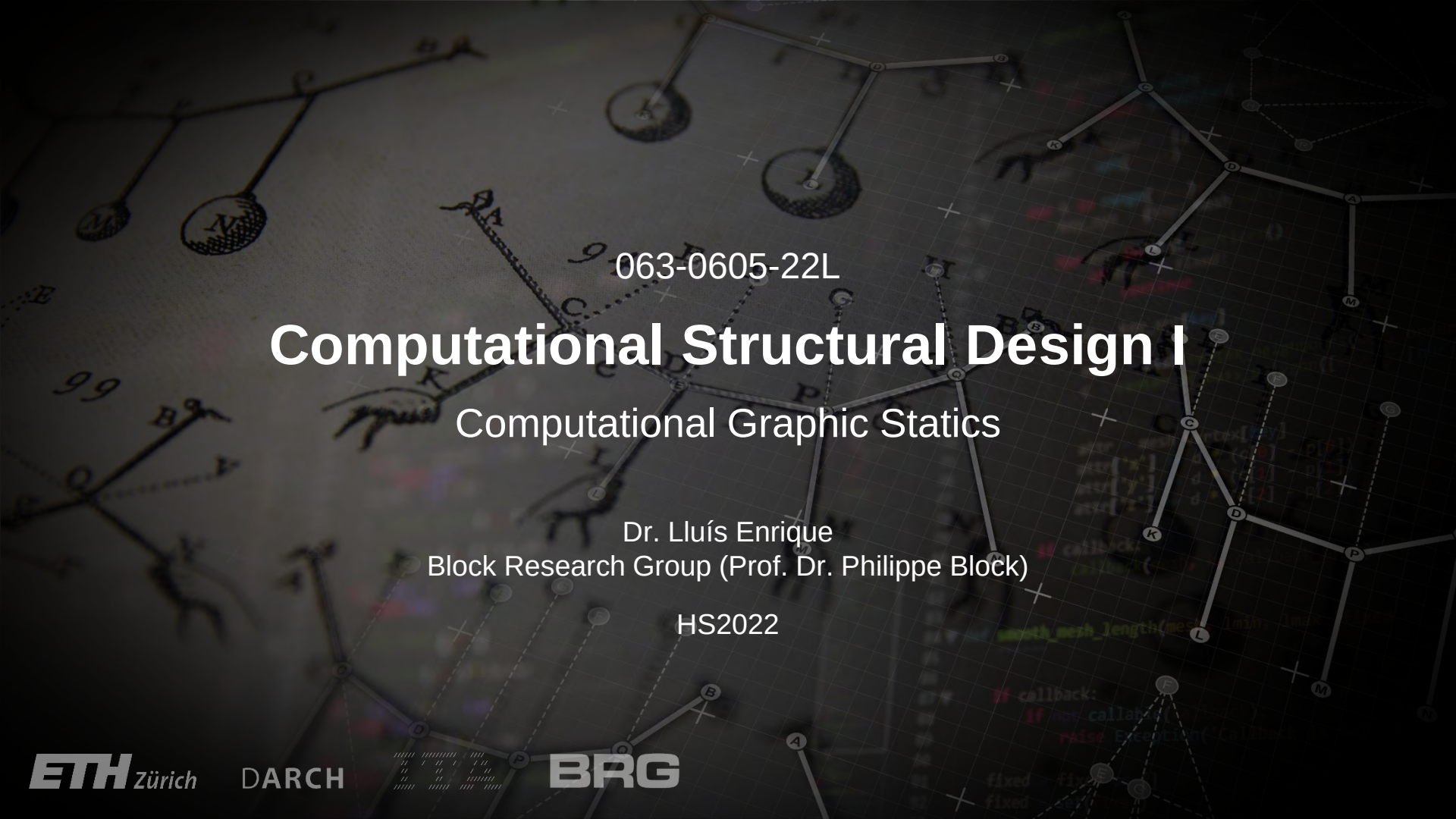


063-0605-22L

Computational Structural Design I

Computational Graphic Statics

Dr. Lluís Enrique
Block Research Group (Prof. Dr. Philippe Block)

HS2022

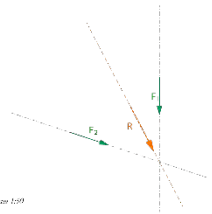
ETH Zürich DARCH BRG

Week 2 Friday, September 30th

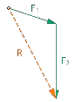
9:45 – 10:30	Recap of last week’s exercise **Lecture: Algorithmic design & thinking**
10:30 – 10:45	Break
10:45 – 11:30	**Tutorial: Procedural graphic statics I**
11:30 – 11:45	Break
11:45 – 12:30	**Tutorial: Procedural graphic statics I**

Task 1 Resultant of two non-parallel forces

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



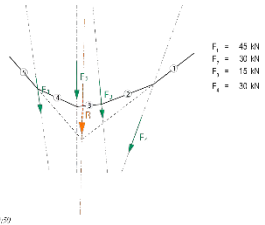
form diagram 1:50



force diagram 1cm ≙ 15kN

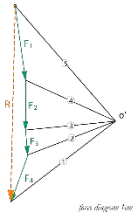
Task 2 Resultant of several non-parallel forces

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



F_1 = 45 kN
F_2 = 30 kN
F_3 = 15 kN
F_4 = 30 kN

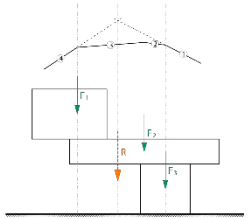
form diagram 1:50



force diagram 1cm ≙ 15kN

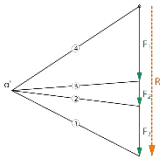
Task 3 Resultant of several parallel forces

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



F_1 = 80 kN
F_2 = 80 kN
F_3 = 40 kN

form diagram 1:50

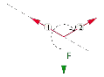


force diagram 1cm ≙ 15kN

Task 4 Drawing subsystems

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

a)



F = 30 kN
N_1 = 30 kN
N_2 = 30 kN

subsystem

force diagram
1cm ≙ 10kN

b)

F = 30 kN
N_1 = -30 kN
N_2 = -30 kN

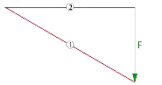
subsystem

force diagram
1cm ≙ 10kN

c)

F = 30 kN
N = 60 kN
N_2 = 52 kN

subsystem



force diagram
1cm ≙ 10kN

d)

F = 30 kN
N_1 = 30 kN
N_2 = -30 kN

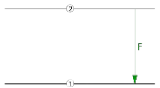
subsystem

force diagram
1cm ≙ 10kN

e)

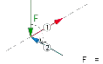
F = 30 kN
N_1 = ∞ kN
N_2 = ∞ kN

subsystem



force diagram
1cm ≙ 10kN

f)

