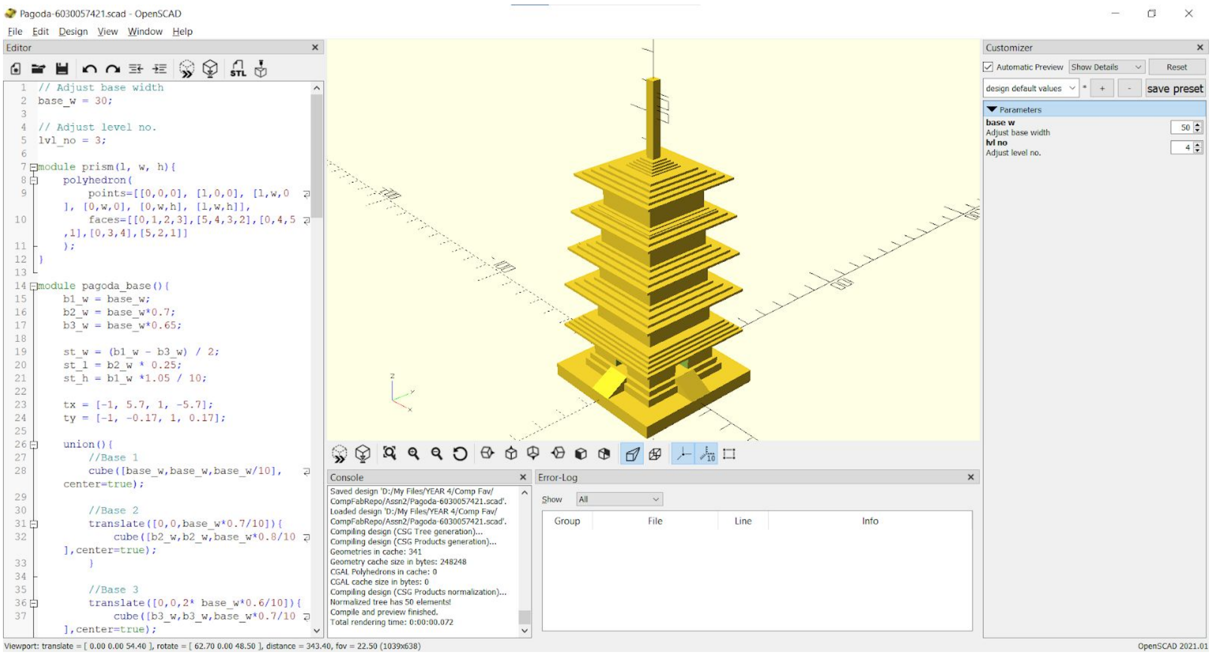
Kueakun Liwcharoenchai 6030057421

Act 2 - openSCAD

Selected model - **Pagoda**

Reference pics:

Results:



```
// Adjust base width
base_w = 30;

// Adjust level no.
lvl_no = 3;

module prism(l, w, h){
    polyhedron(
        points=[[0,0,0], [l,0,0], [l,w,0], [0,w,0], [0,w,h], [l,w,h]],
        faces=[[0,1,2,3],[5,4,3,2],[0,4,5,1],[0,3,4],[5,2,1]]
    );
}

module pagoda_base(){
    b1_w = base_w;
    b2_w = base_w*0.7;
    b3_w = base_w*0.65;

    st_w = (b1_w - b3_w) / 2;
    st_l = b2_w * 0.25;
    st_h = b1_w *1.05 / 10;

    tx = [-1, 5.7, 1, -5.7];
    ty = [-1, -0.17, 1, 0.17];

    union(){
        //Base 1
        cube([base_w,base_w,base_w/10], center=true);

        //Base 2
        translate([0,0,base_w*0.7/10]){
            cube([b2_w,b2_w,base_w*0.8/10],center=true);
        }

        //Base 3
        translate([0,0,2* base_w*0.6/10]){
            cube([b3_w,b3_w,base_w*0.7/10],center=true);
```

```
Saved design 'D:/My Files/YEAR 4/Comp Fav/CompFabRepo/Assn2/Pagoda-6030057421.scad'.
Loaded design 'D:/My Files/YEAR 4/Comp Fav/CompFabRepo/Assn2/Pagoda-6030057421.scad'.
Compiling design (CSG Tree generation)...
Compiling design (CSG Products generation)...
Geometries in cache: 341
Geometry cache size in bytes: 248248
CGAL Polyhedrons in cache: 0
CGAL cache size in bytes: 0
Compiling design (CSG Products normalization)...
Normalized tree has 50 elements!
Compile and preview finished.
Total rendering time: 0:00:00.072
```

