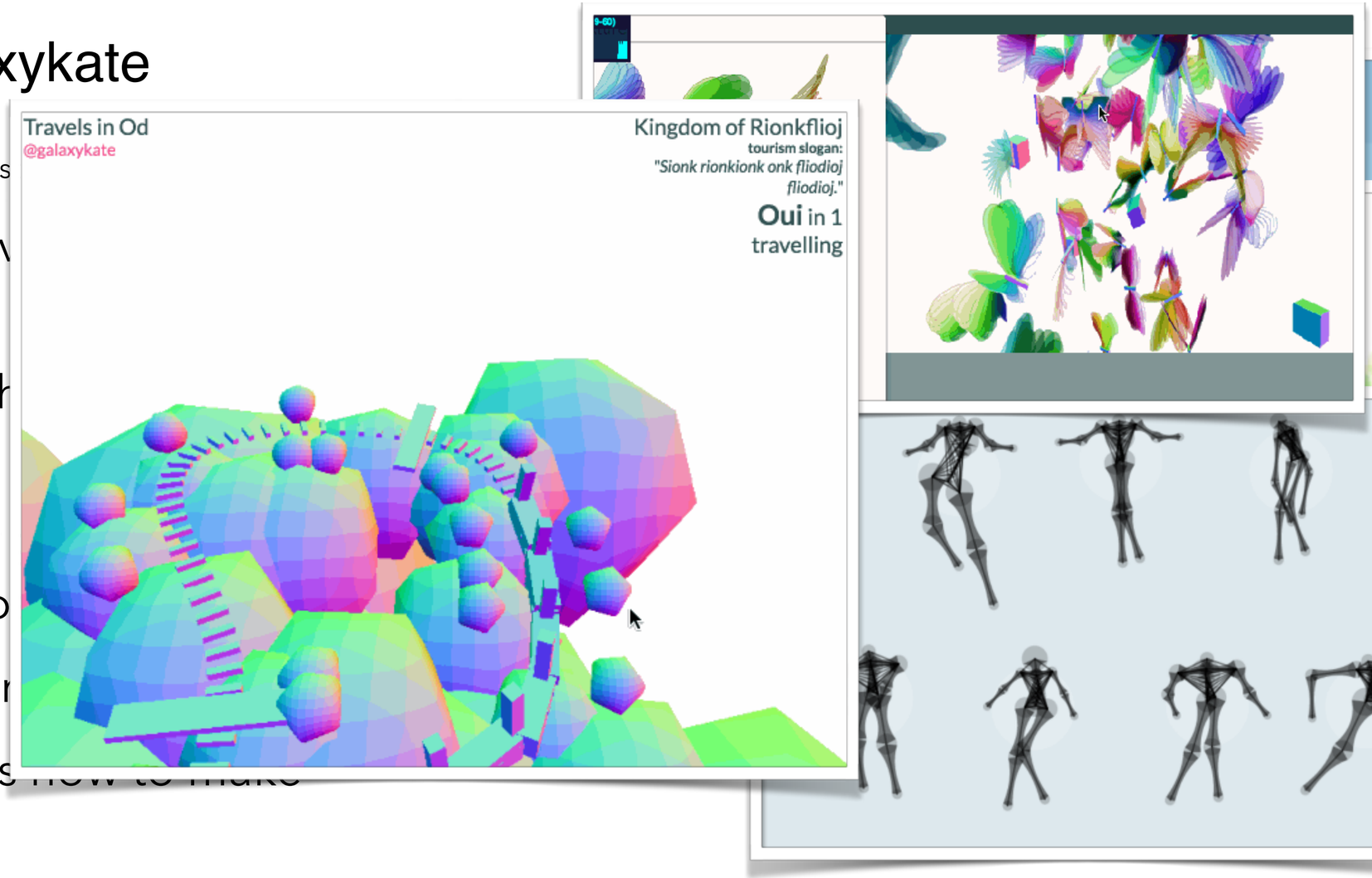
A generative framework for generativity

Kate Compton and Michael Mateas

about me @galaxykate

- 5 years on Spore, SimCity (and s
- 5+ years of a PhD in generativ
  tools for casual users
- Contracting for chatbots on th
  3D printing apps
- Made Tracery, a language for
  for CheapBotsDoneQuick, ho
- I make generative artbots, gar
- I write about and teach others
  generative things.



about me @galaxykate

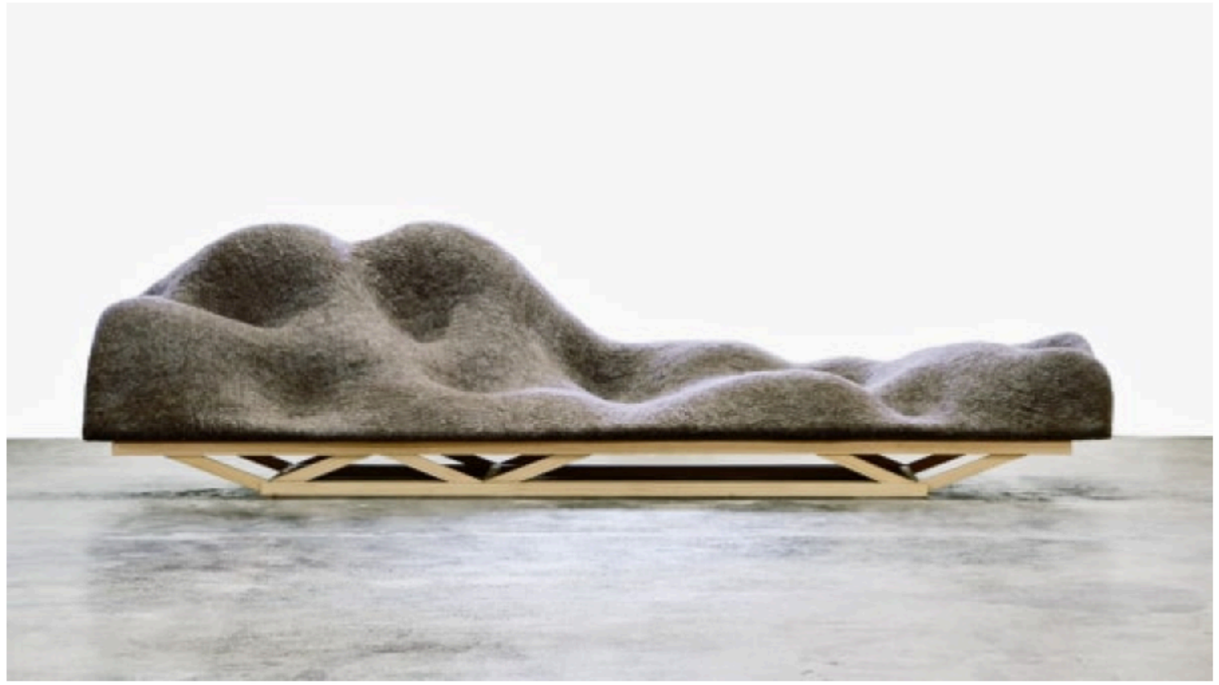
22
2 / 2016

So you want to build a generator...

This is a beginner-level advice essay for people just getting started with building generators. It's also for practiced experts who want a way to

Motivations for the GFG

Italian designer Lucas Maassen wired his brain to an EEG-measuring device and relaxed as he contemplated the word “comfort.” Then he and Belgian Unfold principal Dries Verbruggen translated that brainwave file into instructions for Computer Numerical Control milling machines that carved a block of foam into a biomorphic 3-D couch.



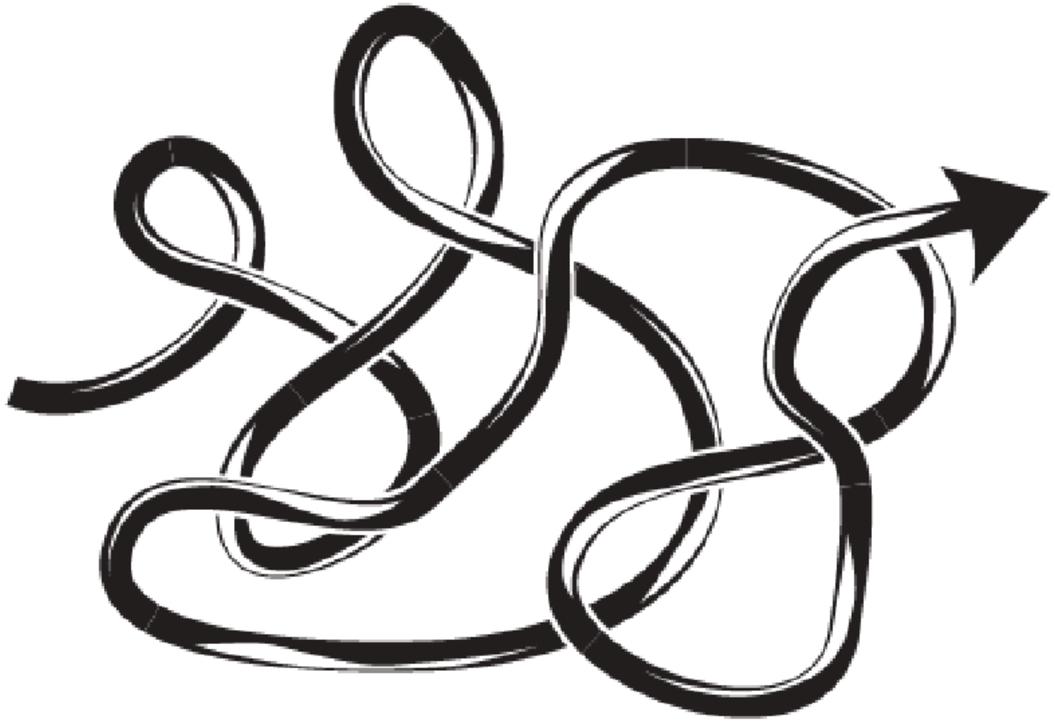
Your Brain_manufacturing

What if you could create a design based on your inner most feelings by measuring your brain activity? Would you like it? How would you relate to it? These where the central questions for Your Brain_manufacturing, a research project by designer Merel Bekking.

Merel Bekking scans the brain of Dezeen founder Marcus Fairs to create his "perfect" design

Motivations for the GFG

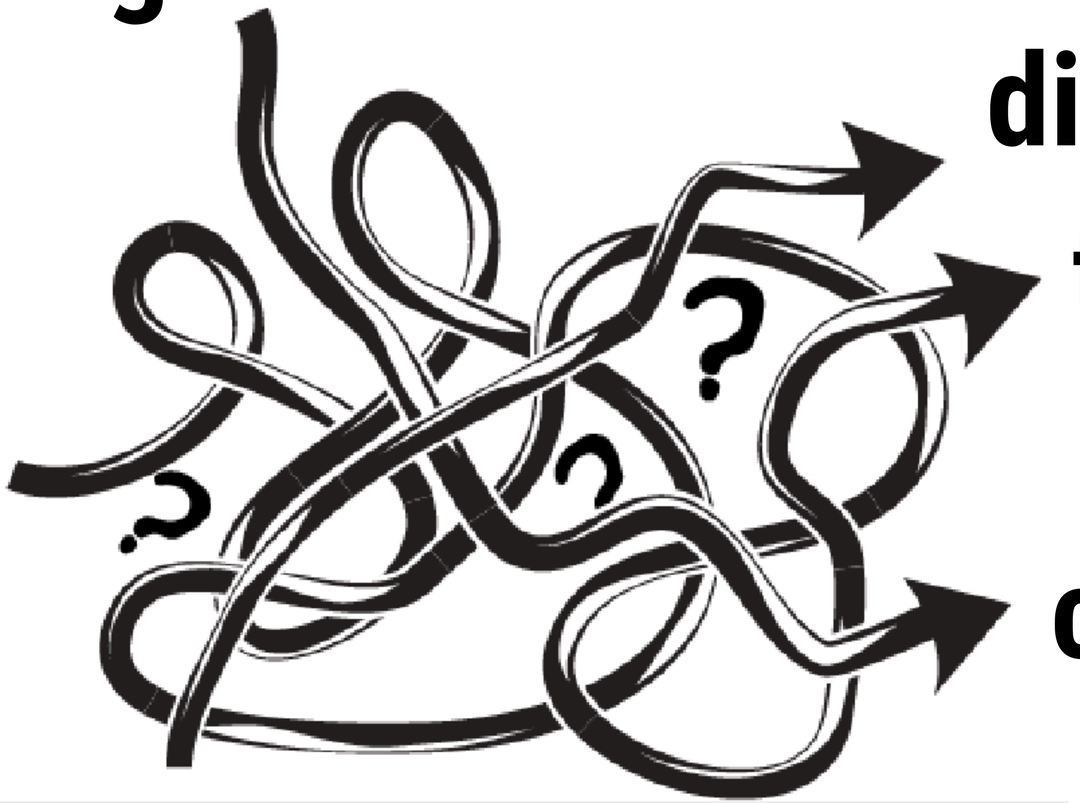
Brain waves



A couch

Motivations for the GFG

“learning”



user actions

difficulty

terrain

characters

simulated ecosystem

Another Framework for PCG?

Another Framework for PCG?

Previous frameworks:

Hendrikx, M.; Meijer, S.; Van Der Velden, J.; and Iosup, A.
Procedural content generation for games: A survey

Togelius, Yanakakis, Stanley, Browne.
Search-based Procedural Content Generation: A Taxonomy and Survey

Smith, Gillian
Understanding procedural content generation: a design-centric analysis of the role of pcg in games

Dorin, A.; McCabe, J.; McCormack, J.; Monro, G.; and Whitelaw, M.
A framework for understanding generative art

Another Framework for PCG?