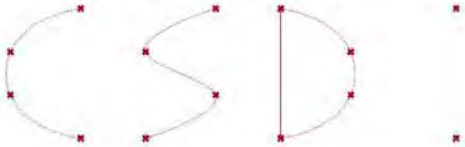


F = 30 kN
N_1 = 30 kN
N_2 = -30 kN

subsystem

force diagram
1cm ≙ 10kN

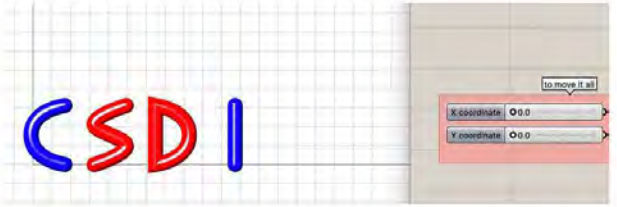
1. Point + line/curve construction



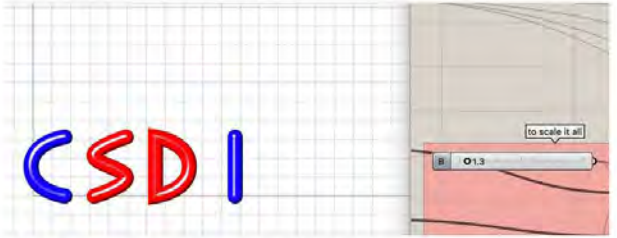
2. Coloured balloons

3. Height, width, spacing

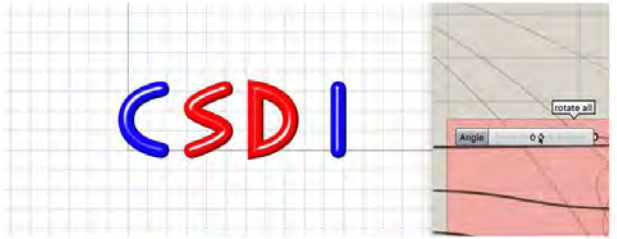
4. Position



5. Scale

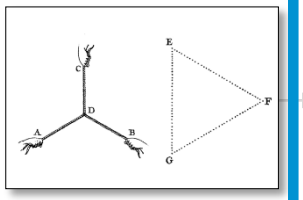


6. Rotation

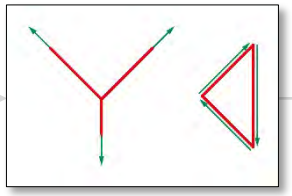


Schedule

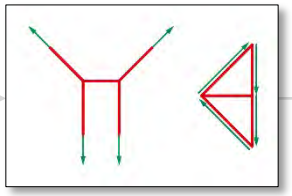
Typology	Topic	Module	Week		Topic	Lead
cables & arches	Introduction		Week 1 (23/9)	Lecture	**Introduction to graphic statics**	Dr. Lluis Enrique
				Tutorial	Introduction to Rhino and Grasshopper	
				Work session	graphic statics + Grasshopper	
	Procedural GS (Grasshopper)	I			Quick recap of last week's exercise	Dr. Lluis Enrique
			Week 2 (30/9)	Lecture	**Algorithmic design & thinking**	
				Tutorial	Single node bridge in Grasshopper	
			Week 3 (7/10)	Work session	Single node bridge in Grasshopper	
		II			Quick recap of last week's exercise	Dr. Lluis Enrique
			Week 4 (14/10)	Lecture	**Computational graphic statics**	
				Tutorial	Multi-node bridge in Grasshopper	
			Week 5 (21/10)	Work session	Multi-node bridge in Grasshopper	
Week 6: Seminar week						
cables & arches	Procedural GS (Python)	III			Quick recap of last week's exercise	Dr. Lluis Enrique
			Week 7 (4/11)	Lecture	**Programming**	
				Tutorial	Multi-node bridge in Python	
			Week 8 (11/11)	Work session	Multi-node bridge in Python	
trusses	AGS	IV			Quick recap of last week's exercise	Dr. Lluis Enrique
			Week 9 (18/11)	Lecture	**Algebraic Graph Statics (AGS)**	Dr. Lluis Enrique
				Tutorial	Interactive Graphic Statics (IGS)	Chaoyu Du
			Week 10 (25/11)	Work session	Truss analysis using IGS	
shells	TNA	V			Quick recap of last week's exercise	Dr. Lluis Enrique
			Week 11 (02/12)	Lecture	**Thrust Network Analysis (TNA)**	Dr. Lluis Enrique
				Tutorial	RhinoVault 2 (rV2)	Selina Bitting
			Week 12 (09/12)	Work session	Shell design using rV2	



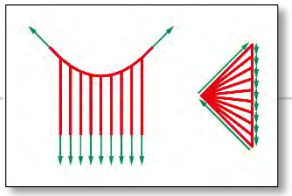
Introduction



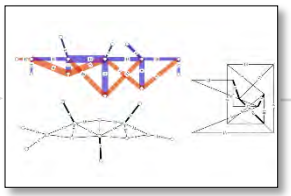
Module I
Procedural Graphic Statics I (single node)



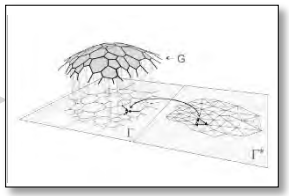
Module II
Procedural Graphic Statics II (multi node)



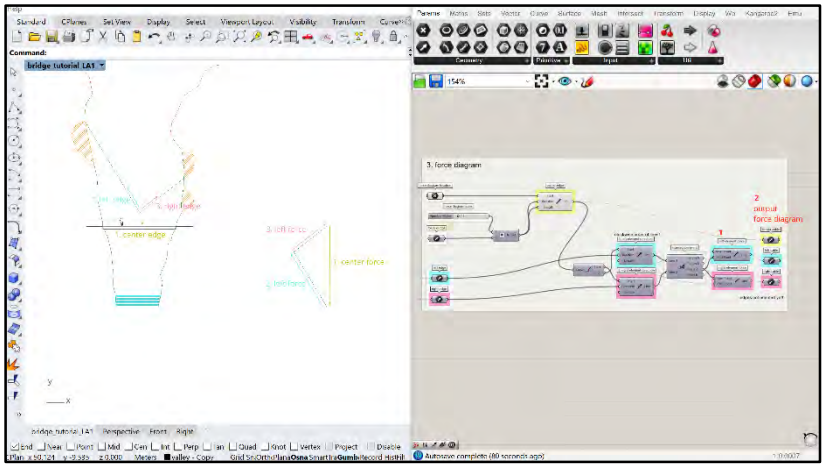
Module III
Programming Graphic Statics



Module IV
Algebraic Graph Statics



Module V
Thrust Network Analysis



+

Algorithmic design & thinking

Friday, September 30th, 2022

Dr. Lluís Enrique

ETH Zürich DARCH BRG

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 …

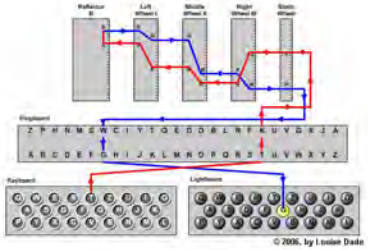
“On Computable Numbers” | Alan Turing (1936)

The Imitation Game (2014)

ATTACKATDAWN

Encryption machine ("Enigma")

"Scrambler" using rotor configurations



Encryption key

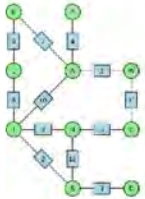
Algorithm

WSNPNLKLSTCS

Decryption machine

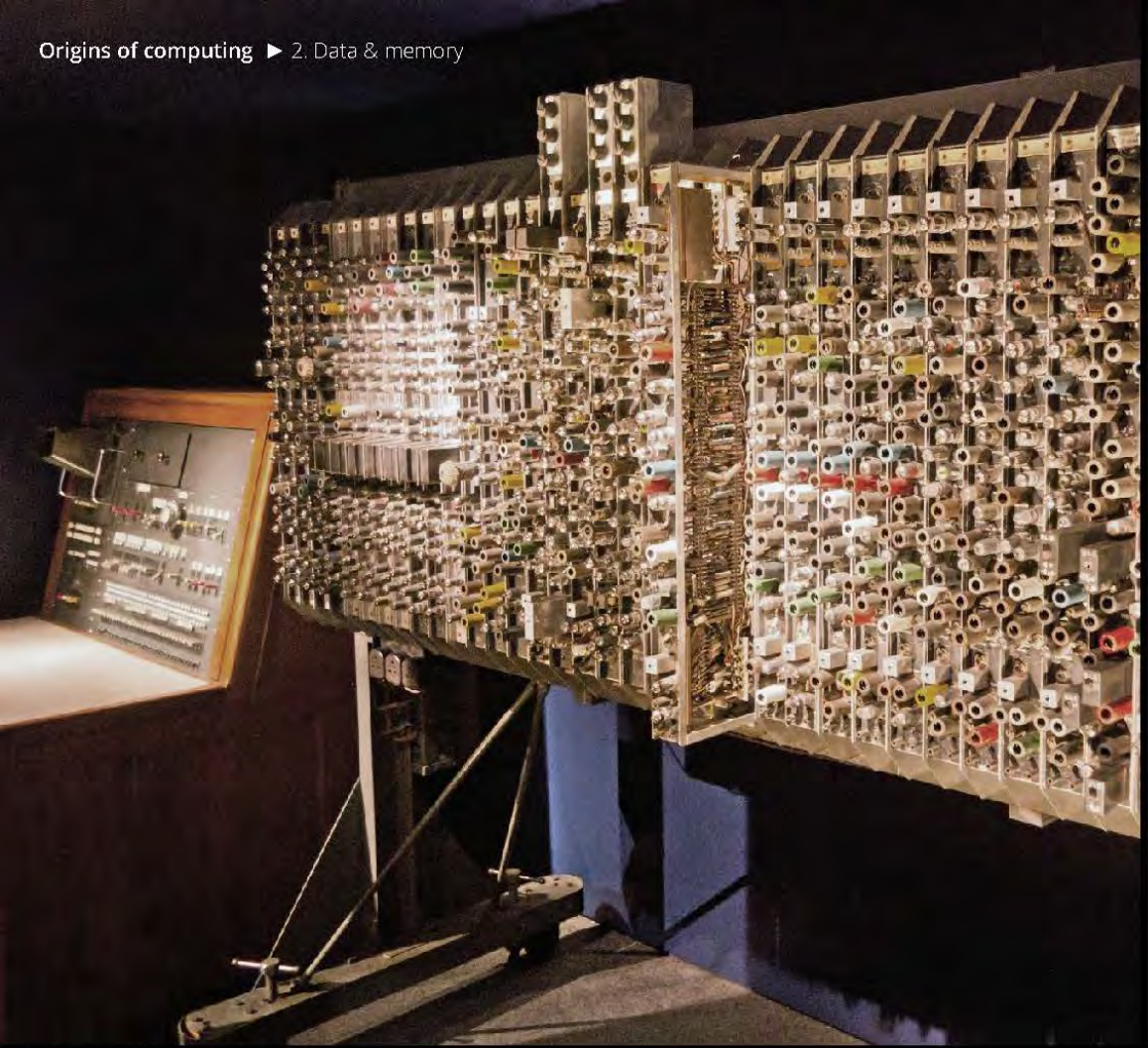
Multiple reverse-engineered Enigma machines

