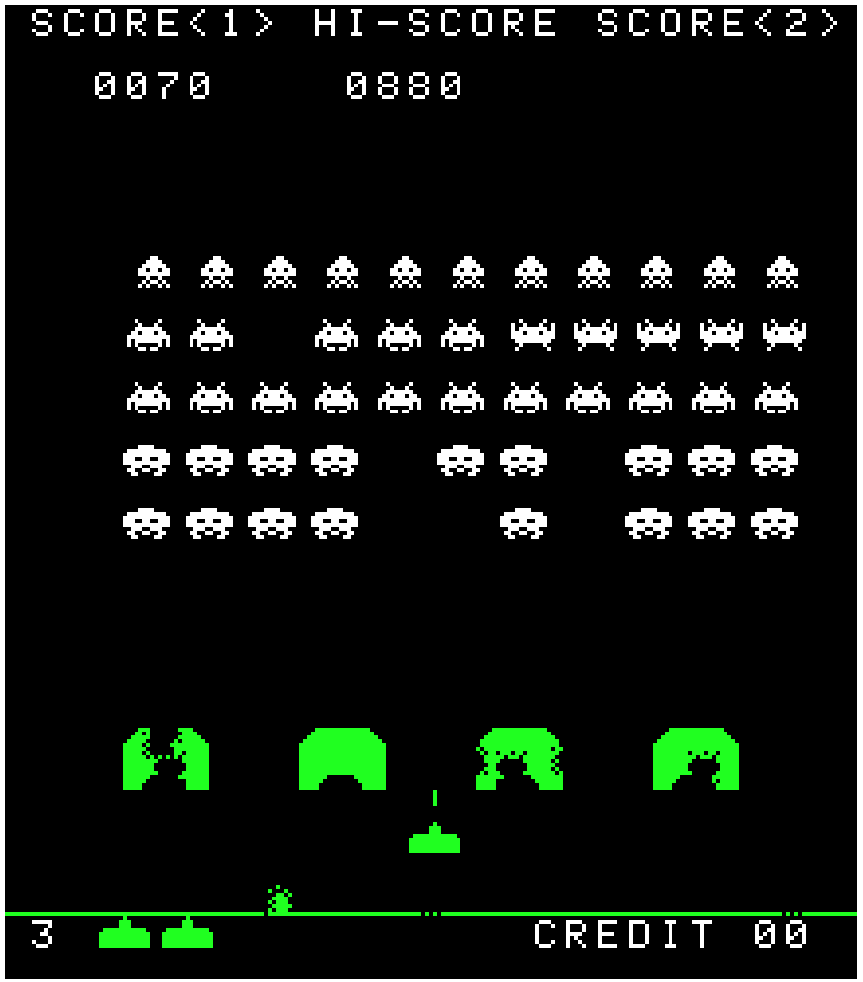


Gestures towards a BNF grammar

- game ⟹ game + result
- result ⟹ system(action) = (Θ , [0 ,] 1)

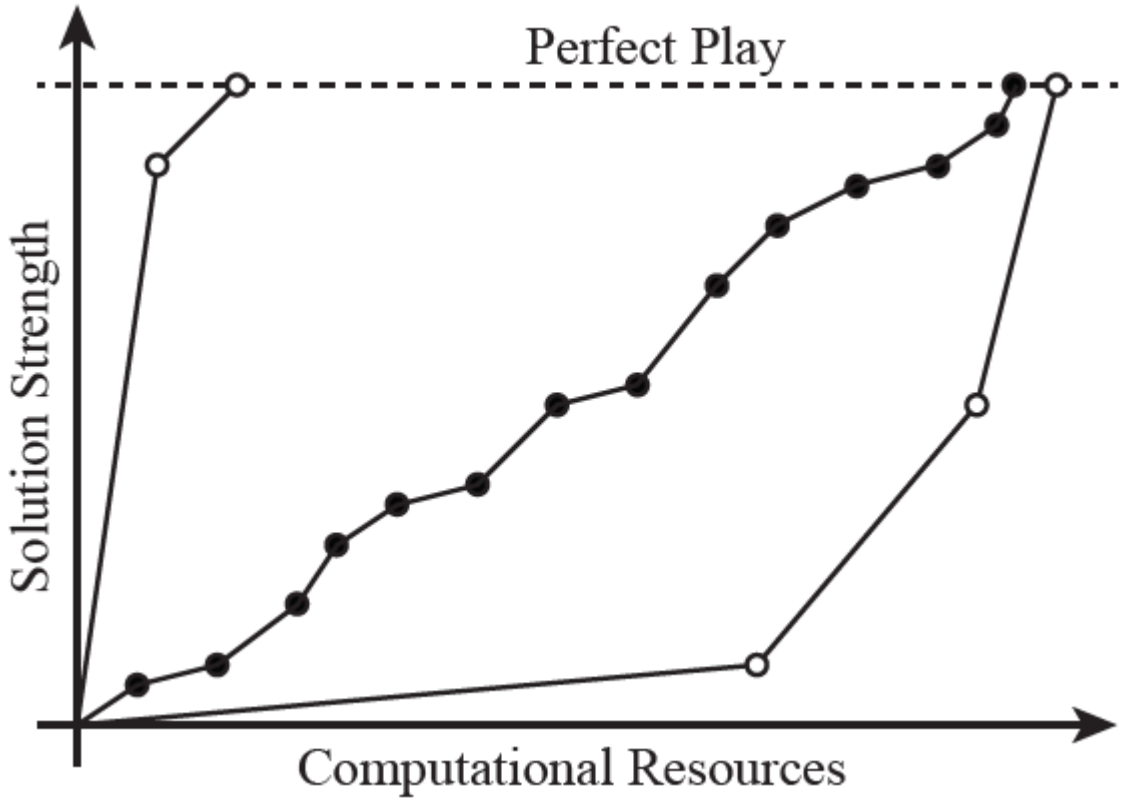
- system ⟹ side + side { [, side] }
- side ⟹ player + mechanic + statistic { , statistic }
- player ⟹ ((mind + experience + body) | AI)

