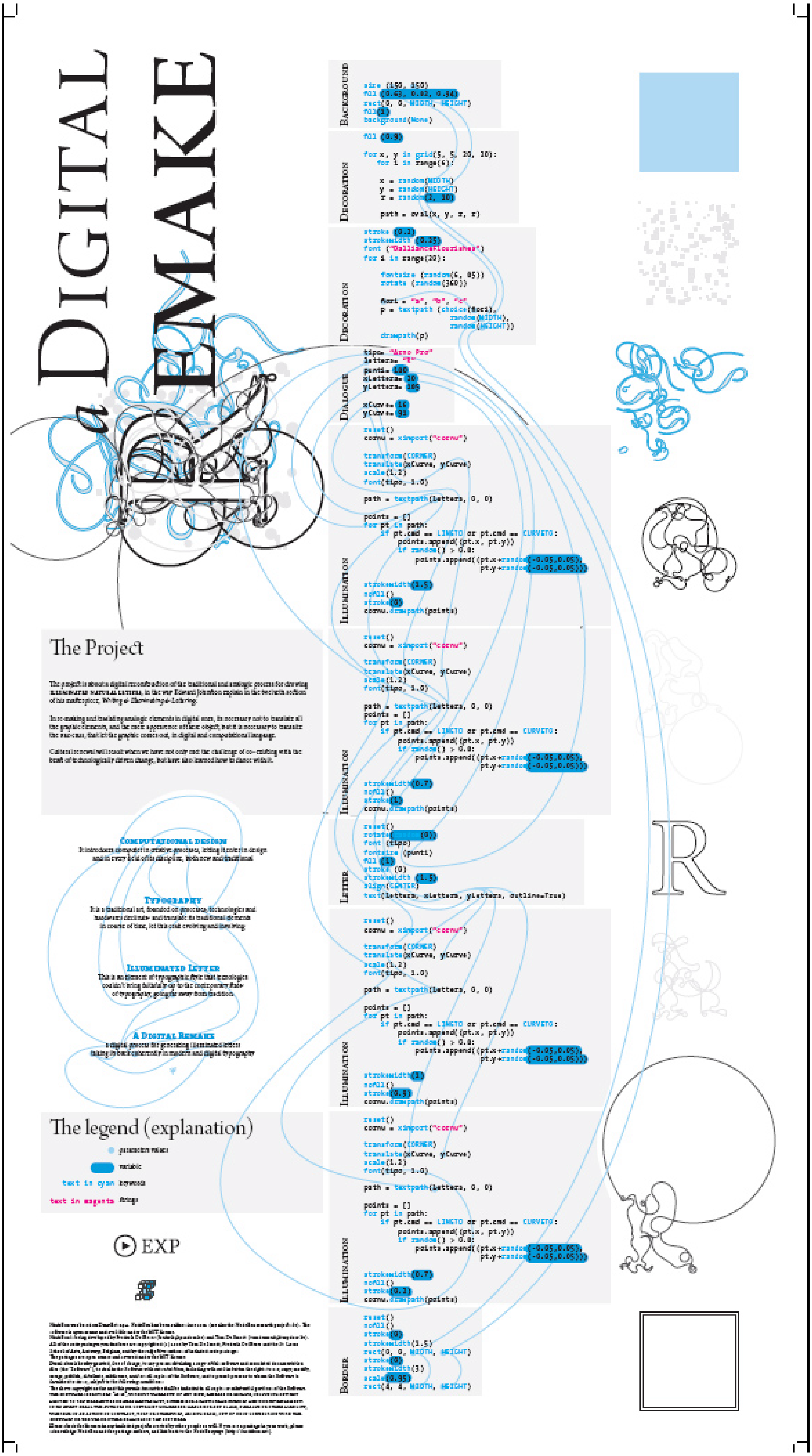


a Digital Remake

The Project

The project is about a digital reconstruction of the traditional and analogic process for drawing ILLUMINATED NATURAL LETTERS, in the way Edward Johnston explains in the twelfth section of his masterpiece, *Writing & Illuminating & Lettering.*

In re-making and translating analogic elements in digital ones, it is necessary not to translate all the graphic elements, and the mere appearance of these object, but it is necessary to translate the structure, that is the graphic construct, in digital and computational language.

Cultural renewal will result whenever we have not only met the challenge of co-existing with the beast of technologically driven change, but have also learned how to dance with it.