Ciphertext	W	S	N	P	N	L	K	L	S	T	C	S
Plaintext "crib"	A	T	T	A	C	K	A	T	D	A	W	N
Message position	1	2	3	4	5	6	7	8	9	10	11	12
Upper drum setting	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
Middle drum setting	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
Lower drum setting	A	B	C	D	E	F	G	H	I	J	K	L



Decryption key

Reverse-algorithm



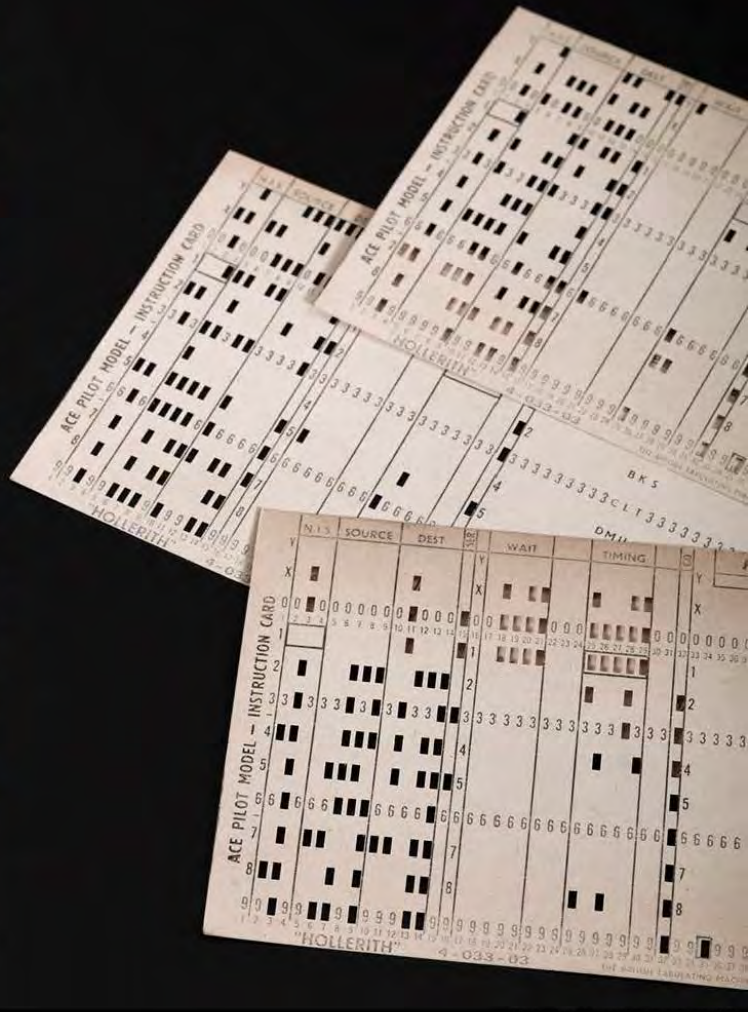
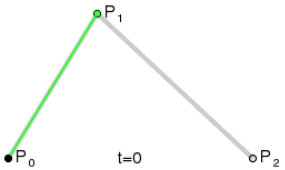


Photo: Antoine Taveneaux

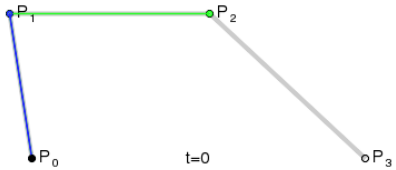
Photo: Science Museum London

TX-2 computer

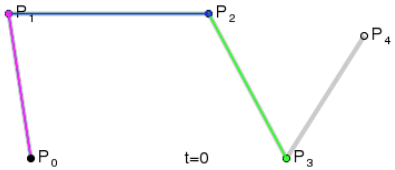
Boeing factory



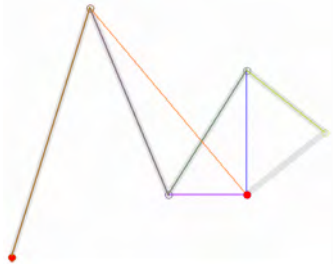
quadratic Bézier curve



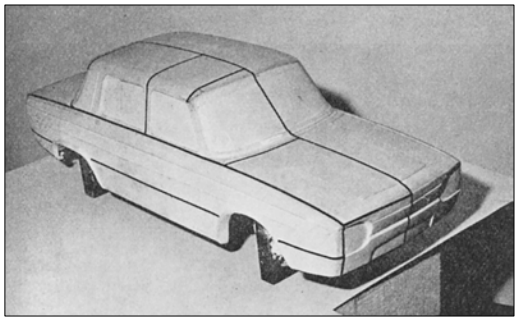
quadratic Bézier curve

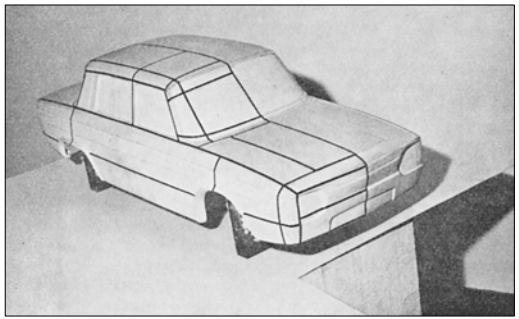


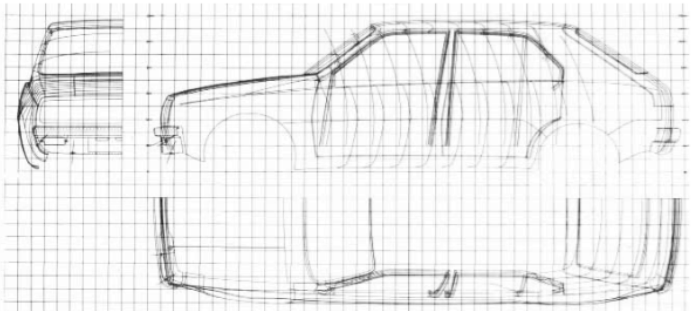
cubic Bézier curve



fifth-order Bézier curve







Bézier-curve-based UNISURZF CAD system for Renault Car Company (1960s)

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)

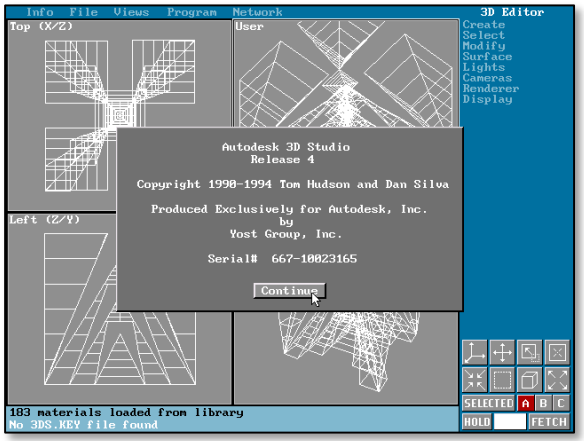
Macintosh (1984)