- mechanic ⟹ rule { , rule } + statistic { , statistic }
- rule ⟹ token { , token } expression { , expression } statistic { , statistic }



Lantz *et al* define (a) depth: *d*

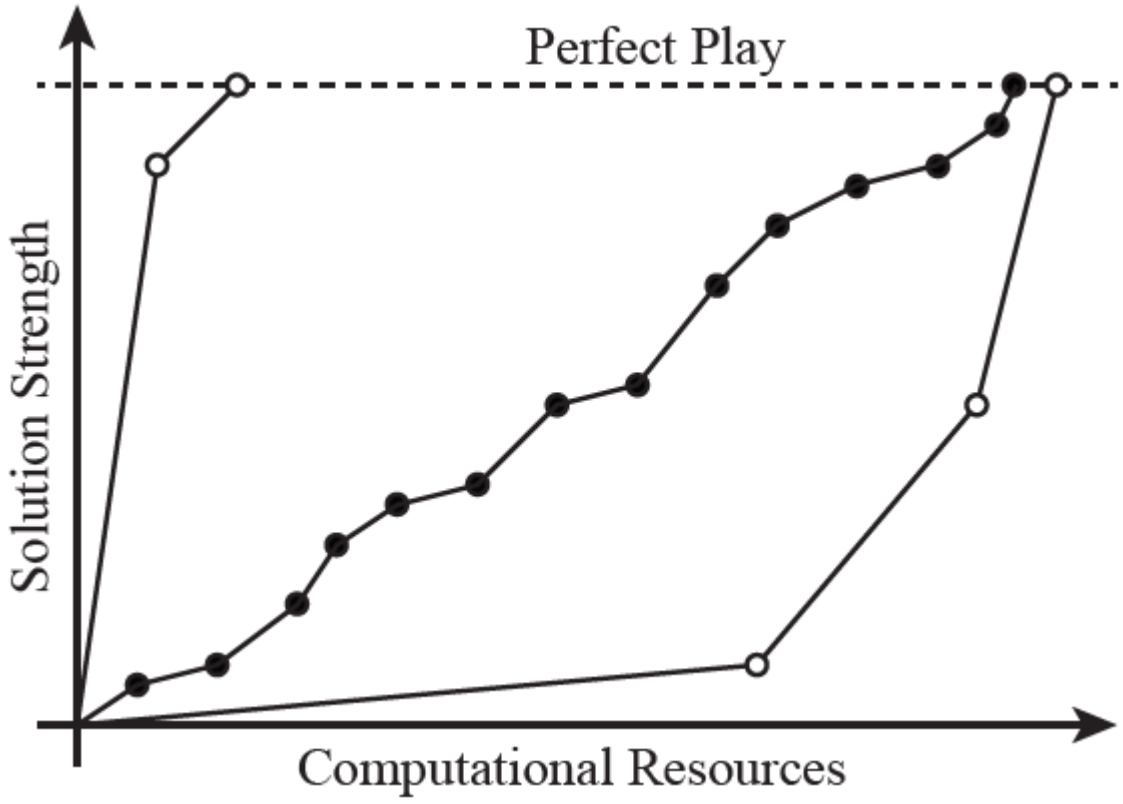
- The capacity for a game system to allow for a ranked population of **strategies** that provide **partial/approximate solutions**.
- The more discrete ranks in such a population, the greater the degree of this property within the game.



https://aaai.org/ocs/index.php/WS/AAAIW17/paper/download/15142/14780

Implications

- A quality of the number of intermediate strategies, or **heuristics** short of optimality, that a player can discover as they proceed to complete mastery.
- A game you can play optimally or near optimally with **a computationally cheap strategy** lacks *d*.
- A game you can't discover new strategies in will also lack *d* because it's based on skill chaining or learning. In a sense, depth by their criteria is based on lemmas, on plateaus.



Heuristics = strategies

- "Control the center" $\Rightarrow$ stay in this area of the state space.

- Regularity in the tree bounds the "good game" space.
  - Too high means degenerate strategy; it's easy to arrive at optimal play.
  - Too low means no heuristics work, because of excessive variability.

- Heuristics are effectively brute force search shortcuts in this view.