Computational design

It introduces computer in creative processes, letting it role in design and in every field of its discipline, both new and traditional.

Typography

It is a traditional art, founded on processes, technologies and hardware to dominate and translate its traditional elements in course of time, let this craft evolving and surviving.

Illuminated Letter

This is an element of typographic style that technologies couldn't bring definitely up to the contemporary state of typography, going far away from tradition.

A Digital Remake

a digital process for generating illuminated letters taking it back coherently in modern and digital typography

The legend (explanation)

●	parameter values
▬	variable
text in cyan	keywords
text in magenta	strings

⊙ EXP

[illegible]

BACKGROUND

```
size (150, 150)
fill (0.63, 0.82, 0.94)
rect(0, 0, WIDTH, HEIGHT)
fill(1)
background(None)
```

DECORATION

```
fill (0.9)

for x, y in grid(5, 5, 20, 10):
    for i in range(6):

        x = random(WIDTH)
        y = random(HEIGHT)
        r = random(2, 10)

        path = oval(x, y, r, r)
```

DECORATION

```
stroke (0.1)
strokewidth (0.25)
font ("CallianceFlourishes")
for i in range(20):

    fontsize (random(6, 85))
    rotate (random(360))

    fiori = "a", "b", "c"
    p = textpath (choice(fiori),
                  random(WIDTH),
                  random(HEIGHT))

    drawpath(p)
```

DIALOGUE

```
tipo= "Arno Pro"
lettera= "R"
punti= 180
xLettera= 30
yLettera= 185

xCurve= 16
yCurve= 93
```

ILLUMINATION

```
reset()
cornu = ximport("cornu")

transform(CORNER)
translate(xCurve, yCurve)
scale(1.2)
font(tipo, 1.0)

path = textpath(lettera, 0, 0)

points = []
for pt in path:
    if pt.cmd == LINETO or pt.cmd == CURVETO:
        points.append((pt.x, pt.y))
        if random() > 0.8:
            points.append((pt.x+random(-0.05,0.05),
                           pt.y+random(-0.05,0.05)))

strokewidth(1.5)
nofill()
stroke(0)
cornu.drawpath(points)
```

ILLUMINATION

```
reset()
cornu = ximport("cornu")

transform(CORNER)
translate(xCurve, yCurve)
scale(1.2)
font(tipo, 1.0)

path = textpath(lettera, 0, 0)
points = []
for pt in path:
    if pt.cmd == LINETO or pt.cmd == CURVETO:
        points.append((pt.x, pt.y))
        if random() > 0.8:
            points.append((pt.x+random(-0.05,0.05),
                           pt.y+random(-0.05,0.05)))

strokewidth(0.7)
nofill()
stroke(1)
cornu.drawpath(points)
```

LETTER

```
reset()
rotate(random(0))
font (tipo)
fontsize (punti)
fill (1)
stroke (0)
strokewidth (1.5)
align(CENTER)
text(lettera, xLettera, yLettera, outline=True)
```

