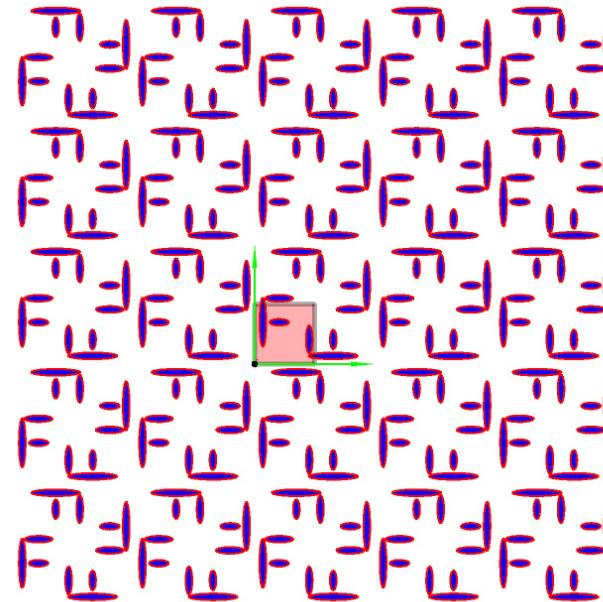


PART 1: 17 PATTERNS FOR “F”



Wallpaper ID 9
Symmetry0 : TRANSLATE
Symmetry1 : TRANSLATE
Symmetry2 : ROTATE 4



CS4497 Spring 2022

Jiaxi Xu

Problem statement

Transform the default “F” pattern to make 17 wallpaper designs, and use Corner-Operated Tran-Similar(COTS) to move the wallpaper.

COMMENTS:

Code folder is code/part_1_2_code

Code shows part2 myGif when opens

Press ‘t’ to change gif and then ‘f’ to show animated F

Press key “F” to start/stop the animation

Solution 1: Solution outline

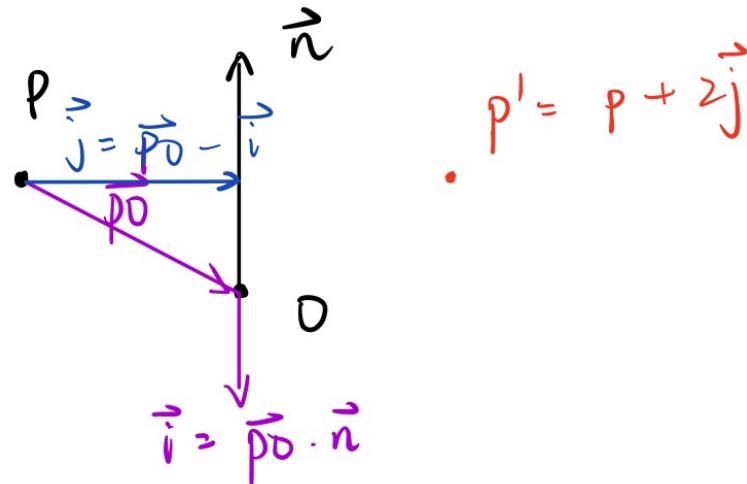
Provide concise outline of your approach in plain English .

- Implement mirror() [1] and glide() [2]
- Use translate(), rotate(), mirror() and glide() to achieve 17 patterns [3]

*** see next page for each [n]

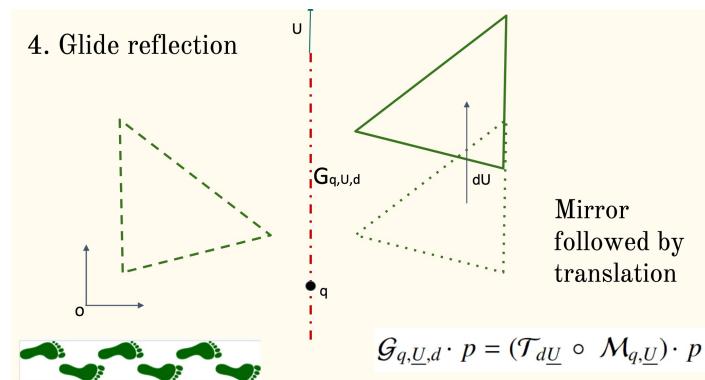
Solution 1: math formulation and derivation

