Starry Night

Python Project with turtle module

(2017.Coding Projects in Python.DK.)

Hacks and tweaks: Using the turtle screen onclick() event

If you want your turtle to draw wherever you click with your mouse on the **trinket** canvas, follow the steps as described below:

1. Create a new **trinket** Python project.

2. In your new **trinket**, copy the *imports* and the *draw_star(size,color,points,x,y)* function from your Starry Night **trinket** project.

3. Create a new function and call it *draw* and pass it two parameters *x* and *y*.

4. Copy the *randomSize*, *randomColor* and *randomPoints* variables from your Starry Night **trinket** project and paste them inside of the newly created function *draw(x,y)*.

5. Still inside of your *draw(x,y)* function, call the *draw_star(size,color,points,x,y)* function and pass it the *randomSize*, *randomColor*, *randomPoints*, *x* and *y* variables where *x* an *y* will be the coordinates of the point where your mouse is clicked and these variables will be received by the *draw(x,y)* function from an event handler which we will call later in the code.

6. After the *draw(x,y)* function, call the *showturtle()* function on your turtle.

7. Finally let's call our long awaited event handler the *Screen().onclick()* function on your turtle and pass it the *draw(x,y)* function.

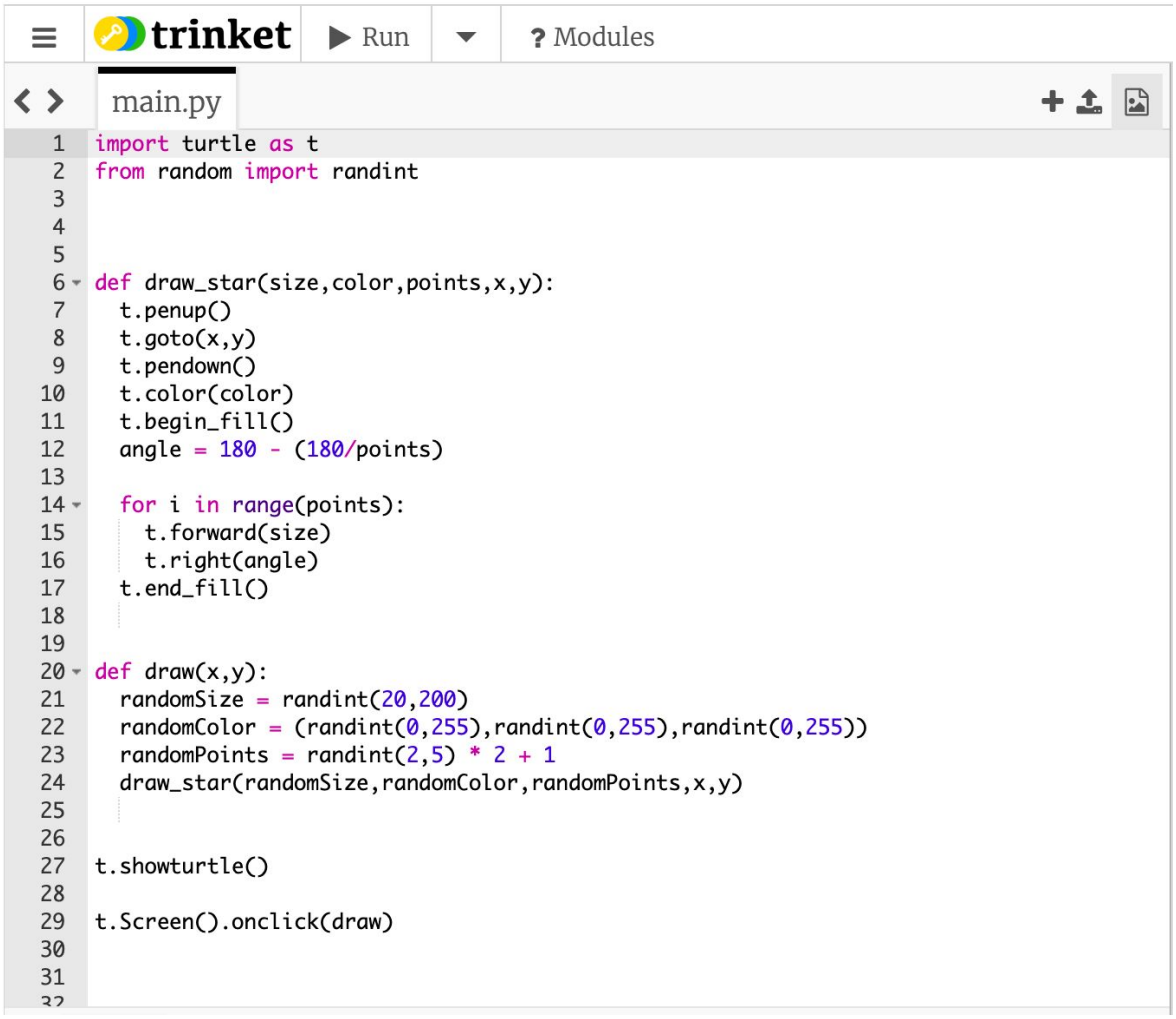
8. Run your program.