First “personal computer”
US$2,495 (equivalent to US$6,100)

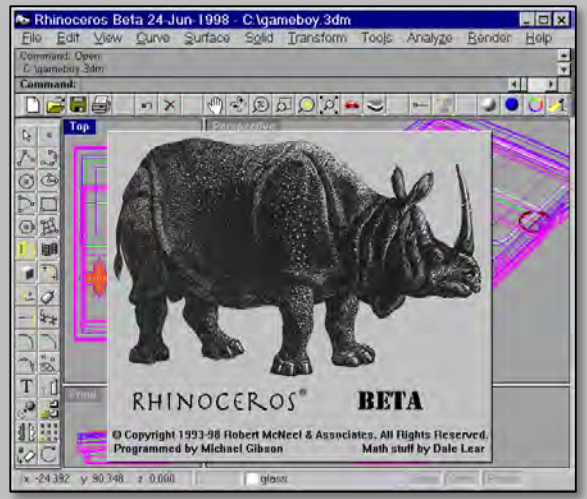
Computer-aided Design

AutoCAD v1.0 (1982)

CATIA 1 (1982)



3D Studio Max (1990)



Rhinoceros/Sculptura 2 (1993)

Maya (1998)

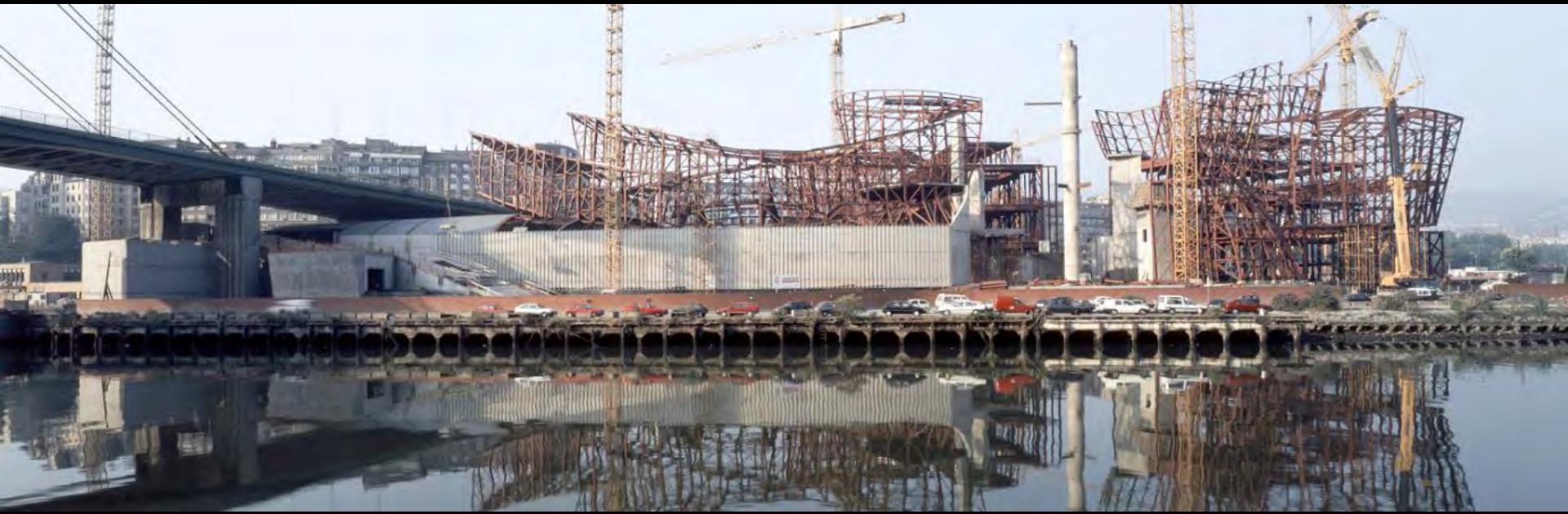
Sage Gateshead by Foster + Partners (2004)

Kunsthaus Graz by Colin Fournier (2003)

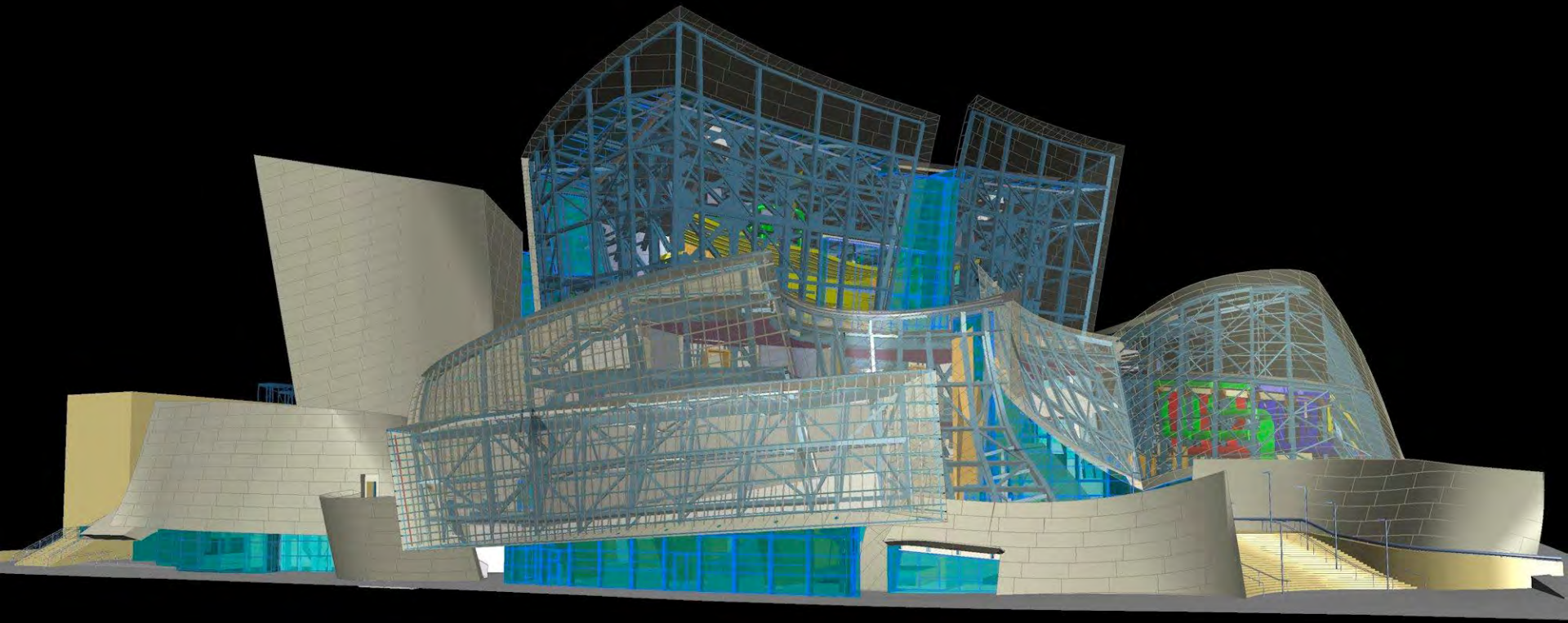
Selfridges Department Store by Future Systems (1999)

Guggenheim Museum Bilbao by Frank Gehry (1997)

