

A Web-based IDE for Generative Design

...

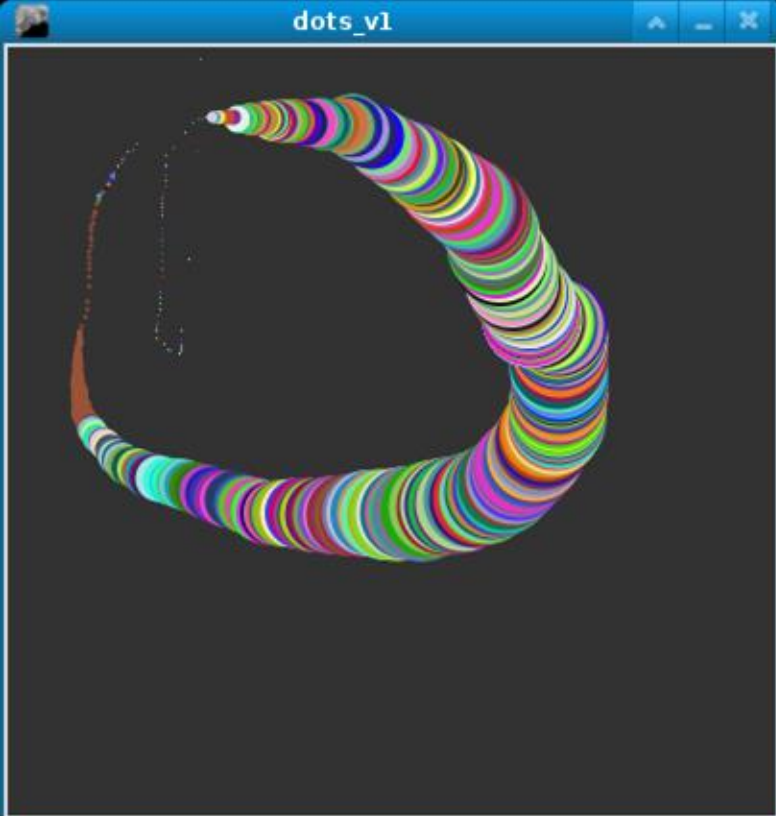
Pedro Alfaiate

pedro.alfaiate@tecnico.ulisboa.pt

Generative Design



IDEs for Generative Design



```
dots_v1 | Processing 2.0b5
File Edit Sketch Tools Help

dots_v1 §
// dots_v1.pde --

int strokeWeight = 20;           // Weight of drawing stroke
int bgcolor = 50;                // Color for canvas background

void setup() {                  // Executed once
  size(400, 400);               // Canvas size
  smooth();                    // Enable antialiasing
  background(bgcolor);         // Set background colour for canvas
                                // (RGB levels all the same, thus grey)
  strokeWeight(strokeWeight);  // Set thickness of stroke

  textSize(20);
  textAlign(CENTER);
  text("Click the mouse here to start", 200, 150);
  text("Space bar: clear canvas", 200, 200);
  text("Left/right arrows: change stroke size", 200, 250);
}

void draw() {                  // Executed continuously
  if (focused)                 // If we have mouse focus
    point(mouseX, mouseY);
}

void keyTyped() {             // Executed when a key is typed
  if (key == int(' '))        // Space bar clears the canvas
    background(bgcolor);
}


```



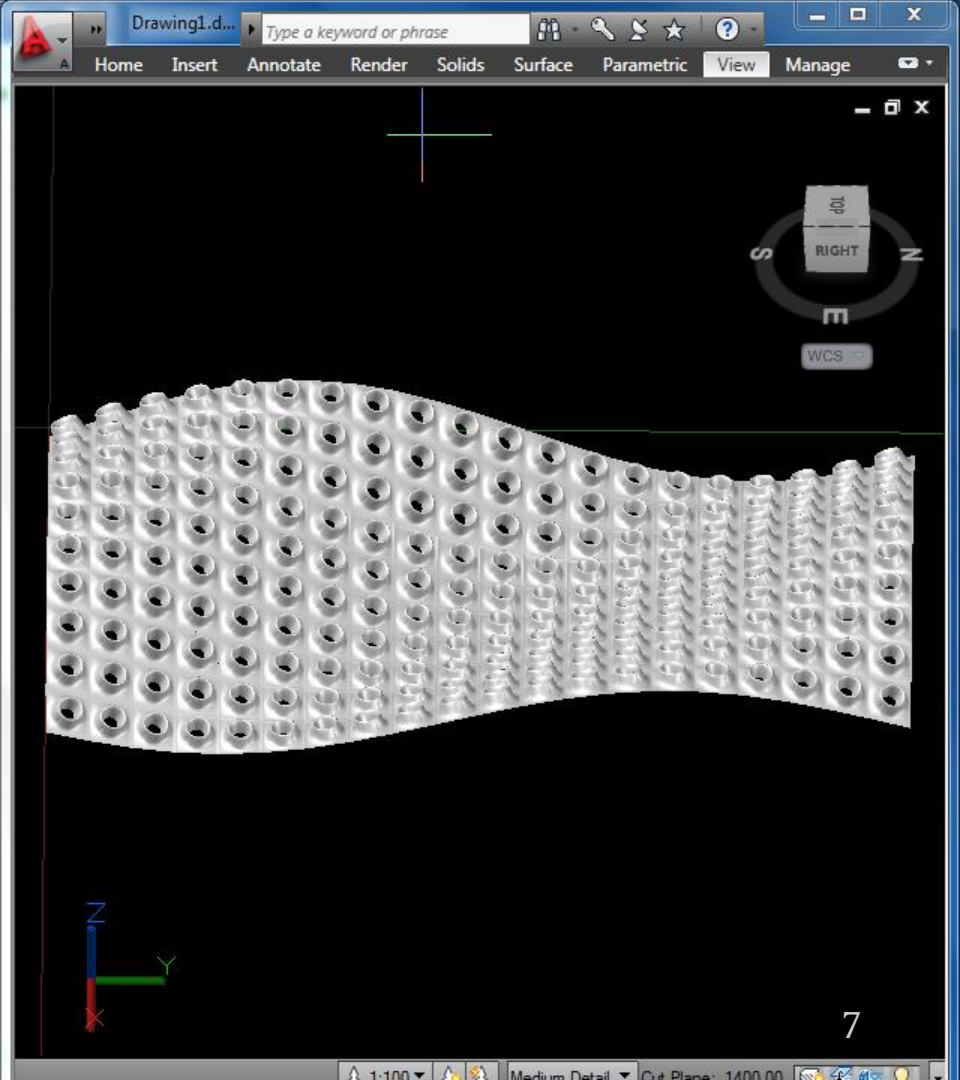
```
crater-tessellation.js - DrRacket
File Edit View Language JavaScript Insert Tabs Help
crater-tessellation.js Debug Check Syntax Run

function rectangleLine(p1, p2, p3, p4) {
  return join(spline(list(p1, p2)),
    spline(list(p2, p3)),
    spline(list(p3, p4)),
    spline(list(p4, p1)));
}

function crater(c, n, p1, p2, p3, p4, radius) {
  return loft(list(rectangleLine(p1, p2, p3, p4),
    move(addC(c, (multC(n, 0.1 * radius))),
      circleN(radius, n)),
    move(addC(c, (multC(n, 0.8 * radius))),
      circleN(0.7 * radius, n)),
    move(addC(c, (multC(n, -0.1 * radius))),
      circleN(0.4 * radius, n))));
}

function wavySkin() {
  return makeSimpleSkin(
    function(u, v) {
      return xyz(u,
        v,
        0.4*sin(u+v)+0.1*abs(u-1)*sin(v-1)); },
    makeUvDomain('closed', 0, 'closed', 3,
      'closed', 0, 'closed', 6));
}

function craterTessellation(skin, nU, nV) {
  return map(function(patch) {
    var p1 = patchObject(patch, uFnMin(), vFnMin());
    var p2 = patchObject(patch, uFnMax(), vFnMin());
    var p3 = patchObject(patch, uFnMax(), vFnMax());
    var p4 = patchObject(patch, uFnMin(), vFnMax());
    return crater(patchObject(patch, uFnHalf(), uFnHalf()),
      patchNormal(patch, uFnHalf(), uFnHalf()),
      p1, p2, p3, p4,
      0.4 * min(xyzR(subC(p1, p2)), xyzR(subC(p1, p4))))
  });
}
```



