

063-0605-22L

# Computational Structural Design I

## Computational Graphic Statics

Dr. Lluís Enrique

Block Research Group (Prof. Dr. Philippe Block)

HS2022

## Week 2 Friday, September 30th

**9:45 – 10:30**      Recap of last week's exercise  
**Lecture: Algorithmic design & thinking**

**10:30 – 10:45**      Break

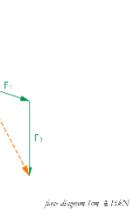
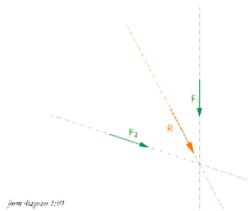
**10:45 – 11:30**      **Tutorial: Procedural graphic statics I**

**11:30 – 11:45**      Break

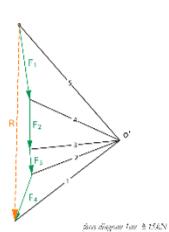
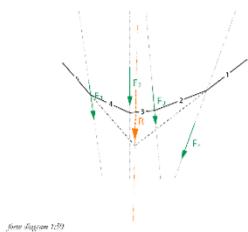
**11:45 – 12:30**      **Tutorial: Procedural graphic statics I**

**Task 1 Resultant of two non-parallel forces**

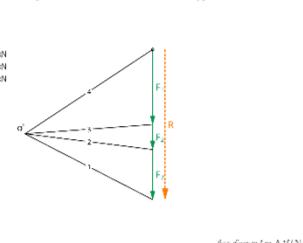
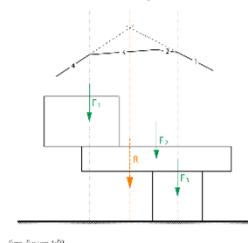
Find the position, direction and magnitude of the resultant in the force and force diagram with the help of vector addition.

**Task 2 Resultant of several non-parallel forces**

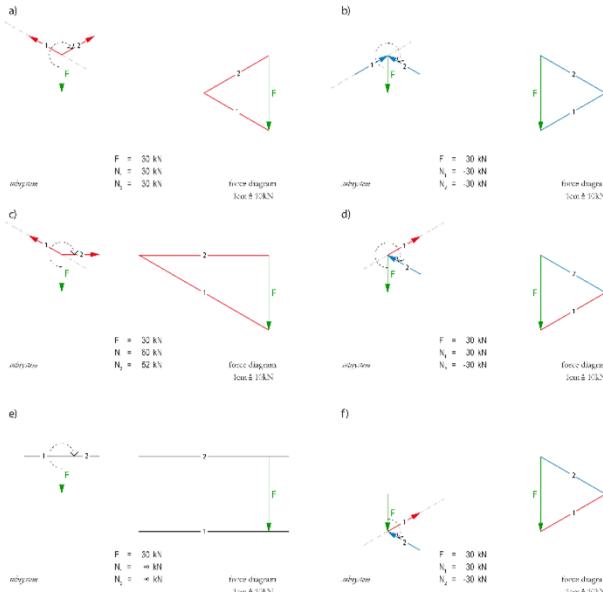
Find the position, direction and magnitude of the resultant in the force and force diagram with the help of a closed force polygon.

**Task 3 Resultant of several parallel forces**

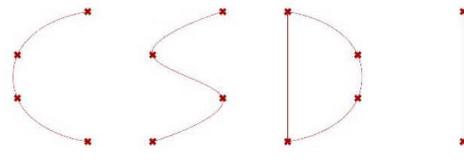
Find the position, direction and magnitude of the resultant in the force and force diagram with the help of a closed force polygon.

**Task 4 Drawing subsystems**

Draw a corresponding force diagram for each subsystem (a-f). Determine the magnitude (N) of each force and mark its direction in the subsystem. Indicate tensile forces with red and compression forces with blue. Explain the solution of situation e).



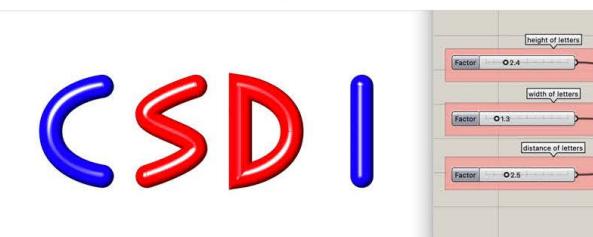
### 1. Point + line/curve construction



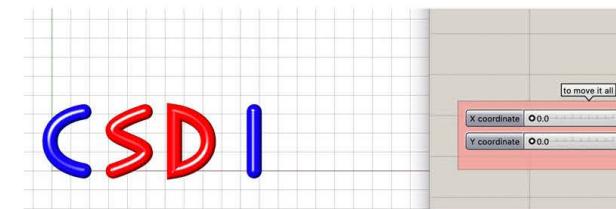
### 2. Coloured balloons



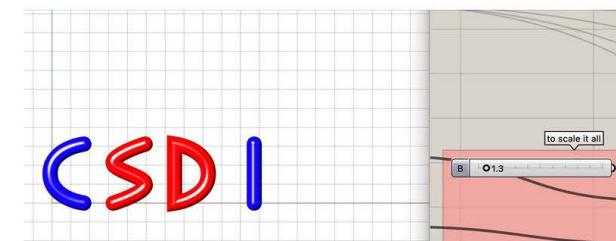
### 3. Height, width, spacing



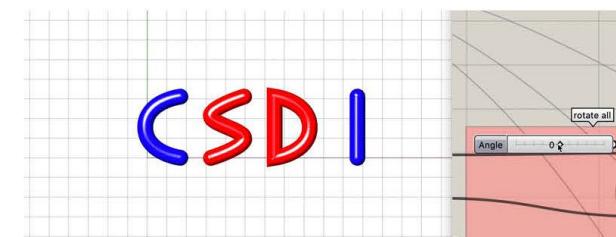
### 4. Position



### 5. Scale

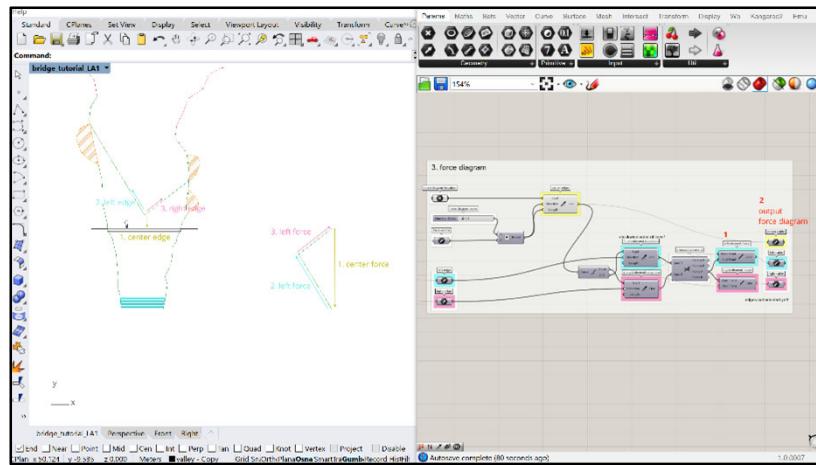
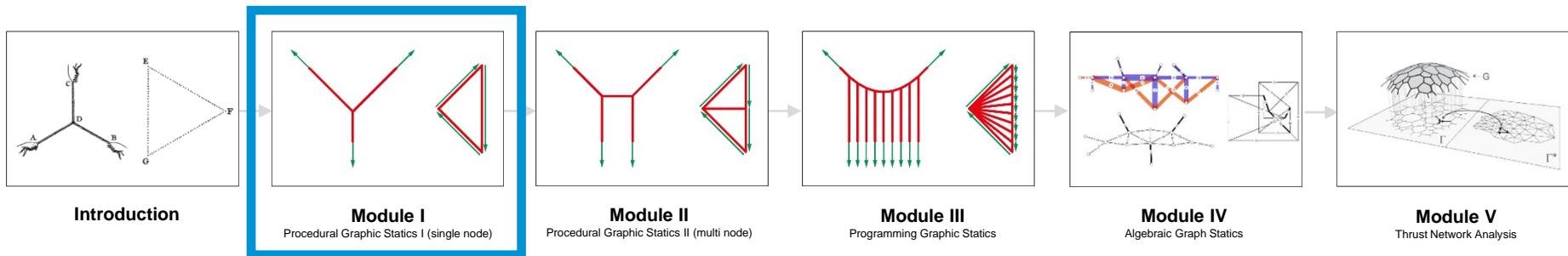


