

蒙特卡洛法求圆周率



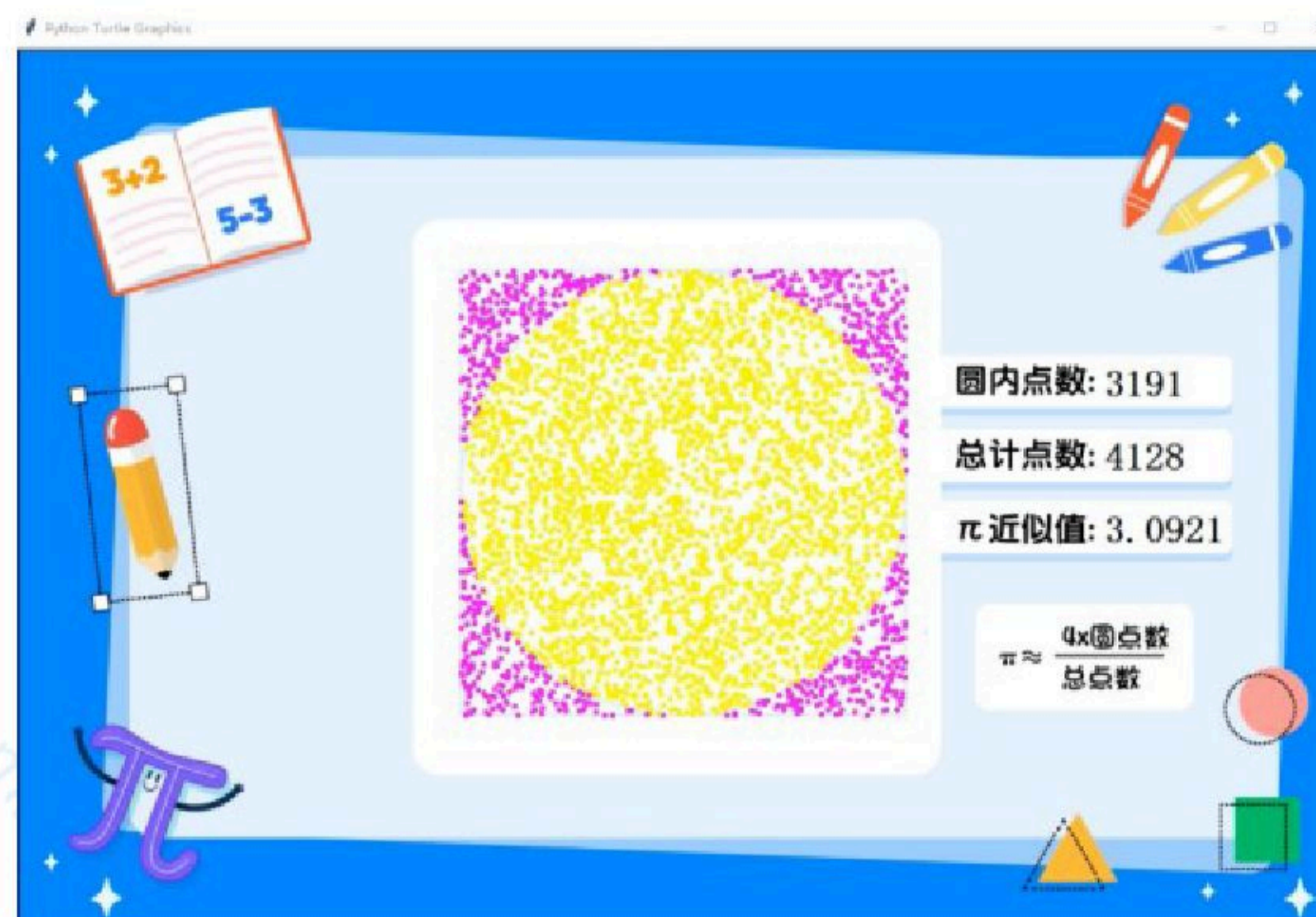


1. 探索新知

项目流程

实现随机打点计算圆周率效果