IDE in the Web



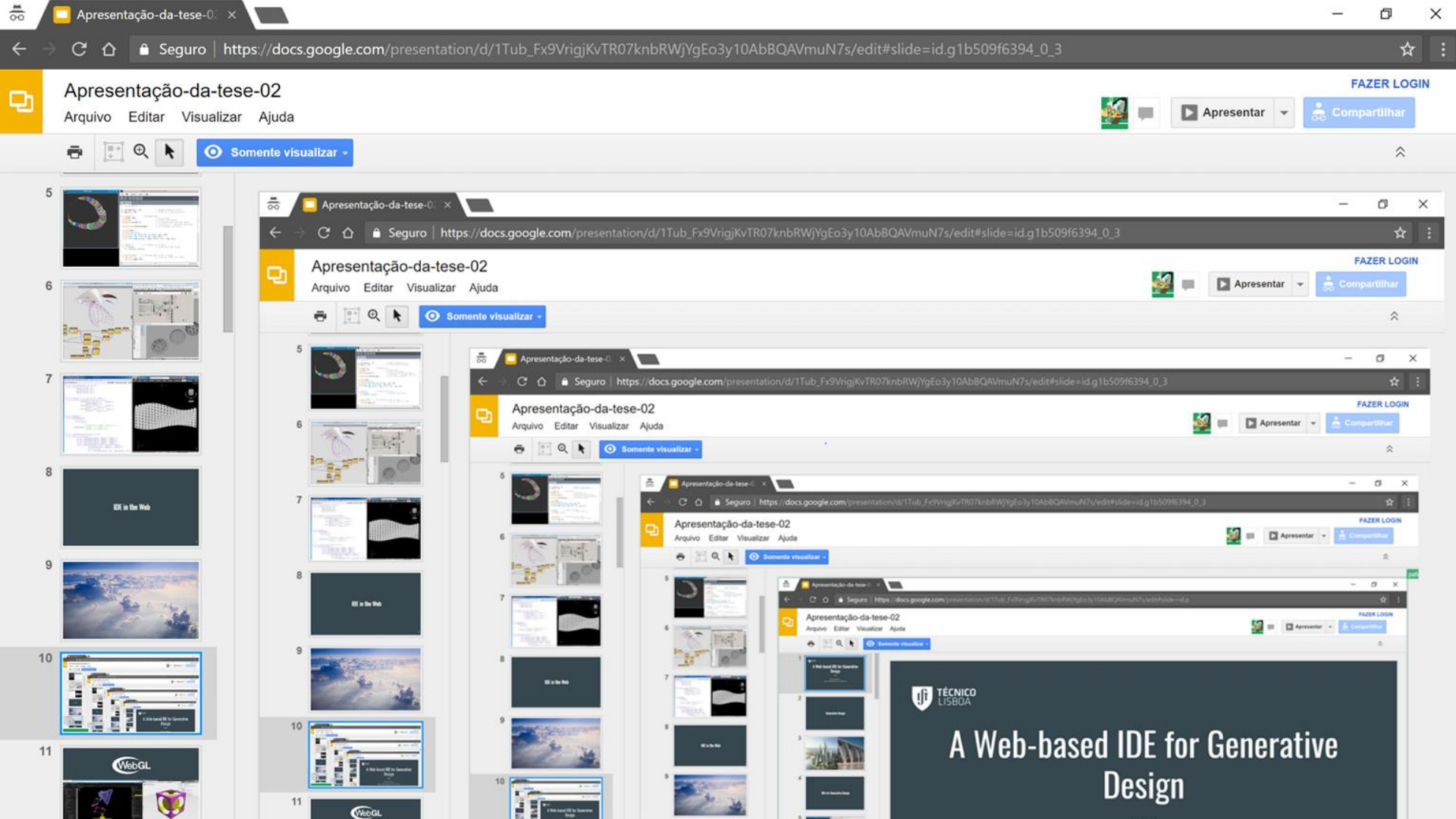
- 1 A Web-based IDE for Generative Design
- 2 Generative Design
- 3
- 4 IDEs for Generative Design
- 5
- 6
- 7