Previous frameworks:

Nyman (1999)	Experimental music	Categorises works according to how an artist outlines a *situation* in which sounds may occur, a *process* of generating action	Chance determination processes, people processes, contextual processes (dependent on unpredictable conditions, such as the selection of new pitches in Cardew's *The Great Learning*, *Paragraph 7* (1971), see below), repetition processes, and lastly, electronic processes
Dorin (1999)	Kinetic art	An attempt to categorise process types and their elements at a high level of abstraction	A pulse, stream, continuum, increase, decrease and complex processes
Galanter (2003)	Complexity, generative art	Analyses generative processes in terms of their *effective complexity*	Arranges works according to the order/disorder of the processes they employ and plots this against the works' complexity
Wooller *et al.* (2005)	Algorithmic music	Two axes of description: *function* (analytic to generative); and *context* (narrow to broad)	Arranges works according to function (analytic, transformational, generative) of the processes they employ and plots this against the breadth of the works' context
Dodge *et al.* (2008)	Dance	Focus on patterns of movement: hierarchical taxonomy of movement primitives; temporal, spatial and spatio-temporal classifications	Primitive, compound and behavioural movement patterns, synchronisation

Thompson (1917)	Biology	Description of form from a physical and biological perspective; focus on physical processes that generate shape and form	Animal morphology; minimal surfaces; parameterised structural relationships of shells and horns
Stevens (1979)	Nature and natural processes, architecture	Decomposition of all structural form to five basic process types	Spirals, meanders, branches, explosions, non-linear packing and cracking, flows, minimal surfaces
Volk (1995)	Space, time and mind	Discusses reappearance of meta-patterns across a variety of physical and conceptual systems	Sheets, tubes, spheres, arrows, breaks, cycles, layers
Ball (2001)	Biology, chemistry, physics, morphology	A broad survey of pattern-generating systems in the physical sciences	Shapes, flows, branches
Hayter (1965)	Kinetic art	A reductive approach to understanding motion, breaking it into components	Orientation (with respect to a spectator), direction (where the spectator or their eye moves, for instance, across a picture plane), cheirality [sic] (handedness), velocity and rhythm
Rickey (1965)	Kinetic art	A classification of techniques by which an artist introduces elements of movement into their work	Optical phenomena; transformations where movement seems to dematerialise an object; movable works where the spectator rearranges the elements; machines; light play dependent on movement; movement itself
Popper (1968)	Kinetic art	A *typology of movement* classified according to 'procedures' employed by the artist	The means of figuration of movement; methods for movement's representation; photographic or filmic techniques employed; movement expressed by movement itself

Table 1

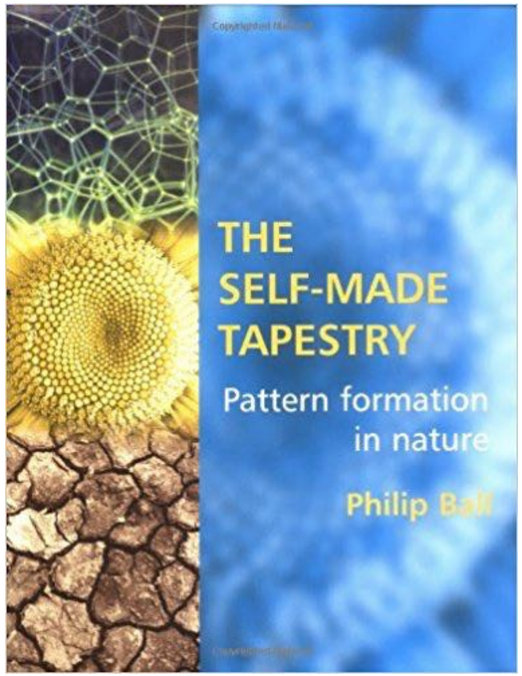
Author	Inspired by . . . applied to . . .	F
Boden and Edmonds (2009)	Electronic, digital, interactive, robotic and virtual reality art	Classification of a to generative a technology
Gamma (1995)	Software engineering	Classification of r designs into co reuse; inspired *Pattern Langu*
Havemann and Fellner (2009)	Computer graphics	Methods for build computers; all on a small num design patterns
Roy and Haridi (2004)	Computer programming	Methods for settin procedures in s

Table 1. Some process taxonomies from biology, the kinetic

Dorin, A.; McCabe, J.; McCormack, J.; Monro, G.; and Whitelaw, M.
A framework for understanding generative art

Another Framework for PCG?

Previous frameworks:



Philip Ball
The Self-made Tapestry

Chapters

1. Patterns
2. Bubbles
3. Waves
4. Bodies
5. Branches
6. Breakdowns
7. Fluids
8. Grains
9. Communities
10. Principles

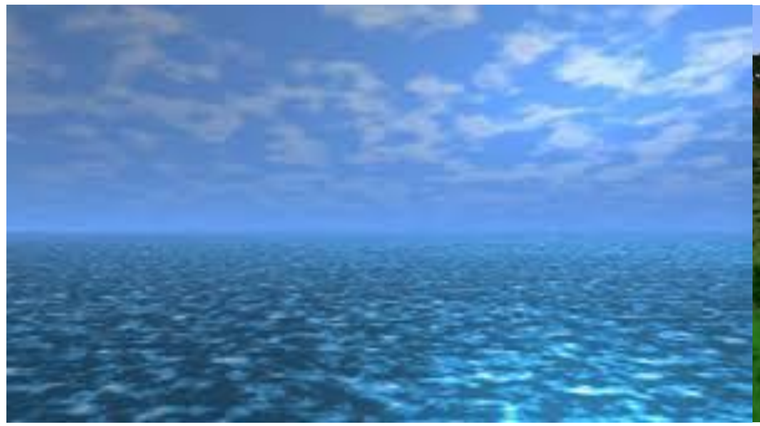
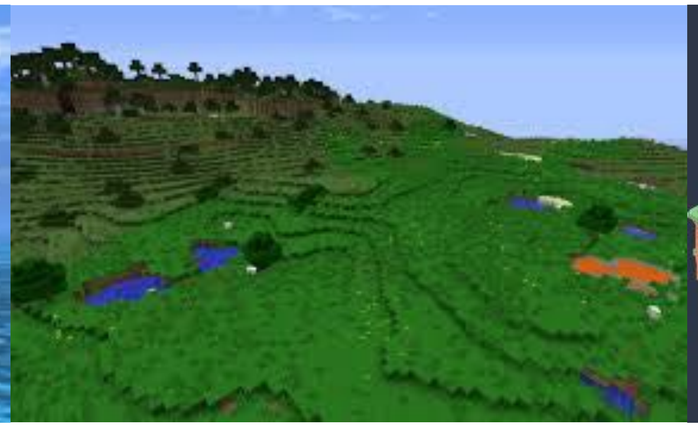
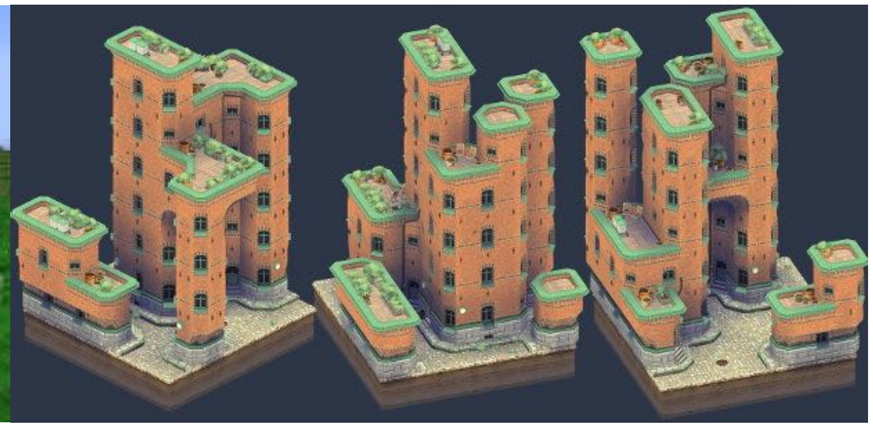
Principles

- Competing forces
- Symmetry breaking
- Non-equilibrium
- Dissipative structures
- Instabilities, thresholds and bifurcations
- Model behaviour
- Pattern selection
- Bit by bit
- Close to the edge
- The race for dominance
- Increasing complexity with increasing driving
- Correlations

The project of this framework

Other frameworks are concerned with

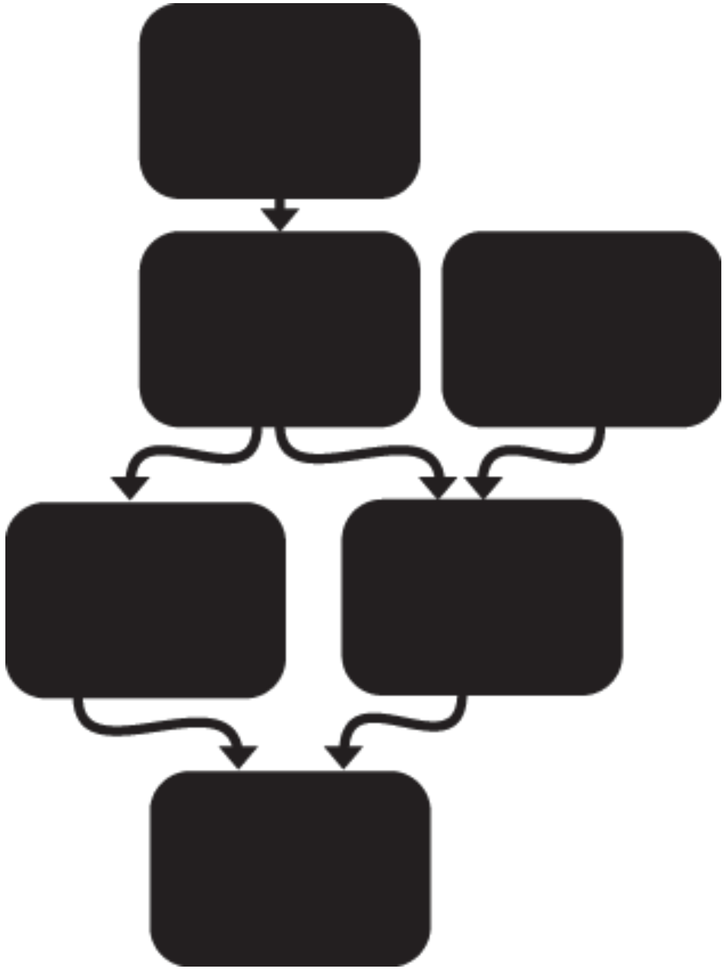
- In what ways can procedural content be used in games (Hendrikx)
- How does procedural content change or create gameplay? (Smith)
- Deep dive into a single-step generative process (“Search-based”)

The project of this framework

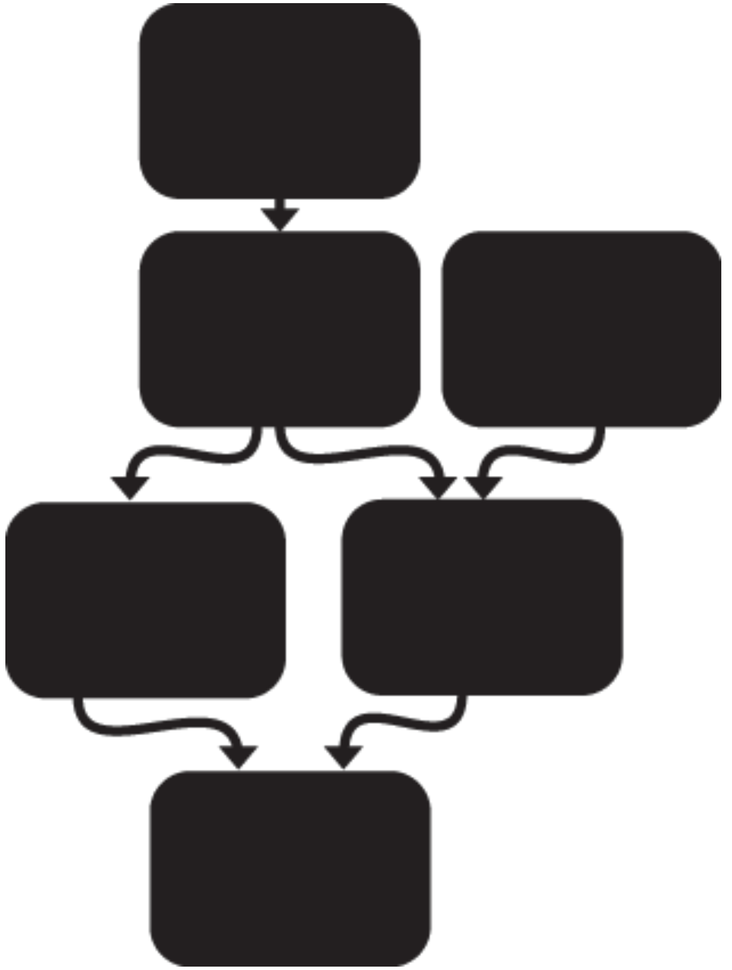
This framework is concerned with

- What are the basic “components” of generativity?
- What are the input and output conditions of those components?
- How can we chain those compents into complex pipelines?
- How can those pipelines be consumed or used? (by other pipelines, even!)



The project of this framework

- It is very rare to have a finished, shipped system which does not use multiple techniques
- even simple “toy” generators are composed of multiple techniques
- by the time a system reaches a user, it will use many generative techniques chained together



The project of this framework

- There is no pipeline so small or uncomplicated
  - …that does not contain the **possibility** of being expanded
  - simple generative pipelines can **become** components in a larger system
- All generative pipelines are interesting and worth examining, even the ones that are not interesting **on their own**

The project of this framework

- What is PCG?
  - Content that's generated (so what's "generativity"?)
- "Generative Methods" Compton, Osborne, Mateas
- "So you want to build a generator" (Feb 22, 2016)
- "Practical Procedural Generation" GDC 2017
- Generominos (summer 2016-current)

The project of this framework

Questions

- What are the basic “components” of generativity?
- What are the input and output conditions of those components?
- How can we chain those compents into complex pipelines?
- How can those pipelines be consumed or used? (by other pipelines, even!)