[1] Mirror



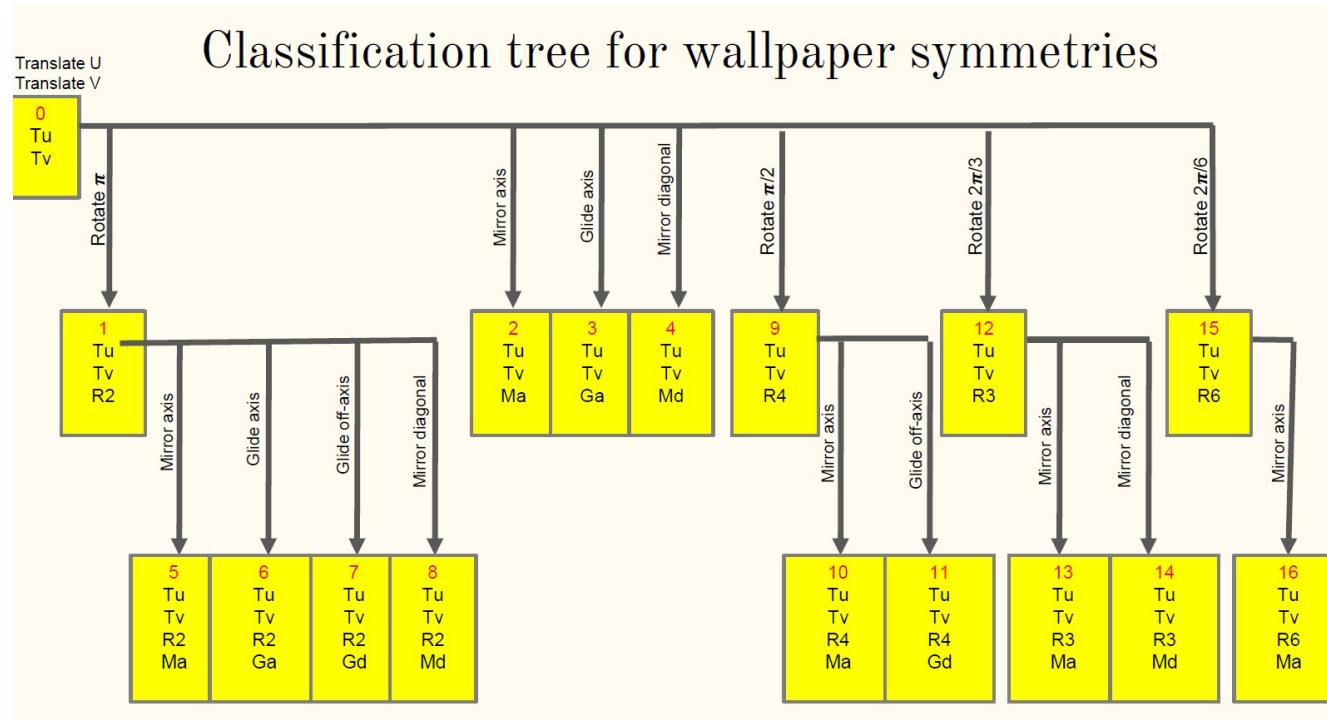
[2] Glide

Glide = mirror then translate



Solution 1: math formulation and derivation

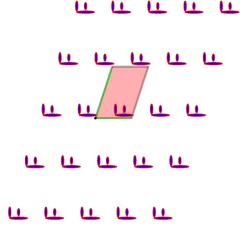
[3] 17 patterns



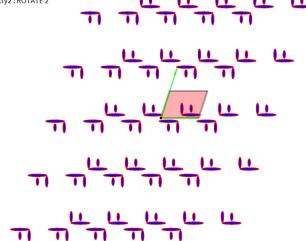
Solution 1: Examples, applications, and limitations

Nice examples (0 ~ 7)

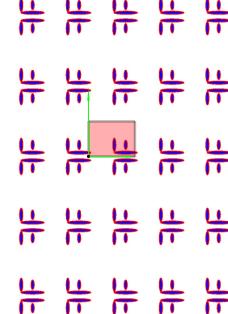
Wallpaper ID 0
Symmetry0 : TRANSLATE
Symmetry1 : TRANSLATE



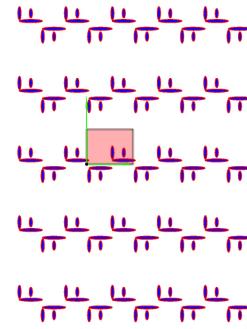
Motifpaper ID 1
Symmetry0 : TRANSLATE
Symmetry1 : TRANSLATE
Symmetry2 : ROTATE 2



Wallpaper ID 2
Symmetry0 : TRANSLATE
Symmetry1 : TRANSLATE
Symmetry2 : MIRROR AXIS



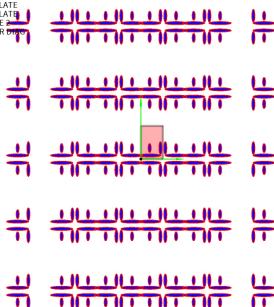
Wallpaper ID 3
Symmetry0 : TRANSLATE
Symmetry1 : TRANSLATE
Symmetry2 : GLIDE AXIS