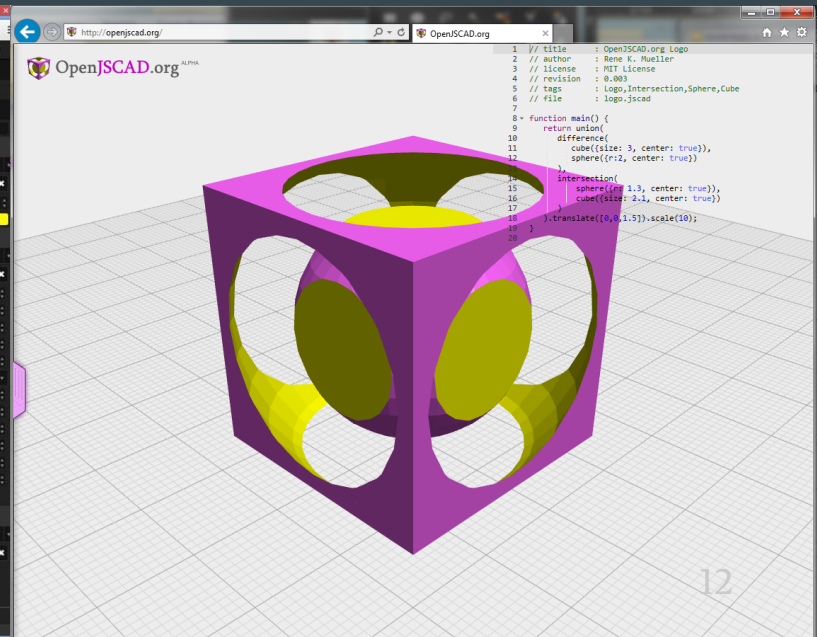
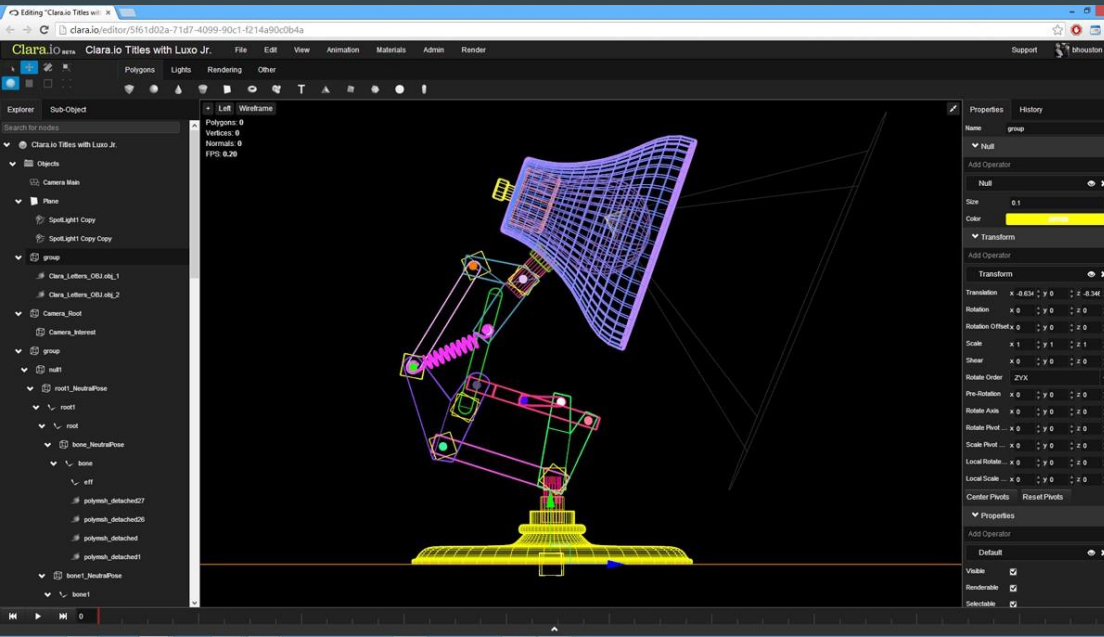
A Web-based IDE for Generative Design

...