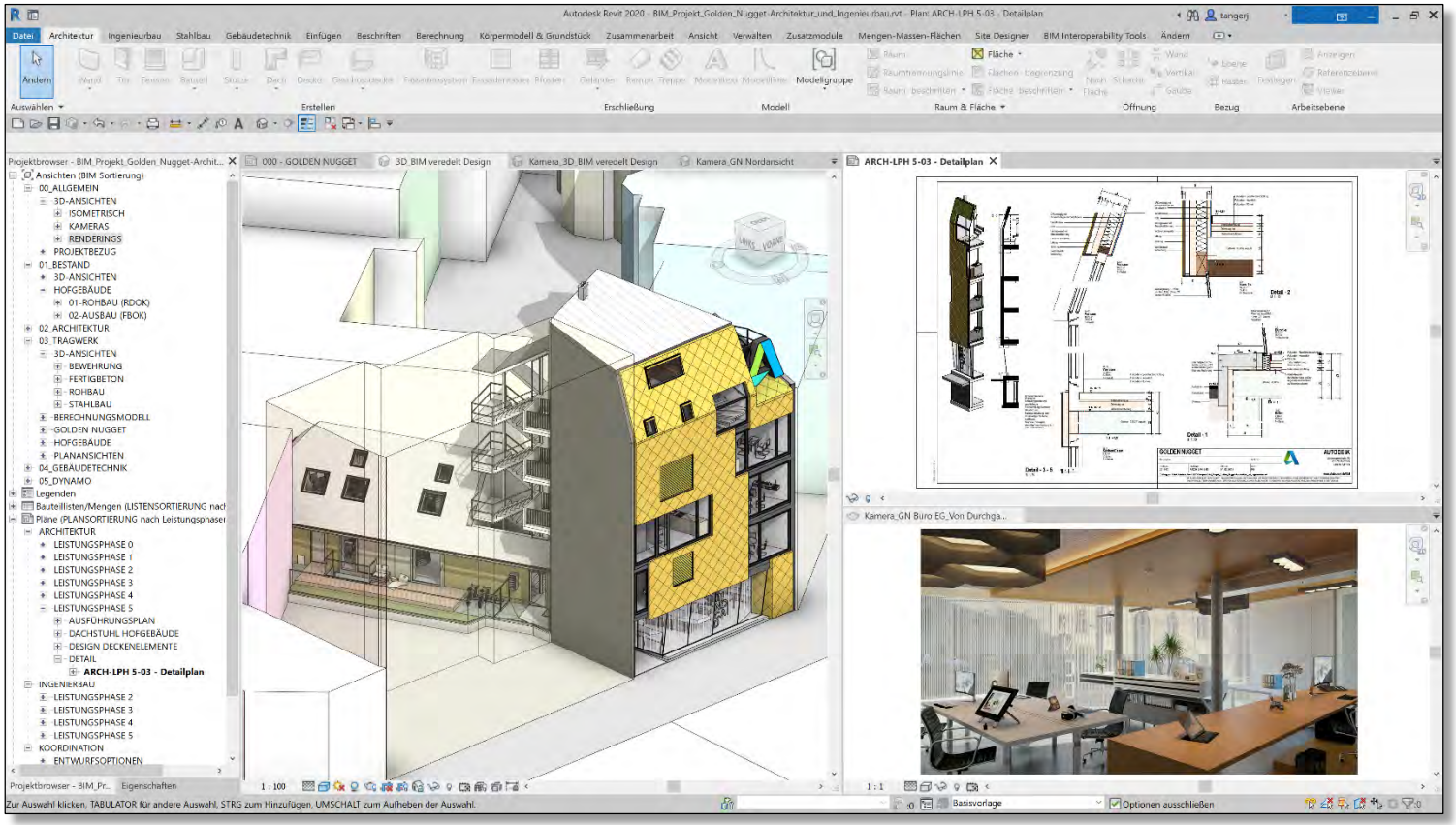

← Frank Gehry



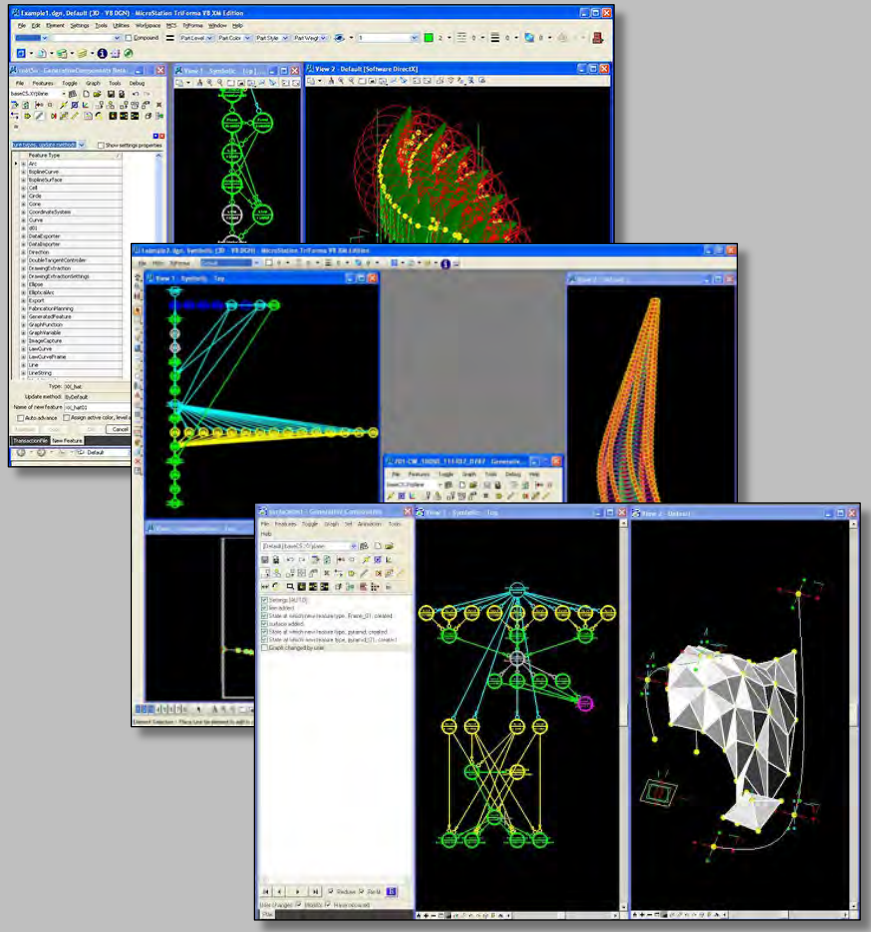
Construction of Guggenheim Museum, Bilbao, Spain, 1997 | Frank Gehry



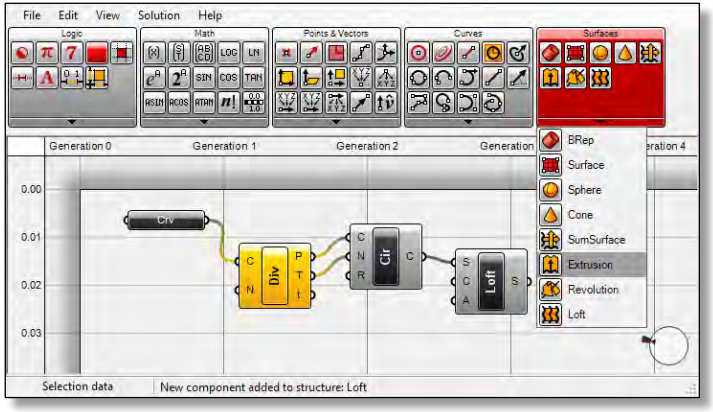
Walt Disney Concert Hall (Digital Project 3D model), Los Angeles, USA, 2003, Frank Gehry