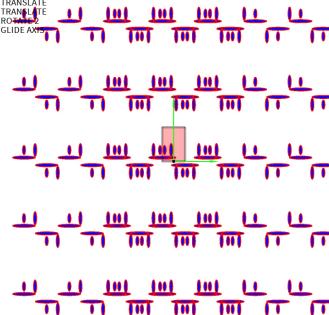
Wallpaper ID 4
Symmetry0 : TRANSLATE
Symmetry1 : TRANSLATE
Symmetry2 : MIRROR DIAG



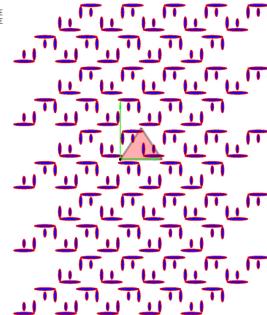
Wallpaper ID 5
Symmetry0 : TRANSLATE
Symmetry1 : TRANSLATE
Symmetry2 : ROTATE
Symmetry3 : MIRROR



Wallpaper ID 6
Symmetry0 : TRANSLATE
Symmetry1 : TRANSLATE
Symmetry2 : ROTATE
Symmetry3 : GLIDE AX



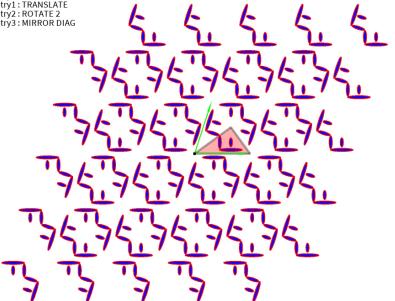
Wallpaper ID 7
Symmetry0 : TRANSLATE
Symmetry1 : TRANSLATE
Symmetry2 : ROTATE 2
Symmetry3 : GLIDE OFF



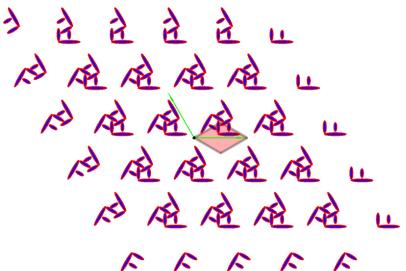
Solution 1: Examples, applications, and limitations

Nice examples (8 ~ 16)

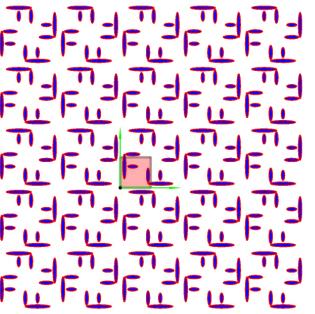
Wallpaper ID 8
Symmetry0: TRANSLATE
Symmetry1: TRANSLATE
Symmetry2: ROTATE 2
Symmetry3: MIRROR DIAG



Wallpaper ID 12
Symmetry0: TRANSLATE
Symmetry1: TRANSLATE
Symmetry2: ROTATE 3



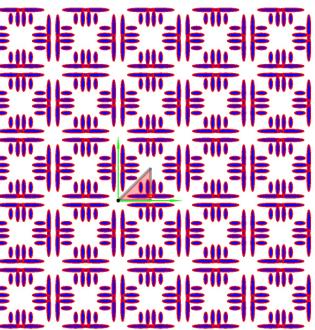
Wallpaper ID 9
Symmetry0: TRANSLATE
Symmetry1: TRANSLATE
Symmetry2: ROTATE 2
Symmetry3: ROTATE 4



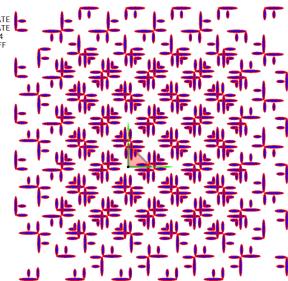
Wallpaper ID 13
Symmetry0: TRANSLATE
Symmetry1: TRANSLATE
Symmetry2: ROTATE 3
Symmetry3: MIRROR DIAG



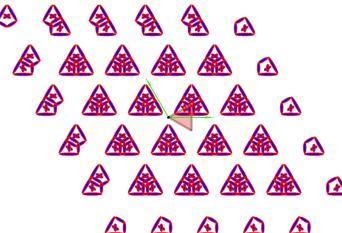
Wallpaper ID 10
Symmetry0: TRANSLATE
Symmetry1: TRANSLATE
Symmetry2: ROTATE 4
Symmetry3: MIRROR DIAG