- How do generators **work?**
- How can we find similarity between generativity in different fields?
- How can you expand, modify, improve or reuse generators?

Generativity is

a pipeline of successive data transformations, with each transformation either generating, transforming, or pruning away information

A generator is a pipeline of **data transformations**

so we care about

- **the data**
- **the transformations**

Generativity is

a pipeline of successive data transformations, with each transformation either generating, transforming, or pruning away information

This doesn't tell us

- how generative or "transformational" it is
- its appropriateness for its purpose
- how the end products will be used (and their appropriateness for that)

It does tell us

- the space of what can be built
- potential shared tools and properties between pipelines

Generativity is

Categories of Transformation

- Content selection
- Tiles
- Grammars
- Parametric
- Geometric transformations
- Distributions
- Machine-learned statistical models

Consumers of pipelines

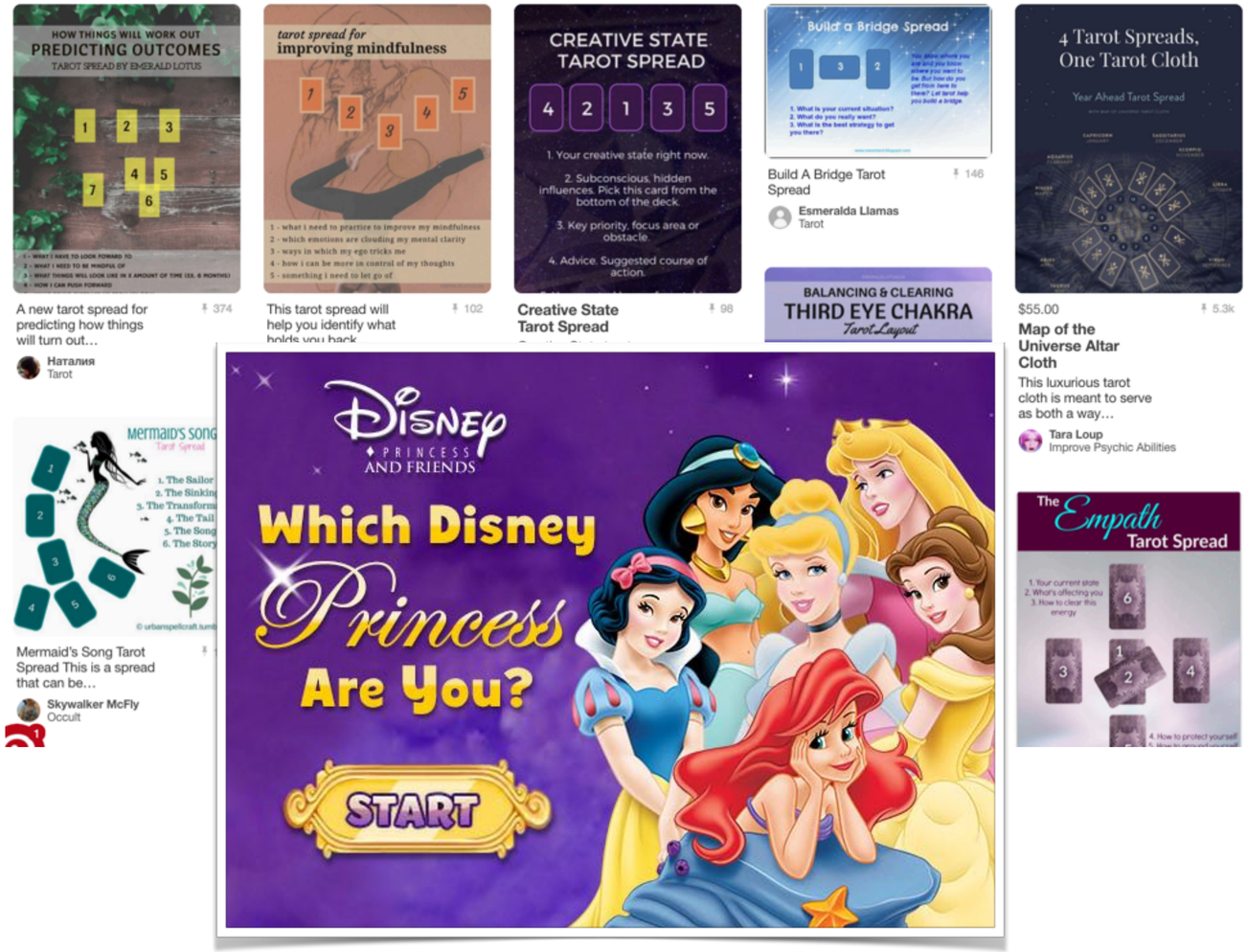
- Search-based techniques
  - Brute force
  - Optimization
  - Genetic algorithms
- Constraint solving
- Interactivity
  - User-generated content
  - Exploration

Categories of Transformations

- Content selection
- Tiles
- Grammars
- Parametric
- Geometric transformations
- Distributions
- Machine-learned statistical models

Content Selection

Requires: some collection of content
May require: a heuristic for selecting content
Returns: some content



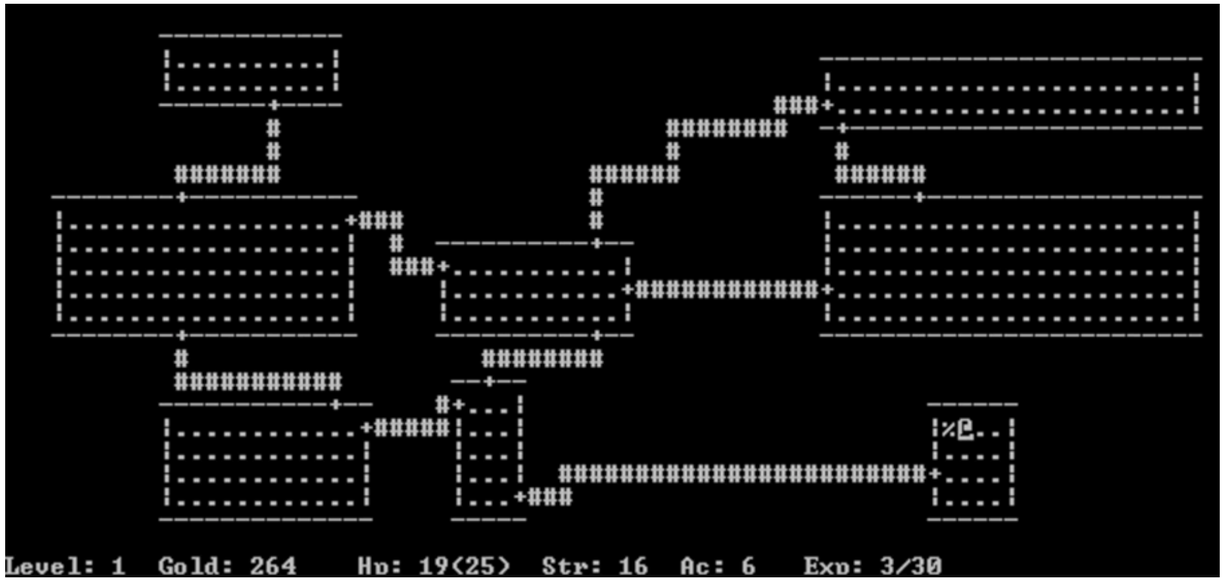
Content Selection

Requires: some collection of content
May require: a heuristic for selecting content
Returns: some content

Tiles

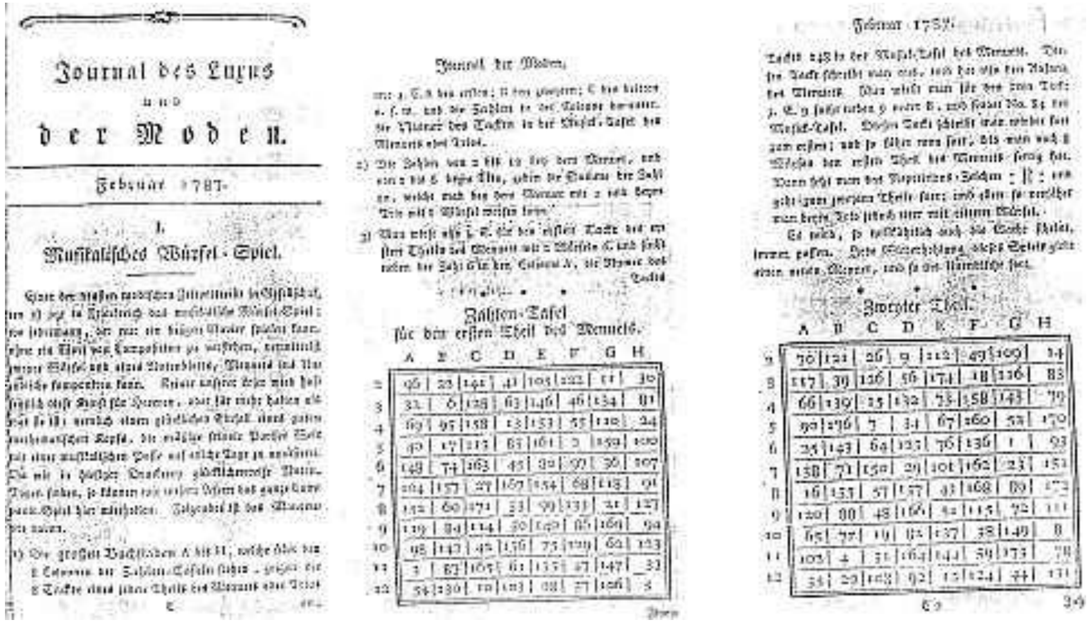
Requires: A set of possible tiles, a template of slots or sockets
May require: selection logic (”content selection”), constraints (which tiles can fill which sockets, orientation or proximity constraints)
Returns:the selected tile (or none) for each socket

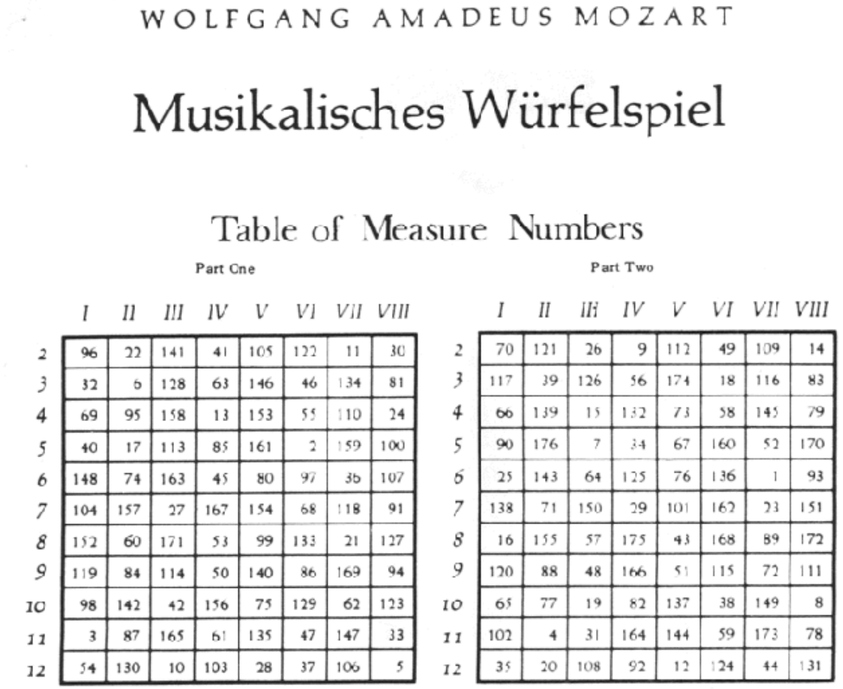
Tiles



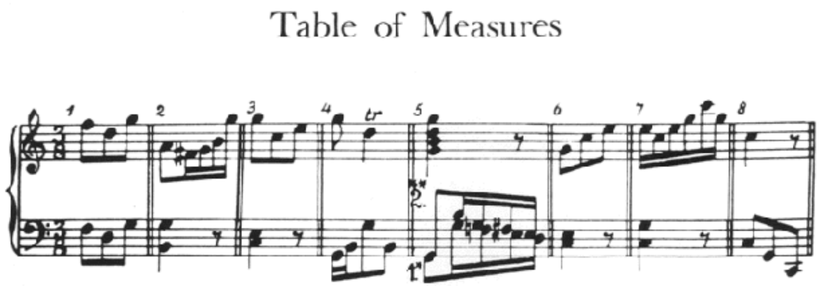
Requires: A set of possible tiles, a template of slots or sockets
May require: selection logic (”content selection”), constraints (which tiles can fill which sockets, orientation or proximity constraints)
Returns:the selected tile (or none) for each socket