Pedro Alfaiate
pedro.alfaiate@tecnico.ulisboa.pt

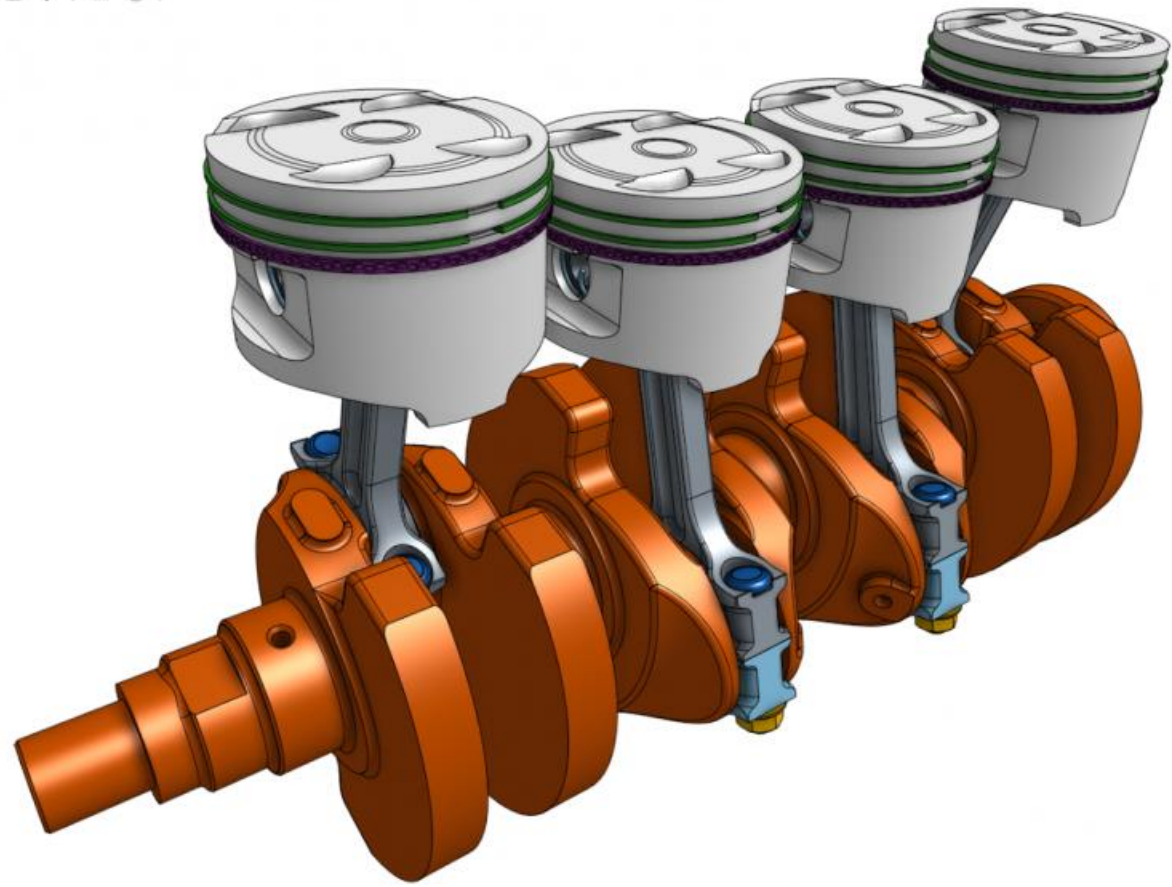


WebGL

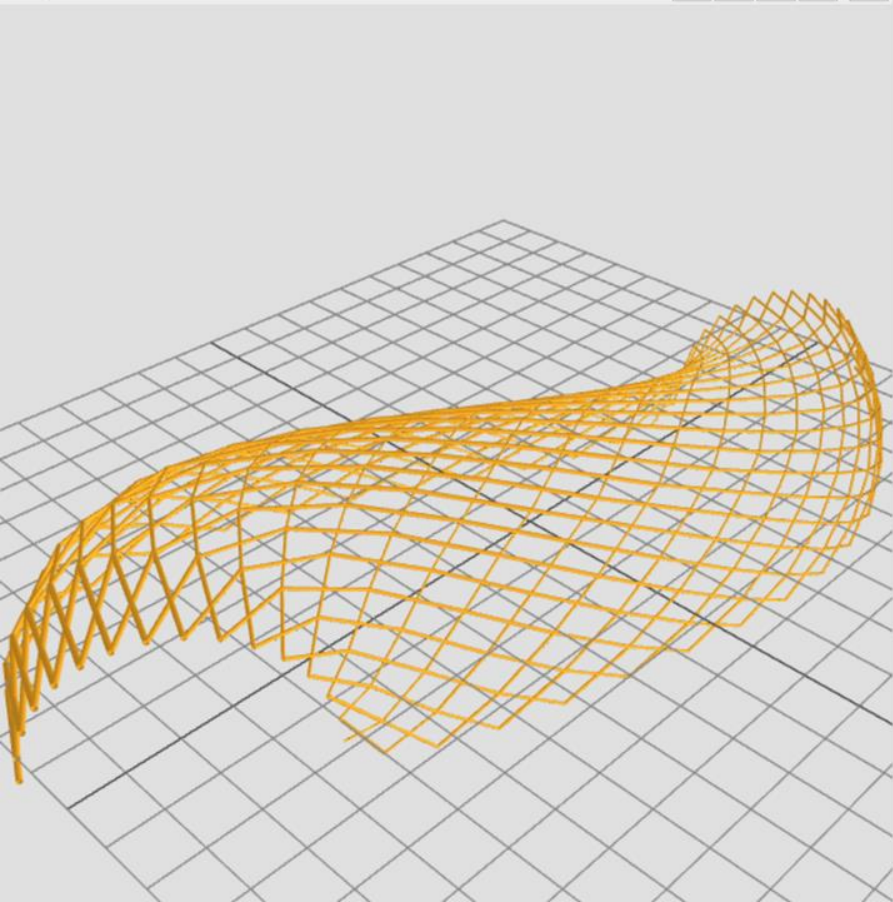




- Instances (63)
- Origin
 - Main_Cap <1>
 - Main_Cap <2>
 - Main_Cap <3>
 - Main_Cap <4>
 - Main_Cap <5>
 - Head <1>
 - crankshaft <1>
 - con_rod <1>
 - con_rod_bearing <1>
 - con_rod_cap <1>
 - con_rod_nut <1>
 - con_rod_nut <2>
 - con_rod_stud <1>
 - con_rod_stud <2>
 - > piston_pin <1>
 - con_rod <2>
 - con_rod_bearing <2>
 - con_rod_cap <2>
 - con_rod_nut <3>
 - con_rod_nut <4>
 - con_rod_stud <3>
 - con_rod_stud <4>
 - > piston_pin <2>



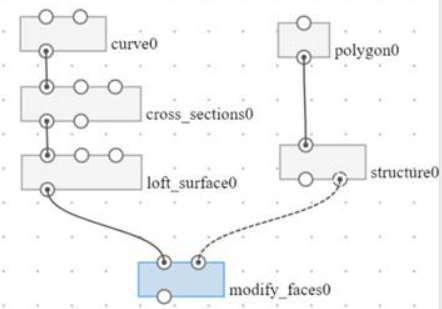
Perspective View



Parameters

This node needs no inputs

Graph



Console

16:24:25:Example loaded

Node Designer

Input: mesh option: none

Input: func option: none

modify_faces *

Output envelope = []

```

for each face in mesh.faces
  lst.extend ( envelope ,
              func( face ).structure )

```

Toolkit

Node Design

- Add Input
- Add Output
- Add Variable

Controls

- for each
- if else

Functions

Search ...