### 6. Rotation



## Schedule

Typology	Topic	Module	Week		Topic	Lead
cables & arches	Introduction	I	Week 1 (23/9)	Lecture	Introduction to graphic statics	Dr. Lluis Enrique
				Tutorial	Introduction to Rhino and Grasshopper	
				Work session	graphic statics + Grasshopper	
			Week 2 (30/9)	Lecture	Quick recap of last week's exercise	Dr. Lluis Enrique
				Tutorial	Algorithmic design & thinking	
	Procedural GS (Grasshopper)	II	Week 3 (7/10)	Work session	Single node bridge in Grasshopper	
			Week 4 (14/10)	Lecture	Quick recap of last week's exercise	Dr. Lluis Enrique
				Tutorial	Computational graphic statics	
				Work session	Multi-node bridge in Grasshopper	
			Week 5 (21/10)		Multi-node bridge in Grasshopper	
Week 6: Seminar week						
cables & arches	Procedural GS (Python)	III	Week 7 (4/11)	Lecture	Quick recap of last week's exercise	Dr. Lluis Enrique
				Tutorial	Programming	
			Week 8 (11/11)	Work session	Multi-node bridge in Python	
trusses	AGS	IV	Week 9 (18/11)	Lecture	Quick recap of last week's exercise	Dr. Lluis Enrique
				Tutorial	Algebraic Graphic Statics (AGS)	Dr. Lluis Enrique
				Work session	Interactive Graphic Statics (IGS)	Chaoyu Du
shells	TNA	V	Week 10 (25/11)	Lecture	Quick recap of last week's exercise	Dr. Lluis Enrique
				Tutorial	Algebraic Graphic Statics (AGS)	Dr. Lluis Enrique
				Work session	Truss analysis using IGS	
shells	TNA	V	Week 11 (02/12)	Lecture	Quick recap of last week's exercise	Dr. Lluis Enrique
				Tutorial	Thrust Network Analysis (TNA)	Dr. Lluis Enrique
				Work session	RhinoVault 2 (rv2)	Selina Bitting
			Week 12 (09/12)		Shell design using rv2	



+





# Algorithmic design & thinking

Friday, September 30th, 2022

Dr. Lluís Enrique

# Computational Graphic Statics

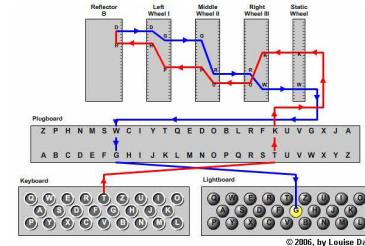
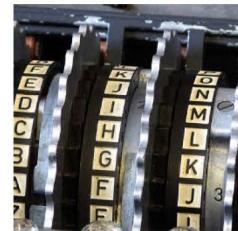


## Universal Computing “Turing” Machine

... a machine capable of computing anything that is computable by executing instructions (program) stored on tape (memory), allowing the machine to be programmable ...

“On Computable Numbers” | Alan Turing (1936)

