```
0
1   ########################################
2   ########################################
3
4   #NOTES: WEEK 3 CIRCLE ANIMATION EXAMPLE
5   #NAME: Maho Kobayashi
6
7   ########################################
8   ########################################
9
10  #IMPORT LIBRARIES
11
12  import rhinoscriptsyntax as rs
13  import random as rnd
14
15  ########################################
16
17  #input the number of frames you want to produce
18  num_of_frames = rs.GetInteger('Number of Frames to Output?', 240)
19
20  #loop through the number of frames creating the animation
21  for frame in range(num_of_frames):
22
23  #######________YOUR CODE GOES (INDENTED) BELOW THIS LINE________########

24          #'frame' is your iteration variable - its simply counting
    from 0 to the maximum
25          #number of frames that you input (num_of_frames)
26          #use the 'frame' variable in your code to change something
      - to produce the
27          #animation.
28
29  ########################################
30  ########################################
31
32          #create an empty list / dictionary
33          ptDict = {}
34          crvList = []
35
36          ########################################
37
38          #input values for imax and jmax
39          imax = rs.GetInteger('input number in x direction',10)
40          jmax = rs.GetInteger('input number in y direction',10)
41
42          #incremental loop to generate points
43          for i in range(imax):
44                  for j in range(jmax):
45                          #define x in terms of i
46                          #define y in terms of j
47                          x = i*6+(rnd.random()*frame/3)
48                          y = j*6+(rnd.random()*frame/3)
49                          z = 0
50
51                          #render point in rhinospace
52                          #rs.AddPoint(x,y,z)
53
54
```

```
54                          #save point values in a dictionary using (i,j)
    as a key
55                          ptDict[(i,j)] = (x,y,z)
56
57          #loop through dictionary to create geometry
58          for i in range (imax):
59                  for j in range(jmax):
60
61                          #CREATE GEOMETRY
62                          if i > 0 and j > 0:
63                                  #find centroid of module using midPt
    of constructed line
64                                  constLine = rs.AddLine(ptDict[(i,j)],ptDict
    [(i-1,j-1)])
65                                  centroid = rs.CurveMidPoint(constLine)
66
67                                  #delete constructed line
68                                  rs.DeleteObject(constLine)
69
70                                  #POINTS
71
72                                  #       2-------1       1:  (i,j)
73                                  #       |               |       2:  (i-1,j)

74                                  #       |    mid    |       3:  (i-1,j-1)
75                                  #       |               |       4:  (i,j-1)
76                                  #       3-------4
77
78                                  #draw line from 1 to centroid to 2
79                                  crvList.append(rs.AddCurve((ptDict[(i,j)],
    centroid, ptDict[(i-1,j)])))
80
81                                  #draw line from 2 to centroid to 3
82                                  crvList.append(rs.AddCurve((ptDict[(i-1,j)]
    , centroid, ptDict[(i-1,j-1)])))
83
84                                  #draw line from 1 to 4 to 3
85                                  crvList.append(rs.AddCurve((ptDict[(i,j)],
    ptDict[(i,j-1)],
86                                  ptDict[(i-1,j-1)])))
87
88                                  #draw line from 1 to 2 to 3
89                                  crvList.append(rs.AddCurve((ptDict[(i,j)],
    ptDict[(i-1,j)],
90                                  ptDict[(i-1,j-1)])))
91
92                                  #construct a closed curve from corner
    points
93                                  crvList.append(rs.AddCurve((ptDict[(i,j)],centroid
    ,ptDict[(i-1,j-1)],
94                                  ptDict[(i,j-1)],ptDict[(i,j)])))
95
96
97
98  ########################################
99  ########################################
100
101
```

```
101
102 #######________YOUR CODE GOES (INDENTED) ABOVE THIS LINE________########
103
104         #Specify local folder to output frames -- you will need
    to change this to a
105         #correct path on your computer
106         render_folder = "C:\\Users\\mahok\\Desktop\\python_rhino\\summer_22\\
    3\\part_2\\render\\"
107
108         def render_step(render_folder, sequence_num):
109                 #Captures screenshots of the scene frame
110                 file_name = str(int(sequence_num)).zfill(5)
111                 file_path = "  " + render_folder + file_name + ".png
    "
112                 rs.Command("_-ViewCaptureToFile" + file_path + "  _Enter
    ")
113
114         #Call function to render frame
115         render_step(render_folder, frame)
116
117         #Clear canvas for the next frame --
118         #YOU HAVE TO DELETE ALL THE OBJECTS YOU ARE RENDERING
119         #This could also be optional if you want to overlay the
    frames of your animation
120         #If you're deleteing a single object use rs.DeleteObject()
121         #If you're deleteing a list of objects use rs.DeleteObjects()
122         rs.DeleteObjects(crvList)
123
124
```