- msc
- expression
- degToRad
- radToDeg
- sin
- cos
- sigDig
- print
- frm

byXYPoint 14

byXZPoints

Problem

Requirements

No installation or updates

Help the programming process

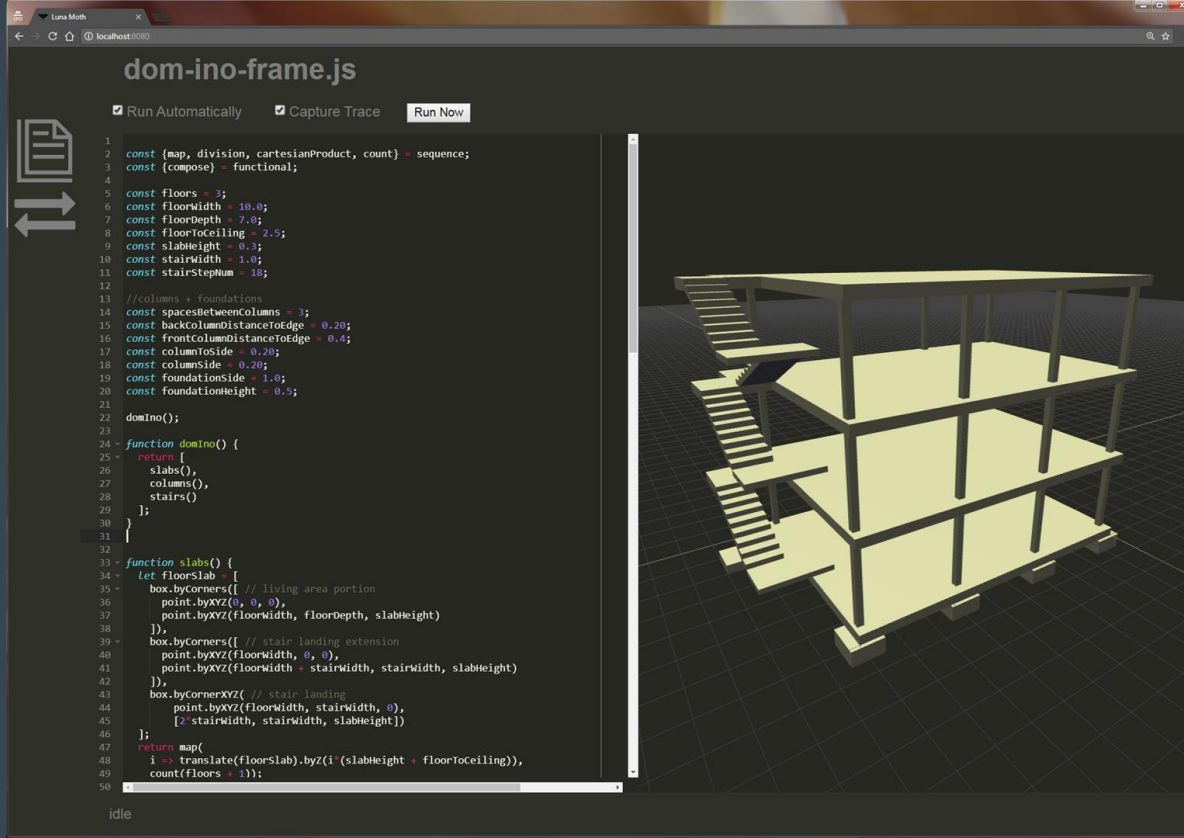
- Traceability

- Immediate feedback

- Intuitive parameter adjustment

Integrate into the architect's workflow

Luna Moth



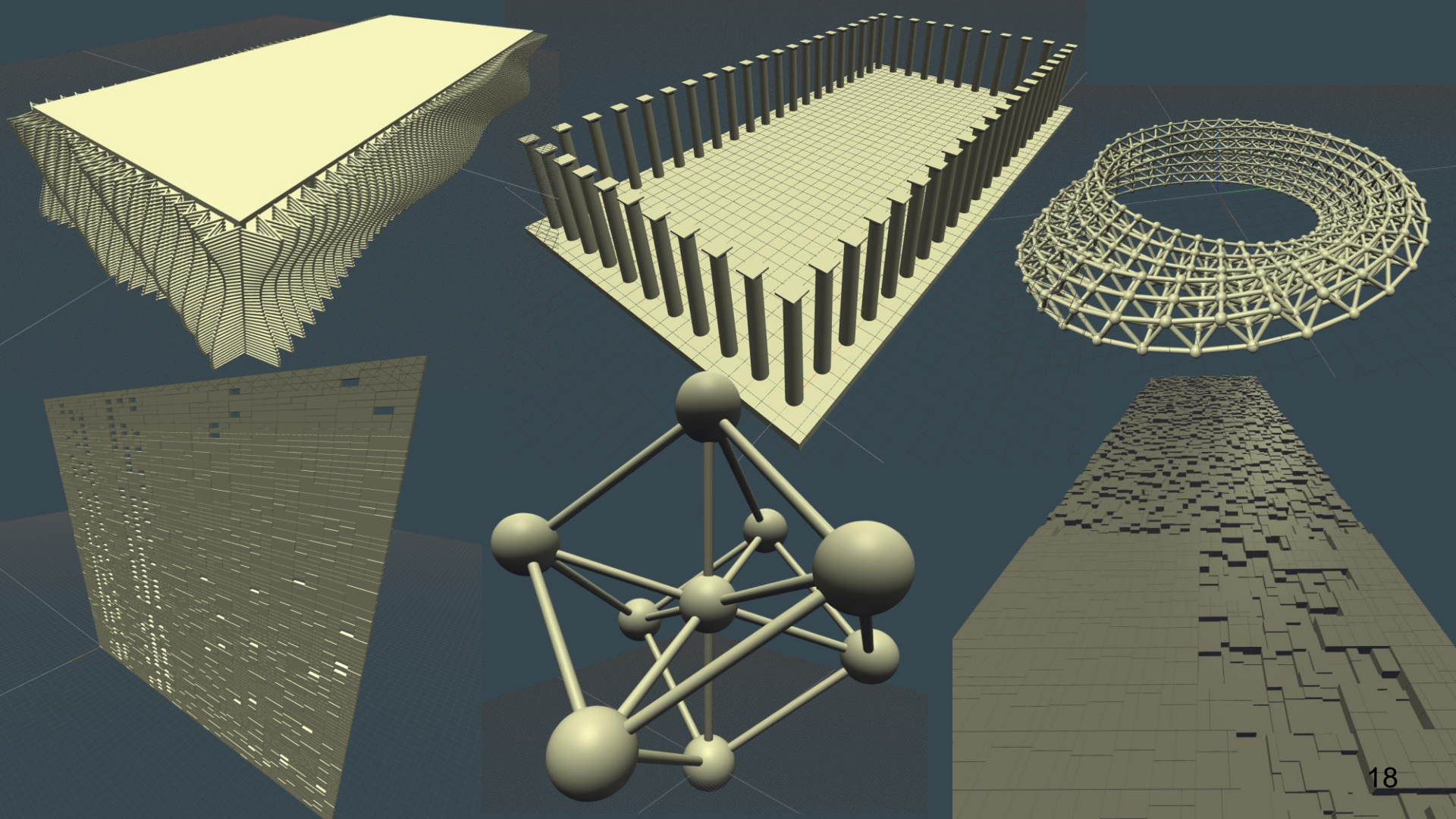
The screenshot shows a web browser window with the title "Luna Moth" and the URL "localhost:3080". The page displays a 3D architectural visualization of a three-story building structure. The browser window is split into two panes: a code editor on the left and a 3D view on the right.

The code editor shows the following JavaScript code:

```
1
2 const {map, division, cartesianProduct, count} = sequence;
3 const {compose} = functional;
4
5 const floors = 3;
6 const floorWidth = 10.0;
7 const floorDepth = 7.0;
8 const floorToCeiling = 2.5;
9 const slabHeight = 0.3;
10 const stairWidth = 1.0;
11 const stairStepNum = 18;
12
13 //columns + foundations
14 const spacesBetweenColumns = 3;
15 const backColumnDistanceToEdge = 0.20;
16 const frontColumnDistanceToEdge = 0.4;
17 const columnToSide = 0.20;
18 const columnSide = 0.20;
19 const foundationSide = 1.0;
20 const foundationHeight = 0.5;
21
22 domIno();
23
24 - function domIno() {
25 -   return [
26     slabs(),
27     columns(),
28     stairs()
29   ];
30 }
31 |
32
33 - function slabs() {
34 -   let floorSlab = [
35 -     box.byCorners([ // living area portion
36       point.byXYZ(0, 0, 0),
37       point.byXYZ(floorWidth, floorDepth, slabHeight)
38     ]),
39     box.byCorners([ // stair landing extension
40       point.byXYZ(floorWidth, 0, 0),
41       point.byXYZ(floorWidth + stairWidth, stairWidth, slabHeight)
42     ]),
43     box.byCornerXYZ( // stair landing
44       point.byXYZ(floorWidth, stairWidth, 0),
45       [2*stairWidth, stairWidth, slabHeight])
46   ];
47   return map(
48     i => translate(floorSlab).byZ(i*(slabHeight + floorToCeiling)),
49     count(floors + 1));
50 }
```