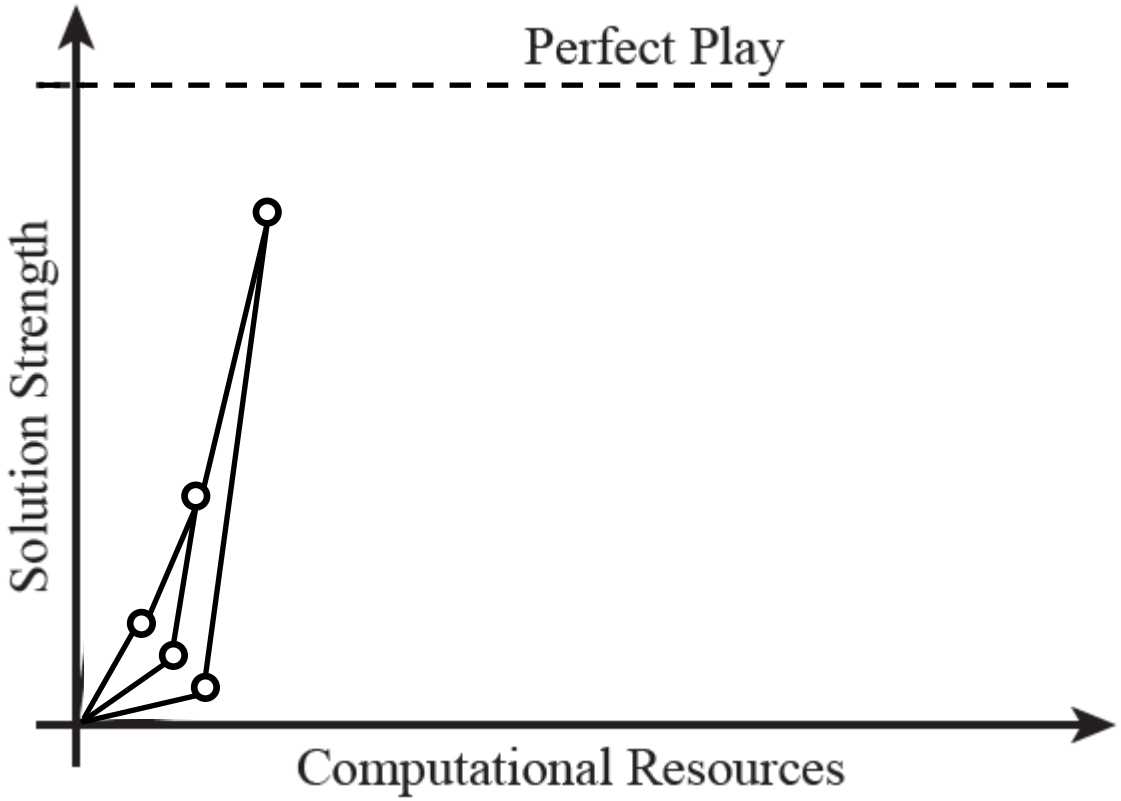
Flaws as utility function for designers

- Heuristic abandonment isn't represented.
  - A classic form of learning is to reach a lemma and be obliged to abandon previous mental models.
- Non-transitive strategies aren't accounted for.
  - Rock-paper-scissors structures at the low end, rich structures in fighting games.
- We don't have any good way to measure computational complexity per heuristic.
  - And if we solve the game, all heuristics but one are retroactively disproven.
- The Dan Cook critique: "Any rubric for depth that does not include a psychological filter will always result in examples that are interesting for a machine, but uninteresting to a human."

What about this game?

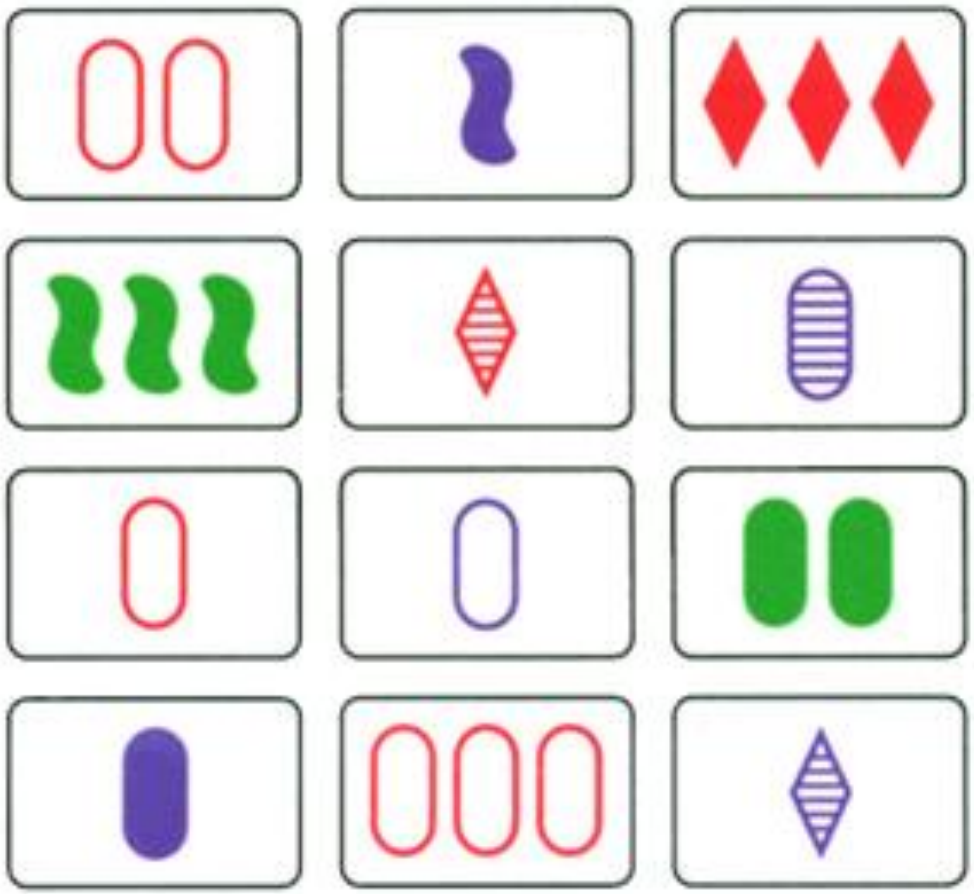
- The game offers many heuristics…
  - All of roughly the same strength.
  - All of roughly the same computational resources.

