Generative design

Professor: Danil Nagy
Monday 11:00am-1:00pm
202 Fayerweather

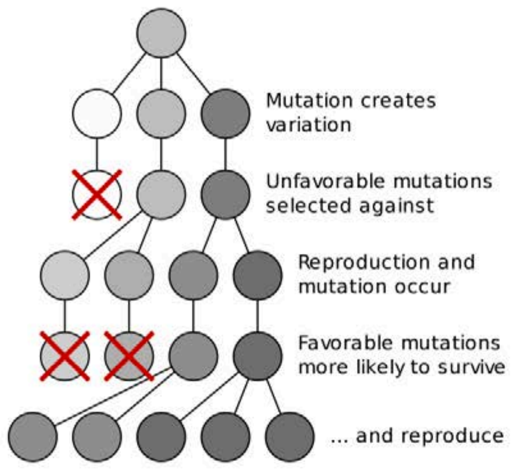
Look deep into nature, and then you will understand everything better.

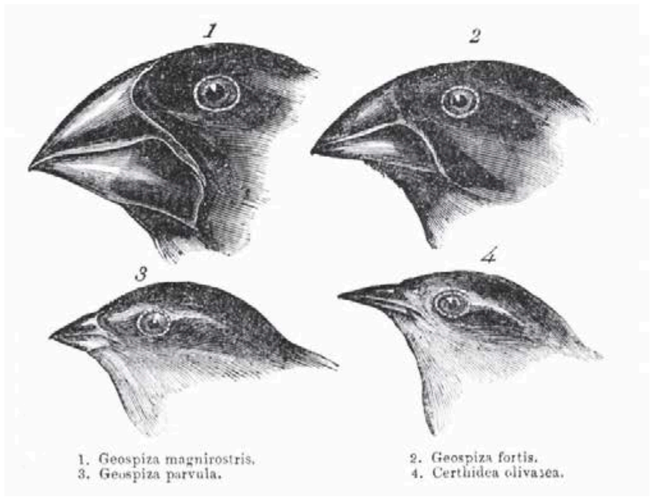
- *Albert Einstein*

I. Intelligent machines

In the past decade, our interaction with the world has been deeply affected by artificial intelligence. Many industries including finance, science, and manufacturing have been revolutionized by developments in Machine Learning, optimization, and other artificial intelligence technologies, which have allowed them to leverage the power of computing to solve complex problems in new, innovative ways.

Meanwhile, architectural design practice has been barely impacted by these developments. Although almost all designers use computers in their practice, the tools they rely on have not leveraged these emerging technologies. As a result, the design profession has not substantially evolved since computers were first introduced to the design world nearly four decades ago.

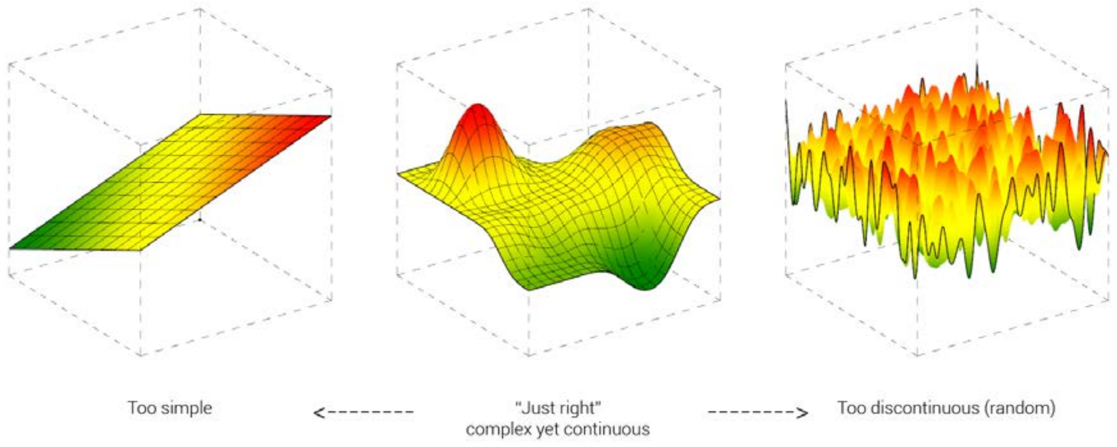

Mutation creates variation
Unfavorable mutations selected against
Reproduction and mutation occur
Favorable mutations more likely to survive
... and reproduce



II. Learning from nature

Perhaps the greatest opportunity for artificial intelligence in design practice today is its ability to leverage another, much older form of intelligence - natural intelligence. Designers have always been inspired by the forms of nature, and their abilities to solve difficult problems in novel and beautiful ways. However, up to this point our inspiration from nature has been limited to 'bio-mimicry', or the reproduction of nature's physical forms in new designs. Can we go a step further and actually design like nature?