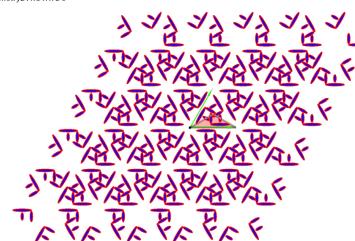
Wallpaper ID 11
Symmetry0: TRANSLATE
Symmetry1: TRANSLATE
Symmetry2: ROTATE 2
Symmetry3: GLIDE OFF



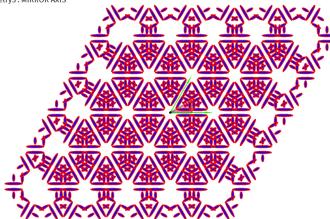
Wallpaper ID 14
Symmetry0: TRANSLATE
Symmetry1: TRANSLATE
Symmetry2: ROTATE 3
Symmetry3: MIRROR DIAG



Wallpaper ID 15
Symmetry0: TRANSLATE
Symmetry1: TRANSLATE
Symmetry2: ROTATE 6



Wallpaper ID 16
Symmetry0: TRANSLATE
Symmetry1: TRANSLATE
Symmetry2: ROTATE 6
Symmetry3: MIRROR AXIS



Solution 1: Examples, applications, and limitations

Potential applications:

Suggest applications (if any) for which it might be of value.

- Complicated pattern design without repetitive work

Limitations:

State limitations of your solution.

- The patterns are all symmetrical/periodic

Solution 1: Your code

Paste an image of your code showing line numbers.

```
21 public pt Mirror(pt P, pt reflection_axis_pt, vec reflection_axis_vec)
22 {
23     //System.out.println("mirror() is called");
24     //System.out.format("P in is %f, %f \n", P.x, P.y);
25     vec P0 = V(P, reflection_axis_pt);
26     vec n = U(reflection_axis_vec);
27     vec pToV = M(P0, V(dot(P0, n), n));
28     return P(P, V(2, pToV));
29 }
30
31 // GLIDE: Recall that a glide can be thought of as a composition of a mirror followed by a
32 // This function takes as input a point P, the mirror axis and translation distance (dist).
33 // The output should be a new point obtained as a result of glide reflection of P along giv
34 // STUDENT CODE BELOW
35 public pt Glide(pt P, pt glide_axis_pt, vec glide_axis_vec, float dist)
36 {
37     return Translate(Mirror(P,glide_axis_pt, glide_axis_vec),U(glide_axis_vec).mul(dist));
38 }
39
```

```
switch(WP.wallpaper_id) {
    case 0:
        WP.addSymmetry(new symmetry("TRANSLATE", U));
        WP.addSymmetry(new symmetry("TRANSLATE", V));
        break;
    case 1:
        WP.addSymmetry(new symmetry("TRANSLATE", U));
        WP.addSymmetry(new symmetry("TRANSLATE", V));
        WP.addSymmetry(new symmetry("ROTATE", 0, PI));
        break;
    case 2:
        WP.addSymmetry(new symmetry("TRANSLATE", U));
        WP.addSymmetry(new symmetry("TRANSLATE", V));
        WP.addSymmetry(new symmetry("MIRROR", 0, U));
        break;
    case 3:
        WP.addSymmetry(new symmetry("TRANSLATE", U));
        WP.addSymmetry(new symmetry("TRANSLATE", V));
        WP.addSymmetry(new symmetry("GLIDE", 0, U, n(U)/2));
        break;
    case 4:
        WP.addSymmetry(new symmetry("TRANSLATE", U));
        WP.addSymmetry(new symmetry("TRANSLATE", V));
        WP.addSymmetry(new symmetry("MIRROR", P(0, U), M(U, V)));
        break;
    case 5:
        WP.addSymmetry(new symmetry("TRANSLATE", U));
        WP.addSymmetry(new symmetry("TRANSLATE", V));
        WP.addSymmetry(new symmetry("ROTATE", 0, PI));
        WP.addSymmetry(new symmetry("MIRROR", 0, V));
        break;
```