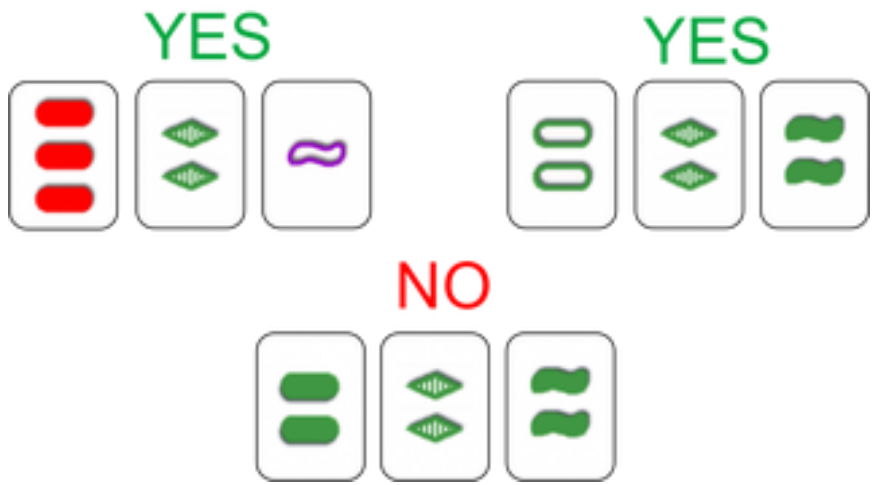
- Rather than heuristics chaining, higher solution strength is achieved by heuristic **selection and combination**.
  - And a lot of this is about **affordances** not inherent complexity.



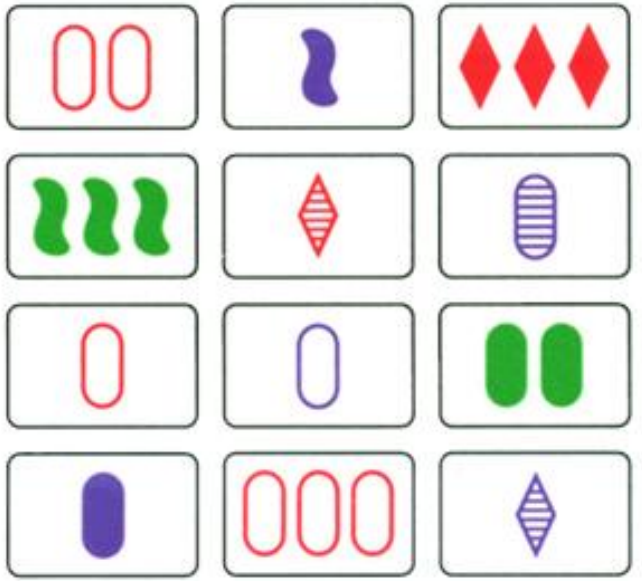
Example: Set


20
BEST
GAME
AWARDS
SET
THE FAMILY GAME
OF VISUAL PERCEPTION




YES
YES
NO

Computer algorithm



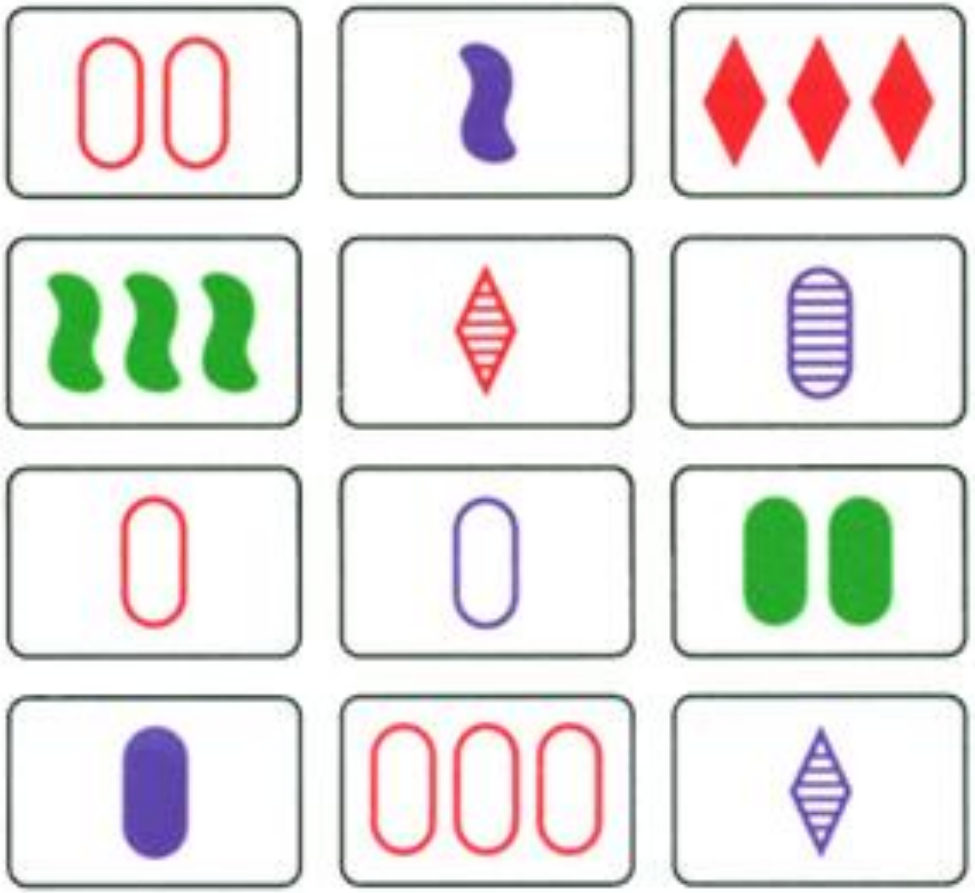
- Equal capability to group things, given image recognition or data.
- No preference in algorithm, except the ordering of IF statements.
- 12 card spread is trivial; can brute search everything.