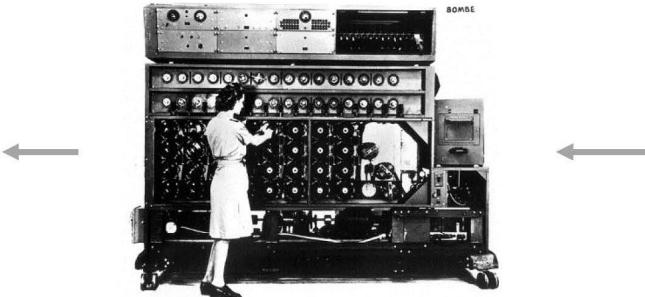
ATTACKATDAWN



WSNPNLKLSTCS

**Encryption machine ("Enigma")**  
"Scrambler" using rotor configurations

**Encryption key**  
Algorithm



**Decryption key**  
Reverse-algorithm

**Decryption machine**  
Multiple reverse-engineered Enigma machines

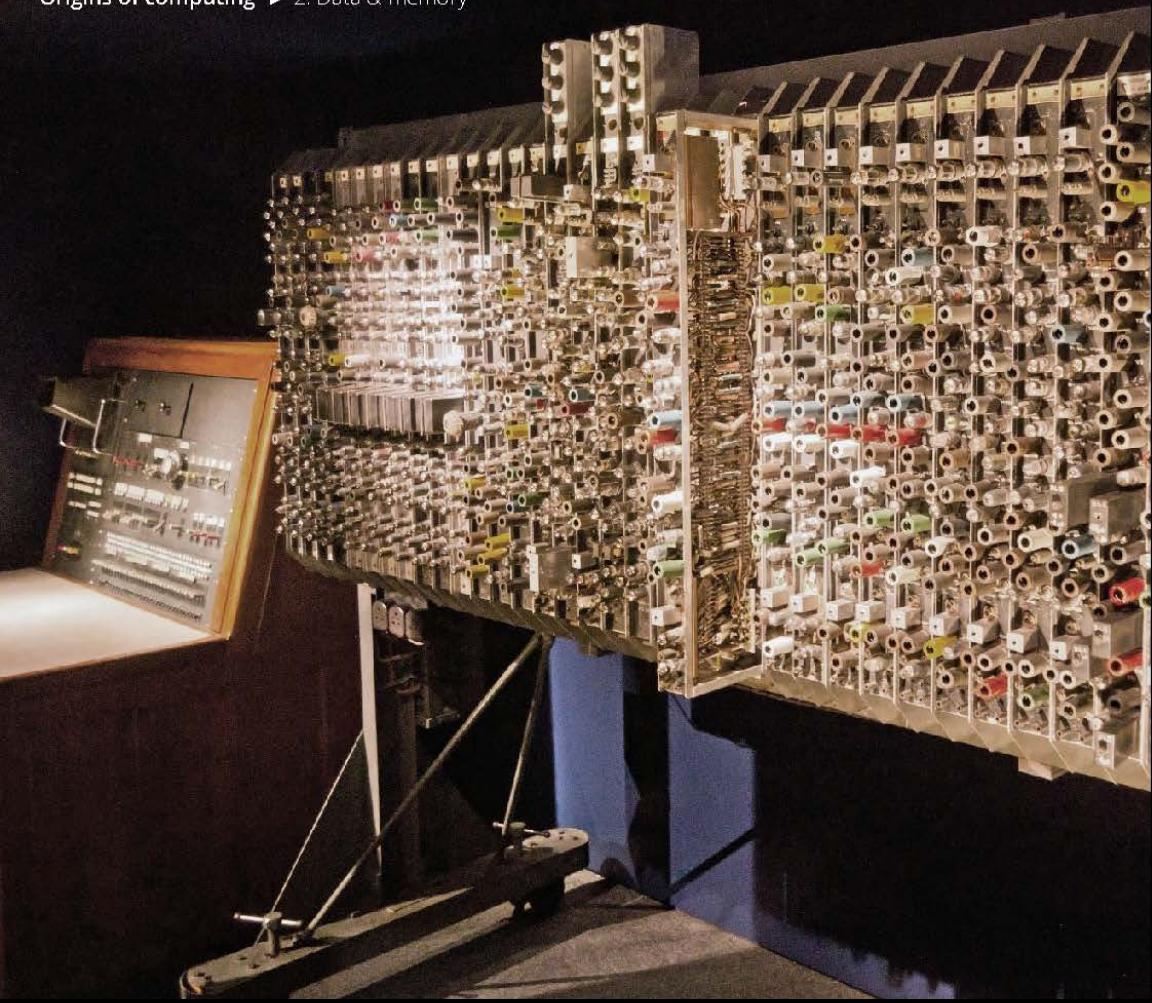


Photo: Antoine Taveneaux

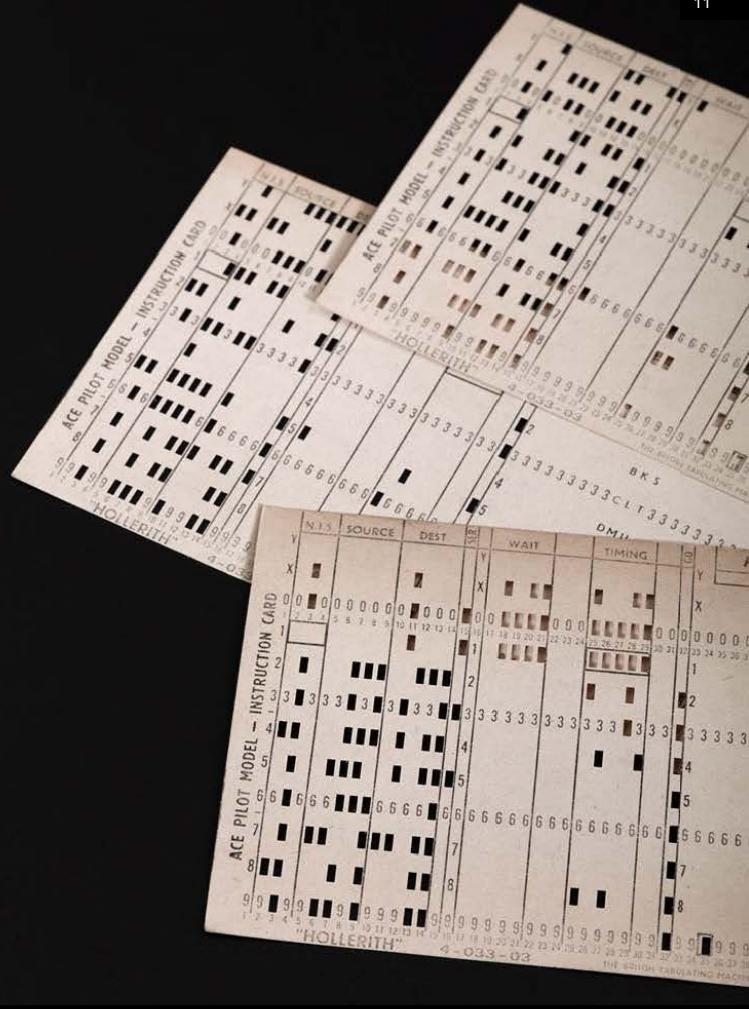


Photo: Science Museum London



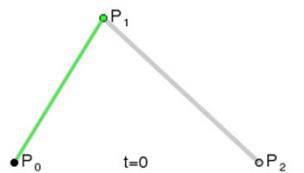


TX-2 computer

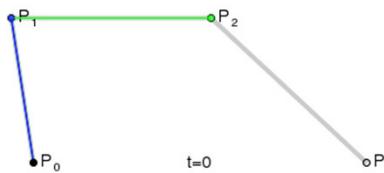




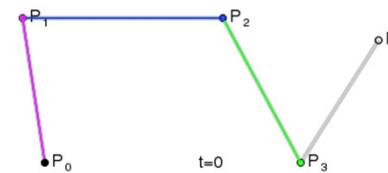
Boeing factory



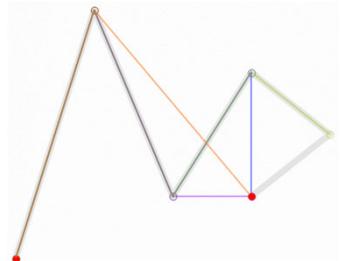
quadratic Bézier curve



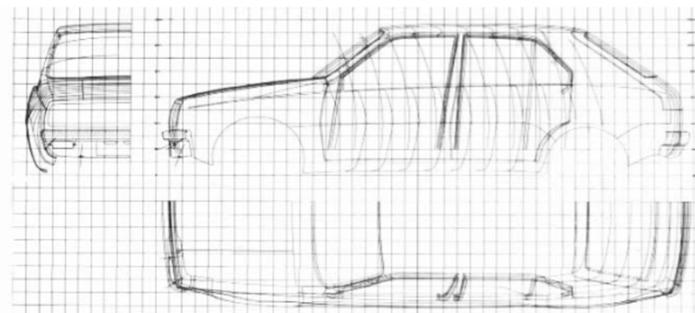
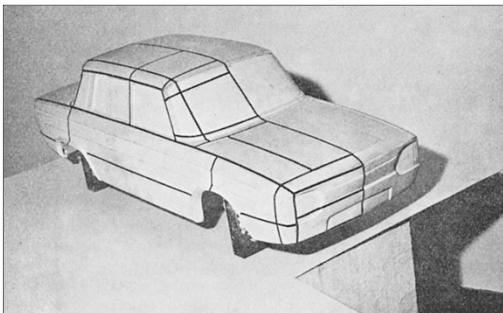
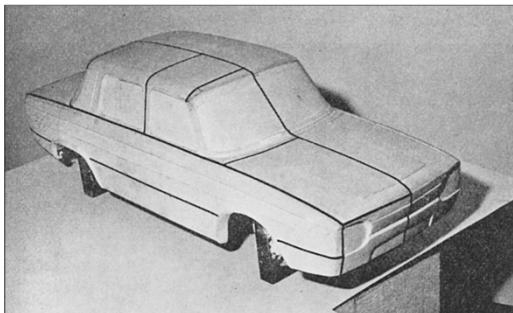
quadratic Bézier curve



cubic Bézier curve



fifth-order Bézier curve





### Xerox Alto (1979)

First "desktop" with graphical user interface (GUI)  
US\$32,000 in 1979 (equivalent to US\$112,726)



### Epson HX-20 (1982)

first "true" laptop computer  
US\$795 (equivalent to US\$2040.76)



### Apple Lisa (1983)

Individual computing with GUI  
US\$9,995 (equivalent to US\$25,811.18)



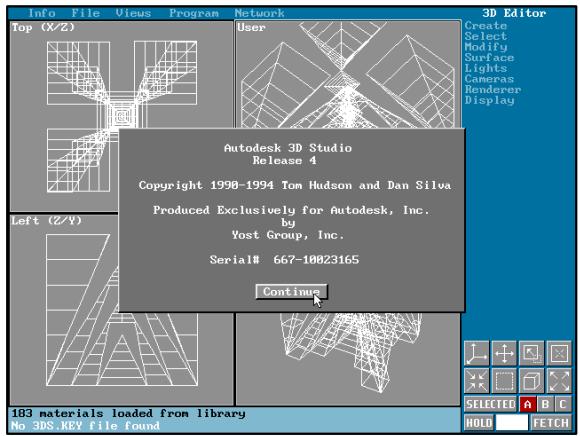
### Macintosh (1984)

First "personal computer"  
US\$2,495 (equivalent to US\$6,100)