Tiles

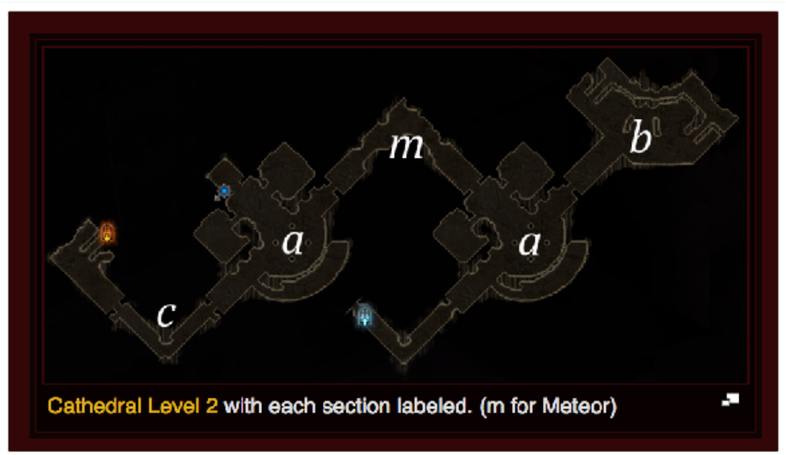




Requires: A set of possible tiles, a template of slots or sockets
May require: selection logic (”content selection”), constraints (which tiles can fill which sockets, orientation or proximity constraints)
Returns:the selected tile (or none) for each socket



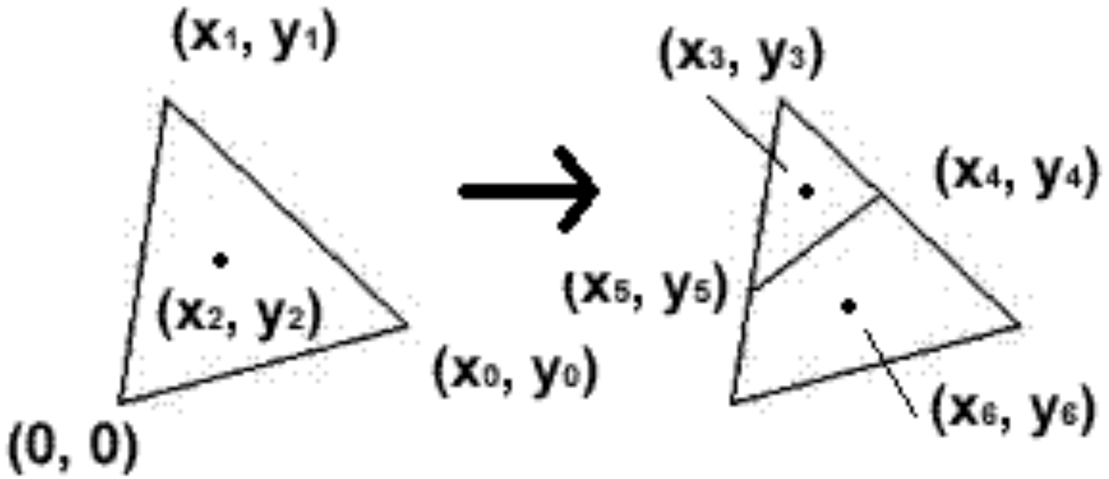
Tiles

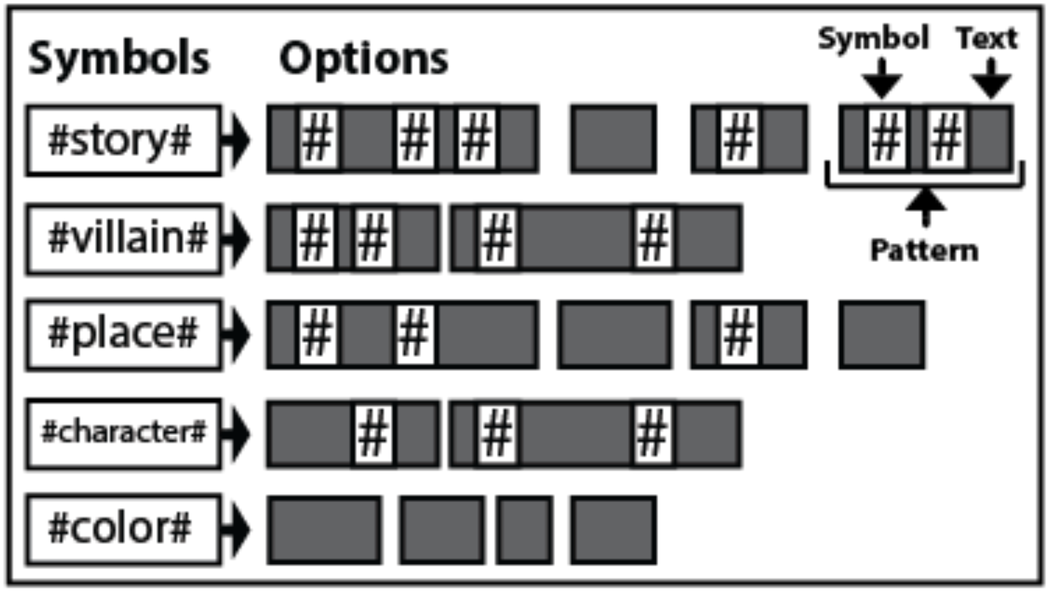


Requires: A set of possible tiles, a template of slots or sockets
May require: selection logic (”content selection”), constraints (which tiles can fill which sockets, orientation or proximity constraints)
Returns:the selected tile (or none) for each socket

https://diablo.gamepedia.com/Map_Generation

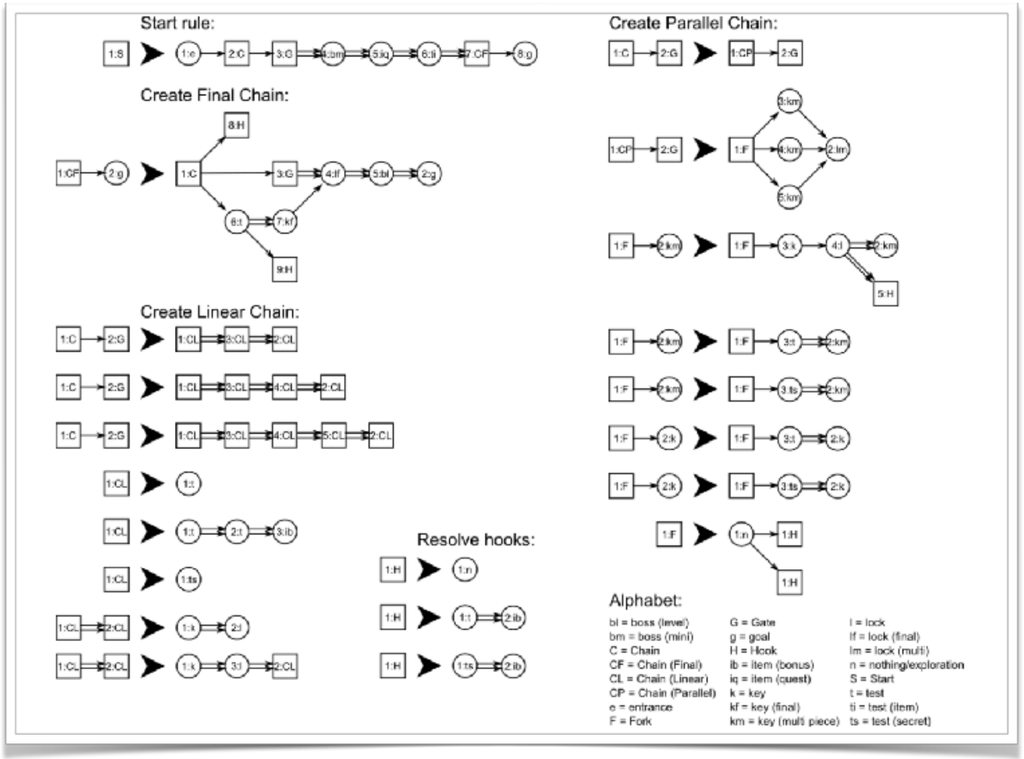
Grammars





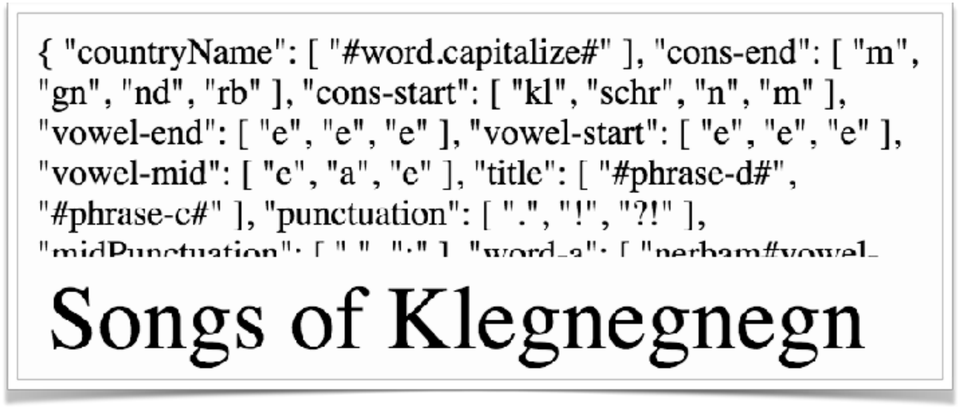
Requires: A set of replacement rules (a grammar)
May require: non-context free rules that govern preferences or constraints (if so, some form of a constraint solver is required)
Returns: A tree of recursive choices.

Grammars



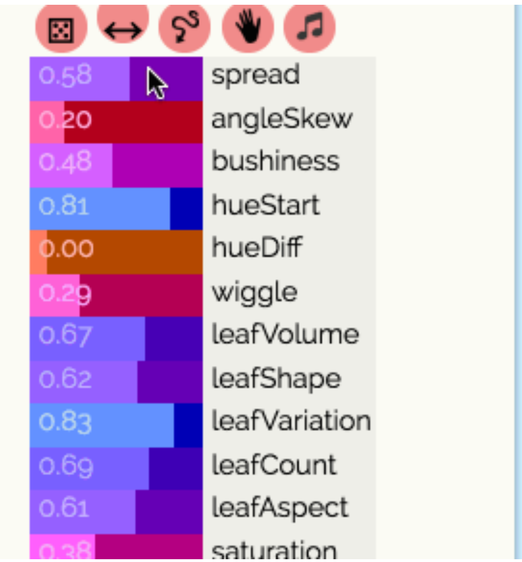
Joris Dormans
Adventures in Level Design: Generating Missions and Spaces for Action Adventure Games

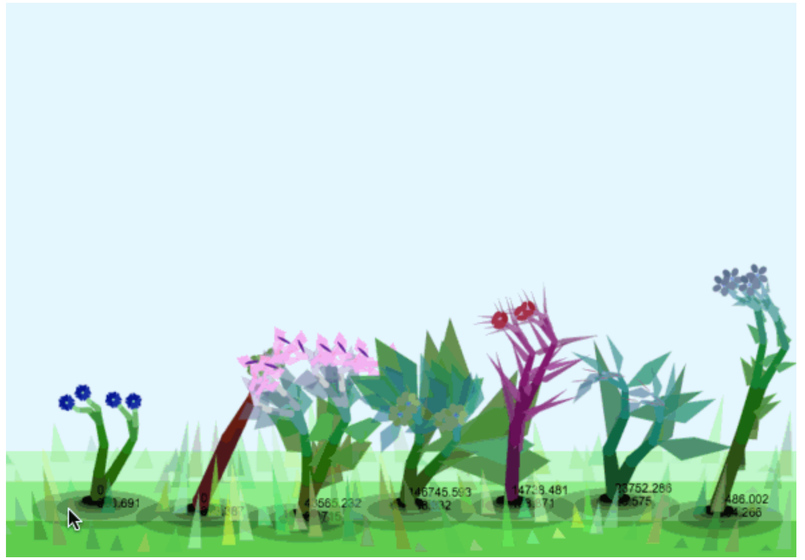
Kate Compton
Kambamanx: songs in an unknown language



Requires: A set of replacement rules (a grammar)
May require: non-context free rules that govern preferences or constraints (if so, some form of a constraint solver is required)
Returns: A tree of recursive choices.

Parametric

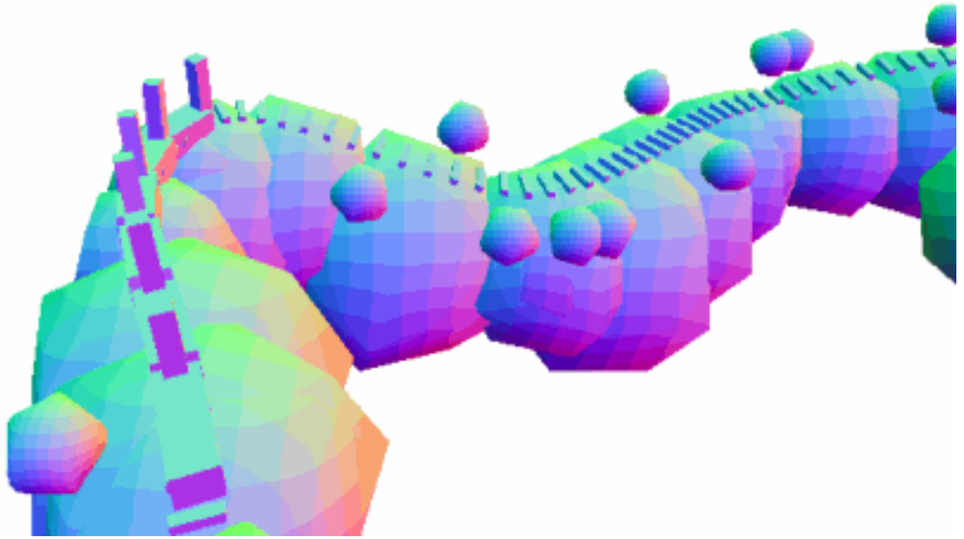




Requires: An array of floats, usually normalized
Returns: Widely variable: L-systems create graphs in space, but other parametric generators may create curves (superellipses), 3D geometry, or waveforms or vectorfields.