9. Now whenever you will click on the **trinket** canvas, the *Screen().onclick()* function will be triggered.

10. On triggering, it will pick up the mouse click coordinates and call the *draw(x,y)* function. It will pass the mouse click coordinates to the *draw(x,y)* function in *x* and *y* respectively.

11. Your *draw(x,y)* function will then run and start drawing the star at the coordinates it received.

12. Your final code should look something like this:




```
import turtle as t
from random import randint



def draw_star(size,color,points,x,y):
  t.penup()
  t.goto(x,y)
  t.pendown()
  t.color(color)
  t.begin_fill()
  angle = 180 - (180/points)

  for i in range(points):
    t.forward(size)
    t.right(angle)
  t.end_fill()


def draw(x,y):
  randomSize = randint(20,200)
  randomColor = (randint(0,255),randint(0,255),randint(0,255))
  randomPoints = randint(2,5) * 2 + 1
  draw_star(randomSize,randomColor,randomPoints,x,y)


t.showturtle()

t.Screen().onclick(draw)
```

Challenge

There is a bug in this code. Try to find it and fix it. **(See solution on last page)**

Resources

https://docs.python.org/3.3/library/turtle.html?highlight=turtle#turtle.onclick

Solution to Challenge

While your code is running, whenever you click on the **trinket** canvas, the t.Screen().onclick() function is triggered. If you don't wait for the turtle to complete drawing the star and click again anywhere else on the canvas, the turtle will start drawing at the new clicked position leaving the star that it was drawing earlier. The previous star will look incomplete and the turtle will seem confused.

To fix this issue, let's add some lines of code:

1. After your *import* statements, create a global variable called *working* and assign it the value of *False*.

2. In the *draw(x,y)* function, we will use the variable *working* to add a logic that would prevent our turtle from drawing a new star on clicking the screen if a star is already being drawn.

3. To use the variable *working* inside the *draw(x,y)* function, we use the keyword *global* before the variable name *working* as the first line in the function. If you do not do this, the *draw(x,y)* function will not recognize the variable when it will be used in our logic.

4. Our logic will use an *if-statement* as follows:

    a. Check if *working* is *False*,

        i. assign *working* the value of *True*,

        ii. call the *draw_star(size,color,points,x,y)* function,

        iii. Then assign *working* the value of *False*.

5. Your final code should look something like this:

trinket ▶ Run ? Modules

main.py

```
import turtle as t
from random import randint

working = False

def draw_star(size,color,points,x,y):
  t.penup()
  t.goto(x,y)
  t.pendown()
  t.color(color)
  t.begin_fill()
  angle = 180 - (180/points)

  for i in range(points):
    t.forward(size)
    t.right(angle)
  t.end_fill()


def draw(x,y):
  global working

  randomSize = randint(20,200)
  randomColor = (randint(0,255),randint(0,255),randint(0,255))
  randomPoints = randint(2,5) * 2 + 1
  if working == False:
    working = True
    draw_star(randomSize,randomColor,randomPoints,x,y)
    working = False


t.showturtle()

t.Screen().onclick(draw)
```