- Strategy vs equal competitor would be about choosing IF statement ordering.
  - Select an algorithm that reduces the available sets that opponents search for first, so that computation speed increases over iterations.
  - E.g., if all computers search color first, the computer that searches for sets that tend to reduce color commonality sets would gradually come out ahead.
  - But a computer that selects search method randomly wouldn't be vulnerable to this attack.

Human strategies

- Humans find some types of congruency easier to see.
  - All the same on any axis is the easiest to match.
  - Next easiest is quantity sequence.
  - Then come shape and shading.

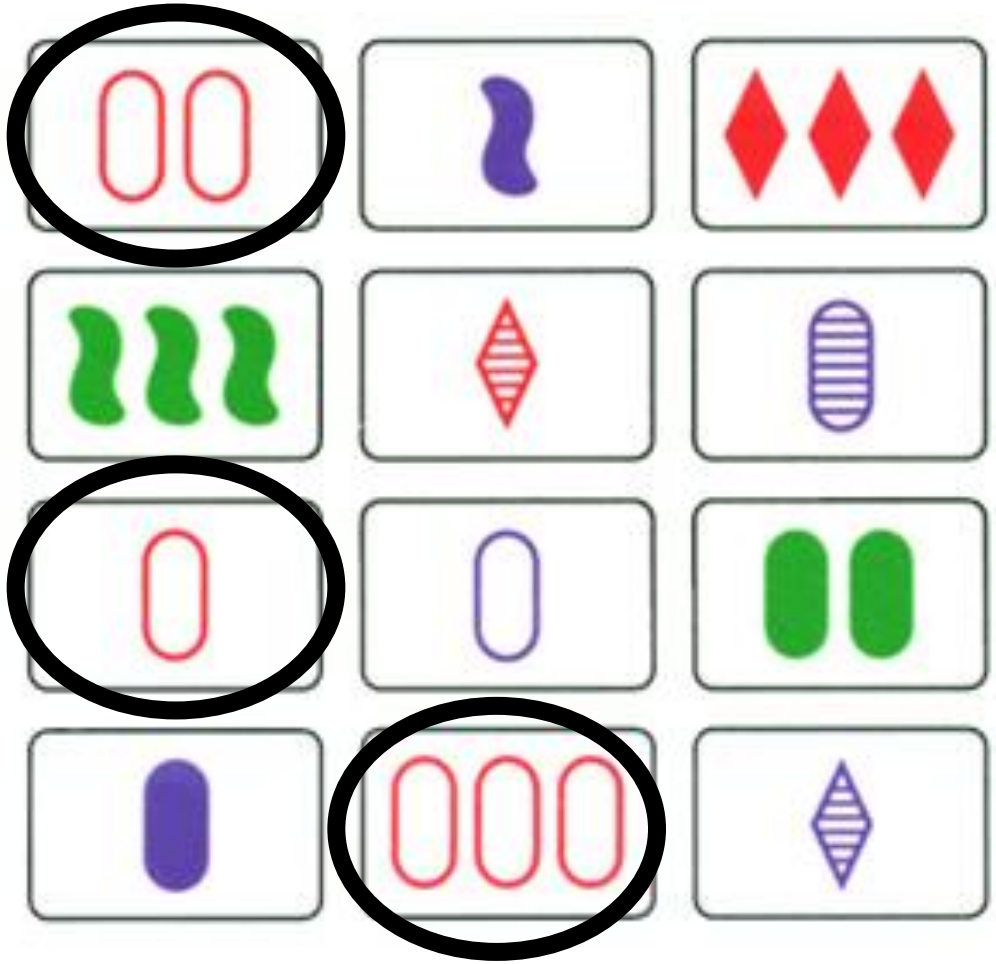
- Human cognition clusters, rather than run brute force search.
  - A player who focuses on shape difference sets first may disrupt other players by destroying viable sets remaining in the deck.



Human strategies

- Humans find some types of congruency easier to see.
  - All the same on any axis is the easiest to match.
  - Next easiest is quantity sequence.
  - Then come shape and shading.