Show: All

Group	File	Line	Info

☑ Automatic Preview | Show Details | Reset

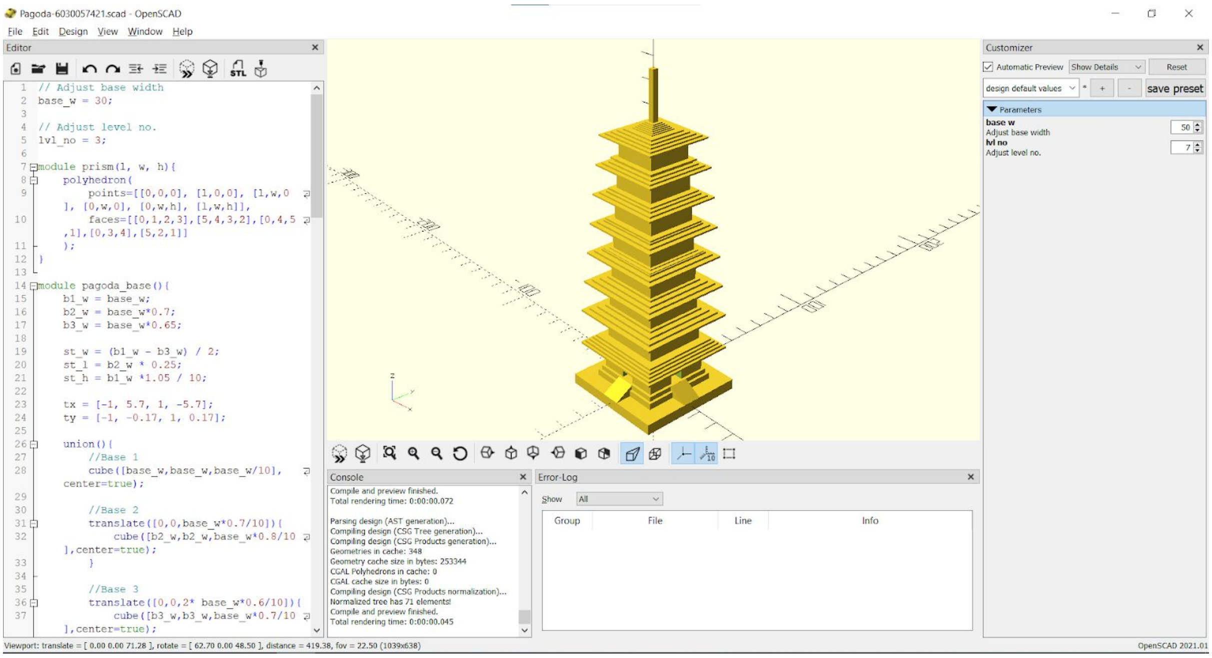
design default values * | + | - | save preset

base w
Adjust base width: 50

lvl no
Adjust level no.: 4

Viewport: translate = [0.00 0.00 54.40], rotate = [62.70 0.00 48.50], distance = 343.40, fov = 22.50 (1039x638)

OpenSCAD 2021.01



```
// Adjust base width
base_w = 30;

// Adjust level no.
lvl_no = 3;

module prism(l, w, h){
    polyhedron(
        points=[[0,0,0], [l,0,0], [l,w,0], [0,w,0], [0,w,h], [l,w,h]],
        faces=[[0,1,2,3],[5,4,3,2],[0,4,5,1],[0,3,4],[5,2,1]]
    );
}

module pagoda_base(){
    b1_w = base_w;
    b2_w = base_w*0.7;
    b3_w = base_w*0.65;

    st_w = (b1_w - b3_w) / 2;
    st_l = b2_w * 0.25;
    st_h = b1_w *1.05 / 10;

    tx = [-1, 5.7, 1, -5.7];
    ty = [-1, -0.17, 1, 0.17];

    union(){
        //Base 1
        cube([base_w,base_w,base_w/10], center=true);

        //Base 2
        translate([0,0,base_w*0.7/10]){
            cube([b2_w,b2_w,base_w*0.8/10],center=true);
        }

        //Base 3
        translate([0,0,2* base_w*0.6/10]){
            cube([b3_w,b3_w,base_w*0.7/10],center=true);
```

```
Compile and preview finished.
Total rendering time: 0:00:00.072

Parsing design (AST generation)...
Compiling design (CSG Tree generation)...
Compiling design (CSG Products generation)...
Geometries in cache: 348
Geometry cache size in bytes: 253344
CGAL Polyhedrons in cache: 0
CGAL cache size in bytes: 0
Compiling design (CSG Products normalization)...
Normalized tree has 71 elements!
Compile and preview finished.
Total rendering time: 0:00:00.045
```

☑ Automatic Preview

base w
Adjust base width 50

lvl no
Adjust level no. 7

Viewport: translate = [0.00 0.00 71.28], rotate = [62.70 0.00 48.50], distance = 419.38, fov = 22.50 (1039x638)

Adjustable Parameters:

- base_w (base width) → Control base width ; every element width is based on this parameter
- lvl_no → Adjust how many floors that the pagoda have

Referenced materials:

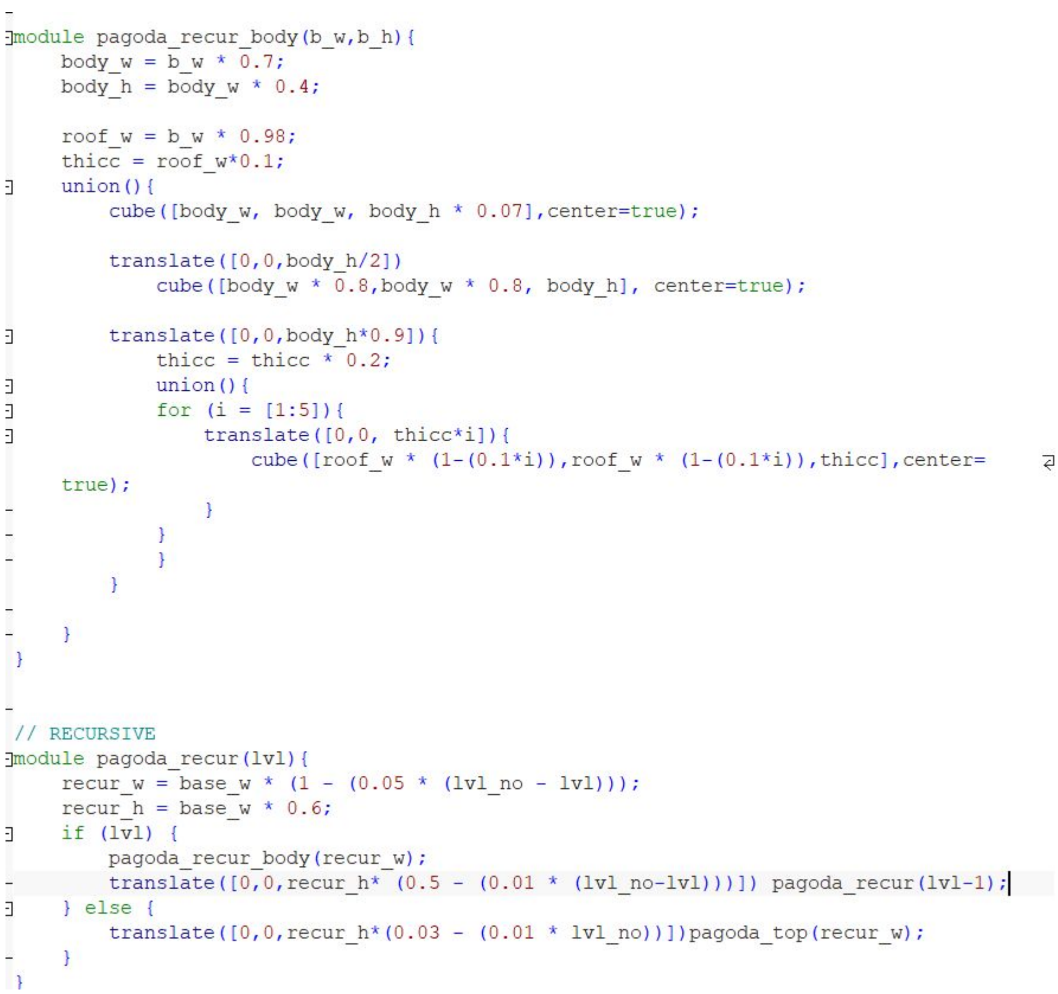
- Cheatsheet : http://www.openscad.org/cheatsheet/index.html
- Recursive tutorial: recursion - Recursively constructing an object in OpenSCAD - Stack Overflow

Problems in design:

- If we adjust lvl_no to a large number (>10) the gap between the floor will be larger, therefore the floor won't connect
    - May need to adjust translation parameters so that the floor is connect

Extra credits:

- Recursive module
    - pagoda_recur() function use to build floors recursively, stack on top of each other



```
module pagoda_recur_body(b_w,b_h){
    body_w = b_w * 0.7;
    body_h = body_w * 0.4;

    roof_w = b_w * 0.98;
    thicc = roof_w*0.1;
    union(){
        cube([body_w, body_w, body_h * 0.07],center=true);

        translate([0,0,body_h/2])
            cube([body_w * 0.8,body_w * 0.8, body_h], center=true);

        translate([0,0,body_h*0.9]){
            thicc = thicc * 0.2;
            union(){
            for (i = [1:5]){
                translate([0,0, thicc*i]){
                    cube([roof_w * (1-(0.1*i)),roof_w * (1-(0.1*i)),thicc],center=
    true);
                }
            }
            }
        }

    }
}


// RECURSIVE
module pagoda_recur(lvl){
    recur_w = base_w * (1 - (0.05 * (lvl_no - lvl)));
    recur_h = base_w * 0.6;
    if (lvl) {
        pagoda_recur_body(recur_w);
        translate([0,0,recur_h* (0.5 - (0.01 * (lvl_no-lvl)))]) pagoda_recur(lvl-1);
    } else {
        translate([0,0,recur_h*(0.03 - (0.01 * lvl_no))])pagoda_top(recur_w);
    }
}
```

Comments:

- Maybe some more decorating around pagoda for beautifulness