1. 了解蒙特卡洛法
2. 推导 π 的计算公式
3. 模拟随机打点过程
4. 显示结果



1. 蒙特卡洛法

蒙特卡洛法也被称为随机模拟或统计试验法，通过构造与实际问
题相似的模型，并在模型上进行随机抽样和统计分析，来研究原
问题或系统。

过程：模拟 \rightarrow 抽样 \rightarrow 估值



2. 求圆的周长

假设没有圆周率 π ，怎么测出一个圆周长呢？



滚轮法



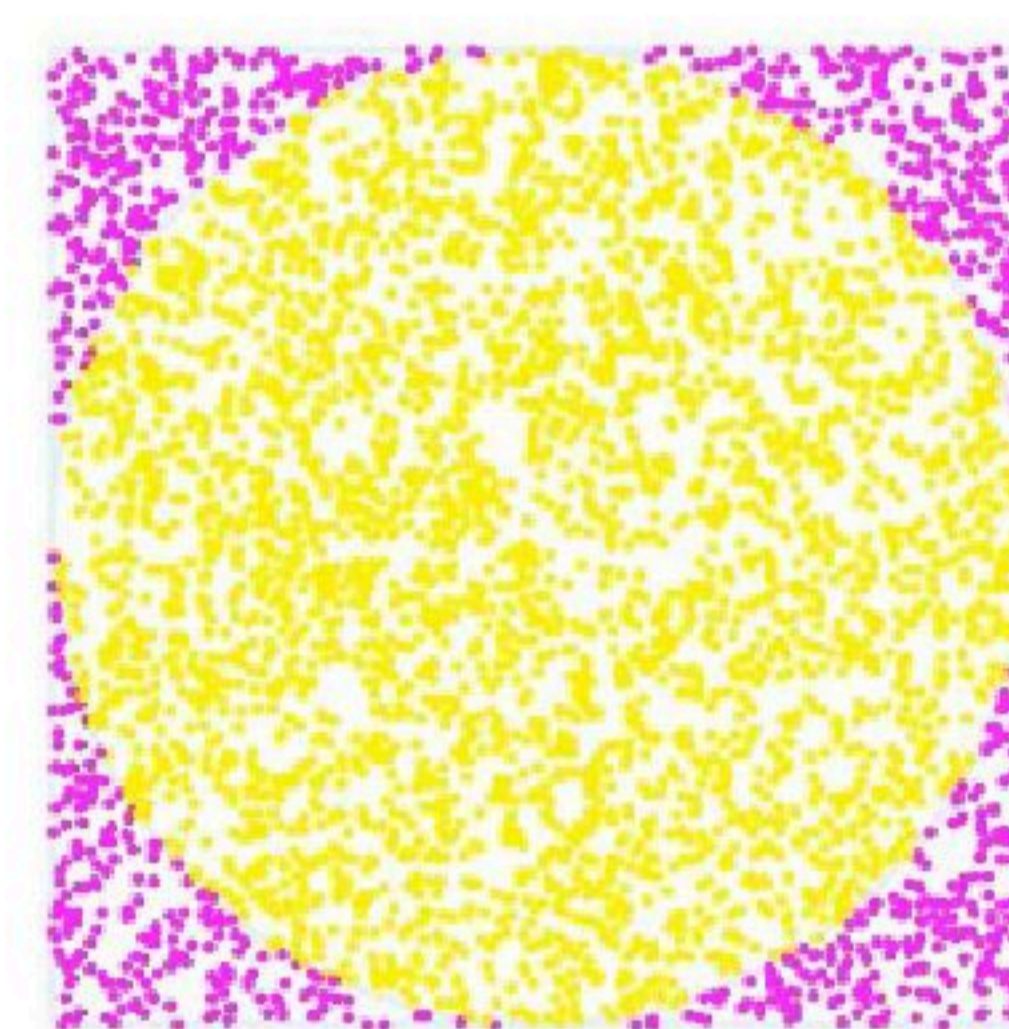
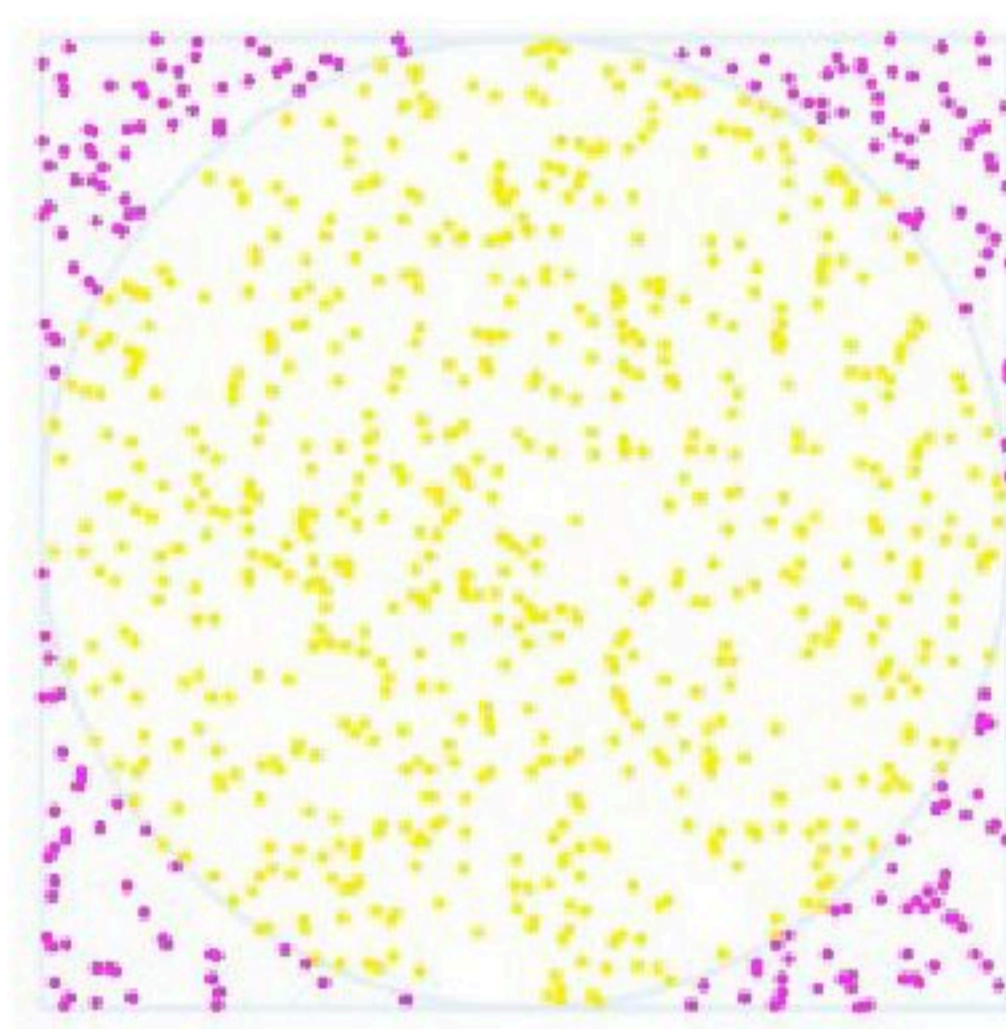