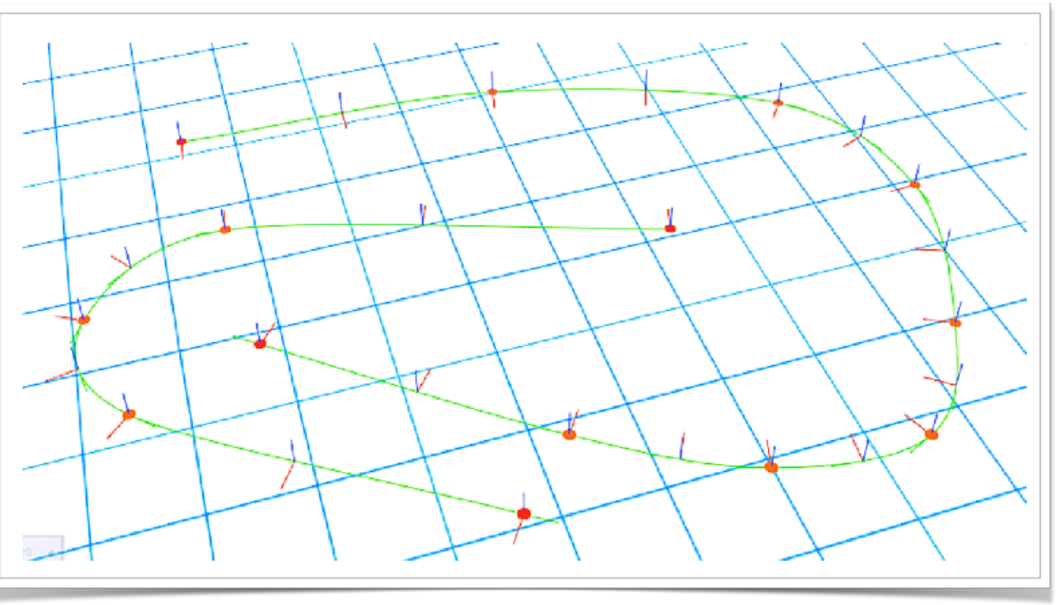
Geometric transformations

IOBrush
Tangible Media group

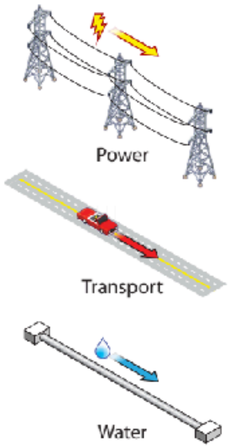


Tiltbrush
Google

Geometric transformations



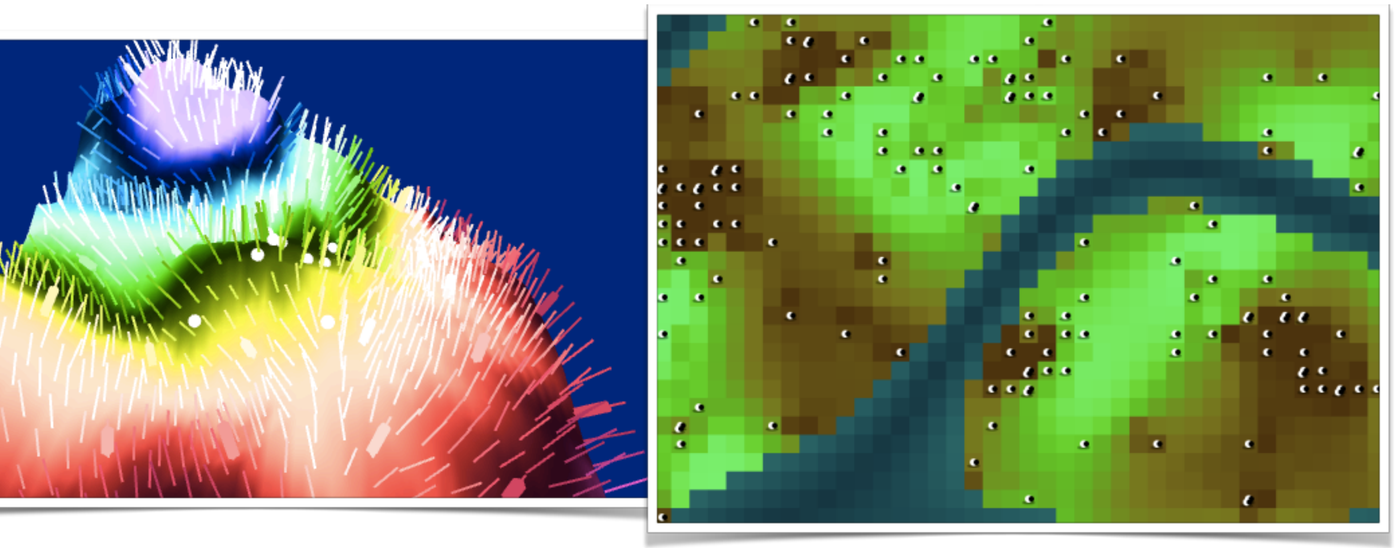
- **Points** connected by **Segments** make up **Paths** make up **Path Sets**
- Fully 3D, spline-based, rich set of operations
- Typically player created
- Curvy roads!
- But also: power lines, water pipes, flight paths



Glassbox: A new simulation architecture
Andrew Willmott, Ocean Quigley, Dan Moskowitz

Distributions

Requires: Content to be distributed, rules to govern selection and distribution,
Returns: Location (and possibly other annotations) of the new positions for the content



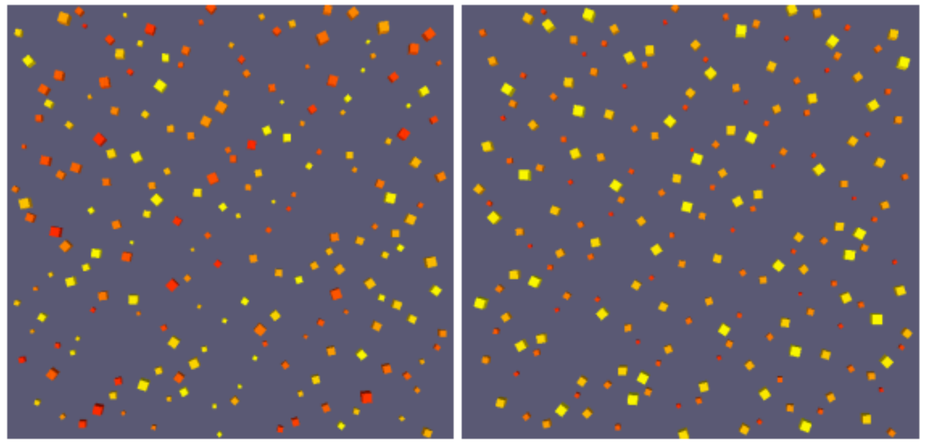
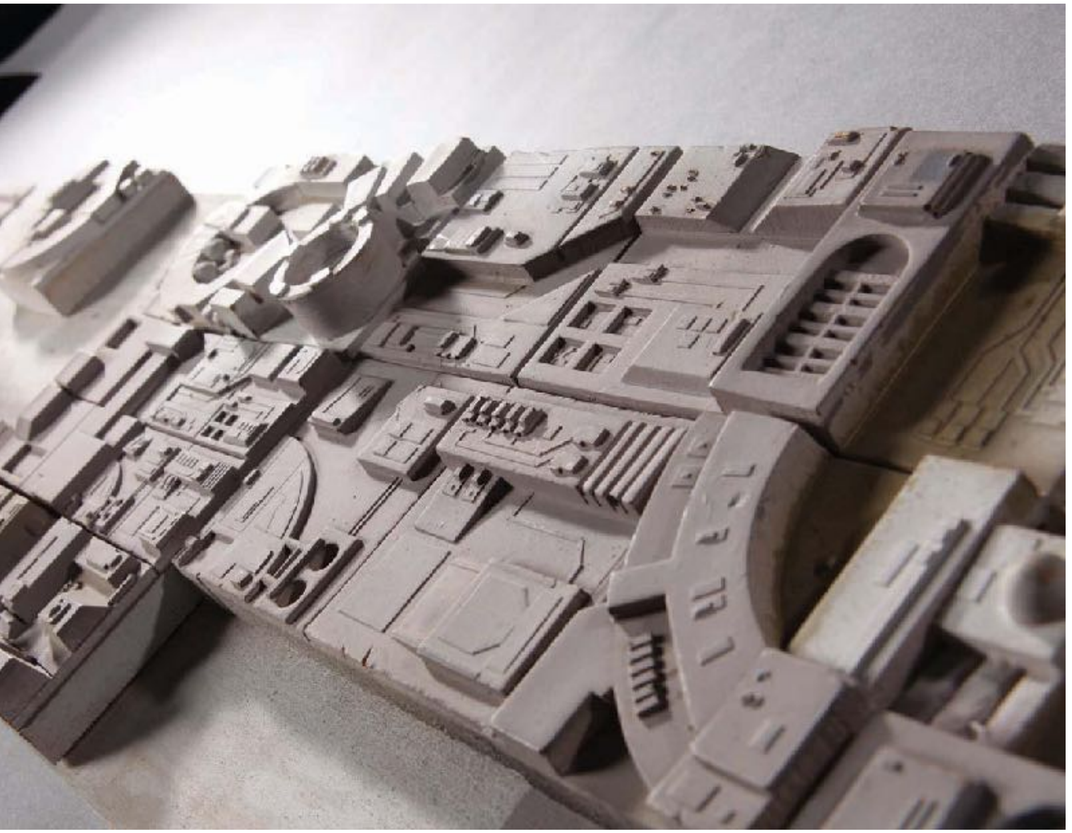


Figure 1: *Left, random variation, right, indexed variation*

https://www.cs.cmu.edu/~ajw/s2007/0312-ObjectDistribution.pdf

Distributions

Requires: Content to be distributed, rules to govern selection and distribution,
Returns: Location (and possibly other annotations) of the new positions for the content



- **greebling:** a term originating in Star Wars production history, meaning that the uneven distribution of object over a surface makes the surface look ”real”
- **footing:** the tendency for the intersection between two naturally-occurring objects to be noticeably different than non-intersection spaces, such as grass growing higher against a fence post, moss accumulating the the cracks of boulders, trash getting caught against a park bench, or sand pooling where a river bank meets the river.
- **barnacling:** a term-of-art on Spore used to describe the aesthetic value that a monumental object is often best framed by smaller simpler versions of itself, or by some contrasting object.

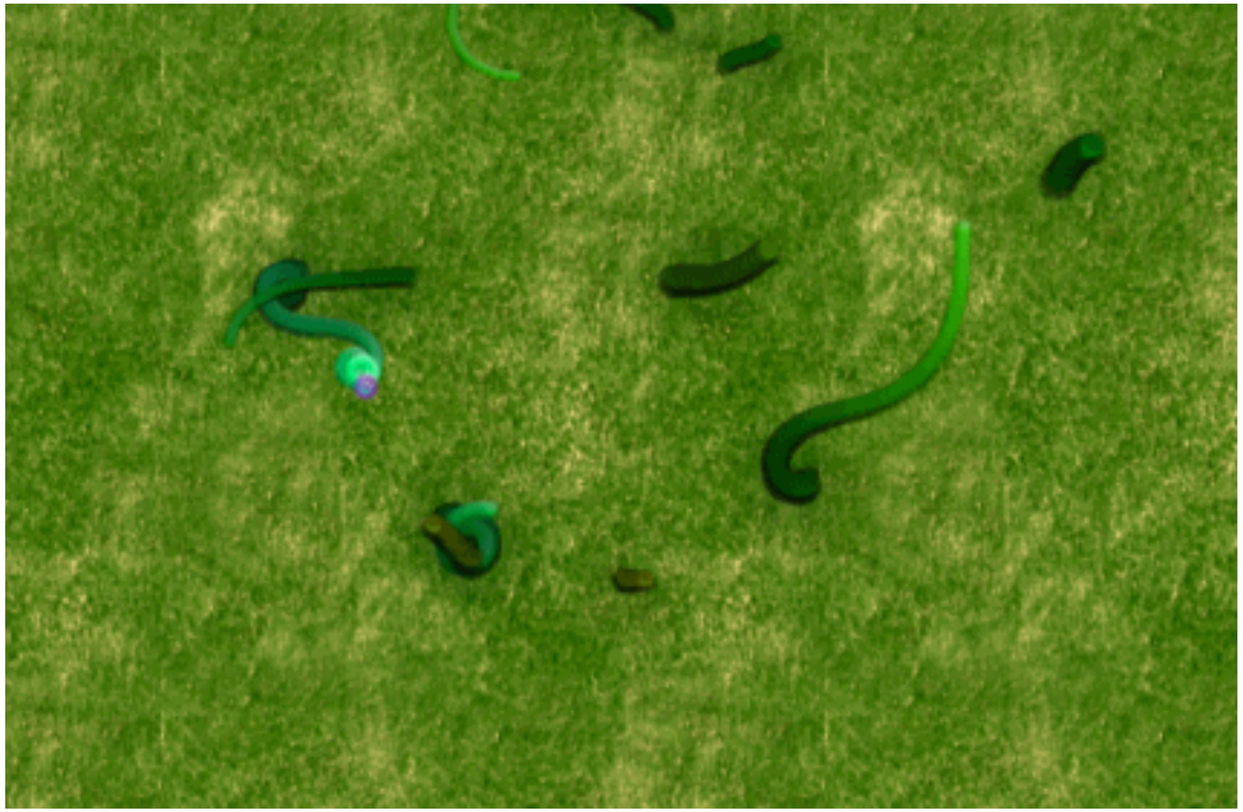
Distributions

Requires: Content to be distributed, rules to govern selection and distribution,
Returns: Location (and possibly other annotations) of the new positions for the content

- **greebling:** a term originating in Star Wars production history, meaning that the uneven distribution of object over a surface makes the surface look ”real”
- **footing:** the tendency for the intersection between two naturally-occurring objects to be noticeably different than non-intersection spaces, such as grass growing higher against a fence post, moss accumulating the the cracks of boulders, trash getting caught against a park bench, or sand pooling where a river bank meets the river.
- **barnacling:** a term-of-art on Spore used to describe the aesthetic value that a monumental object is often best framed by smaller simpler versions of itself, or by some contrasting object.