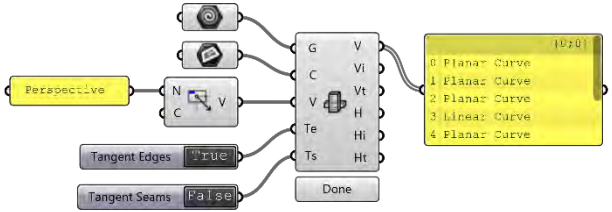

Autodesk Revit 2020 - BIM_Projekt_Golden_Nugget-Architektur_und_Ingenieurbau.rvt - Plan: ARCH-LPH 5-03 - Detailplan
Datei
Architektur
Ingenieurbau
Stahlbau
Gebäudetechnik
Einfügen
Beschriften
Berechnung
Körpermodell & Grundstück
Zusammenarbeit
Ansicht
Verwalten
Zusatzmodule
Mengen-Massen-Flächen
Site Designer
BIM Interoperability Tools
Ändern
Projektbrowser - BIM_Projekt_Golden_Nugget-Archit...
Ansichten (BIM Sortierung)
00_ALLGEMEIN
3D-ANSICHTEN
ISOMETRISCH
KAMERAS
RENDERINGS
PROJEKTBEZUG
01_BESTAND
3D-ANSICHTEN
HOFGEBÄUDE
01-ROHBAU (RDOK)
02-AUSBAU (FBOK)
02_ARCHITEKTUR
03_TRAGWERK
3D-ANSICHTEN
BEWEHRUNG
FERTIGBETON
ROHBAU
STAHLBAU
BERECHNUNGSMODELL
GOLDEN NUGGET
HOFGEBÄUDE
PLANANSICHTEN
04_GEBÄUDETECHNIK
05_DYNAMO
Legenden
Bauteillisten/Mengen (LISTENSORTIERUNG nach
Pläne (PLANSORTIERUNG nach Leistungsphase
ARCHITEKTUR
LEISTUNGSPHASE 0
LEISTUNGSPHASE 1
LEISTUNGSPHASE 2
LEISTUNGSPHASE 3
LEISTUNGSPHASE 4
LEISTUNGSPHASE 5
AUSFÜHRUNGSPLAN
DACHSTUHL HOFGEBÄUDE
DESIGN DECKENELEMENTE
DETAIL
ARCH-LPH 5-03 - Detailplan
INGENIERBAU
LEISTUNGSPHASE 2
LEISTUNGSPHASE 3
LEISTUNGSPHASE 4
LEISTUNGSPHASE 5
KOORDINATION
ENTWURFSOPTIONEN
000 - GOLDEN NUGGET
3D_BIM veredelt Design
Kamera_3D_BIM veredelt Design
Kamera_GN Nordansicht
ARCH-LPH 5-03 - Detailplan
Kamera_GN Büro EG_Von Durchga...
1 : 100
1 : 1
Basisvorlage
Optionen ausschließen
Zur Auswahl klicken. TABULATOR für andere Auswahl. STRG zum Hinzufügen. UMSCHALT zum Aufheben der Auswahl.

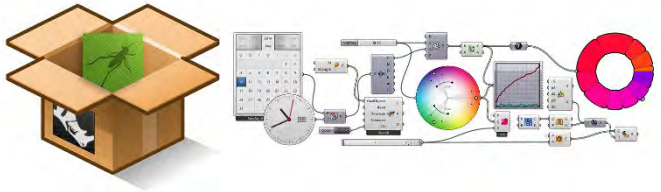
Computational ~~Computer-aided~~ Design



Generative Design (2003)







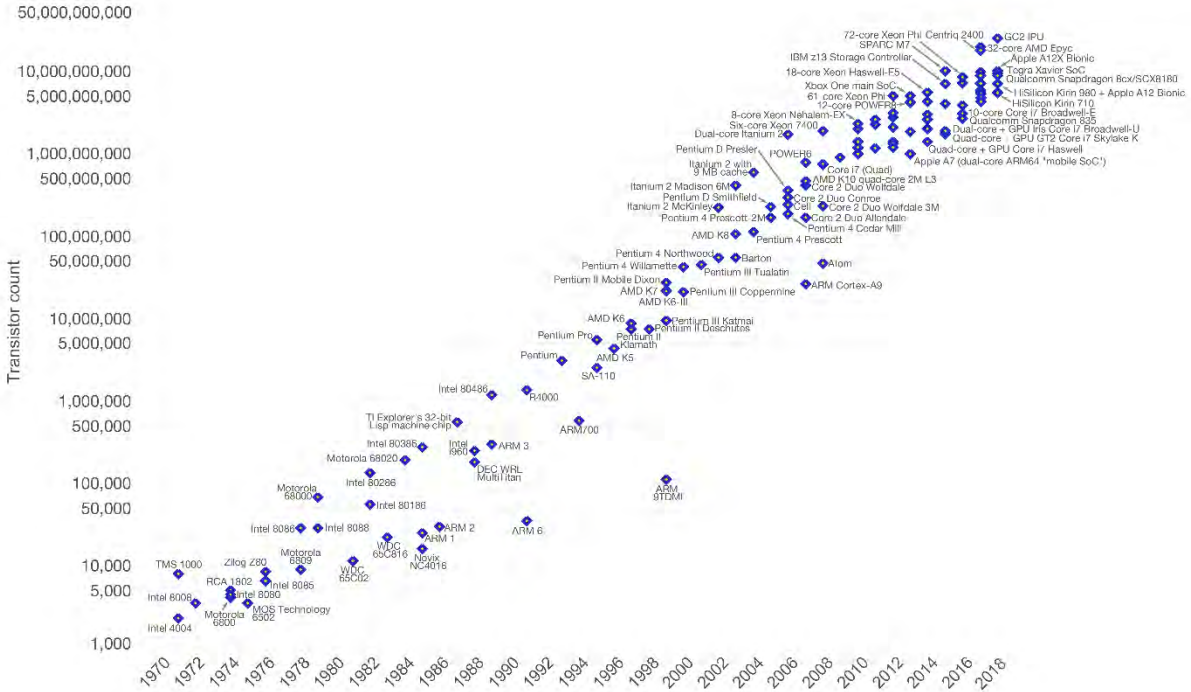
Explicit History / Grasshopper (2007)

