The CyberAntHill

by Evan Raskob <e.raskob@gold.ac.uk>

The CyberAnthill is both a generative sculpture and a Live Computational Sculpting (LCS) system that uses a 3D printer and custom software to build plastic sculptures out of layered cellular automata. As the title alludes to, the cellular automata are inspired by Langston's Ant and the light cycle racers in the cult 1980's science-fiction movie Tron. Instead of the normal process of printing exacting, predetermined 3D models, the 3D printer generates its plastic forms by running unpredictable computer code, shown below. Lines of code are highlighted and then executed on demand, part of a process called *livecoding*.

https://github.com/pixelpusher/liveprinter

```
// The CyberAntHill
// By Evan Raskob < info@pixelist.info>
// <e.raskob@gold.ac.uk>
// Copyright 2019, all rights reserved

attachScript("the-cyber-anthill/ant.js"); // run this line  first by itself
attachScript("the-cyber-anthill/grid.js"); // run this line second
attachScript("the-cyber-anthill/antgrid-api.js"); // run  this line 3rd

lp.start(190); // start up printer, home axes, set print  head
        temperature and bed temp

lp.moveto({ x: lp.cx + 10, y: lp.miny + 32 })
lp.speed(50);

lp.sync()

const minZ = 4;
lp.moveto({ z: minZ })

lp.down(0.1).go();
lp.extrude({ e: 20, speed: 8 })

lp.retract(9)
lp.extrude({ e: 9, speed: 20 })

lp.downto(5).go()

lp.fan(0)
lp.bed(40);

lp.upto(3.2).go()

lp.moveto({ x: lp.cx, y: lp.cy })
lp.turnto(180).dist(20).go()
lp.dist(10).go()
lp.moveto({ x: 171.5, y: 12, z: 3.6 })
// 171.5, 12, 3.6
//x 30, y 188 min, 220 max -- overhang is 21mm
lp.minx = 41;
lp.maxy = 188;
lp.miny = 15;
loginfo(lp.cx)

lp.moveto({ x: lp.cx + lp.minx })
lp.moveto({ x: lp.maxx })
lp.speed(14); // set print speed to a bit slower, 35 for  higher layers

lp.lh(0.1); //.2, 0.15, .1
```

```
lp.fan(60)

// 10,16,22

let dims = 22; // dimensions of underlying grid -  careful not to
        make too big!
let grid = new Grid(dims, dims * 2); // grid for ants  to fill up
let ant = null; // our current ant walker
let paths = []; // array of paths walked
let layers = 3; // layers per ant trail
let iteration = 0; // how many time's we've run this  before (start at 0)
const w = lp.maxx - lp.minx;
const h = lp.maxy - lp.miny;
const sectW = w / 3;
const gridH = 2 * h / 3;

const gridW = sectW / 2;
const offsetx = sectW / 4; // 1/3 of 1/4
const offsety = h / 6;

const index = 2; //0,1,2 for all 3
const startx = lp.minx + offsetx + index *
        (2 * offsetx + gridW);
const starty = lp.miny + offsety;
const z = 3 + lp.layerHeight;

while (ant = createAnt(grid)) {
  if (ant) {
    ant.maxLife = 36;
    while (ant.alive) {
      move(ant, grid);
    }
    paths.push(ant.path);
  }
}

//lp.unretract();
//print all paths
lp.printPaths({ paths: paths, x: startx, y: starty, z: z +
        lp.layerHeight * layers
        * iteration, w: gridW, h: gridH, passes: layers });

lp.up(100).go(); // move up at end
```