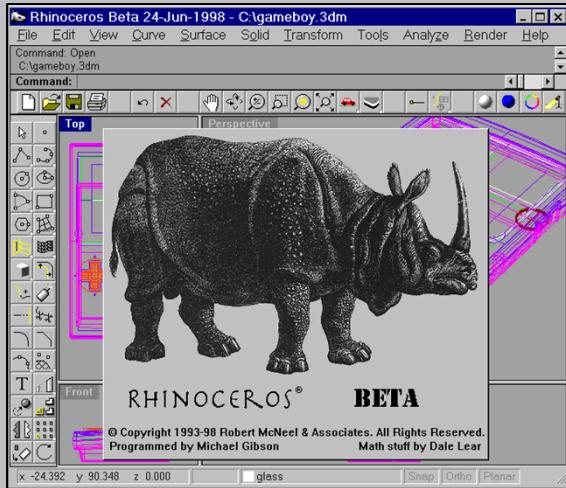
# Computer-aided Design







3D Studio Max (1990)



Rhinoceros/Sculptura 2 (1993)



Maya (1998)



Sage Gateshead by Foster + Partners (2004)



Kunsthaus Graz by Colin Fournier (2003)



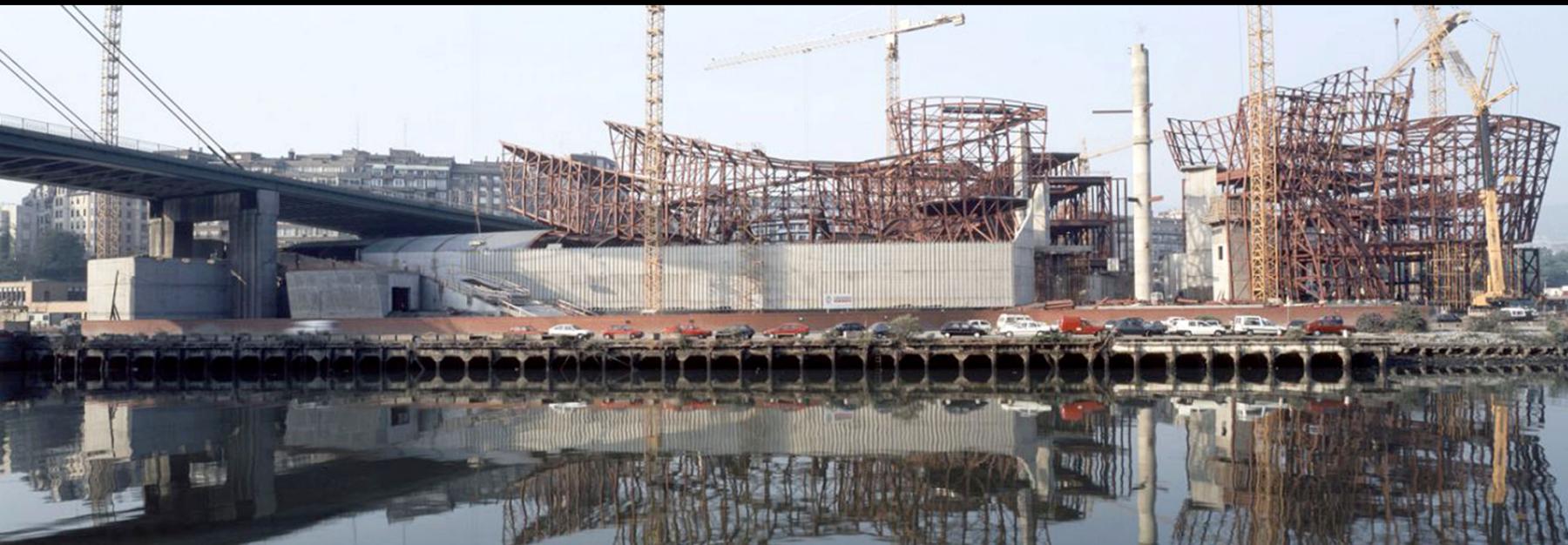
Selfridges Department Store by Future Systems (1999)



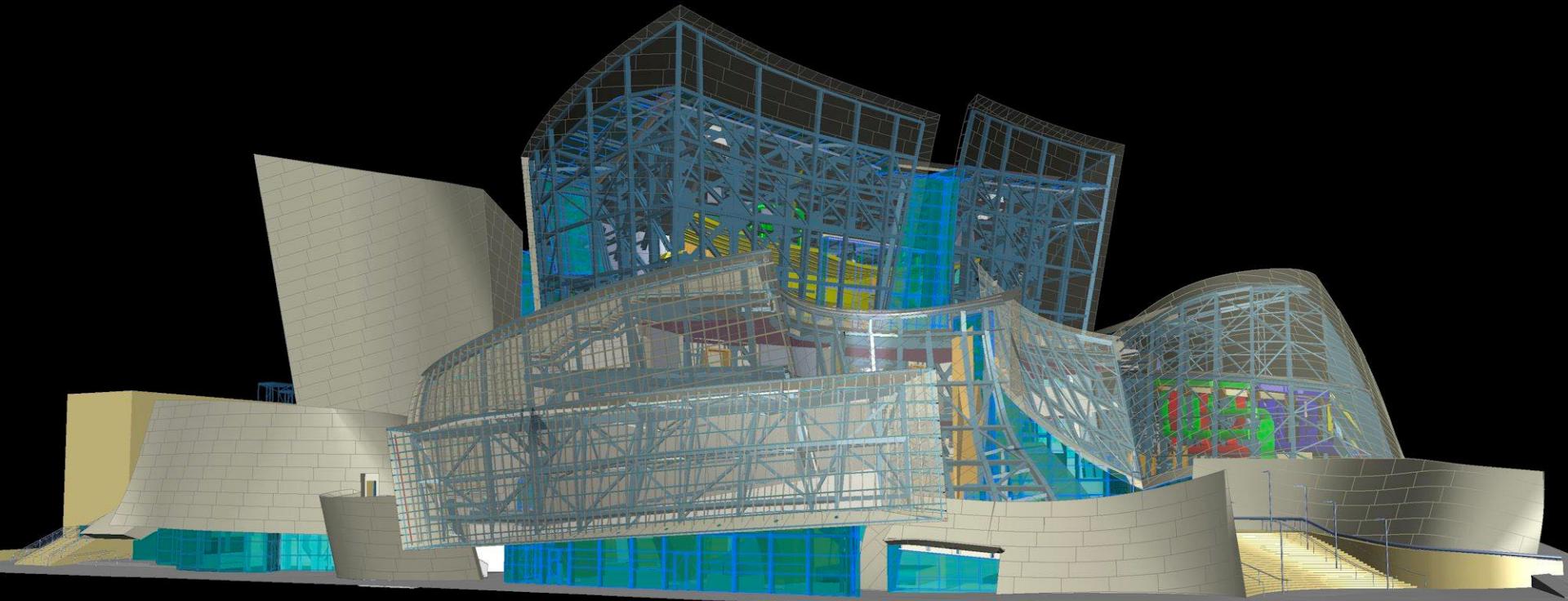
Guggenheim Museum Bilbao by Frank Gehry (1997)



