SCHOOL OF CODING

Elderberry

SAATHI HOUSE

symplexis

Info def INSTITUTO PARA EL FOMENTO DEL DESARROLLO Y LA FORMACIÓN

SILVERSAP Coopérative de services à la personne et de la Silver économie

coding

4 MIGRANT WOMEN RETURNERS

Music and Art

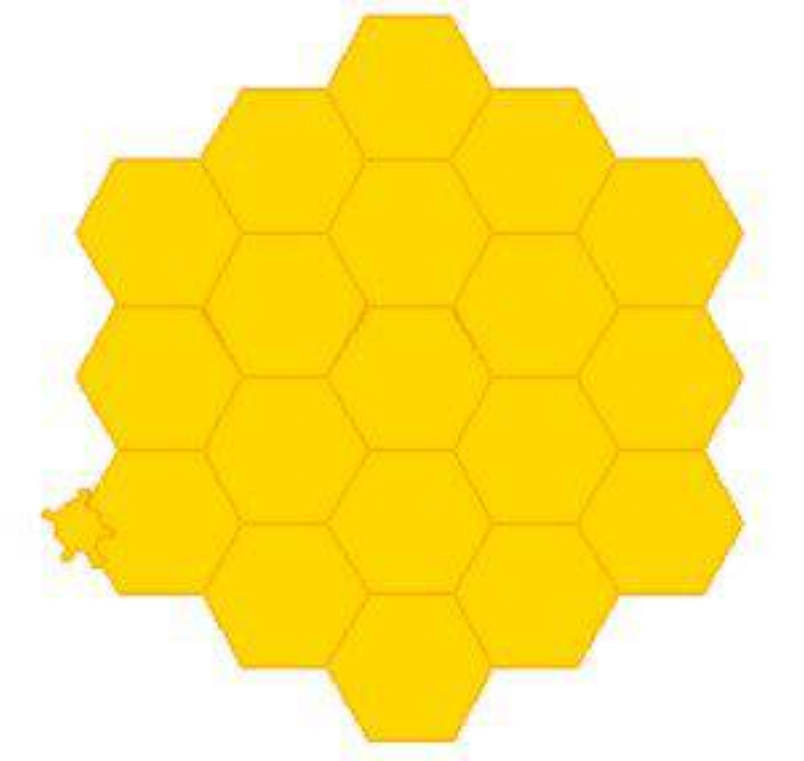
Intermediate

Activity scenario summary:

Music and art are important sectors within our modern day society. The teaching and acknowledgment of the importance of the creative sector as an integral part of human development, can lead to the creation of new and innovative solutions to some of the worlds most complex problems. Over the 3 difficulty levels in this project, different styles of art and music will be explored in the context of coding and computer science, leading to 3 separate, fully developed projects.

In the intermediate section there will be a focus on tessellating shapes, such as honeycombs in a bee hive, using turtle graphics.

What will happen…



We will use different angles to fit together our honeycomb shape and build a small section of a beehive

How does turtle graphics work?

- Python is a high level programming language which allows its users to use different programming styles to create simple or complex programs.
- It is designed to be easy to read and simple to implement.

- Python libraries are a set of useful functions that eliminate the need for writing codes from scratch.
- The library we will be importing is ‘Turtle’ which allows the user to create graphic shapes.
- The turtle understands simple commands such as left and right turns, moving forwards and moving backwards.

Step 1

Importing turtle

In the code here we have imported the turtle library and changed the shape of the cursor to look like a turtle too!

Also, by using a delay we can control the speed of the turtle to make it go as fast, or as slow, as we want. The last part of this code sets the colour's for the outline and fill of the shapes we will create.

```
1  #Beehive and tesselations
2  from turtle import *
3  #changes cursor shape
4  shape("turtle")
5  #Changes the speed of the turtle
6  delay(0)
7  #Subroutine to build the hexagon
8  pencolor("orange")
9  #Changes the outline of the shapes
10 fillcolor("gold")
11 begin_fill()
12 #Changes the fill colour of the shape
```

Subroutines

Subroutines are sets of instructions designed to perform a frequently used operation within a program.

Subroutines can store code and will only be run when ‘called’

There are two main types of subroutine: procedures and functions.

Procedures are not required to return a value, whereas functions must return a value

```
1
2 def greeting():
3     print("Hello World!")
4     print("How are you today?")
5
6
7 greeting()
8
```

```
Hello World!
How are you?
```

Subroutines are great ways of writing more maintainable code and leads to more structured, organized and understandable programs.

coding
4 MIGRANT WOMEN RETURNERS

Step 2

Creating a subroutine

To create the shape itself subroutines and for loops have been used. The use of the subroutine means that once the code has been written at the top it can simply be called multiple times in the program with just the heading name. In addition, the for loops are used to reduce the amount of repetitive code that is written since a hexagon is a regular shape.