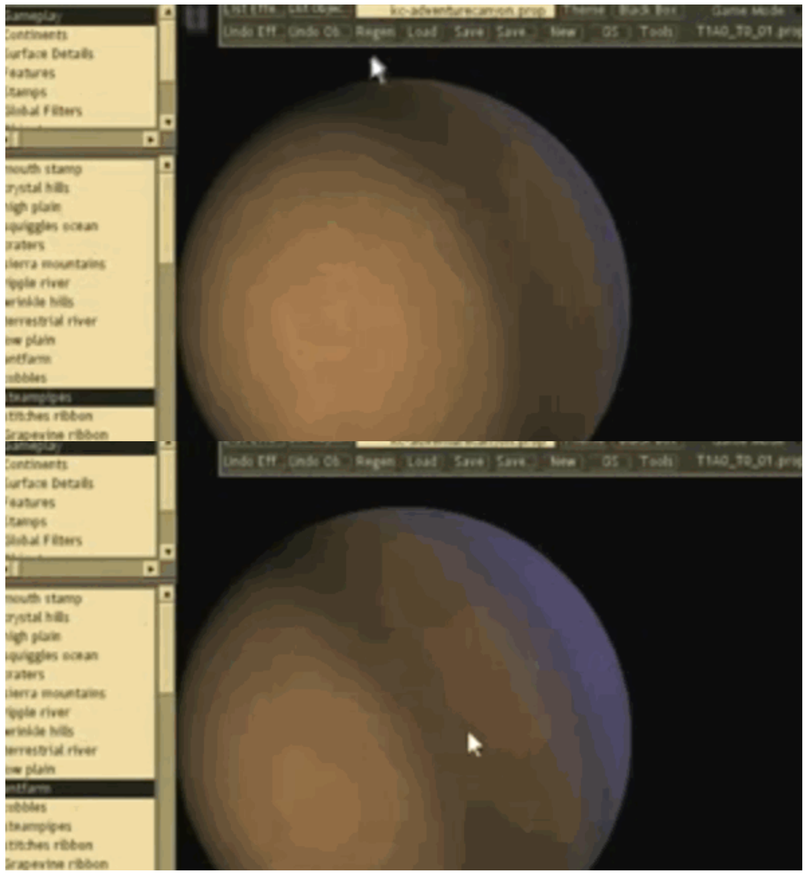
Particles, cellular automata, and graph-based simulations

Requires: A directed graph with update rules, or forces and distribution(particles), or cell update rules (cellular automata)
Returns: Particle trails, or the new states of the graph or grid-cells



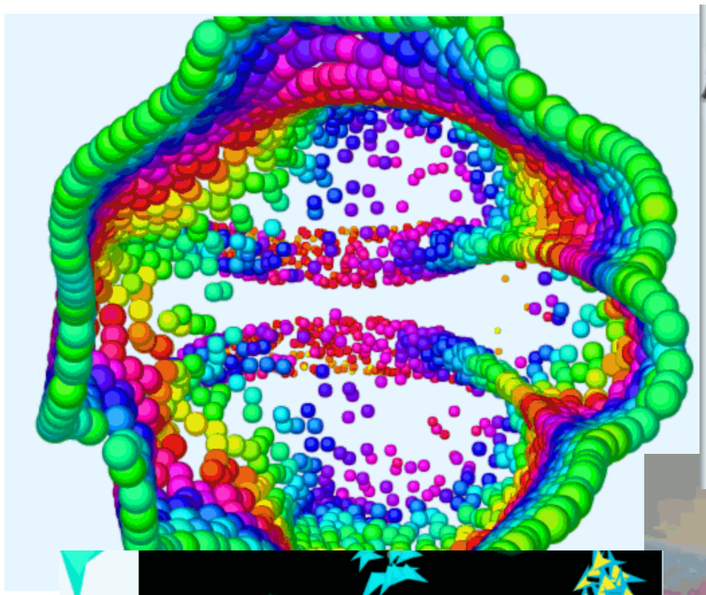
Particles, cellular automata, and graph-based simulations

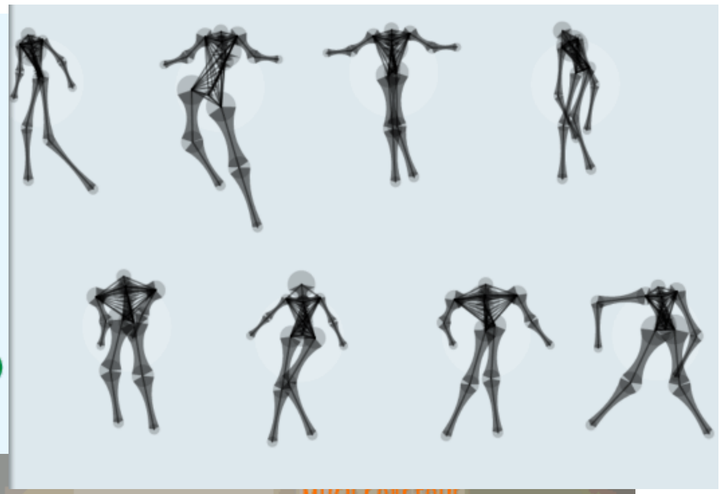
Requires: A directed graph with update rules, or forces and distribution(particles), or cell update rules (cellular automata)
Returns: Particle trails, or the new states of the graph or grid-cells

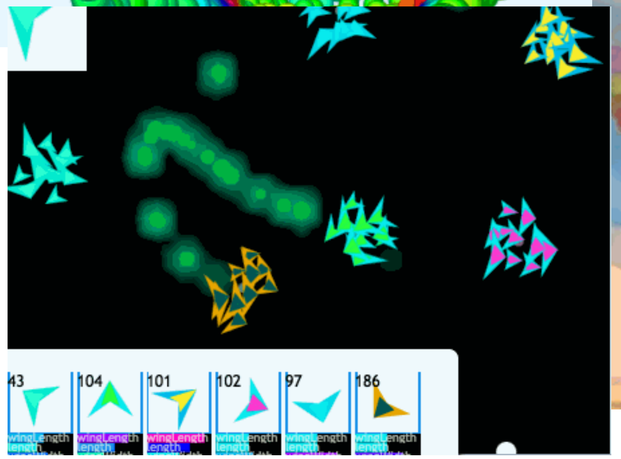


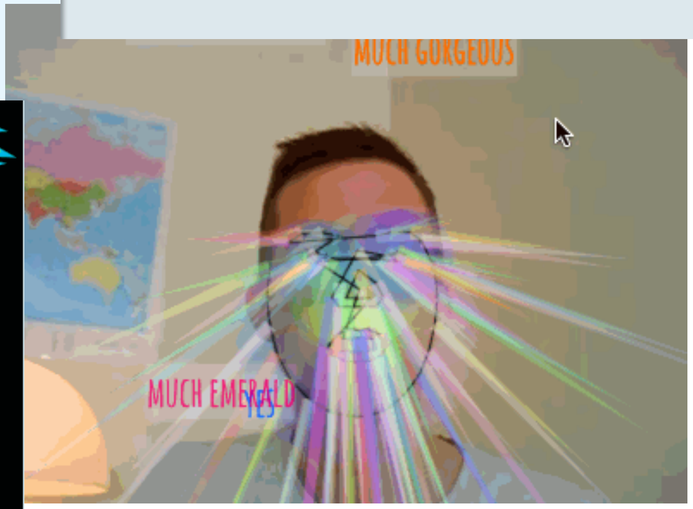
Particles, cellular automata, and graph-based simulations

Requires: A directed graph with update rules, or forces and distribution(particles), or cell update rules (cellular automata)
Returns: Particle trails, or the new states of the graph or grid-cells

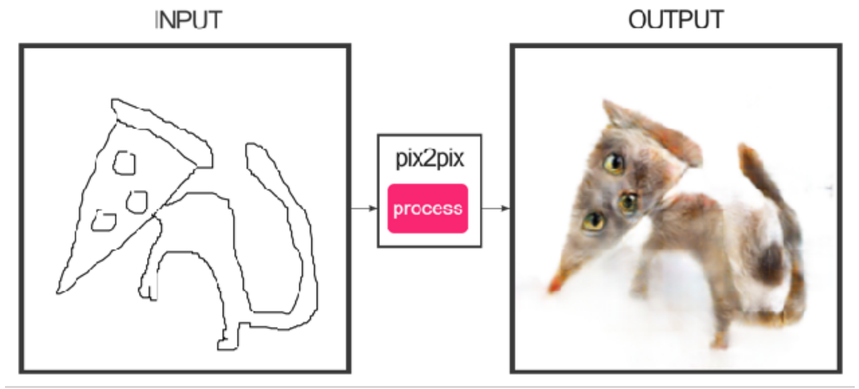


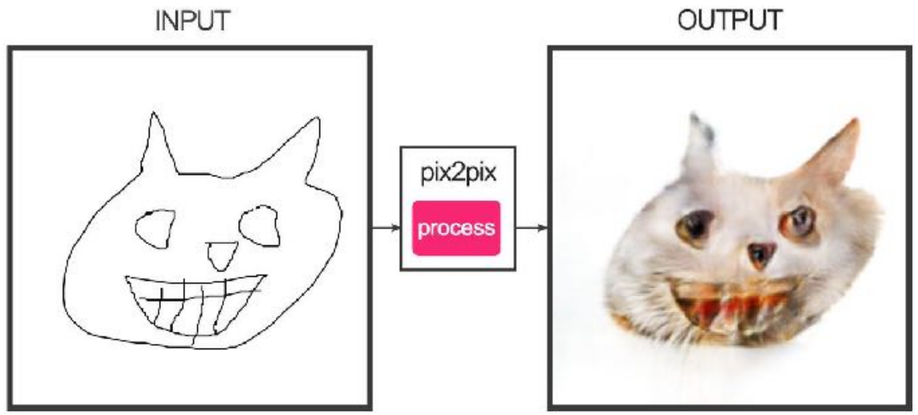






Machine-learned statistical models



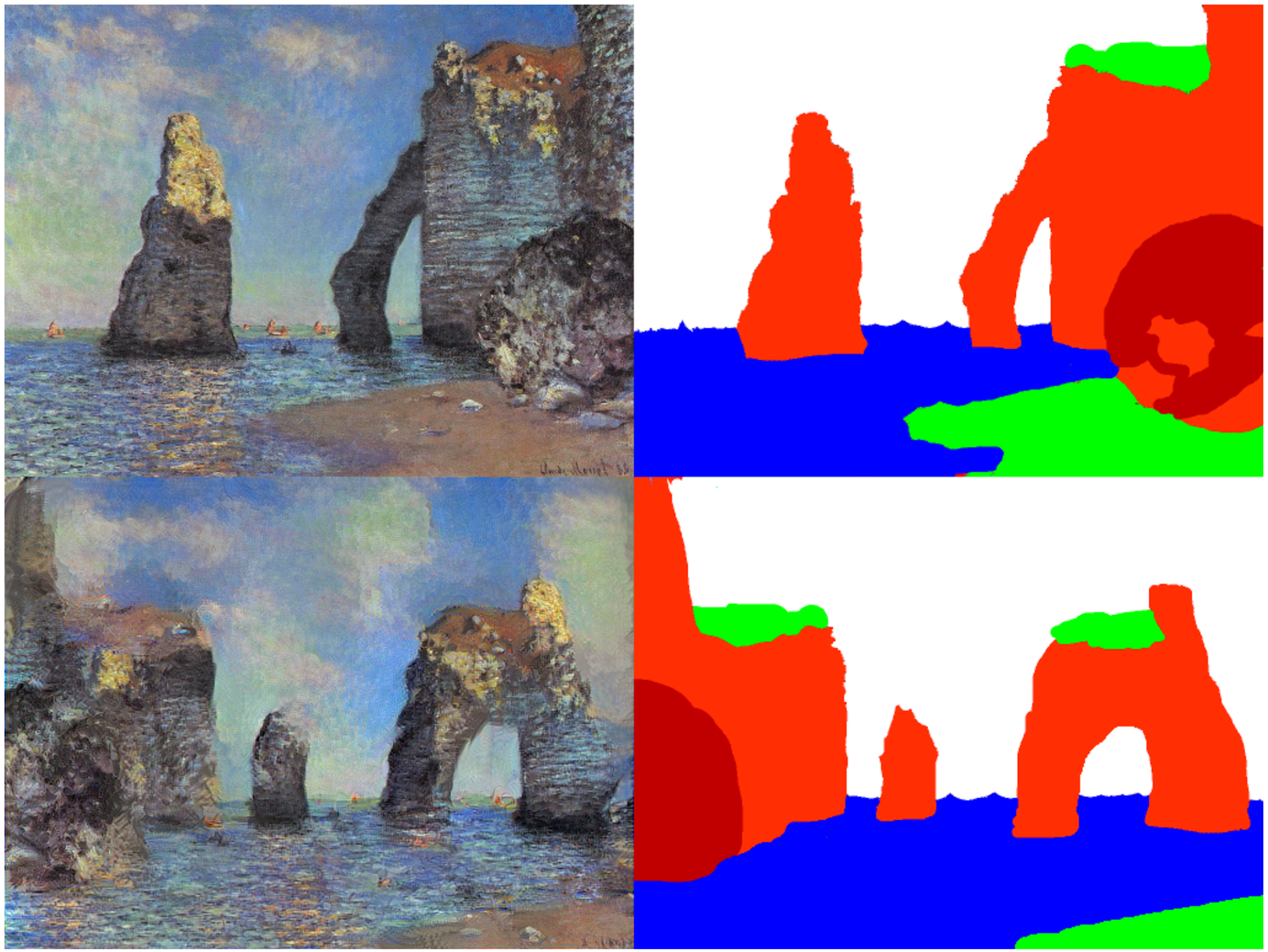


https://affinelayer.com/pixsrv/index.html

Model construction:
Requires: Lots of content
Returns: A model representing a set of statistical patterns in that content