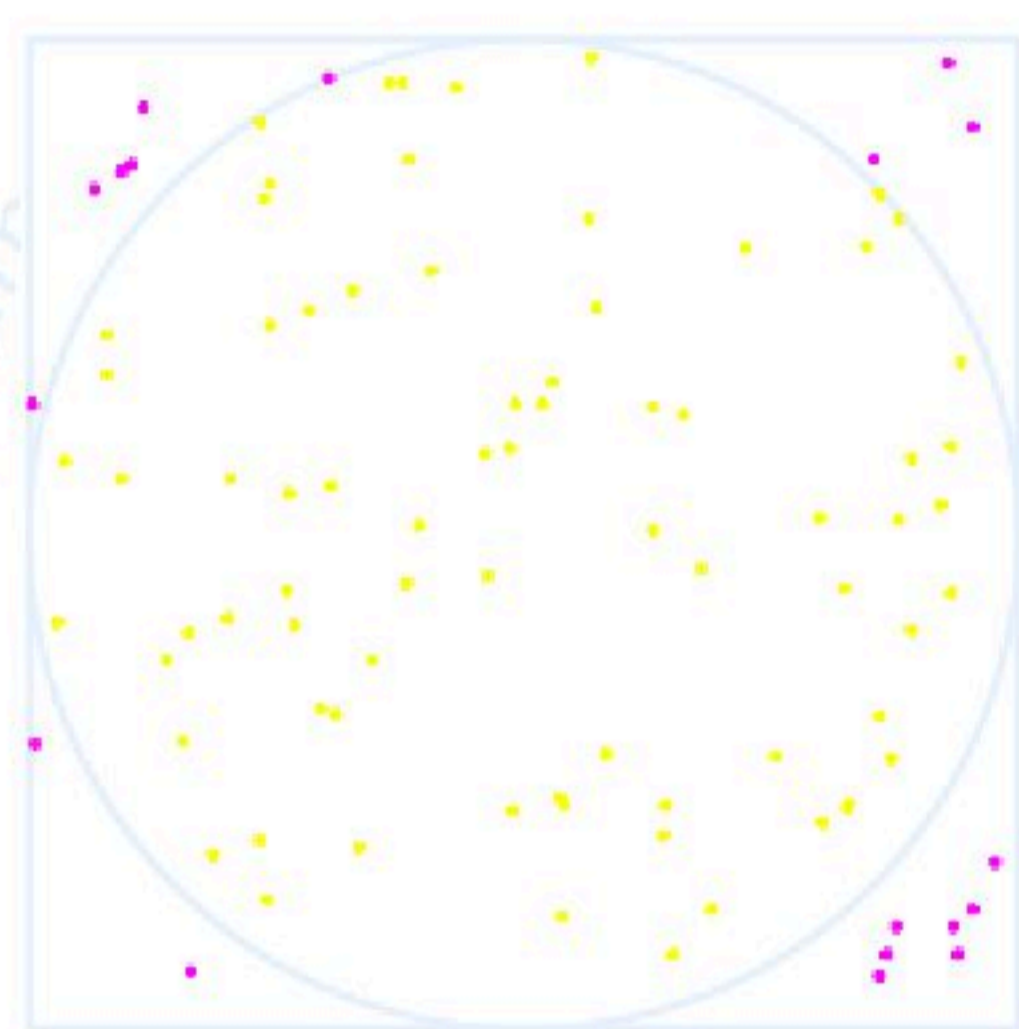
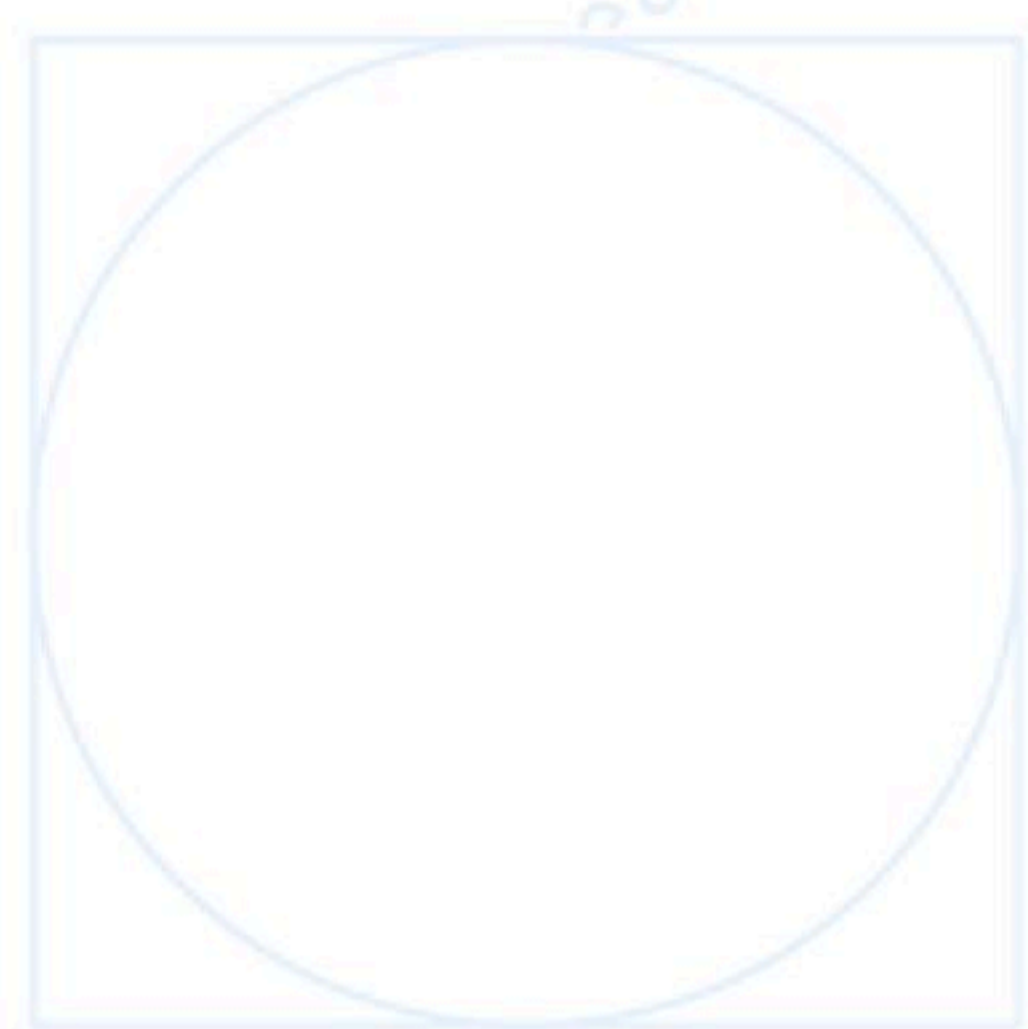
绕绳法