Solution 1: Your code

Paste an image of your code showing line numbers.

```
104 case 6:
105     WP.addSymmetry(new symmetry("TRANSLATE", U));
106     WP.addSymmetry(new symmetry("TRANSLATE", V));
107     WP.addSymmetry(new symmetry("ROTATE", 0, PI));
108     WP.addSymmetry(new symmetry("GLIDE", 0, U, n(U)/2));
109     break;
110 case 7:
111     WP.addSymmetry(new symmetry("TRANSLATE", U));
112     WP.addSymmetry(new symmetry("TRANSLATE", V));
113     WP.addSymmetry(new symmetry("ROTATE", P(0, V(0.5, U)), PI));
114     WP.addSymmetry(new symmetry("GLIDE", P(0, V(0.25, V)), U, n(U)/2));
115     break;
116 case 8:
117     WP.addSymmetry(new symmetry("TRANSLATE", U));
118     WP.addSymmetry(new symmetry("TRANSLATE", V));
119     WP.addSymmetry(new symmetry("ROTATE", P(0, V(0.5, U)), PI));
120     WP.addSymmetry(new symmetry("MIRROR", 0, V(0.5, A(U, V))));
121     break;
122
123 case 9:
124     WP.addSymmetry(new symmetry("TRANSLATE", U));
125     WP.addSymmetry(new symmetry("TRANSLATE", V));
126     WP.addSymmetry(new symmetry("ROTATE", P(P(0, V(0.5, V)), V(0.5, U)), P
127     break;
128 case 10:
129     WP.addSymmetry(new symmetry("TRANSLATE", U));
130     WP.addSymmetry(new symmetry("TRANSLATE", V));
131     WP.addSymmetry(new symmetry("ROTATE", P(P(0, V(0.5, V)), V(0.5, U)), P
132     WP.addSymmetry(new symmetry("MIRROR", 0, V(0.5, A(U, V))));
133     break;
```

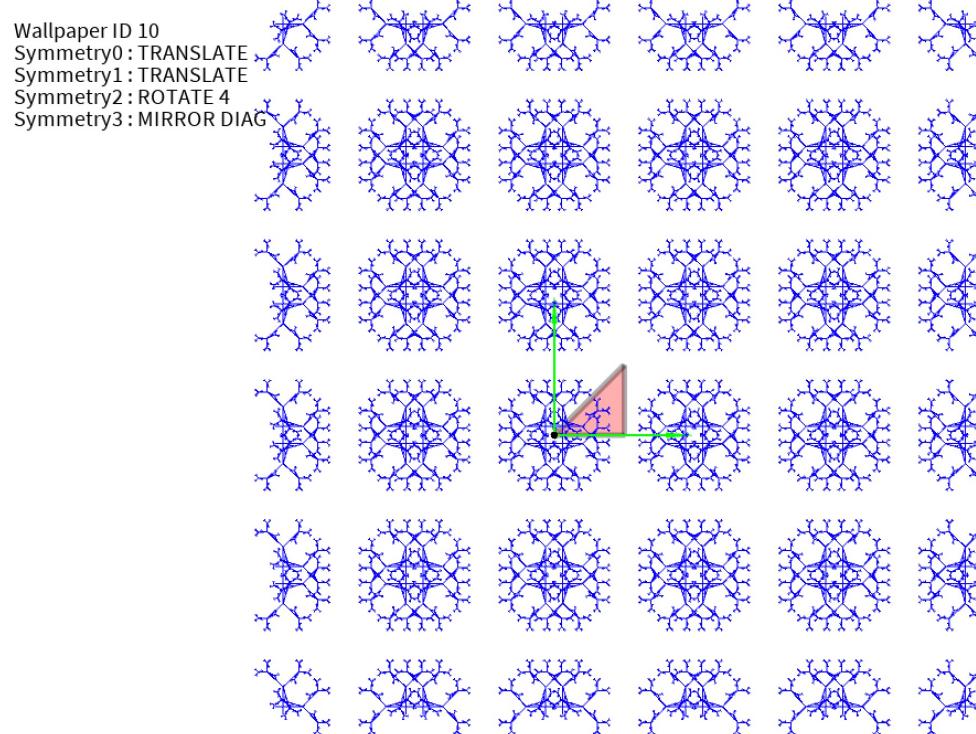
```
case 11:
WP.addSymmetry(new symmetry("TRANSLATE", U));
WP.addSymmetry(new symmetry("TRANSLATE", V));
WP.addSymmetry(new symmetry("ROTATE", P(P(0, V(0.5, V)), V(0.5, U)), PI*0.5));
WP.addSymmetry(new symmetry("GLIDE", P(0, V(0.5, V)), V(0.5, A(U, V)), 0));
break;
case 12:
WP.addSymmetry(new symmetry("TRANSLATE", U));
WP.addSymmetry(new symmetry("TRANSLATE", V));
WP.addSymmetry(new symmetry("ROTATE", 0, PI*2/3));
break;
case 13:
WP.addSymmetry(new symmetry("TRANSLATE", U));
WP.addSymmetry(new symmetry("TRANSLATE", V));
WP.addSymmetry(new symmetry("ROTATE", 0, PI*2/3));
WP.addSymmetry(new symmetry("MIRROR", 0, V(0.5, A(U, V))));
break;
case 14:
WP.addSymmetry(new symmetry("TRANSLATE", U));
WP.addSymmetry(new symmetry("TRANSLATE", V));
WP.addSymmetry(new symmetry("ROTATE", 0, PI*2/3));
WP.addSymmetry(new symmetry("MIRROR", P(0, U), M(U, V)));
break;
case 15:
WP.addSymmetry(new symmetry("TRANSLATE", U));
WP.addSymmetry(new symmetry("TRANSLATE", V));
WP.addSymmetry(new symmetry("ROTATE", 0, PI/3));
break;
case 16:
WP.addSymmetry(new symmetry("TRANSLATE", U));
WP.addSymmetry(new symmetry("TRANSLATE", V));
WP.addSymmetry(new symmetry("ROTATE", 0, PI/3));
WP.addSymmetry(new symmetry("MIRROR", 0, U));
break;
```