Transformation via model:
Requires: A trained model. Deep-learning activation-optimization may also require a starting piece of content to optimize, just as Markov models may require a starting sequence.
Returns: For a Markov model: a statistically-viable walk through a Markov chain (ie, a text string or other string of symbols). For a neural network: an activation-optimized version of the source content

Machine-learned statistical models

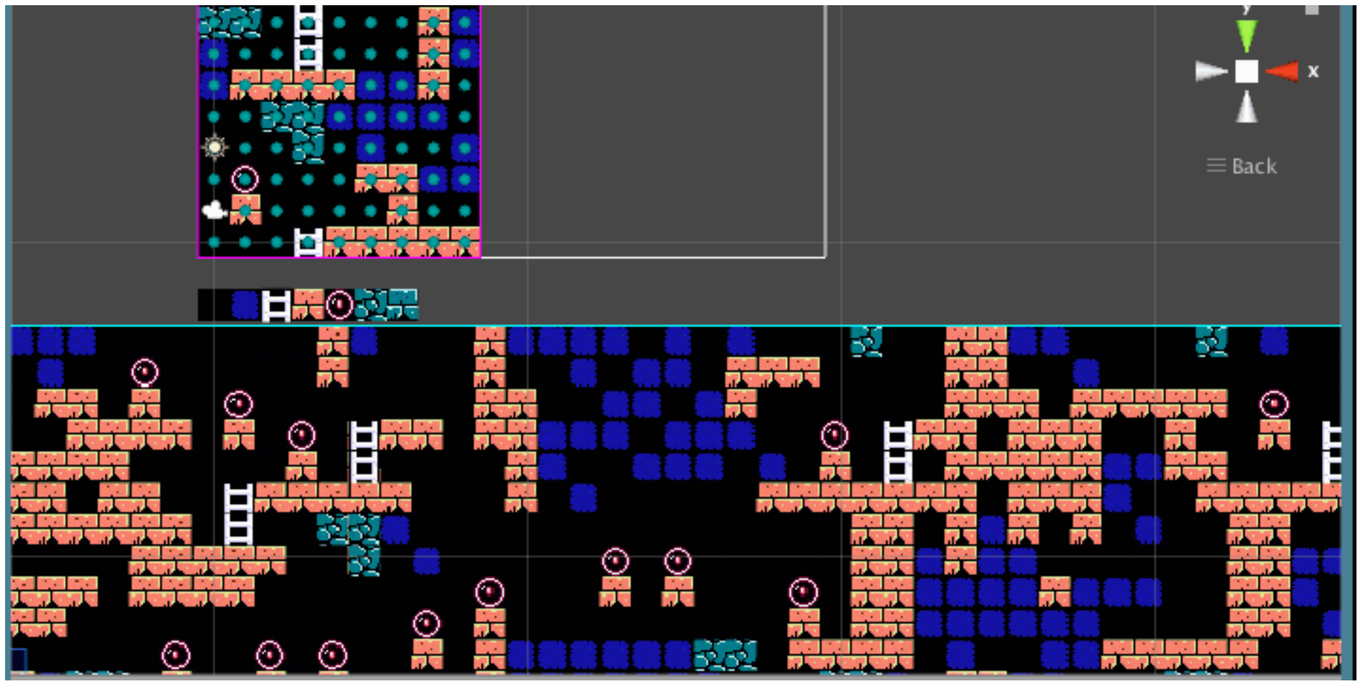


Alex Champandard, **Neural Doodle**

Model construction:
Requires: Lots of content
Returns: A model representing a set of statistical patterns in that content

Transformation via model:
Requires: A trained model. Deep-learning activation-optimization may also require a starting piece of content to optimize, just as Markov models may require a starting sequence.
Returns: For a Markov model: a statistically-viable walk through a Markov chain (ie, a text string or other string of symbols). For a neural network: an activation-optimized version of the source content

Machine-learned statistical models



Wave Function Collapse

Model construction:
Requires: Lots of content
Returns: A model representing a set of statistical patterns in that content

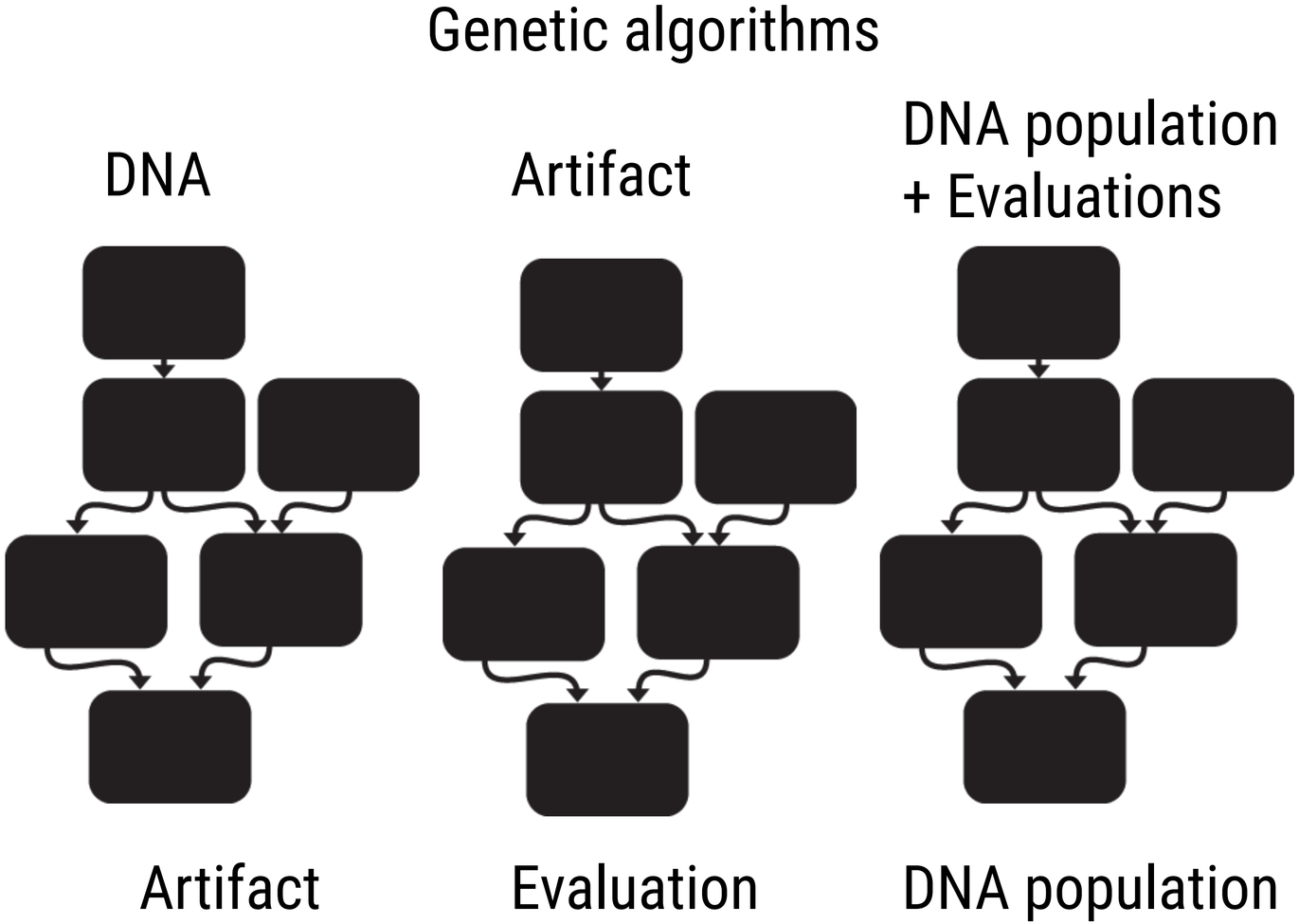
Transformation via model:
Requires: A trained model. Deep-learning activation-optimization may also require a starting piece of content to optimize, just as Markov models may require a starting sequence.
Returns: For a Markov model: a statistically-viable walk through a Markov chain (ie, a text string or other string of symbols). For a neural network: an activation-optimized version of the source content

Consumers of pipelines

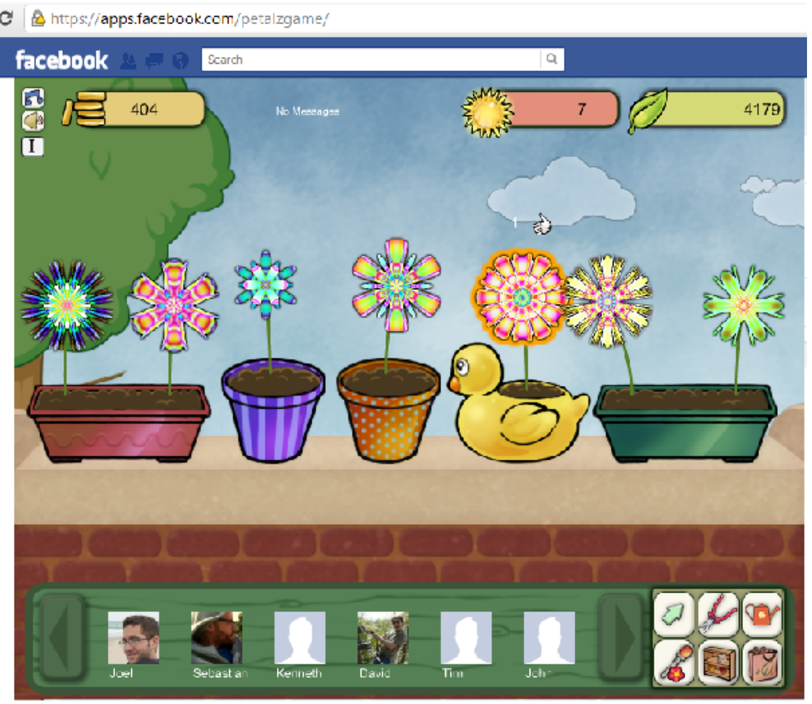
- Search-based techniques
  - Brute force
  - Optimization
  - Genetic algorithms
- Constraint solving
- Interactivity
  - User-generated content
  - Exploration

Consumers of pipelines

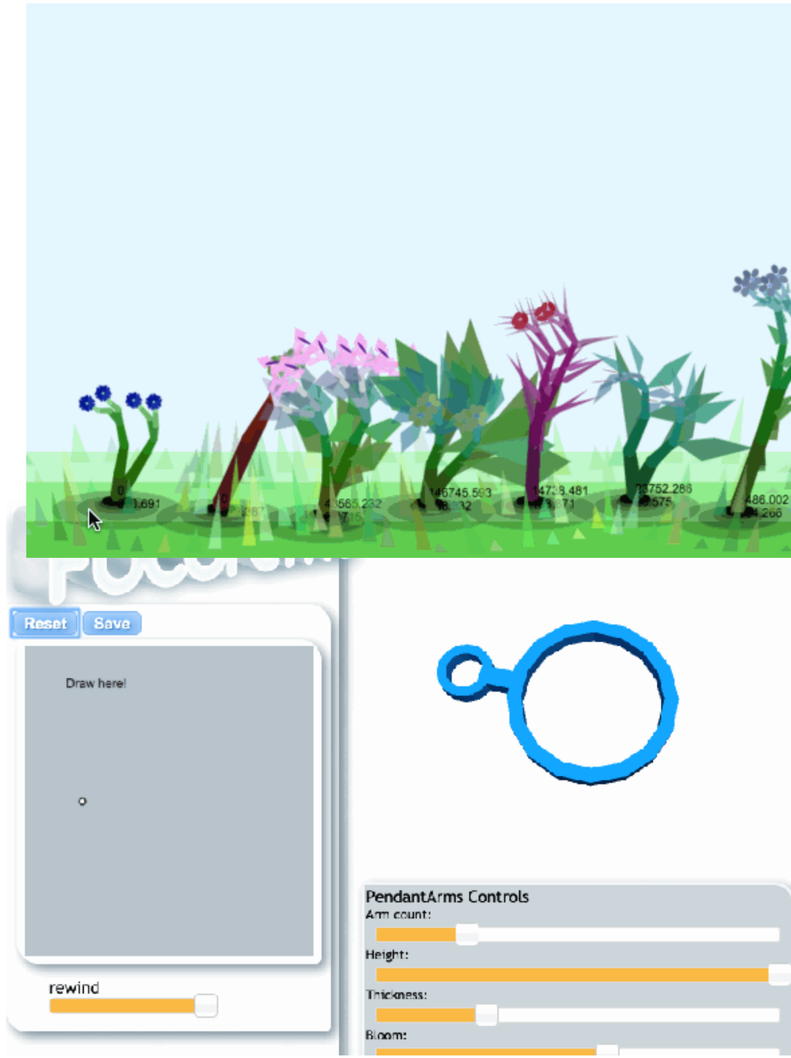
- Search-based techniques
  - Brute force
  - Optimization
  - Genetic algorithms