← Frank Gehry

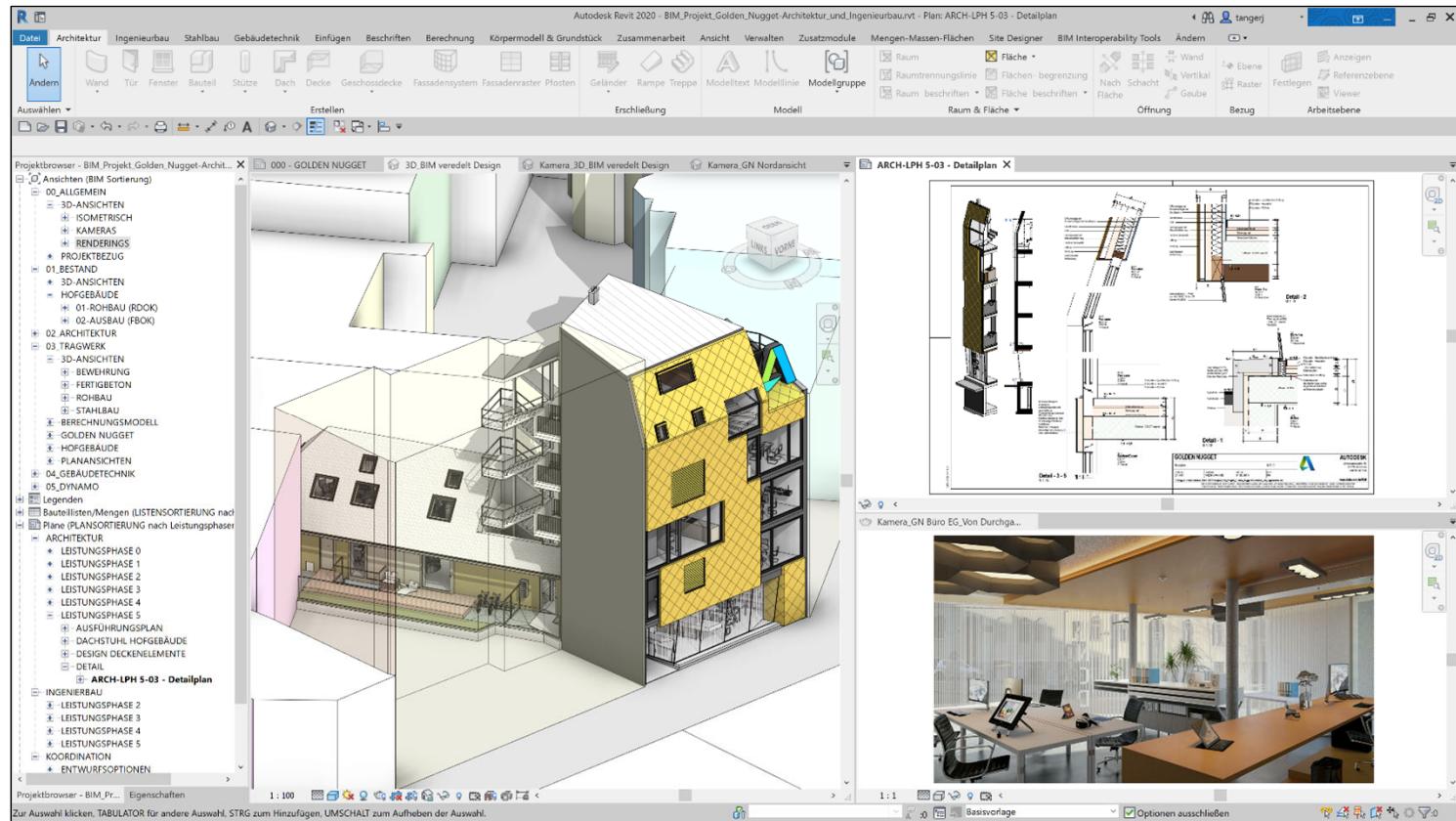


Construction of Guggenheim Museum, Bilbao, Spain, 1997 | Frank Gehry

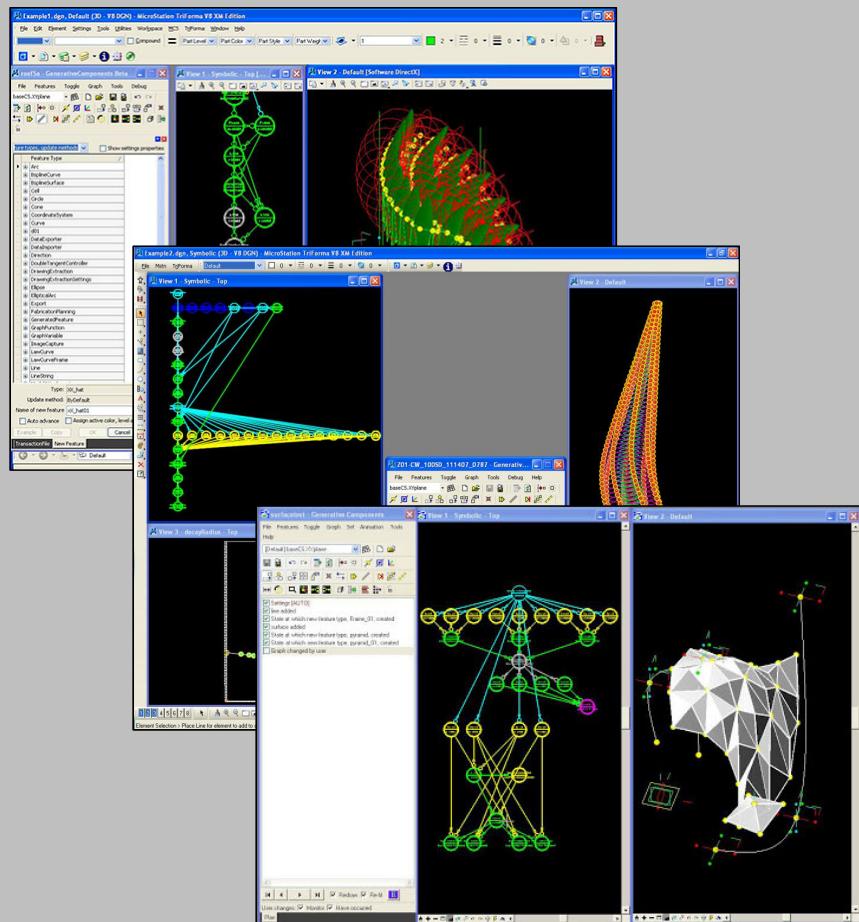


Walt Disney Concert Hall (Digital Project 3D model), Los Angeles, USA, 2003, Frank Gehry

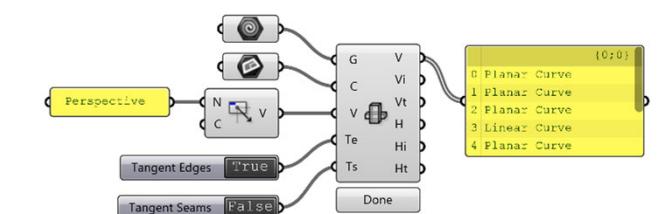
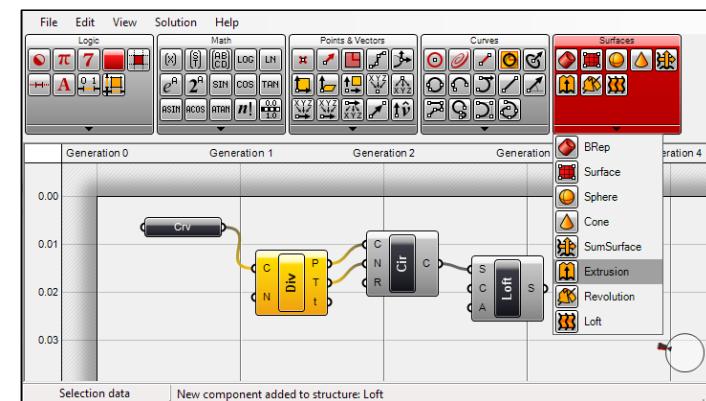




# Computational ~~Computer-aided~~ Design



Generative Design (2003)



Explicit History / Grasshopper (2007)