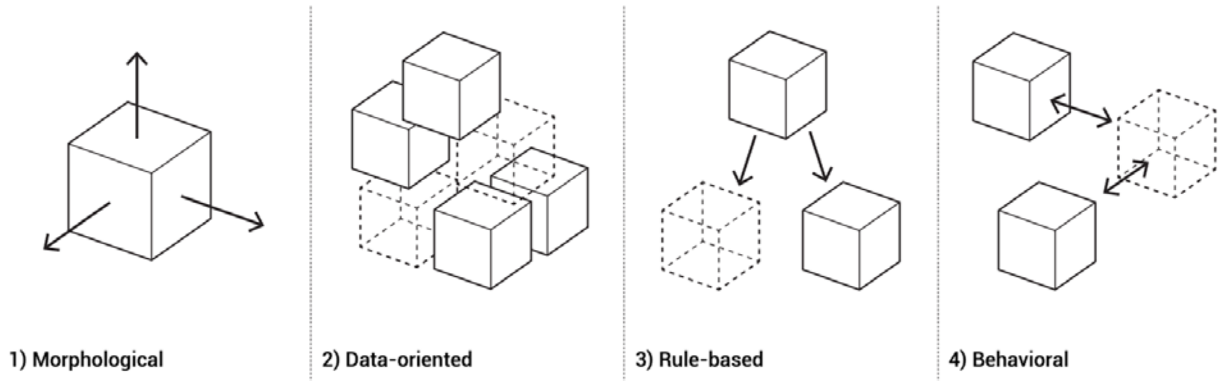
To do this we have to first understand how nature designs. The basic element of nature's design is the species, a kind of *model* which encodes all of the unique properties and abilities of its individual members. The basic tool of nature's design is evolution, which is an *iterative process* by which species are able to adapt and improve based on interaction with other species and their environment.



III. <u>Automating design</u>

This class will explore how we can use new technology to leverage nature's design methods to create new design workflows:

1. Instead of designing objects, we will learn to *design systems* which encode the full range of possibilities of a particular design concept
2. We will then learn methods for *measuring and quantifying the performance* of these systems so that each design can be evaluated automatically by the computer
3. Finally, we will create *automated evolutionary processes* which will allow the computer to search through our design systems to find novel and high-performing designs.