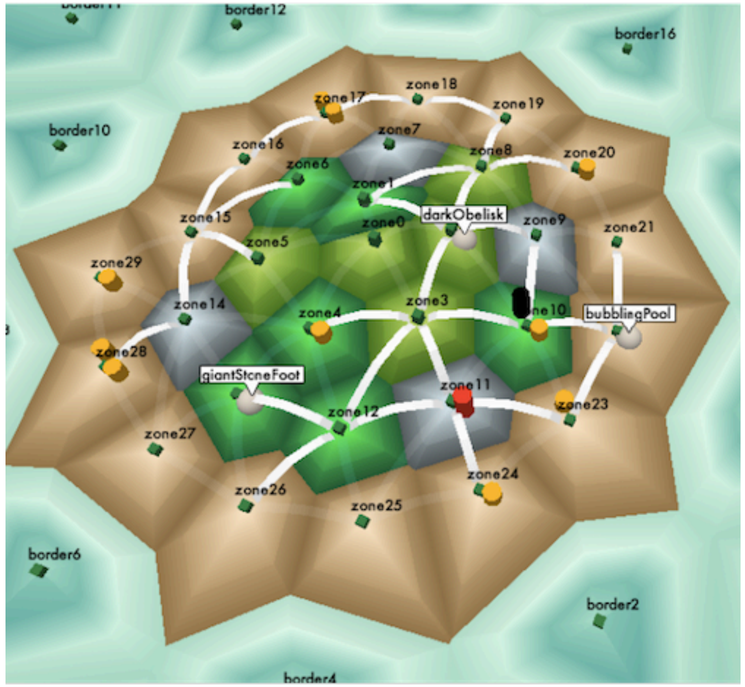
Consumers of pipelines



- Interactivity
  - User-generated content
  - Exploration



Consumers of pipelines



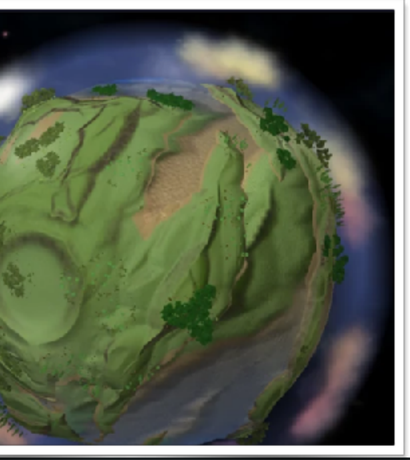
- Constraint solving

Where do your constraints come from?

Spore and No Mans Sky

Random Seed Value

Use as terrain under generated creatures



Spore and No Mans Sky

Random Seed Value

Select Brush script

Image to stamp

Particle behavior

Extrude along curve or stamp pattern into height map

Heightmap -> geometry

Create Voxel layer

Create Perlin noise map over sphere (global lumpiness)

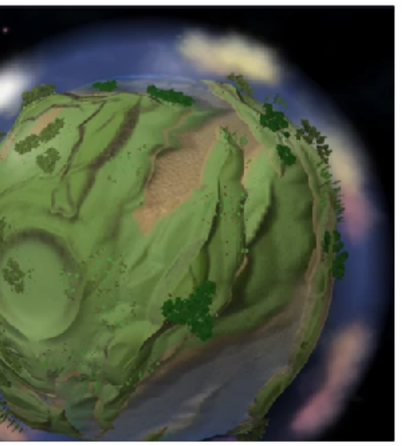
Use 3D Perlin noise to add or remove voxels (tunnels and “Perlin worms”)

Turbulence!

Offset voxel data

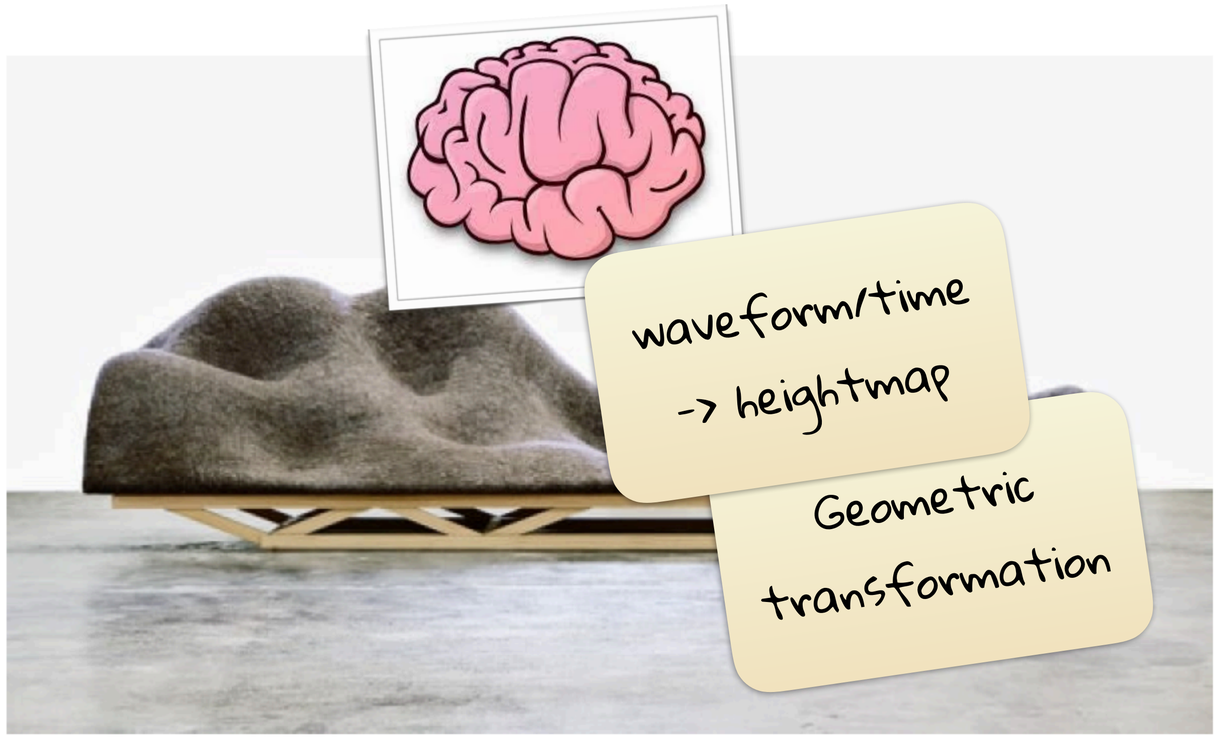
Voxel -> Poly Mesh

Use as terrain under generated creatures



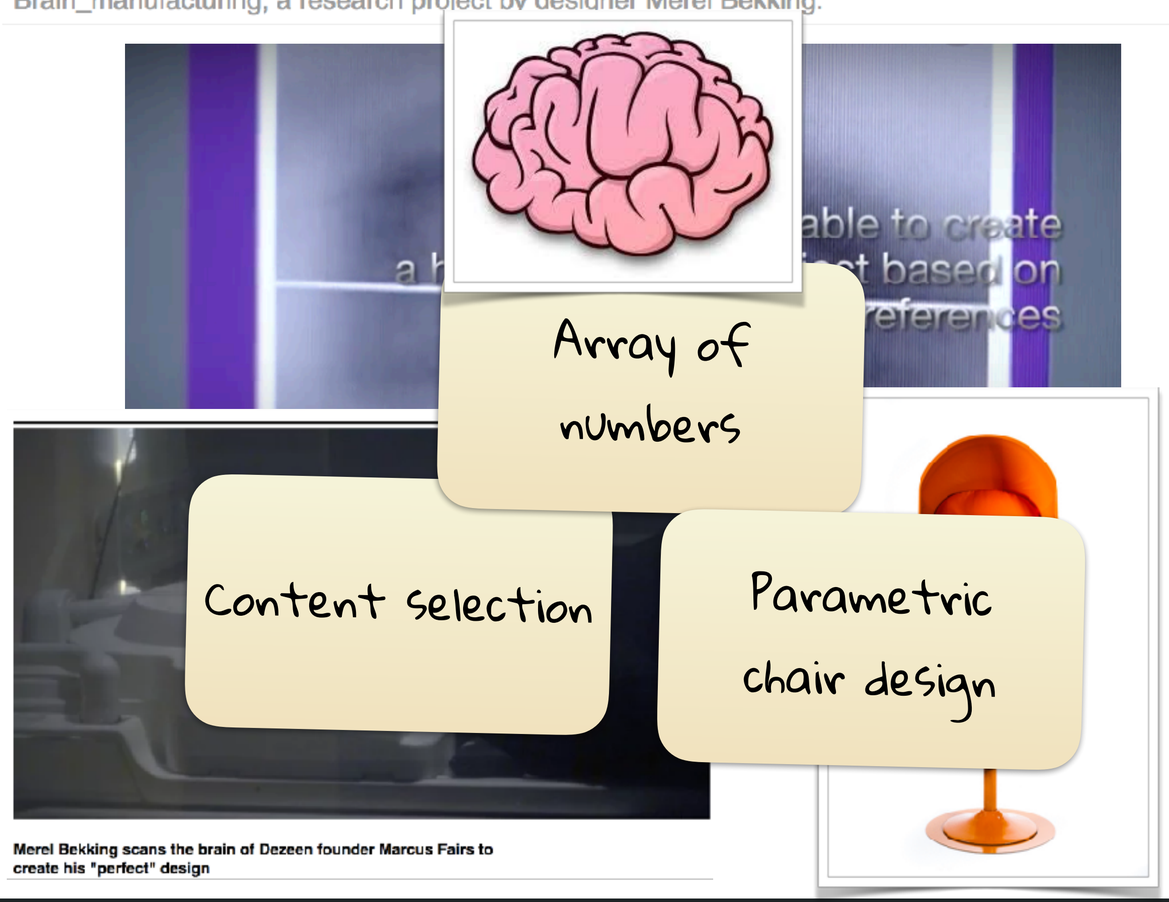
Revisiting the chairs...

Italian designer Lucas Maassen wired his brain to an EEG-measuring device and relaxed as he contemplated the word “comfort.” Then he and Belgian Unfold principal Dries Verbruggen translated that brainwave file into instructions for Computer Numerical Control milling machines that carved a block of foam into a biomorphic 3-D couch.



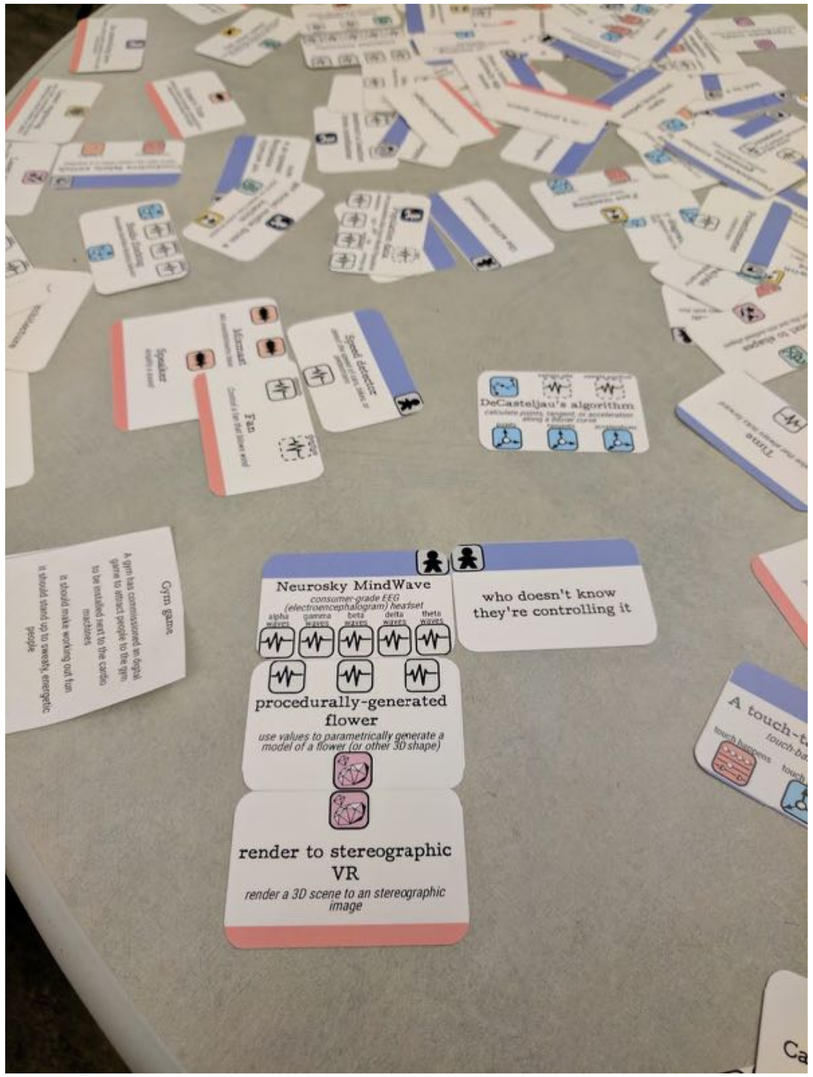
Your Brain_manufacturing

What if you could create a design based on your inner most feelings by measuring your brain activity? Would you like it? How would you relate to it? These where the central questions for Your Brain_manufacturing, a research project by designer Merel Bekking.



Merel Bekking scans the brain of Dezeen founder Marcus Fairs to create his "perfect" design

Future work….


Neurosky MindWave
consumer-grade EEG (electroencephalogram) headset
alpha waves
gamma waves
beta waves
delta waves
theta waves
who doesn't know they're controlling it
procedurally-generated flower
use values to parametrically generate a model of a flower (or other 3D shape)
render to stereographic VR
render a 3D scene to an stereographic image
DeCasteljau's algorithm
Gym game
A touch-b

Questions?

"Nothing by itself is really interesting enough, the key is having this toolkit of loads different techniques that you can use in different areas until you get something that looks like the goal you’re seeking."

Innes McKendrick, GDC 17