Metropol Parasol by Jürgen Mayer

Galaxy SOHO by Zaha Hadid Architects

Serpentine Gallery Pavillion by BIG

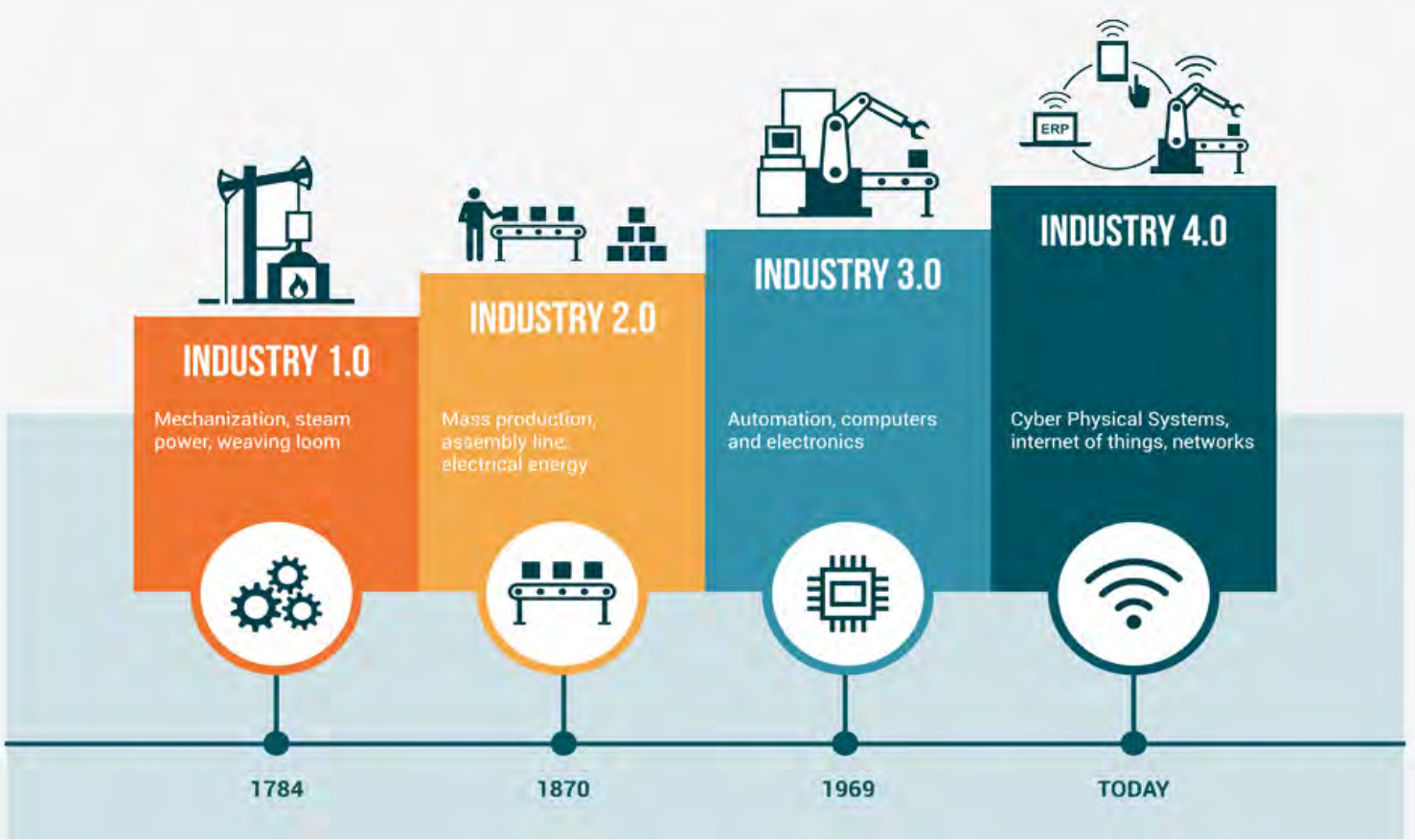
Pillar of Dreams by Marc Fornes

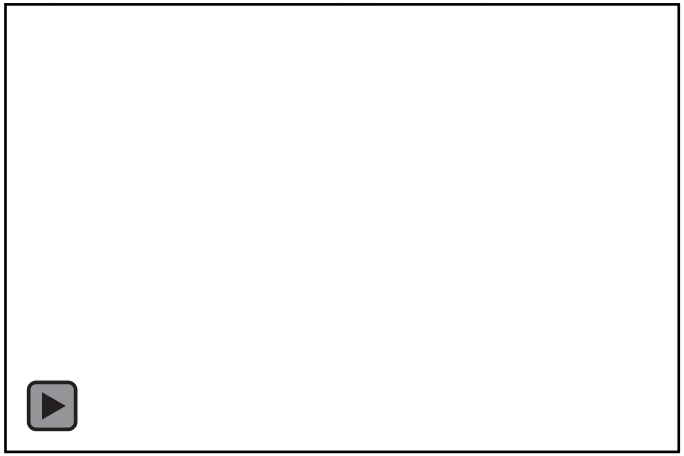


Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

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.

Max Roser via OuWorldInData.org

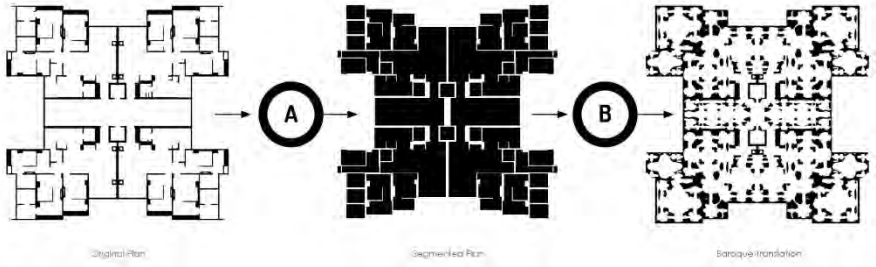

INDUSTRY 1.0
Mechanization, steam power, weaving loom
1784
INDUSTRY 2.0
Mass production, assembly line, electrical energy
1870
INDUSTRY 3.0
Automation, computers and electronics
1969
INDUSTRY 4.0
Cyber Physical Systems, internet of things, networks
ERP
TODAY



Layout + spatial configuration algorithms

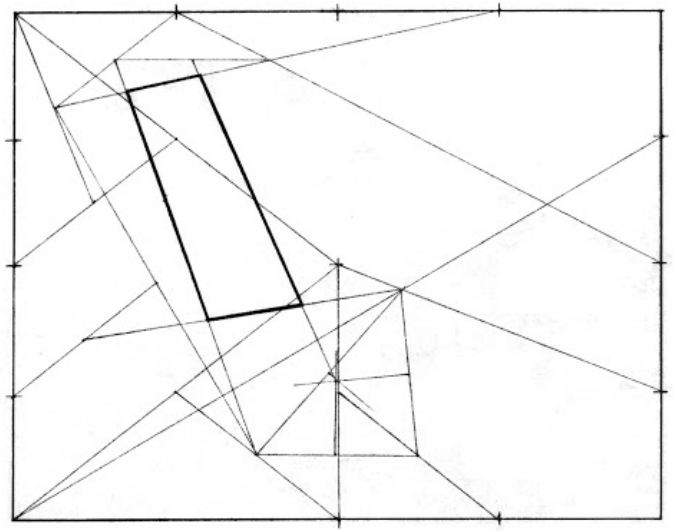
Generative design + topology optimisation

Pattern recognition + cataloguing



Style crossover

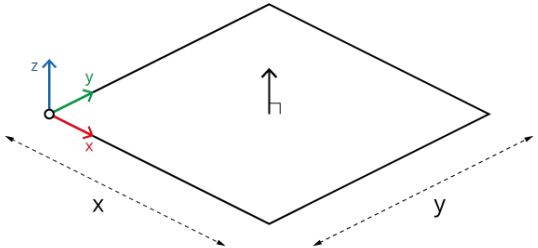
Computational Design
Algorithmic



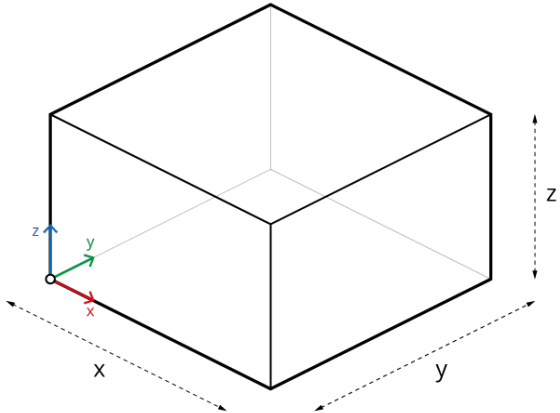
A quadrangle which is formed and enclosed by four lines, the first of which is drawn from a point halfway between a point halfway between the center of the wall and the upper left corner and the midpoint of the left side and the upper left corner to a point halfway between the midpoint of the top side 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