Solution 1: Sources

List and provide links or bib references to all sources of inspiration that you have consulted and used or find useful.

[https://gatech.instructure.com/courses/249698/files/folder/PROJECTS/P4%20patterns
?preview=31622363](https://gatech.instructure.com/courses/249698/files/folder/PROJECTS/P4%20patterns?preview=31622363) (lecture slide)

PART 2: REPLACE “F” WITH A GIF



CS4497 Spring 2022

Jiaxi Xu

Problem statement

Design a cool asymmetrical animated gif to replace the figure “F”

COMMENTS:

Press ‘t’ to change gif and then ‘f’ to show animated myGif

Press key “F” to start/stop the animation

Solution 1: Solution outline

Provide concise outline of your approach in plain English .

- Use recursion to generate a fractal tree-like pattern
- It branches out by theta degrees and for animation, theta changes with time in the range of [0, 90]

Solution 1: math formulation and derivation

Provide math for your constructions.

- Pseudo code for the recursive step
 - Parameter h (h = 120 in my case)
 - Terminate condition: every time $h *= 0.4$, stops when $h \leq 2$

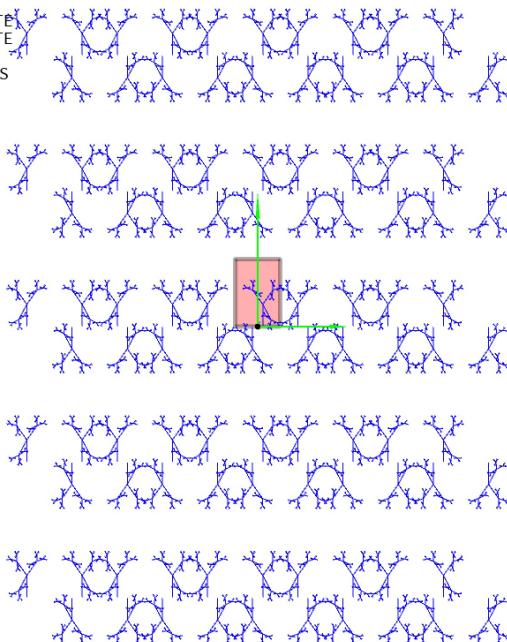
```
pushMatrix(); // Save the current state of transformation (i.e. where are we now)
rotate(theta); // Rotate by theta
line(0, 0, 0, -h); // Draw the branch
translate(0, -h); // Move to the end of the branch
branch(h); // recursive call
popMatrix(); // Whenever we get back here, we "pop" in order to restore the previous matrix so
• Repeat the above steps to branch left and bottom right by changing rotate(theta);
```