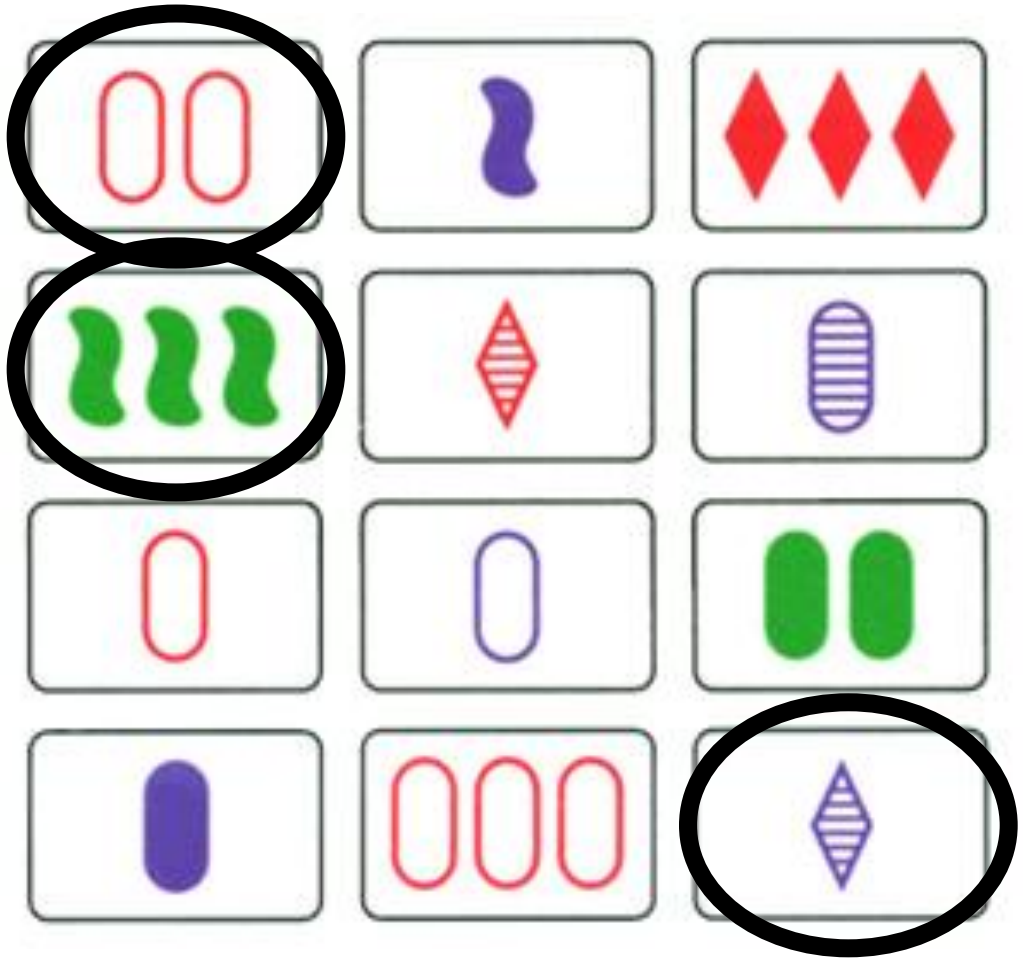
- Human cognition clusters, rather than run brute force search.
  - A player who focuses on shape difference sets first may disrupt other players by destroying viable sets remaining in the deck.
  - E.g. a player who searches non-congruity first prefers high entropy spreads.



Human strategies

- Humans find some types of congruency easier to see.
  - All the same on any axis is the easiest to match.
  - Next easiest is quantity sequence.
  - Then come shape and shading.

- Human cognition clusters, rather than run brute force search.
  - A player who focuses on shape difference sets first may disrupt other players by destroying viable sets remaining in the deck.



Forms of indeterminacy affect computational resources

- **Input randomness:** where the landscape for the system is randomized.
  - Forces use of heuristic selection; lots of possible *x*'s for f(x).
- **Output randomness:** where a system result is randomized rather than simulated.
  - Forces contingency planning.
- **Hidden information:** where a portion of the critical data for a decision isn't available and has a wide spread of possible values.
  - Increases raw search.
- **Practical non-computability:** where perception time, reaction time, and execution time are all so tight that you can't actually use anything other than instinct and reflex.
  - Analogizes to computational complexity, but admits of the human.

Is depth about…

- …the actual complexity of the game?
  - Systems that are NP-Hard, and so on.

- …the ability to learn more ways to approach the game?
  - If so, we can't leave out "how the player learns these."

- …building mental models of the opponent's mental model?
  - "Yomi" (cf David Sirlin, fighting game community)
  - For that matter, what about mental models of **teammates?**

Alternative takes on depth

- A count of iterations where parity is not expressible as **Θ** or 1 and the parent wasn't also indeterminate.
  - This would capture high complexity systems, eliminate trivial branching, and also capture other forms of indeterminacy.

- The number of hidden affordances (e.g., possible actions)
  - A measure of the complexity level for **perceiving** each affordance.

- The number of possible intents a player could possibly have.
  - Which isn't in any way measurable, but is driven by systemic properties (inputs, etc).

Moment #37

- Daigo is down to one pixel of health. His opponent has plenty left.

https://www.youtube.com/watch?v=KS7hkwbKmBM

Was this deep?

What are the design takeaways?

Moment #37

- Daigo is down to one pixel of health. His opponent has plenty left.
- Daigo precisely anticipates which move his opponent will choose
  - A super attack that involves fifteen separate blows, any one of which would be fatal.

https://www.youtube.com/watch?v=KS7hkwbKmBM

Was this deep?

What are the design takeaways?

Moment #37

- Daigo is down to one pixel of health. His opponent has plenty left.
- Daigo precisely anticipates which move his opponent will choose
  - A super attack that involves fifteen separate blows, any one of which would be fatal.
- He also anticipates when it will begin to within 7ms accuracy.
- He then executes fifteen perfect parries, again with 7ms accuracy **each**.

https://www.youtube.com/watch?v=KS7hkwbKmBM

Was this deep?

What are the design takeaways?