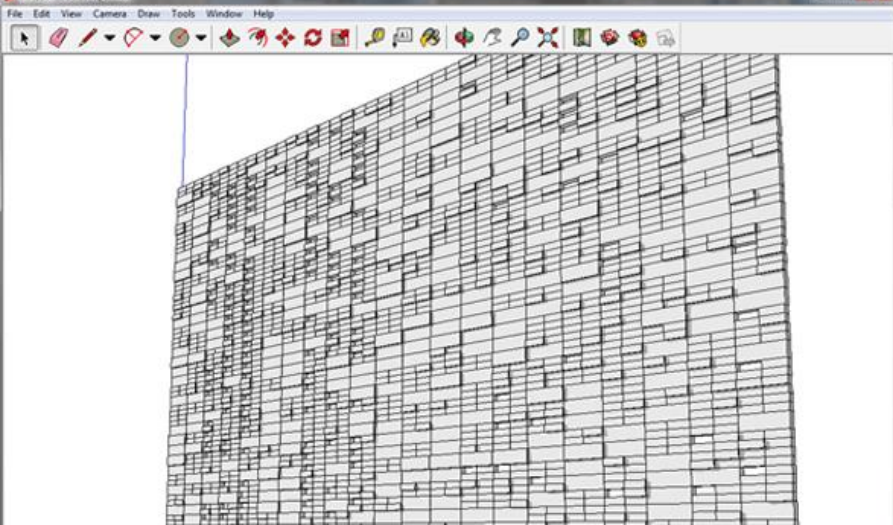
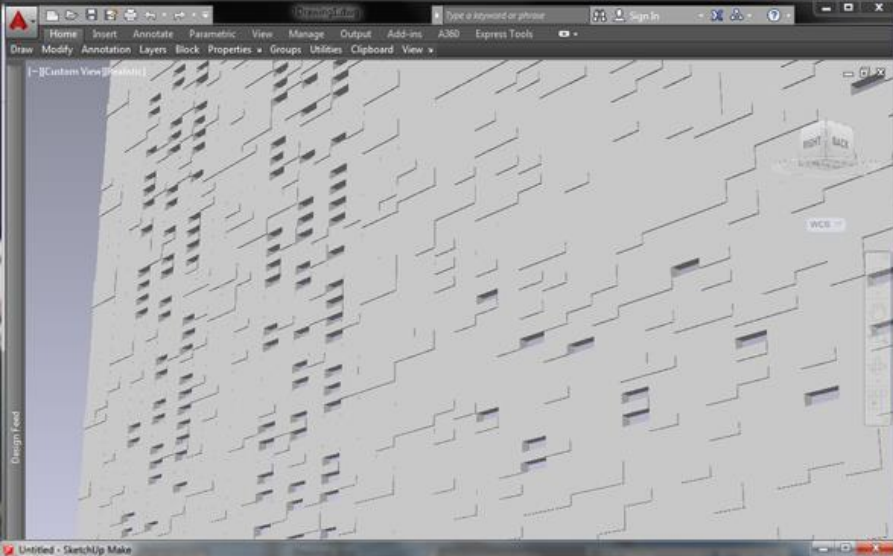
The 3D view shows a three-story building structure with a grid of columns and slabs. The building is rendered in a light yellow color. The stairs are located on the left side of the building. The background is a dark gray grid.

idle



Luna Moth - Demo

Workflow Integration



Luna Moth

localhost:8080

ines-wall.js

Run Automatically Capture Trace

```

1
2 //pedidas convertidas no REVIT
3 const cRevit = 1.93333;
4 const hRevit = 0.66666;
5 const eRevit = 0.322;
6
7 function rightCuboid(bottom, width, height, z) {
8   let p = brickMovement(bottom);
9   let c1 = point.add(p, vector.byXY(width / 2, height / 2 + 0.01));
10  let c2 = point.add(p, vector.byXYZ(width / 2, height / 2 + 0.01, z - 0.01));
11  return box.byCorners([c1, c2]);
12 }
13
14
15 const j = 0;
16 const salienciaRevit = 0.002;
17 function brickMovement(pt) {
18   return random.Integer(4) == 0
19     ? point.add(pt, vector.byX(salienciaRevit))
20     : pt;
21 }
22
23 //FUNCAO QUE CRIA QUATRO TIPOLOS PEQUENOS NO ESPACO DE 1 TIPOLO GRANDE (3x1x1cm)
24 function brick4(p, lado, alt, e) {
25   const lado2 = (lado - j) / 2.0;
26   const alt2 = (alt - j) / 2.0;
27   const pp = point.add(p, vector.byXY(c / 2.0, lado2 / 2.0 + j));
28   const p1 = point.add(pp, vector.byZ(lado2));
29   const p2 = point.add(pp, vector.byZ(alt2));
30   const p3 = point.add(pp, vector.byYZ(lado2, alt2));
31   return [
32     rightCuboid(pp, e, lado2, alt2),
33     rightCuboid(p1, e, lado2, alt2),
34     rightCuboid(p2, e, lado2, alt2),
35     rightCuboid(p3, e, lado2, alt2)
36   ];
37 }
38
39 //FUNCOES QUE CRIAM SO TRES TIPOLOS PEQUENOS NO ESPACO DE UM GRANDE
40 function brick3(p, lado, alt, e) {
41   const lado2 = (lado - j) / 2.0;
42   const alt2 = (alt - j) / 2.0;
43   const pp = point.add(p, vector.byXY(c / 2.0, lado2 / 2.0 + j));
44   const p1 = point.add(pp, vector.byZ(lado2));
45   //const p2 = point.add(pp, vector.byZ(alt2));
46   const p3 = point.add(pp, vector.byYZ(lado2, alt2));
47   return [
48     rightCuboid(pp, e, lado2, alt2),
49     rightCuboid(p1, e, lado2, alt2),
50     rightCuboid(p3, e, lado2, alt2)
51   ];
52 }
53
54 //FACHADA NORTE
55 //DIMENSÕES
56 const length = 12.0;
57 const height = 10.8;
58 //DIMENSÕES BRICKS
59 const cBrick = 0.62; //largura dos tijolos grandes
60 const hBrick = 0.25; //altura dos tijolos grandes
61 const eBrick = 0.03; //espessura dos tijolos
62 const nBricks = 21; //numero em x
63 const mBricks = 54; //numero em z
64 //FATORES DENSIDADE
65 const r50 = 2; //para as zonas de opacidade 100%
66