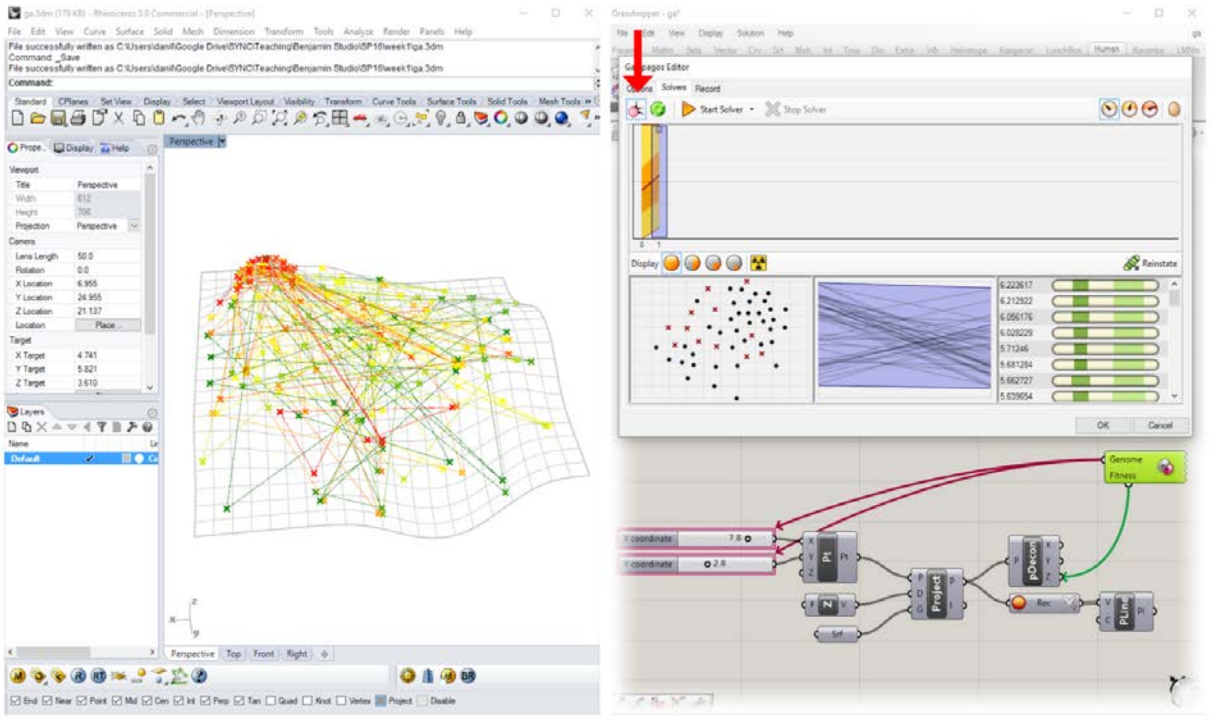


IV. <u>From tools to partners</u>

These new workflows will allow us to explore a much wider space of design than possible through traditional intuitive methods, and lead not only to the discovery of novel and unexpected solutions, but to a deeper understanding of the design problem itself.

To take advantage of these possibilities, we will have to learn how to work with computers in new ways. Instead of thinking of computers as *tools* that accomplish specific tasks in predictable ways, we will think of computers and algorithms as *partners* in our design process.

We will discover that orchestrating such a human/machine design collaboration is actually quite difficult. Artificial and human intelligences work in very different ways, and in order to work together we will have to be much more explicit in how we describe our design concepts and intentions to the computer. However, if we succeed, this interaction will not only create new opportunities for design, but will make us more thoughtful, more responsible, and better human designers.



```
File successfully written as C:\Users\danil\Google Drive\SYNC\Teaching\Benjamin Studio\SP16\week1\ga.3dm
Command: _Save
File successfully written as C:\Users\danil\Google Drive\SYNC\Teaching\Benjamin Studio\SP16\week1\ga.3dm
Command:
```

Title	Perspective
Width	612
Height	706
Projection	Perspective
Lens Length	50.0
Rotation	0.0
X Location	6.955
Y Location	24.955
Z Location	21.137
X Target	4.741
Y Target	5.821
Z Target	3.610

```
6.223617
6.212922
6.056176
6.028229
5.71246
5.681284
5.662727
5.639054
```

V. Technology

The course will teach students new workflows for generating, evaluating, and evolving physical designs using custom *Python* scripts running on top of the *Grasshopper* environment for *Rhino*. Prior knowledge of computer programming in Python is encouraged but not required. A working understanding of Rhino and Grasshopper are a prerequisite for taking the course. For basic training in Grasshopper students are encouraged to work through the relevant tutorials on the http://skilltree.gsapp.org/ prior to the first day of class.

VI. Schedule

Session A - Design computation.
The first session will introduce students to the basic concepts of generative design and teach them how to create complex models that can be controlled and evaluated by an automated search algorithm. The Python programming language will be introduced as a way to amplify the generative complexity of parametric models in Grasshopper. This session will also cover techniques for evaluating designs including using third-party Grasshopper plugins for structural and environmental analysis.

	Date	Location	Module	Topic

1	Jan 23	Avery 115	Introduction to generative design	
2	Jan 30	Fayer. 202	Fundamentals of computational design	
3	Feb 6	Fayer. 202	Generate	Basic elements of programming (Python)
4	Feb 13	Fayer. 202	Generate	Control strategies for computational design
5	Feb 20	Fayer. 202	Evaluate	Structural analysis
6	Feb 27	Fayer. 202	Evaluate	Environmental analysis + CFD (optional)

- March 6th - Session A assignments due

Session B - Design evolution.
The second session will dive deeper into the generative design workflow, and focus primarily on the automated search engine itself. Students will learn how to use state-of-the-art genetic algorithms to automatically search through their design models for high-performing solutions, and how to evaluate the search process to derive new knowledge about their design.