Solution 1: Examples, applications, and limitations

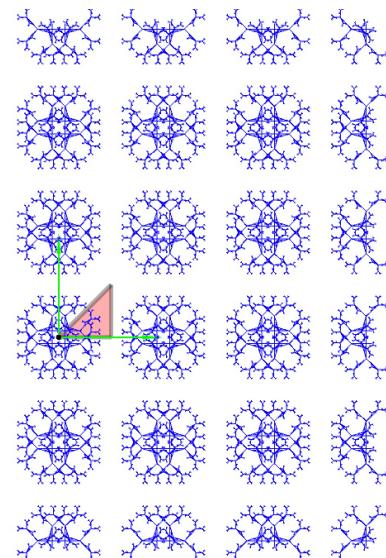
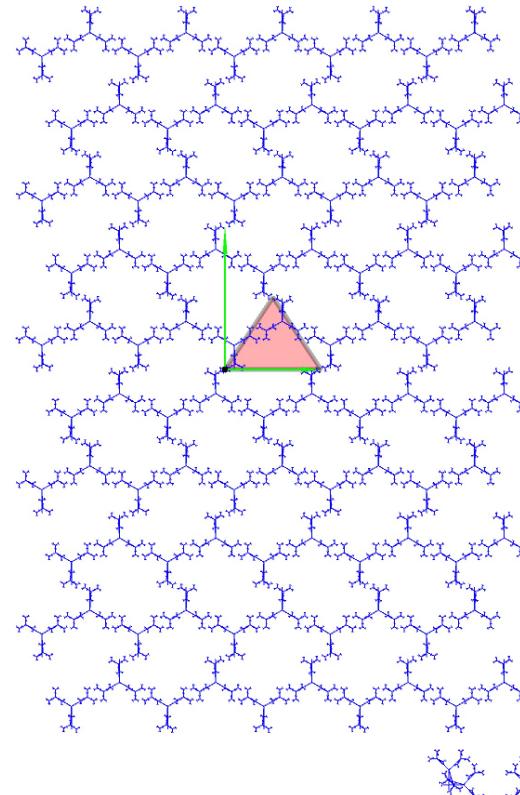
Nice examples

Show nice examples of you

Wallpaper ID 6
Symmetry0: TRANSLATE
Symmetry1: TRANSLATE
Symmetry2: ROTATE 2
Symmetry3: GLIDE AXIS



Wallpaper ID 7
Symmetry0: TRANSLATE
Symmetry1: TRANSLATE
Symmetry2: ROTATE 2
Symmetry3: GLIDE OFF



Solution 1: Examples, applications, and limitations

Potential applications:

Suggest applications (if any) for which it might be of value.

- To generate beautiful patterns
- To simulate periodic motions

Limitations:

State limitations of your solution.

- Such asymmetrical recursive pattern may look messy for complication transformations (e.g. case 16). A better solution may be to adjust recursive steps and degrees to accommodate different transformations.

Solution 1: Your code

```
213 float theta;
214 void drawMyPattern(float a) {
215   stroke(blue);
216
217   //float a = (mouseX / (float) width) * 90f;
218   theta = radians(a);
219   translate(2.*a/5, -a/24);
220   line(0,0,0,-60);
221   translate(0,-60);
222   branch(120); // Line 222
223 }
224
225 void branch(float h) {
226   h *= 0.4;
227
228   if (h > 2) {
229     pushMatrix();
230     rotate(theta);
231     line(0, 0, 0, -h);
232     translate(0, -h);
233     branch(h);
234     popMatrix();
235
236     // branch off to the "left"
237     pushMatrix();
238     rotate(-theta);
239     line(0, 0, 0, -h);
240     translate(0, -h);
241     branch(h);
242     popMatrix();
243
244     // branch off to the "left"
245     pushMatrix();
246     rotate(theta + 90);
247     line(0, 0, 0, -h);
248     translate(0, -h);
249     branch(h);
250     popMatrix();
251   }
252 } // Line 252
253
```

Solution 1: Sources

List and provide links or bib references to all sources of inspiration that you have consulted and used or find useful.

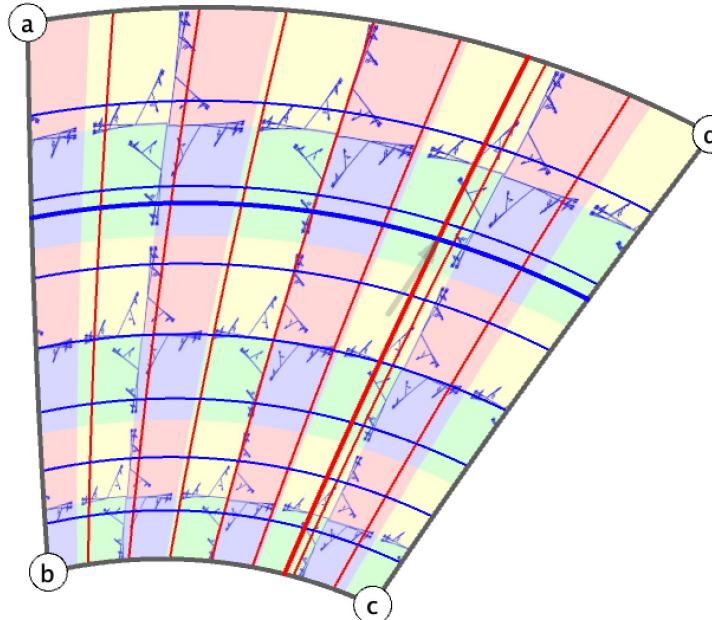
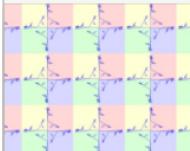
<https://processing.org/examples/tree.html>

PART 3: COTS MAP



CS4497 Spring 2022
Jiaxi Xu

http://www.cc.gatech.edu/~jarek



Problem statement

Apply the COTS mapping to the wallpaper pattern created in Parts 1 and 2.