Metropol Parasol by Jürgen Mayer



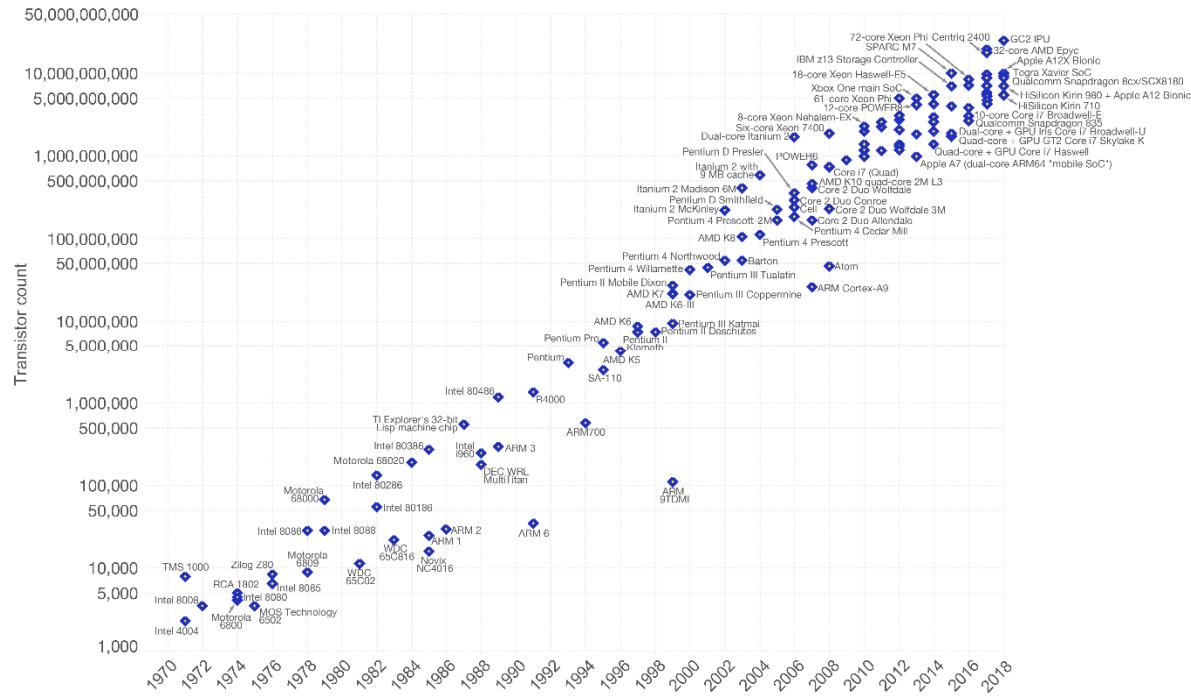
Galaxy SOHO by Zaha Hadid Architects



Serpentine Gallery Pavilion by BIG

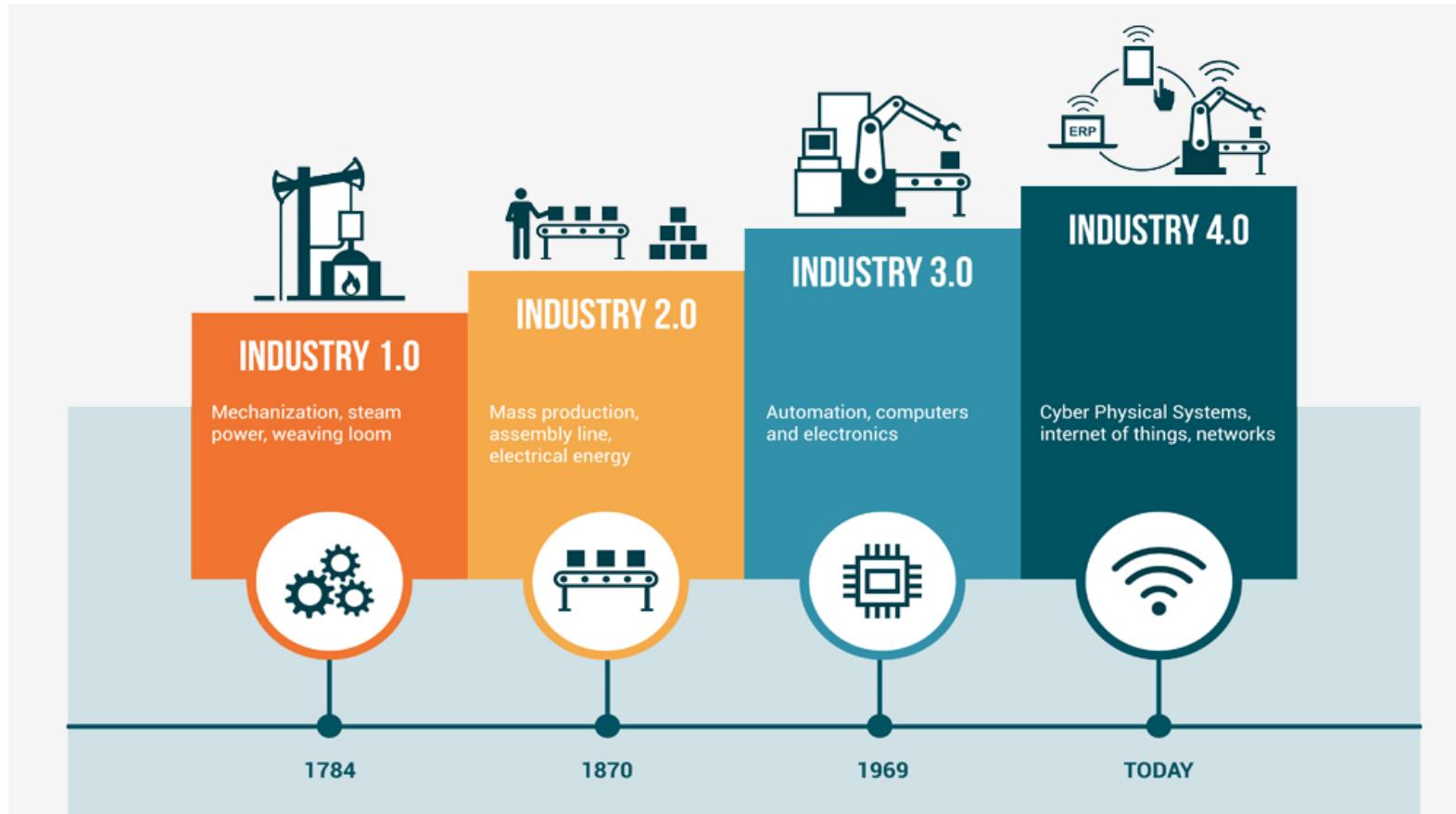


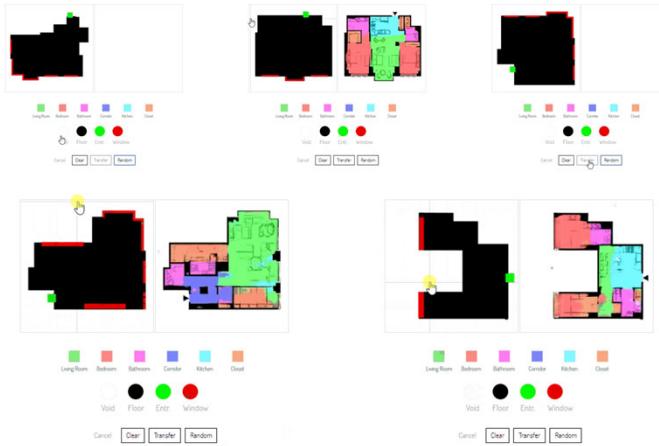
Pillar of Dreams by Marc Fornes



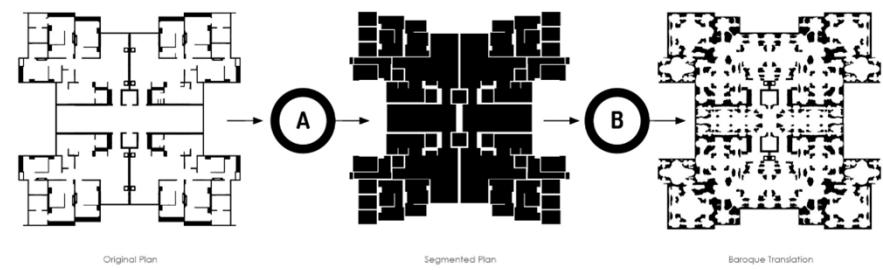
## Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