	Date	Location	Module	Topic
7	Mar 6	Fayer. 202	Evolve	Introduction to design optimization
	Mar 13	SPRING BREAK - NO CLASS		
8	Mar 20	Fayer. 202	Evolve	Elements of genetic algorithms
9	Mar 27	Fayer. 202	Evolve	Discover I - Setting up an optimization
10	Apr 3	Fayer. 202	Evolve	Discover II - Input types, single vs. multiple objective optimization
11	Apr 10	Fayer. 202	Evolve	Discover III - Series and permutations
12	Apr 17	Fayer. 202	Evolve	Discover IV - Constraints and visualization

- Thursday, May 4, 6-8pm - Session B final review - 300 Buell S

VII. Links

- http://bit.ly/sp17-gd-syllabus - this page
- https://github.com/danilnagy/discover - Github page of Discover/Explore tool
- https://medium.com/generative-design - collection of class notes and tutorials
- https://drive.google.com/open?id=0B-JxV5CTg1J9UElyUlNFWUlFWFE - GHPython library
- https://drive.google.com/open?id=0B-JxV5CTg1J9ZEJqczBKS0ZCa2s - ‘explorer.bat’ file

VIII. Reading list

Articles, papers, and book chapters:

1. Alan Turing - Computing Machinery and Intelligence (1950) [pdf]
2. Christopher Alexander - Notes on the Synthesis of Form, *Introduction* (1964) [pdf]
3. Nicholas Negroponte - Computational Design Thinking, *Towards a Humanism Through Machines* (1969) [pdf]
4. William J. Mitchell - Computer-Aided Architectural Design, *Chapter 2: The Computer's Role in Design* (1977) [pdf]
5. Kevin Kelly - Out of Control, *Chapter 15: Artificial Evolution* (1994) [pdf]
6. John Frazer - An Evolutionary Architecture, *Introduction* (1995) [pdf]
7. H. Lipson and J. B. Pollack - *Automatic design and Manufacture of Robotic Lifeforms*, Nature (2000) [pdf]
8. Peter J. Bentley and David W. Corne - Creative Evolutionary Systems, *An Introduction to Creative Evolutionary Systems* (2002) [pdf]
9. A. Konak, D. W. Coit, A. E. Smith - *Multi-Objective Optimization Using Genetic Algorithms: A Tutorial* (2006) [pdf]
10. Ian Keough and David Benjamin - *Multi-objective Optimization in Architectural Design*, SimAUD (2010) [pdf]
11. Daniel Shiffman - The Nature of Code, *Chapter 9: The Evolution of Code* (2012) [link]
12. David Benjamin, Danil Nagy and Carlos Olguin - *Growing Details*, AD (2014) [pdf]
13. Kostas Terzidis - Permutation design, *Introduction* (2015) [pdf]
14. Danil Nagy - *Project Discover: An application of generative design for architectural space planning*, SimAUD (2017) [pdf]

Books

15. James C. Spall - Introduction to Stochastic Search and Optimization (2003) [Amazon]
16. J. J. Schneider and S. Kirkpatrick - Stochastic Optimization (2006) [pdf]
17. Xin-She Yang - Nature-Inspired Metaheuristic Algorithms (2nd edition) (2008) [pdf]
18. Stuart Russell and Peter Norvig - Artificial Intelligence, A Modern Approach (3rd edition) (2010) [pdf]
19. Achim Menges and Sean Ahlquist, ed. - Computational Design Thinking (2011) [Amazon]
20. A.E. Eiben and J.E. Smith - Introduction to Evolutionary Computing (2nd edition) (2015) [pdf]

IX. Semester project

The main deliverable for the class is a design project which students will develop individually or in groups of two during the semester.

- Brief

  Choose a design problem which you want to solve using the tools of generative design. The scale of the problem is up to you, and can relate to any field of design including but not limited to industrial, architecture, or urban design. For example, you might want to optimize the design of a chair, a room, a building, or an entire city block. You can use your studio project or another current or past project as a starting point, but I would recommend that you recalibrate the scope of your design problem specifically for this class. This will allow you to fully explore the generative design methods without being burdened by too many unnecessary factors.

- Components

  1. Design space - create a model that parametrizes the design problem and defines all possible solutions that can be searched by the genetic algorithm. You should be clear in

your choice of parameters, and develop a good intuition for how your model navigates the tradeoffs of bias/variance and complexity/continuity.
2. Design evaluation - define the objectives and constraints of your model. You should be clear about how these measures relate to the requirements of the design problem, how you value your design, and how you communicate these values to the search algorithm.
3. Design evolution - using the tools covered in class, run your model through a series of optimization ‘experiments’ to derive novel and high-performing solutions to your design problem. You should be clear about how you are specifying the algorithm’s parameters before each experiment, how you are analyzing and learning from the results, and how you are adjusting the process each time based on what you have learned.