```

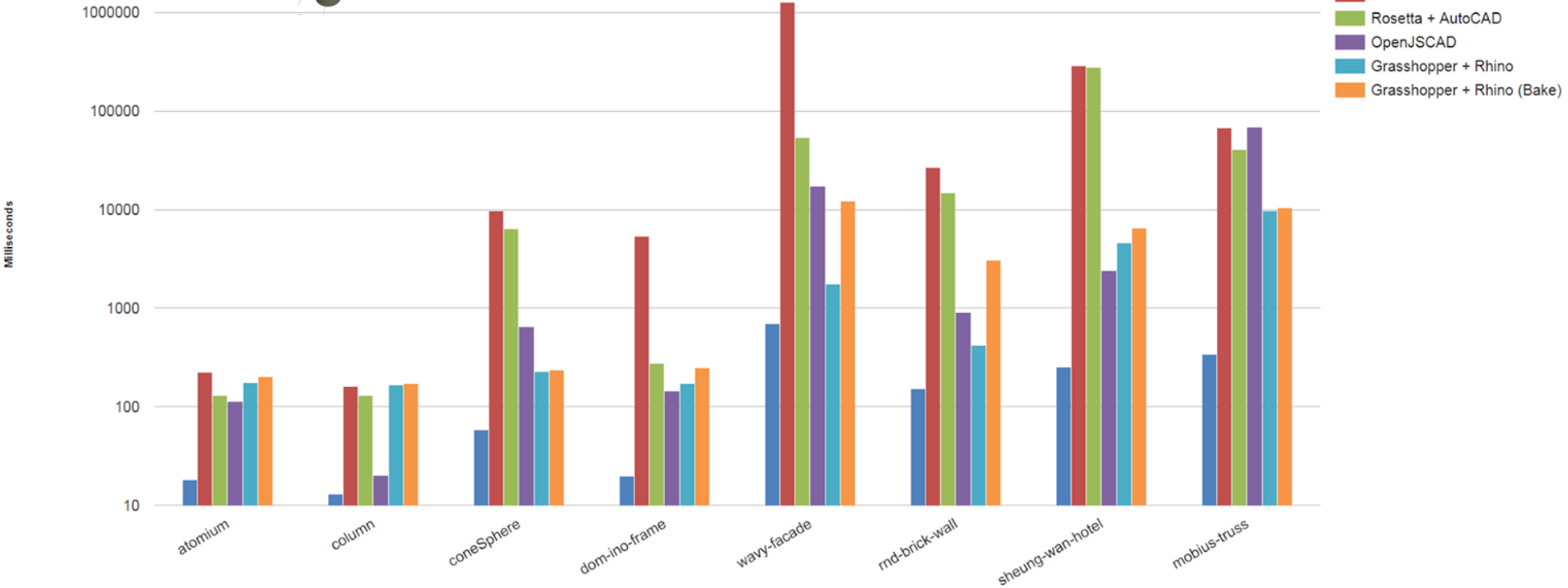
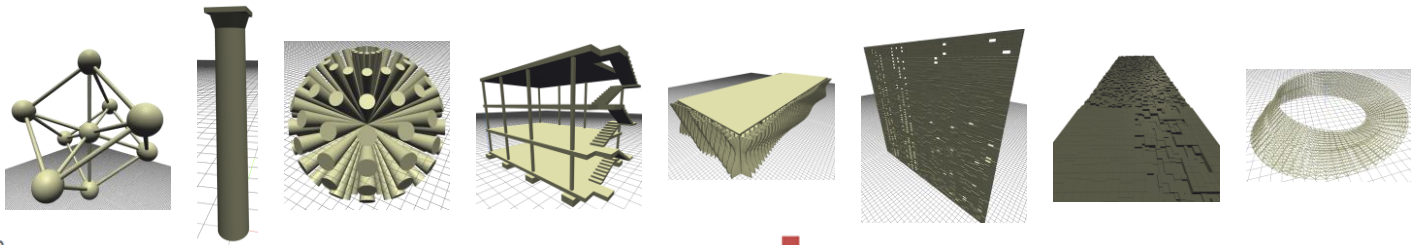

Performance Evaluation

Performance Evaluation

Examples that can be generated by programs

Compared with other IDEs

Grasshopper, Rosetta, OpenJSCAD



Future work

Better integration

faster

more design tools

Better development experience

illustrated programming

timetable traceability

code completion

error reporting

Collaboration

Contributions

Luna Moth: A Web-based Programming Environment for Generative Design

35th eCAADe conference: Sharing of Computable Knowledge (accepted)

Luna Moth: Supporting Creativity in the Cloud

37th ACADIA conference: Disciplines & Disruption (submitted)

Thank you

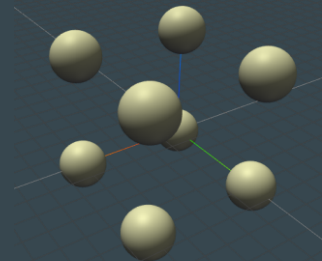
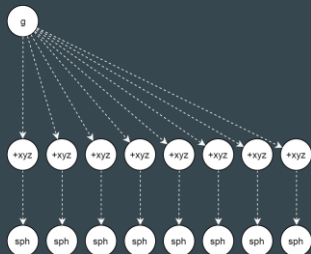
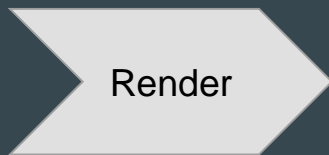
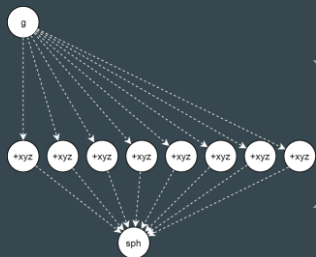
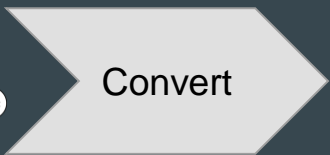
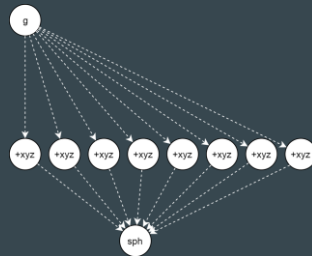
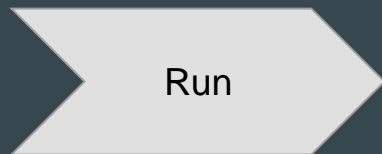


Implementation

Rendering

Rendering

```
function sphereCube(w, sphR) {  
  let sph = sphere.byRadius(sphR);  
  return [  
    translate(sph).byXYZ(0, 0, 0),  
    translate(sph).byXYZ(w, 0, 0),  
    translate(sph).byXYZ(w, w, 0),  
    translate(sph).byXYZ(0, w, 0),  
  ];  
}  
let g = sphereCube(10, 1);  
g;
```