1. 探索新知

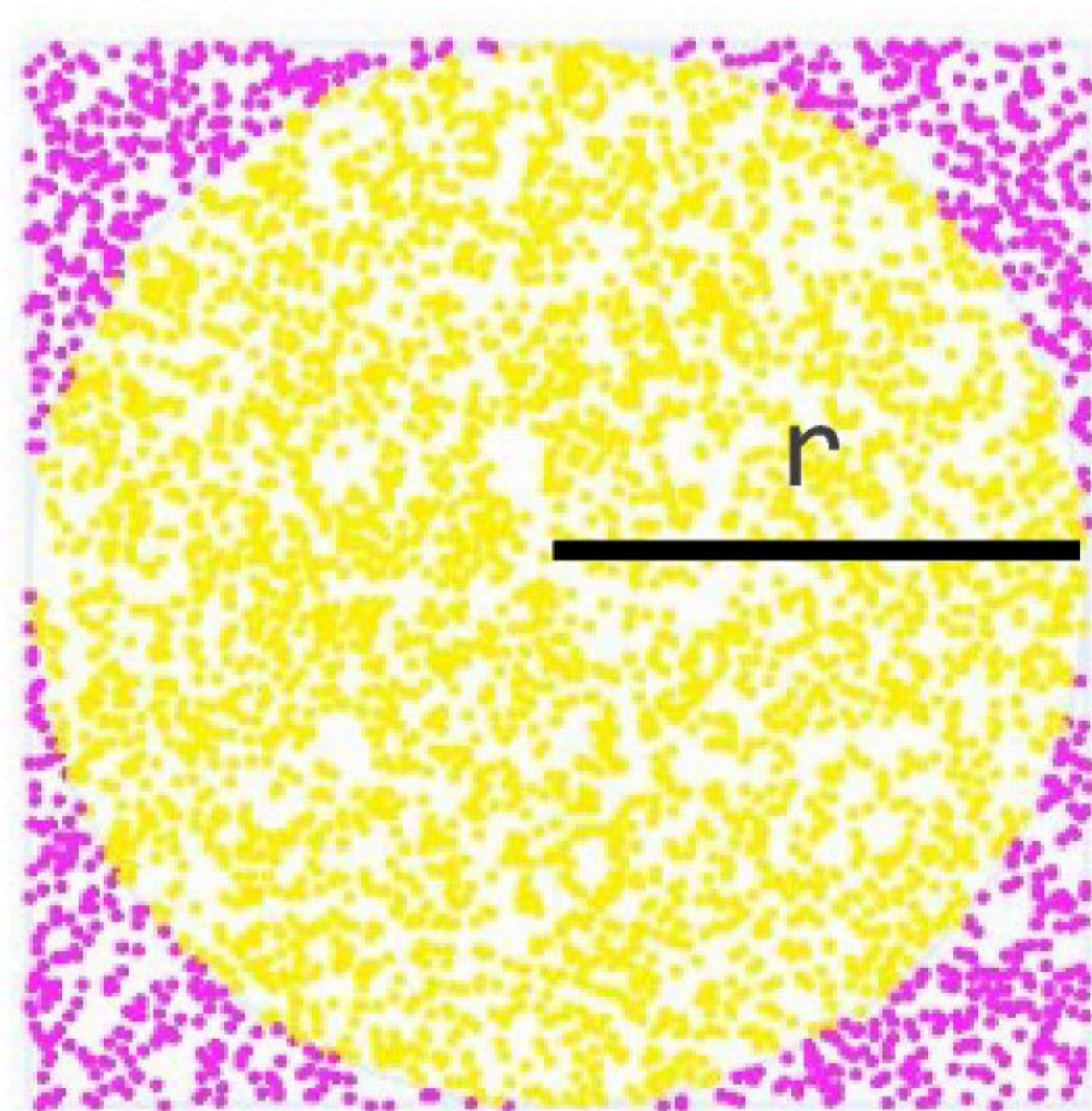
3. 蒙特卡洛法求π



不断地在平面内随机打点，有些会落在圆内，有些落在圆外。
随着点数逐渐增多，我们就可以看似圆和正方形的面积之比 \approx 点数之比

也就是说

$$\frac{\text{圆面积}}{\text{正方形面积}} \approx \frac{\text{圆内点数}}{\text{总点数}}$$



$$\frac{\text{圆面积}}{\text{正方形面积}} \approx \frac{\text{圆内点数}}{\text{总点数}}$$

$$\frac{\pi \cdot r \cdot r}{(2 \cdot r) \cdot (2 \cdot r)} \approx \frac{\text{圆内点数}}{\text{总点数}}$$

$$\frac{\pi}{4} \approx \frac{\text{圆内点数}}{\text{总点数}}$$

$$\pi \approx \frac{4 \times \text{圆内点数}}{\text{总点数}}$$



1. 探索新知

4. 设置窗口

```
from turtle import *  
bgpic("dot.png")  
setup(1200,800)  
  
done()
```