Deliverables

Model description document - due March 6

- Document your design space and system of measures as a unique design strategy for solving your chosen design problem. Your document should include a description of the following aspects of your design:
  1. What are the dimensions (parameters) of your design space?
  2. How does each parameter affect the design?
  3. What variable types are you using for your parameters, and how does this relate to the complexity and continuity of your design space?
  4. What are the measures and how do they relate to the priorities of your design system?
  5. Do you think your design space searchable?
  6. What are the “intuitive” solutions in your design space?
  7. What do you predict will happen during the optimization?

Final review - Thursday, May 4, 6-8pm

The final review will consist of individual project presentations. Your presentation should fully describe your project according to the rough outline below. The goal of your presentation should be to describe both the design problem you chose to address, as well as the way in which you used generative design to solve the problem or understand it better.

Since this is a visual presentation make sure to spend some time developing graphics that can describe both your design problem and the generative design method in a way that is understandable and intuitive to an audience of designers who may not know the technicalities of the tools we have been using. This includes diagrams that show how your model works and how it is measured, data analysis showing the progress of optimization over time, and diagrams, collages, or renderings showing the final outcomes of the generative design process. The outcomes may be a selection of high-performing designs, or a description of a novel set of strategies which you learned through the optimization process.

Each person/group will have 10 minutes for their project. Your presentation should be around 3 minutes long, which will leave sufficient time for discussion with the jury. I will time the presentations and cut you off at 5 minutes to make sure we have enough time for discussion.

Below is a rough outline of a typical presentation. Please use this only as a guide, and feel free to add to or adapt the outline as necessary for your particular project. You can also find a collection of sample graphics here:

http://bit.ly/sp17-gd-examples

Although you can use these as a starting point you should try to invent and develop your own graphic strategies that can best visually communicate your particular project and the generative design workflow. These methods are still relatively new in design, so there are few rules of thumb for the types of graphics or diagrams you should produce. Therefore you should use this presentation as an opportunity to not only hone your own graphic style as a designer, but experiment and speculate on new representational methods that relate specifically to the generative design methodology.

<u>Basic GD outline</u>

1. Design problem
   a. Set of diagrams or precedent images that describe the design problem you are trying to solve

2. Design model
   a. Set of diagrams showing how your model is constructed in Grasshopper
   b. Set of diagrams that show how the inputs control the model (in other words, how the genotype of input variables controls the phenotype or formal manifestation of each design). The diagrams should make it clear what role each input parameter plays in determining each design variation.

3. Design measures
   a. Diagrams that show what you are measuring about each design variation and how the measure is calculated.
      i. Is the measure used as an objective or constraint in optimization?
      ii. If it is an objective, is the goal to minimize or maximize the value?
      iii. If it is a constraint, what is the criteria for a feasible design?
   b. Examples of intuitive designs that you explored before running the optimization. You can show an example of a poorly performing design as well as a relatively high performing design to give us an intuition about how your model works, what are the trade-offs in objectives, how the constraints limit feasibility, etc.

4. Design optimization
   a. Data analysis showing the result of your optimization(s) including one or more of the following types of scatter plots:
      i. *time vs. objective* - is the algorithm learning to pick designs with higher performance over time? (ID or generation along X-axis, objective value along Y-axis, color, and/or size)
      ii. *time vs. input* - is the algorithm narrowing in on particular strategies over time? (ID or generation along X-axis, input value along Y-axis, color, and/or size)
      iii. *objective vs. objective* - is there a tradeoff in your objectives that is creating a Pareto front of optimal designs. Is the algorithm able to target the front more and more over time? (objective(s) along X and Y-axes, ID or generation along color)
      iv. *input vs. objective* - are there particular input values that tend to create higher performing designs? (input(s) along X and Y-axes, objective(s) along color and size)
   b. Animations, collages, or composites showing process of optimization over time. The animations can visualize each design in the order it was generated, or you can create composite that collapse all the designs in a single generation into a single image. Then you can either cycle through generations in an animation, or create a grid of images that show the whole optimization all at once.