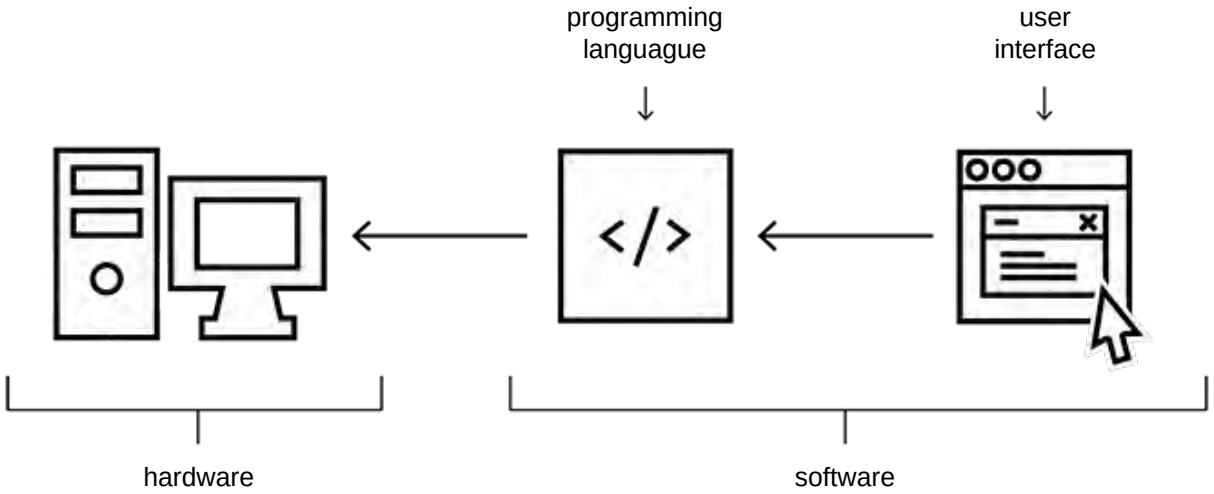
1. Choose a starting point

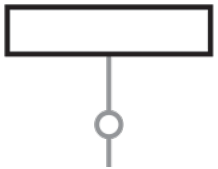


2. Draw a rectangle with dimensions X and Y on the xy plane

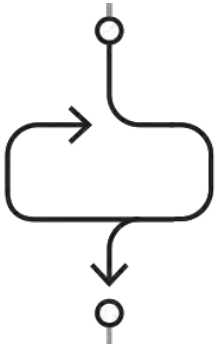


3. Extrude the rectangle in the z-axis with amount z

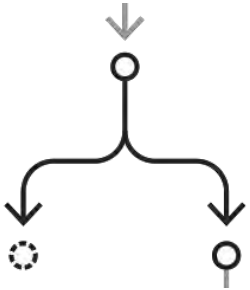

programming
languague
user
interface
hardware
software



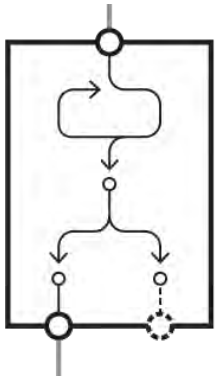
1. variables



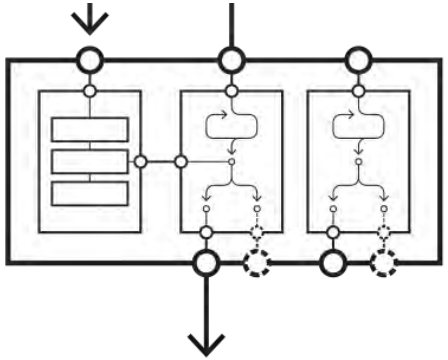
2. loops



3. conditional

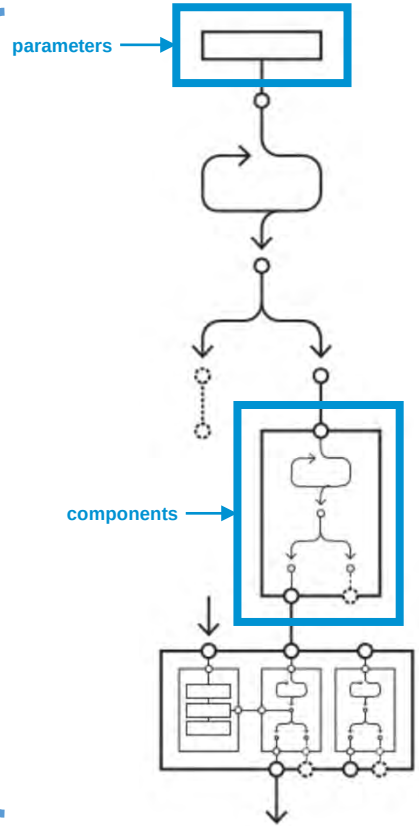


4. functions



5. objects

visual programming
(Grasshopper)



1. variable(s)

```
x = 1
y = 'hello world'
myList = [1, 'b', 3]
myDict = {'a': 1, 'b': 2}
```

2. Loops

```
for i in range(10):
    # do something 10 times

for element in myList:
    # do something for all elements
```

3. Conditional

```
if x > 10 and y > 10:
    # do something
elif x > 5 or y < 5:
    # do something else
else:
    # do something else
```

4. Functions

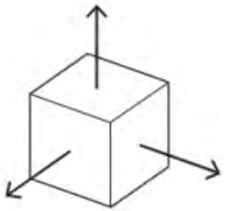
```
def myFunction(input1, intput2):
    # do something
    return output

c = myFunction(a, b)
```

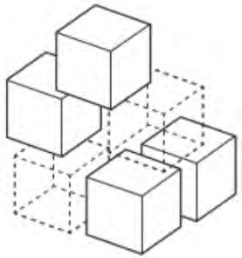
5. Object

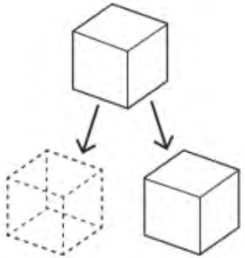
```
class MyClass:
    def __init__(self, input1):
        self.localVar = input1
    def myMethod(self, input2):
        # do something
        return output

classInstance = MyClass(a)
c = classInstance.myMethod(b)
```

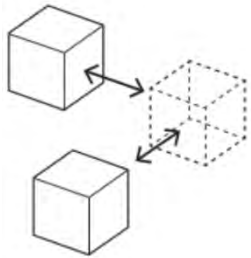
procedural programming
("scripting")

object-oriented programming
("OOP")



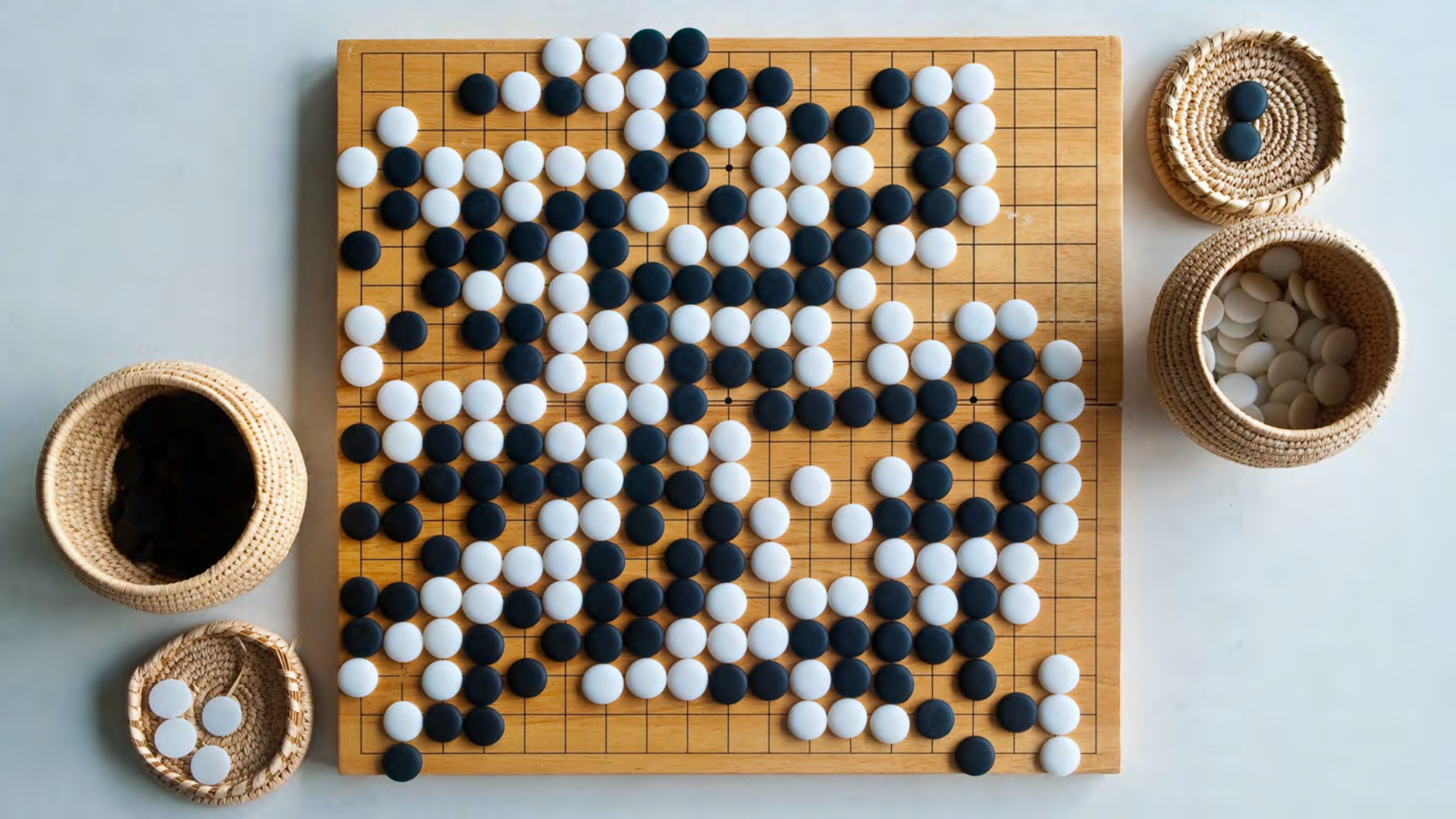






Morphological	State-change	Rule-based	Behavioral
· Continuous measures	· Choices, categories	· L-system, shape grammars, 1D cellular automata	· object-oriented, agent-based behavior models (dynamic)
+ Good top-down control over design	+ can create discontinuous design spaces + control over individual elements	+ reduced number of inputs (abstraction of inputs into rule sets) + can create complexity	+ reduced number of inputs (abstraction of inputs into agent behaviors) + can lead to emergence
- Can usually only generate simple design spaces	- Many inputs required (each element needs to be controlled separately)	- Only top-down control - can't control individual behaviour - can't create emergence - Potentially redundant or incomplete design space	- Little intuitive control over macro design - Potentially redundant or incomplete design space

Adapted from Danil Nagy



37