

minecraft chainmail thingy armor design p5 code SUUUDO code

Algorithm 1 InShape(i, j)

```
1: if  $shape = "circle"$  then
2:    $centerCol \leftarrow (cols - 1)/2$ 
3:    $centerRow \leftarrow (rows - 1)/2$ 
4:   return  $\text{dist}(i, j, centerCol, centerRow) \leq \min(centerCol, centerRow) + 0.01$ 
5: else
6:   return true
7: end if
```

Algorithm 2 LinkCenters()

```
1: for  $i \leftarrow 0$  to  $cols - 1$  do
2:   for  $j \leftarrow 0$  to  $rows - 1$  do
3:      $present[i][j] \leftarrow \text{InShape}(i, j)$ 
4:      $used[i][j] \leftarrow \text{false}$ 
5:   end for
6: end for
7:  $wonkyLinks \leftarrow \emptyset$   $\triangleright$  links centered between four grid points
8: for  $i \leftarrow 0$  to  $cols - 2$  do
9:   for  $j \leftarrow 0$  to  $rows - 2$  do
10:     $squadmates \leftarrow \emptyset$ 
11:     $possibleHomies \leftarrow \{(i, j), (i+1, j), (i, j+1), (i+1, j+1)\}$ 
12:    for each  $bruh \in possibleHomies$  do
13:      if  $present[bruh.x][bruh.y]$  then
14:         $squadmates \leftarrow squadmates \cup \{bruh\}$ 
15:      end if
16:    end for
17:    if  $|squadmates| \geq 2$  then
18:       $wonkyLinks \leftarrow wonkyLinks \cup \{(i + 0.5) \cdot linkPitch, (j + 0.5) \cdot linkPitch\}$ 
19:      for each  $bruh \in squadmates$  do
20:         $used[bruh.x][bruh.y] \leftarrow \text{true}$ 
21:      end for
22:    end if
23:  end for
24: end for
25:  $centers \leftarrow \emptyset$ 
26: for  $i \leftarrow 0$  to  $cols - 1$  do
27:   for  $j \leftarrow 0$  to  $rows - 1$  do
28:     if  $used[i][j]$  then
29:        $centers \leftarrow centers \cup \{(i \cdot linkPitch, j \cdot linkPitch)\}$ 
30:     end if
31:   end for
32: end for
33:  $centers \leftarrow centers \cup wonkyLinks$ 
34: return  $centers$ 
```

Algorithm 3 IsVoxelSolid(i, j, k)

```
1: if  $i \notin [0, 2]$  or  $j \notin [0, 2]$  or  $k \notin [0, 2]$  then
2:   return false
3: end if
4:  $spicyX \leftarrow (i \neq 1)$ 
5:  $spicyY \leftarrow (j \neq 1)$ 
6: if  $spicyX$  and  $spicyY$  then
7:   return true
8: else if  $spicyY$  and  $k = 0$  then
9:   return true
10: else if  $spicyX$  and  $k = 2$  then
11:   return true
12: else
13:   return false
14: end if
```

Algorithm 4 UnitLinkTriangles()

```
1:  $chonk \leftarrow barWidth/2$ 
2:  $xEdges \leftarrow [-linkHalfSize - chonk, -linkHalfSize + chonk, linkHalfSize - chonk, linkHalfSize + chonk]$ 
3:  $yEdges \leftarrow xEdges$ 
4:  $zEdges \leftarrow [0, barThickness, barThickness + gapThickness, sheetThickness]$ 
5:  $triangleSoup \leftarrow \emptyset$ 
6: for  $i \leftarrow 0$  to 2 do
7:   for  $j \leftarrow 0$  to 2 do
8:     for  $k \leftarrow 0$  to 2 do
9:       if not IsVoxelSolid( $i, j, k$ ) then
10:        continue
11:      end if
12:       $(x_{lo}, x_{hi}) \leftarrow (xEdges[i], xEdges[i+1])$ 
13:       $(y_{lo}, y_{hi}) \leftarrow (yEdges[j], yEdges[j+1])$ 
14:       $(z_{lo}, z_{hi}) \leftarrow (zEdges[k], zEdges[k+1])$ 
15:      for each face direction  $yeetDir \in \{+x, -x, +y, -y, +z, -z\}$  do
16:        if neighbor of  $(i, j, k)$  in direction  $yeetDir$  is not solid then
17:          AddQuad(the two triangles forming the face at  $yeetDir$ , using
18:             $x_{lo}, x_{hi}, y_{lo}, y_{hi}, z_{lo}, z_{hi}$ )
19:          append both triangles to  $triangleSoup$ 
20:        end if
21:      end for
22:    end for
23:  end for
24: return  $triangleSoup$ 
```

Algorithm 5 BuildChains()

```
1:  $centers \leftarrow \text{LinkCenters}()$ 
2:  $triangleSoup \leftarrow \text{UnitLinkTriangles}()$ 
3:  $bigChungusMesh \leftarrow \emptyset$ 
4: for each  $center \in centers$  do
5:   for each  $triangle \in triangleSoup$  do
6:     for each vertex  $v \in triangle$  do
7:       emit vertex  $(center.x + v.x, center.y + v.y, v.z)$  into  $bigChungusMesh$ 
8:     end for
9:   end for
10: end for
11: return  $bigChungusMesh$ 
```