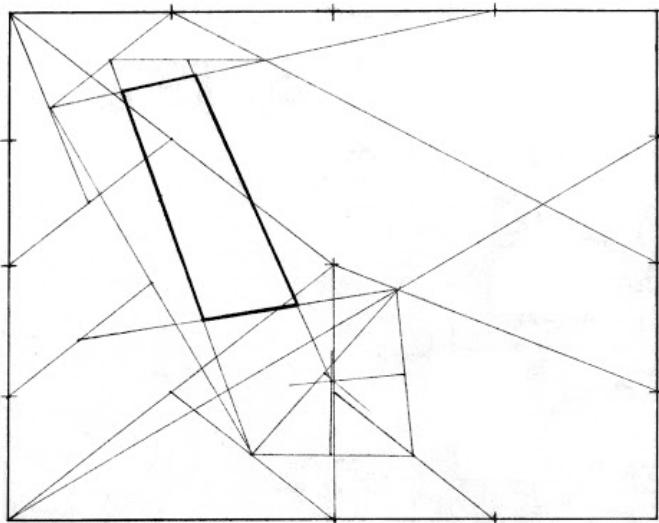


Pattern recognition + cataloguing



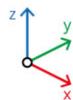
Style crossover

# Computational Design Algorithmic

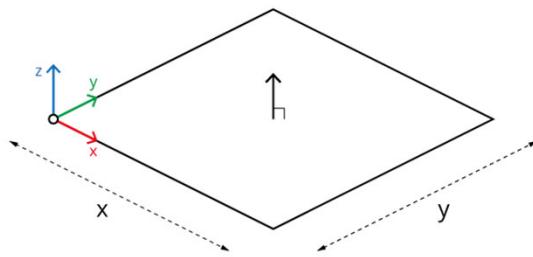


A quadrangle which is formed and enclosed by four lines, the first of which is drawn from a point halfway between a point halfway between the center of the wall and the upper left corner and the midpoint of the left side and the upper left corner to a point halfway between the midpoint of the top side and the upper right corner, the second line from a point halfway between the start of the first line and a point halfway between the midpoint of the top side and the upper left corner to a point halfway between a point halfway between the center of the wall and the lower left corner and the midpoint of the bottom side, the third line from a point halfway between a point halfway between the start of the first line and the end of the second line and a point halfway between the midpoint of the left side and the lower left corner to a point which is on an axis between the lower left corner to a point halfway between the midpoint of the right side and the upper right corner where a line drawn from the center of the wall to a point halfway between the midpoint of the right side and the lower right corner would cross that axis, the fourth line from a point equidistant from the end of the third line, the end of the second line and a point halfway between a point halfway between the center of the wall and the midpoint of the bottom side and a point halfway between the midpoint of the bottom side and the lower right corner to a point halfway between the start of the second line and a point where a line would cross the first line if it were drawn from the midpoint of the right side to a point halfway between the midpoint of the top side and the upper left corner.

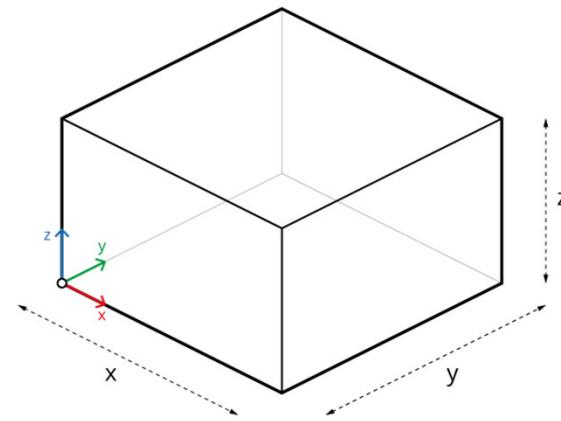
—Sol LeWitt, 1974



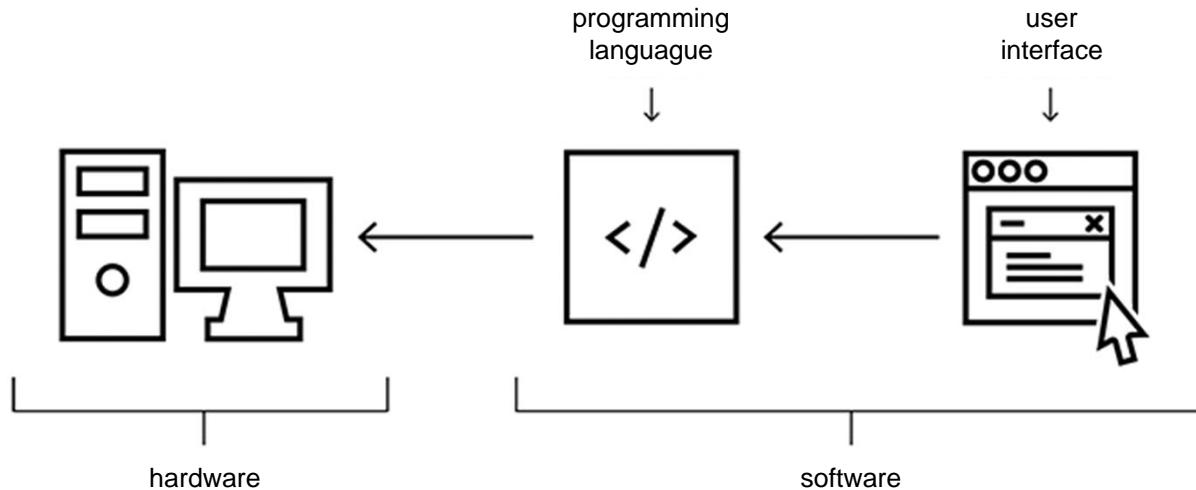
1. Choose a starting point

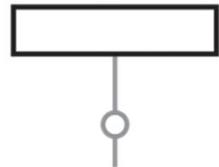


2. Draw a rectangle with dimensions X and Y on the xy plane

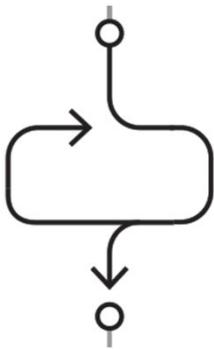


3. Extrude the rectangle in the z-axis with amount z

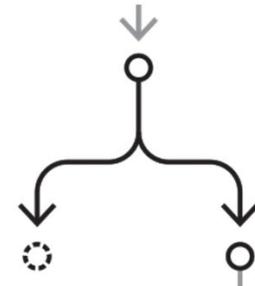




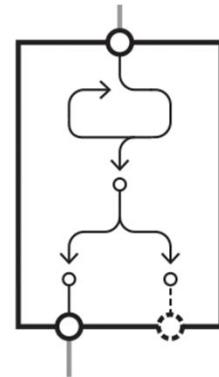
1. variables



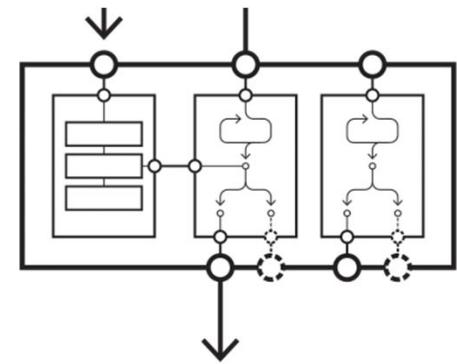
2. loops



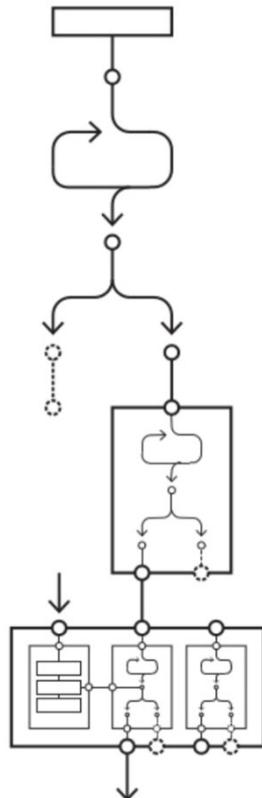
3. conditional

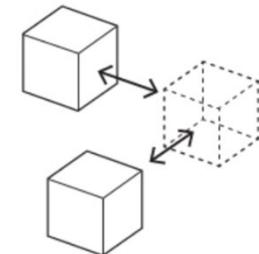
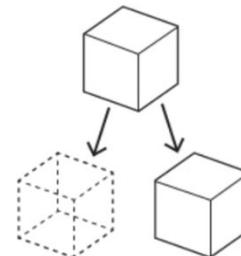
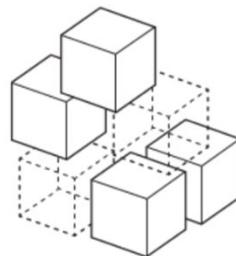
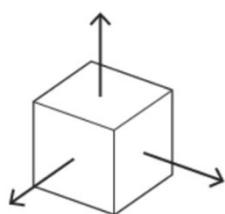


