

```
//Quilt Pattern - v3
//Lucian Novosel - 2014
//Open sourcerer on the internet of things
// WWW: luciannovosel.com GH: github.com/luciannovo

//define the amount of cubes per the grid
int vertCubes = 10;
int horzCubes = vertCubes;
int cubeCount = vertCubes * horzCubes;
//define the relative margin between each cube
float marginPercent = 1 - .2;
float horzSpacingInc;
float vertSpacingInc;

int frameCounts = 10;
int frameOffset = 0;

float[] heightVert = new float[vertCubes * horzCubes];
float[] widthVert = new float[vertCubes * horzCubes];
float[] randomFrame = new float[vertCubes * horzCubes * frameCounts];

PVector[] cubePositionArray = new PVector[vertCubes * horzCubes];
PVector[] cubeTargetArray = new PVector[vertCubes * horzCubes];

void setup() {
    size(500, 500, P3D);
    smooth();
    strokeWeight(2.2);
    noStroke();
    fill(255, 100);

    horzSpacingInc = height/vertCubes;
    vertSpacingInc = width/horzCubes;

    ortho(0, width, 0, height, -2000, 2000);
    background(0);

    for(int i=0; i<cubeCount; i++){
        heightVert[i] = random((horzSpacingInc * marginPercent), (horzSpacingInc *
        widthVert[i] = random((horzSpacingInc * marginPercent), (horzSpacingInc *

    }
}
```

```
for(int i=0; i < vertCubes; i++){
    for(int j=0; j < horzCubes; j++){
        pushMatrix();
        translate(horzSpacingInc/2+ horzSpacingInc * i, vertSpacingInc/2 + ve
        rotateX(-PI/6);
        rotateY(PI/4);
        box(widthVert[(j+(10*i))],heightVert[(j+(10*i))],widthVert[(j+(10*i))
        popMatrix();
    }
}
String imgPath = ("../..//generatedImagesEX3/" + frame.getTitle() + ".png");
saveFrame(imgPath);
}
```