Moment #37

- Daigo is down to one pixel of health. His opponent has plenty left.
- Daigo precisely anticipates which move his opponent will choose
  - A super attack that involves fifteen separate blows, any one of which would be fatal.
- He also anticipates when it will begin to within 7ms accuracy.
- He then executes fifteen perfect parries, again with 7ms accuracy **each**.
- And then executes his own super attack. And wins.

https://www.youtube.com/watch?v=KS7hkwbKmBM

Was this deep?

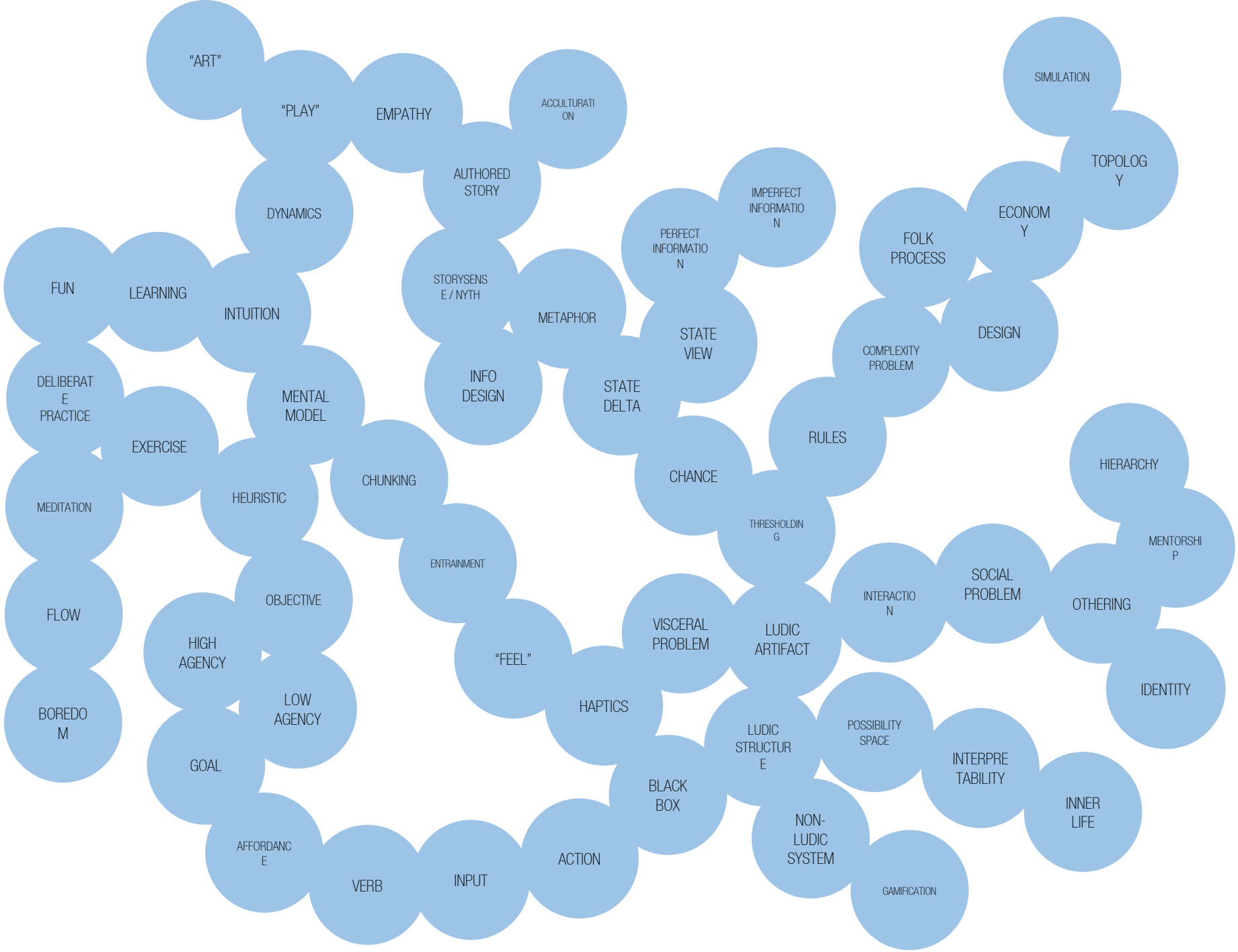
What are the design takeaways?