Here the two subroutine draw hexagons that are the same size but hex1() draws a group of three hexagons whereas hex2() draws a hexagon but repeats round two extra times so that we can go back in the pattern as we go through the code.

```
13 #Creates a subroutine to draw the shape in one place
14 def hex1():
15     for loop in range(3):
16         for loop in range(6):
17             fd(25)
18             rt(60)
19         lt(120)
20 def hex2():
21     for loop in range(8):
22         fd(25)
23         rt(60)
```

coding
4 MIGRANT WOMEN RETURNERS

Step 3

Creating even more subroutines

These are another two subroutines called hex3 () and hex4().

They both draw hexagons that are the same size as the previous ones except one takes left turns and the other takes right turns to complete the shape. Also, hex4() specifically goes around 10 the shape 10 times so that the angle will fit when the main code it written.

```
24 def hex3():
25     for loop in range(8):
26         fd(25)
27         lt(60)
28 def hex4():
29     for loop in range(10):
30         fd(25)
31         rt(60)
32
```

Step 4

Main code

This is where the main code for the honeycomb picture will be done. So far, the for loop draws the top line of shapes but by using this loop we can avoid repeating code in the program. Furthermore, the subroutines have been called into the main code to be displayed once the code is run.

Another command turtle uses is pu() and pd(), which stands for pen up and pen down. This means that the turtle can move without leaving a line behind it.

```
33 #main code
34 for loop in range(3):
35     hex1()
36     pu()
37     fd(25)
38     lt(60)
39     pd()
40     hex2()
41 hex1()
42 pu()
43 rt(60)
44 fd(75)
45 lt(60)
46 fd(25)
47 lt(180)
48 pd()
49 hex2()
50 pu()
51 fd(25)
52 lt(60)
53 pd()
```

Step 5

Moving the turtle

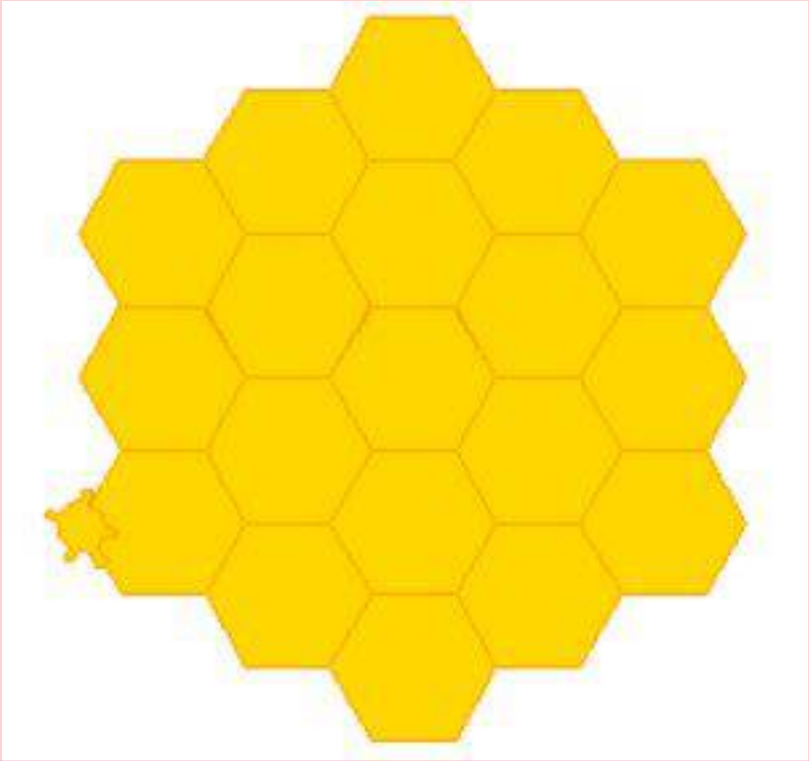
Following on from the last slide, we again are calling subroutines to fit into this tessellated program. The program used a lot of commands between the pu() and pd() statements as the turtle must move through many different lines before getting to the point where another hexagon is needed.

The last line of the program is the end fill button which tells the program to stop the fill colour we coded at the start of the program.

```
54 hex2()
55 pu()
56 lt(120)
57 pd()
58 hex4()
59 for loop in range(2):
60     hex3()
61     rt(120)
62 lt(120)
63 pu()
64 fd(25)
65 rt(60)
66 fd(25)
67 lt(60)
68 fd(25)
69 rt(180)
70 pd()
71 hex2()
72 fd(25)
73 rt(60)
74 hex3()
75 end_fill()
```

The final code …

Run the finished program and see what happens!



Extension:

- Once you have completed this project, try experimenting with the colours and see if you can make the outline thicker on the design.

coding
4 MIGRANT WOMEN RETURNERS

What have you learned how to do?

✔You should have confidently been able to import a library into Python,

✔You should be confident in using subroutines to create regular shapes.

✔You should be comfortable using angles to navigate through regular shapes,

✔You should be confident in manipulating the speed and colour within a turtle graphics program.

Congratulations!

You have completed your very own tessellation program

You are ready to move on to the advanced level…