4. functions



5. objects



**Morphological**

- Continuous measures
- + Good top-down control over design
- Can usually only generate simple design spaces

**State-change**

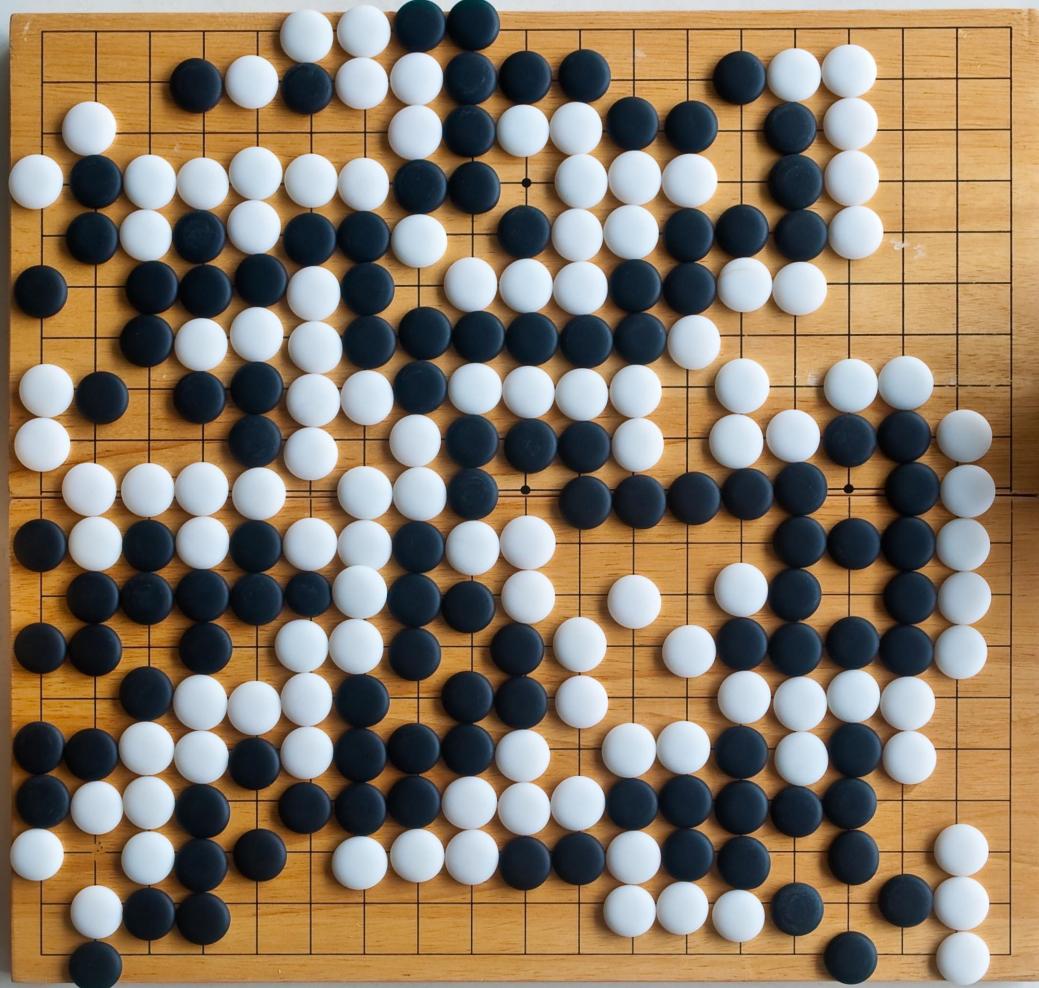
- Choices, categories
- + can create discontinuous design spaces
- + control over individual elements
- Many inputs required  
(each element needs to be controlled separately)

**Rule-based**

- L-system, shape grammars, 1D cellular automata
- + reduced number of inputs  
(abstraction of inputs into rule sets)
- + can create complexity
- Only top-down control
- can't control individual behaviour
- can't create emergence
- Potentially redundant or incomplete design space

**Behavioral**

- object-oriented, agent-based behavior models (dynamic)
- + reduced number of inputs  
(abstraction of inputs into agent behaviors)
- + can lead to emergence
- Little intuitive control over macro design
- Potentially redundant or incomplete design space



Overlay: None PV Prior Rollout VNet Mixed Visits Ownership Critical

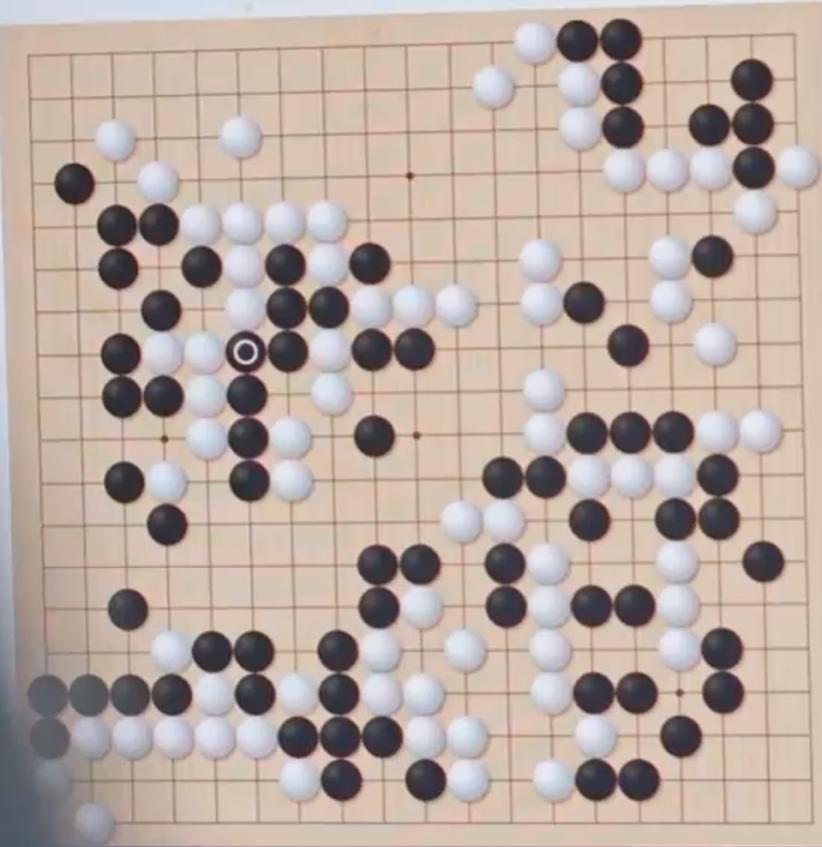


149



Gigo

Rank



Gigo

Rank

## Comments

I0209 01:56:36.261  
I0209 01:56:36.510  
I0209 01:56:36.875  
I0209 01:56:36.875  
I0209 01:56:36.875  
I0209 01:56:36.883  
Move number to pl  
Last move: g9  
Last ko point: in



37