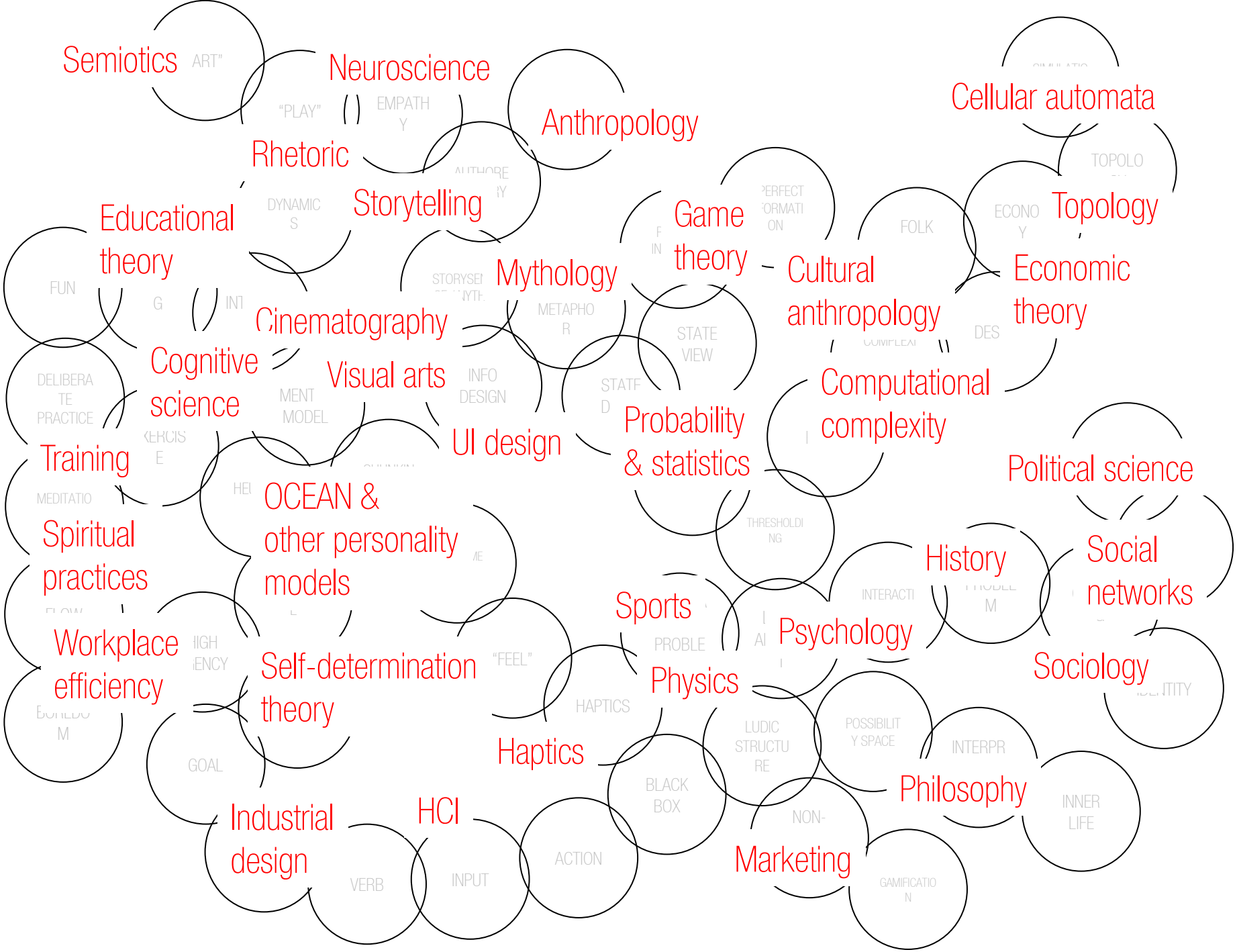
The grammar doesn't say

- game ⟹ result
- game ⟹ game + result
- result ⟹ system(action) = ((Θ, [0,] 1) | 1)
- system ⟹ side + side { [, side] }
- side ⟹ player + mechanic + statistic { , statistic }
- player ⟹ ((mind + experience + body) | AI) + perception
- mechanic ⟹ rule { , rule } + statistic { , statistic }
- rule ⟹ token { , token } expression { , expression } statistic { , statistic }
- action ⟹ verb [{ + action }] (data)
- verb ⟹ intent + input
- input ⟹ affordance + action [+ game]
- affordance ⟹ player (data)

But it's probably ALL of these:

- System of sufficient complexity to permit emergence, nonlinearity, etc.
- Non-transitive strategies.
- Unfolding of heuristics, and heuristic abandonment too.
- Affordances and perceptual limits.
- Yomi, modeling opponent mental models.


"ART"
"PLAY"
EMPATHY
ACCULTURATION
AUTHORED STORY
DYNAMICS
FUN
LEARNING
INTUITION
STORYSENSE / NYTH
METAPHOR
PERFECT INFORMATION
IMPERFECT INFORMATION
STATE VIEW
INFO DESIGN
STATE DELTA
SIMULATION
TOPOLOGY
FOLK PROCESS
ECONOMY
COMPLEXITY PROBLEM
DESIGN
DELIBERATE PRACTICE
MENTAL MODEL
EXERCISE
RULES
CHANCE
HIERARCHY
MEDITATION
HEURISTIC
CHUNKING
THRESHOLDING
MENTORSHIP
ENTRAINMENT
INTERACTION
SOCIAL PROBLEM
OTHERING
OBJECTIVE
FLOW
HIGH AGENCY
"FEEL"
VISCERAL PROBLEM
LUDIC ARTIFACT
IDENTITY
LOW AGENCY
BOREDOM
HAPTICS
LUDIC STRUCTURE
POSSIBILITY SPACE
INTERPRETABILITY
GOAL
BLACK BOX
INNER LIFE
AFFORDANCE
VERB
INPUT
ACTION
NON-LUDIC SYSTEM
GAMIFICATION


Semiotics
Neuroscience
Anthropology
Rhetoric
Storytelling
Educational theory
Cellular automata
Topology
Game theory
Mythology
Cultural anthropology
Economic theory
Cinematography
Cognitive science
Visual arts
Computational complexity
UI design
Probability & statistics
Training
Political science
OCEAN & other personality models
Spiritual practices
History
Social networks
Sports
Psychology
Workplace efficiency
Self-determination theory
Physics
Sociology
Haptics
Philosophy
Industrial design
HCI
Marketing

Thank you!

http://www.raphkoster.com

http://www.theoryoffun.com