**視覺化程式設計-碎形繪圖專題**

**模組化-遞迴函式Fractals**

**學習活動 4：Drawing Fractals by programming**

1. **實作以下程式，繪製Koch Curve 第0~3步驟圖形**

|  |  |
| --- | --- |
| 1. from turtle import \* 2. **def** **f**(t, length, depth): 3. if depth == 0: 4. t.forward(length) 5. return 6. else: 7. f(t, length/3, depth-1) 8. t. left(60) 9. f(t, length/3, depth-1) 10. t. right (120) 11. f(t, length/3, depth-1) 12. t.left(60) 13. f(t, length/3, depth-1) 14. canvas = Screen() # Set up the window and its attributes 15. koch = Turtle() # create koch 16. **f(koch, 200, \_\_\_\_\_\_\_)**  # Call the function to draw the koch Curve 17. canvas.exitonclick() | |
| **第0步驟** | **第1步驟** |
| **第2步驟** | **第3步驟** |

1. **觀察Koch Curve**

|  |  |  |  |
| --- | --- | --- | --- |
| http://atlas-zone.com/complex/fractals/product/pic/Koch_g0.gif | http://atlas-zone.com/complex/fractals/product/pic/Koch_g1.gif | http://atlas-zone.com/complex/fractals/product/pic/Koch_g2.gif | http://atlas-zone.com/complex/fractals/product/pic/Koch_g3.gif |

**def** **f**(t, length, depth):

|  |  |
| --- | --- |
| **起始元：**  **if depth == 0:**  t.forward(length) | http://atlas-zone.com/complex/fractals/product/pic/Koch_g0.gif |
| **生成元：**  **else:**  **f(t, length/3, depth-1)**  **t.left(60)**  **f(t, length/3, depth-1)**  **t.right(120)**  **f(t, length/3, depth-1)**  **t.left(60)**  **f(t, length/3, depth-1)** | http://atlas-zone.com/complex/fractals/product/pic/Koch_g1.gif |

討論：程式呼叫 **f(Koch, 150, 2)** 時，請畫下其f函式呼叫的歷程

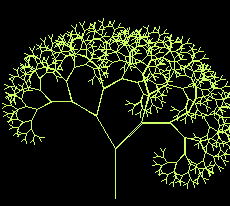
1. **實作以下程式，繪製蕨葉(Fern)。試修改生長規則另生成不同蕨葉**

|  |  |
| --- | --- |
| from turtle import \*  **def** **fern(t, size)**:  **if size < 5 :**  return  t.forward(size/25)  t.left(80)  **fern(t, size\*.3)**  t.right(82)  t.forward(size/25)  t.right(80)  **fern(t, size\*.3)**  t.left(78)  **fern(t, size\*.9)**  t.left(2)  t.back(size/25)  t.left(2)  t.back(size/25)  canvas = Screen() # Set up the window and its attributes  mytree = Turtle() # create mytree  mytree.left(90)  **fern(mytree,305)**  # Call the function to draw the fern  canvas.exitonclick() | |
| http://d.blog.xuite.net/d/b/1/8/14040789/blog_150608/txt/9952555/2.gif | 修改生長規則另生成不同蕨葉： |

1. **實作以下程式，繪製一棵樹(tree)。試修改生長規則另生成不同形態的樹**

|  |
| --- |
| 1. from turtle import \* 2. **def** **tree(t, order, len)**: 3. **if order == 1** : 4. t.forward(len) 5. t.left(30) 6. t.forward(len) 7. t.backward(len) 8. t.right(30) 9. t.right(30) 10. t.forward(len) 11. t.backward(len) 12. t.left(30) 13. t.backward(len) 14. return 15. **else :** 16. t.forward(len) 17. t.left(30) 18. **tree(t, order-1, len\*.75)** 19. t.right(30) 20. t.right(45) 21. **tree(t, order-1, len\*.75)** 22. t.left(45) 23. t.backward(len) 24. canvas = Screen() # Set up the window and its attributes 25. mt= Turtle() # create mytree 26. mt.left(90) 27. mt.hideturtle() # hide the turtle 28. **tree(mt, 9, 50)** # Call the function to draw the tree 29. canvas.exitonclick() |

1. **執行結果：**



1. **試修改生長規則另生成不同形態的樹**

|  |  |
| --- | --- |
| **tree 12 100** | **change “LT” “RT”** |
| **change LT angle and RT angle** | **change \_\_\_\_\_\_** |

1. **建構碎形圖規則，以程式繪製碎形圖**
2. **碎形繪製規則**

|  |  |
| --- | --- |
| **起始元** | **生成元** |

1. **碎形函式**

|  |  |
| --- | --- |
| **起始元** | **生成元** |

1. **依碎形繪製規則疊代建構至第4代碎形**

|  |  |  |  |
| --- | --- | --- | --- |
| **第零步驟** | **第一步驟** | **第二步驟** | **第三步驟** |

1. **碎形程式碼**