COMMENTS:

Code folder is code/part3_code

Press “x” to load my Gif texture

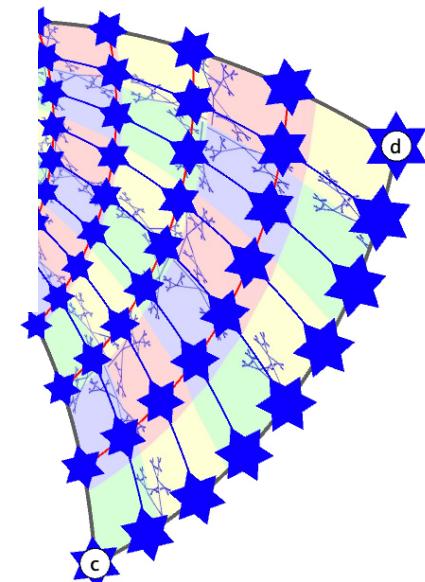
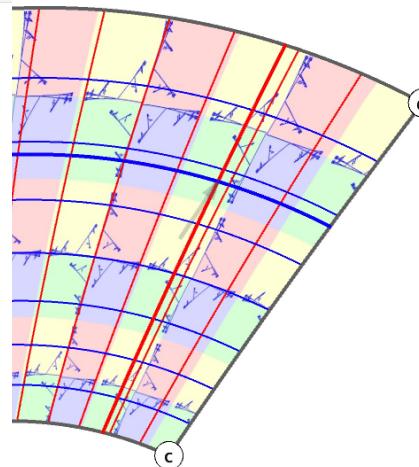
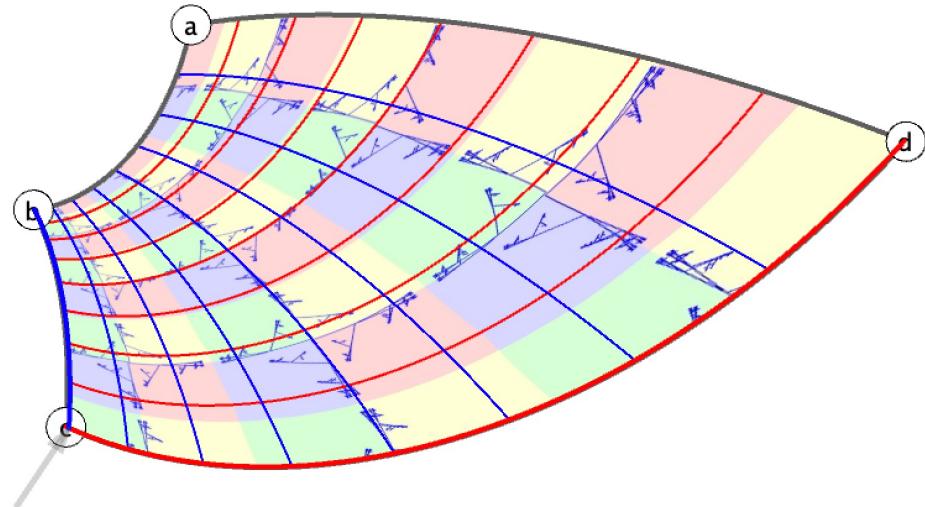
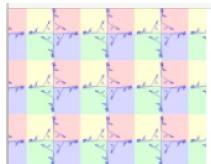
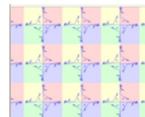
Solution 1: Solution outline

Provide concise outline of your approach in plain English .

- Record 90 frames from part2 gif animation
- Because processing throw errors when loading ‘.tif’ , I change the file format to .jpg
- Load texture in part3 code

Solution 1: Examples, applications, and limitations

Nice examples



Solution 1: Examples, applications, and limitations

Potential applications:

Suggest applications (if any) for which it might be of value.

- Distorted mapping
- Texture accommodates distortion created by projection etc.

Limitations:

State limitations of your solution.

- More variation can be created or make a fully automatic animation

Solution 1: Your code

Paste an image of your code showing line numbers.

```
58 //***** display current frame *****
59 int frame;
60 void draw()
61 {
62     background(white);
63
64     frame = (frame + 1) % 90;
65     if (frame < 10) myFace = loadImage("FRAMES/F000" + frame + ".jpg" );
66     else myFace = loadImage("FRAMES/F00" + frame + ".jpg");
67
68
```