ILLUMINATION

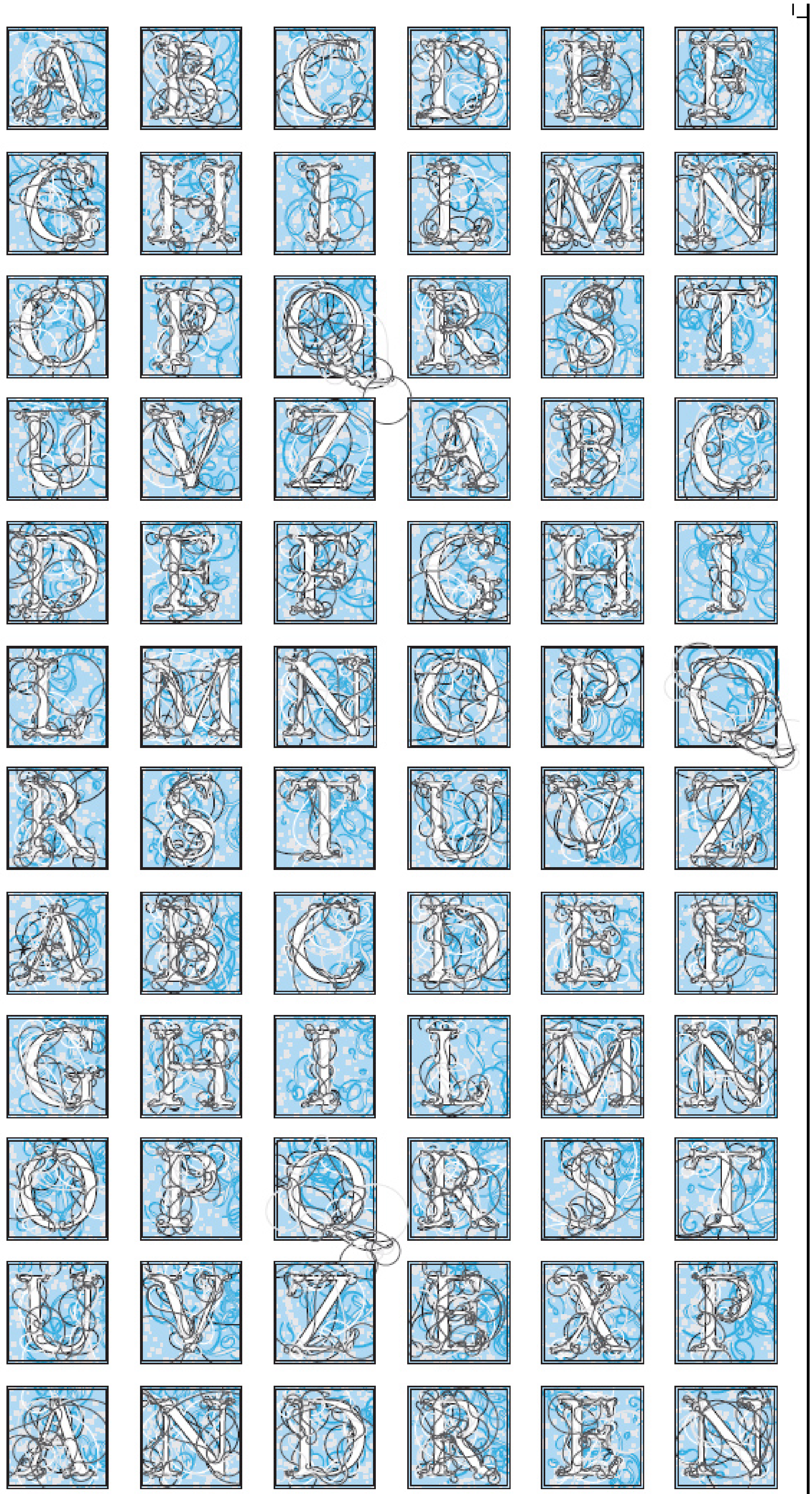
```
reset()
cornu = ximport("cornu")

transform(CORNER)
translate(xCurve, yCurve)
scale(1.2)
font(tipo, 1.0)

path = textpath(lettera, 0, 0)

points = []
for pt in path:
    if pt.cmd == LINETO or pt.cmd == CURVETO:
        points.append((pt.x, pt.y))
        if random() > 0.8:
            points.append((pt.x+random(-0.05,0.05),
                           pt.y+random(-0.05,0.05)))

strokewidth(1)
nofill()
stroke(0.9)
cornu.drawpath(points)
```

ILLUMINATION

```
reset()
cornu = ximport("cornu")

transform(CORNER)
translate(xCurve, yCurve)
scale(1.2)
font(tipo, 1.0)

path = textpath(lettera, 0, 0)

points = []
for pt in path:
    if pt.cmd == LINETO or pt.cmd == CURVETO:
        points.append((pt.x, pt.y))
        if random() > 0.8:
            points.append((pt.x+random(-0.05,0.05),
                           pt.y+random(-0.05,0.05)))

strokewidth(0.7)
nofill()
stroke(0.3)
cornu.drawpath(points)
```

BORDER

```
reset()
nofill()
stroke(0)
strokewidth(1.5)
rect(0, 0, WIDTH, HEIGHT)
stroke(0)
strokewidth(3)
scale(0.95)
rect(4, 4, WIDTH, HEIGHT)
```


A B C D E F
G H I L M N
O P Q R S T
U V Z A B C
D E F G H I
L M N O P Q
R S T U V Z
A B C D E F
G H I L M N
O P Q R S T
U V Z E X P
A N D R E N