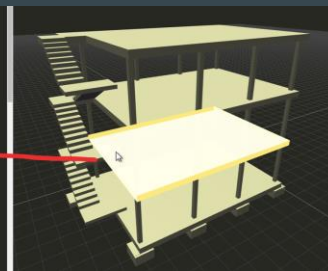
Results

Three.js Scene

Image

Traceability

```
20 const foundationHeight = 0.5;
21
22 domIncl();
23
24 function domIncl() {
25   slabs(),
26   columns(),
27   stairs();
28 };
29
30
31 function slabs() {
32   let floorSlab = [
33     new byCorner([
34       point.byXYZ(0, 0, 0),
35       point.byXYZ(floorsWidth, floorDepth, slabHeight)
36     ]),
37     new byCorner([ // stair landing extension
38       point.byXYZ(floorsWidth, 0, 0),
39       point.byXYZ(floorsWidth + stairWidth, stairWidth, slabHeight)
40     ]),
41     new byCornerXYZ( // stair landing
42       point.byXYZ(floorsWidth, stairWidth, 0),
43       [ [stairWidth, stairWidth, slabHeight] ]
44     );
45   ];
46   return map(
47     i => translate(floorSlab).byZ(i * (slabHeight - floorToCeiling)),
48     count(floors - 1));
49 }
50 }
```



Camera Ray

Rendered Object

Result Value

AST Node

Highlight

Text Coords

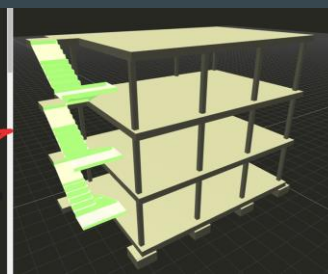
AST Node

Expression Results

Rendered Objects

Highlight

```
1 const floorsWidth = 7.0;
2 const floorDepth = 7.0;
3 const floorToCeiling = 2.5;
4 const slabHeight = 0.5;
5 const stairWidth = 1.0;
6 const stairStepUp = 2.0;
7
8 function domIncl() {
9   slabs(),
10  columns(),
11  stairs();
12 };
13
14 const spacesBetweenColumns = 3;
15 const backColumnDistanceToEdge = 0.5;
16 const frontColumnDistanceToEdge = 0.5;
17 const columnToSide = 0.5;
18 const columnSide = 0.25;
19 const foundationSide = 1.0;
20 const foundationHeight = 0.5;
21
22 domIncl();
23
24 function domIncl() {
25   slabs(),
26   columns(),
27   stairs();
28 };
29
30
31 function slabs() {
32   let floorSlab = [
33     new byCorner([ // living area portion
34       point.byXYZ(0, 0, 0),
35       point.byXYZ(floorsWidth, floorDepth, slabHeight)
36     ]),
37     new byCorner([ // stair landing extension
38       point.byXYZ(floorsWidth, 0, 0),
39       point.byXYZ(floorsWidth + stairWidth, stairWidth, slabHeight)
40     ]),
41     new byCornerXYZ( // stair landing
42       point.byXYZ(floorsWidth, stairWidth, 0),
43       [ [stairWidth, stairWidth, slabHeight] ]
44     );
45   ];
46   return map(
47     i => translate(floorSlab).byZ(i * (slabHeight - floorToCeiling)),
48     count(floors - 1));
49 }
50 }
```

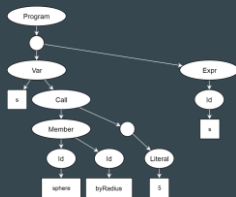
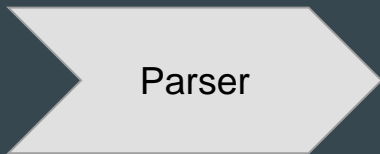


Running for Traceability

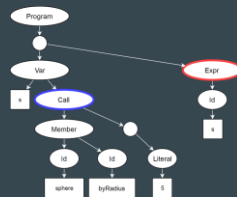
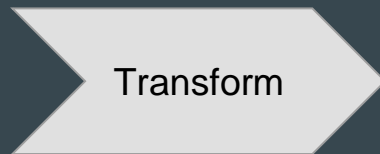
Running for Traceability

```
function sphereCube(w, sphR) {  
  let sph = sphere.byRadius(sphR);  
  return [  
    translate(sph).byXYZ(0, 0, 0),  
    translate(sph).byXYZ(w, 0, 0),  
    translate(sph).byXYZ(w, w, 0),  
    translate(sph).byXYZ(0, w, 0),  
  ];  
}  
let g = sphereCube(10, 1);  
g;
```

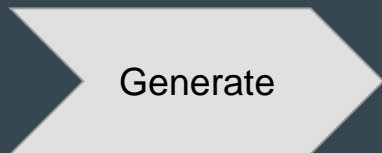
Program



AST

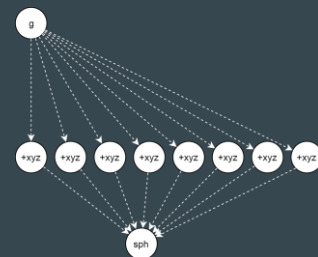
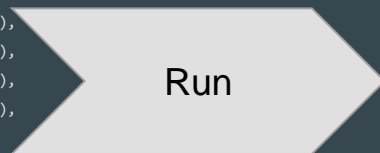


Transformed
AST



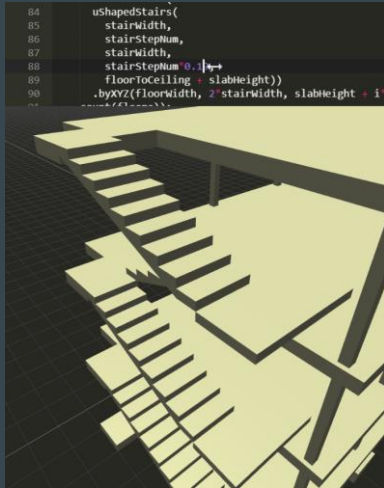
```
function sphereCube(w, sphR) {  
  let sph = sphere.byRadius(sphR);  
  return [  
    translate(sph).byXYZ(0, 0, 0),  
    translate(sph).byXYZ(w, 0, 0),  
    translate(sph).byXYZ(w, w, 0),  
    translate(sph).byXYZ(0, w, 0),  
  ];  
}  
let g = sphereCube(10, 1);  
g;
```

Instrumented
Program



Results
Expr -> Results

Literal Adjustment



0.19 -> 0 19 -> 019 -> +6 025 -> 0.25

str -> str -> int -> int -> str



Workflow Integration



run without traceability -> call operations on Rosetta Remote Service (as in RPC)