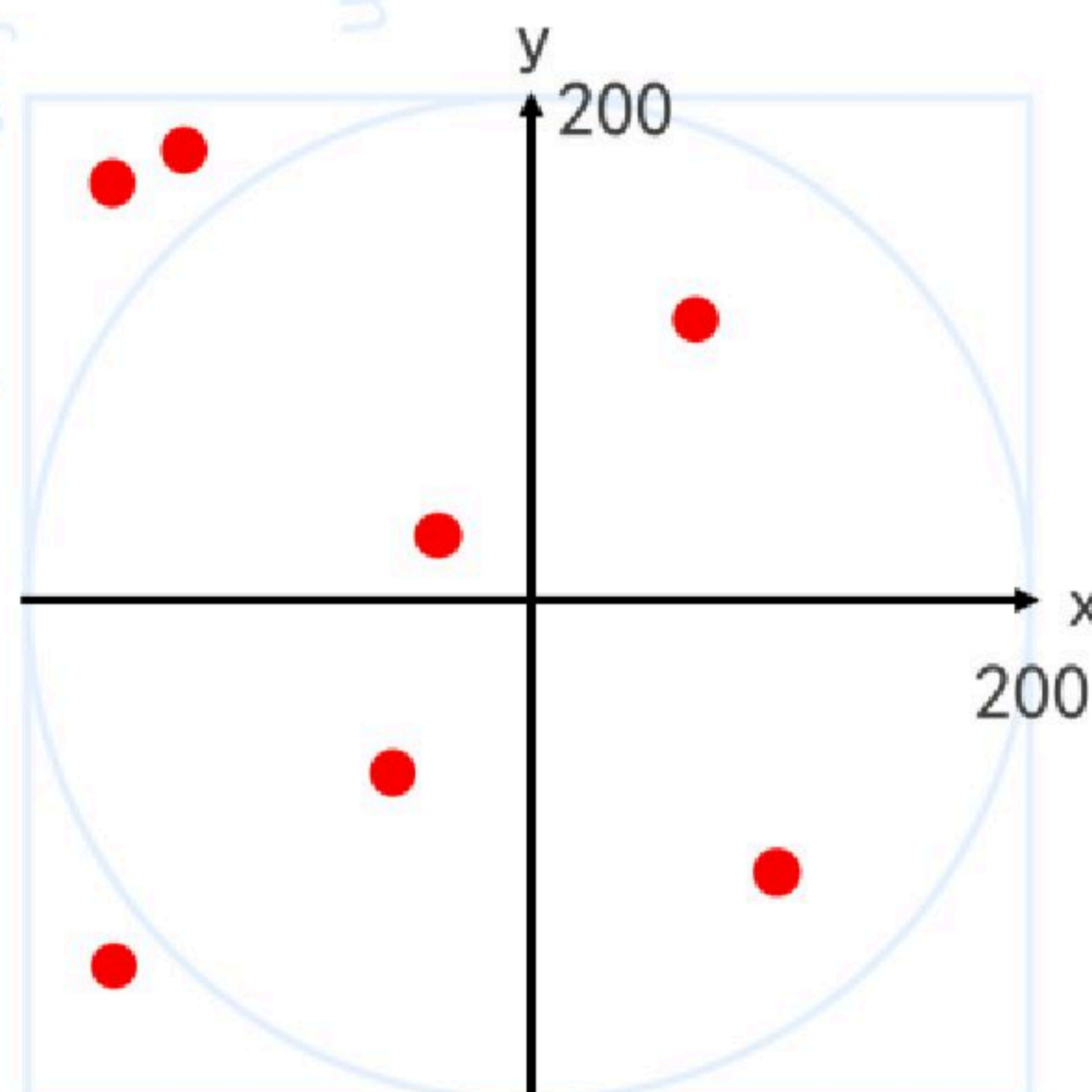


5. 随机打点

随机打点的坐标范围

x: -200 - 200

y: -200 - 200



```
from turtle import *  
import random  
bgpic("dot.png")  
setup(1200,800)  
t=Pen()  
t.ht()  
t.penup()  
  
while True:  
    x=random.randint(-200,200)  
    y=random.randint(-200,200)  
    t.goto(x,y)  
    t.dot(5)
```





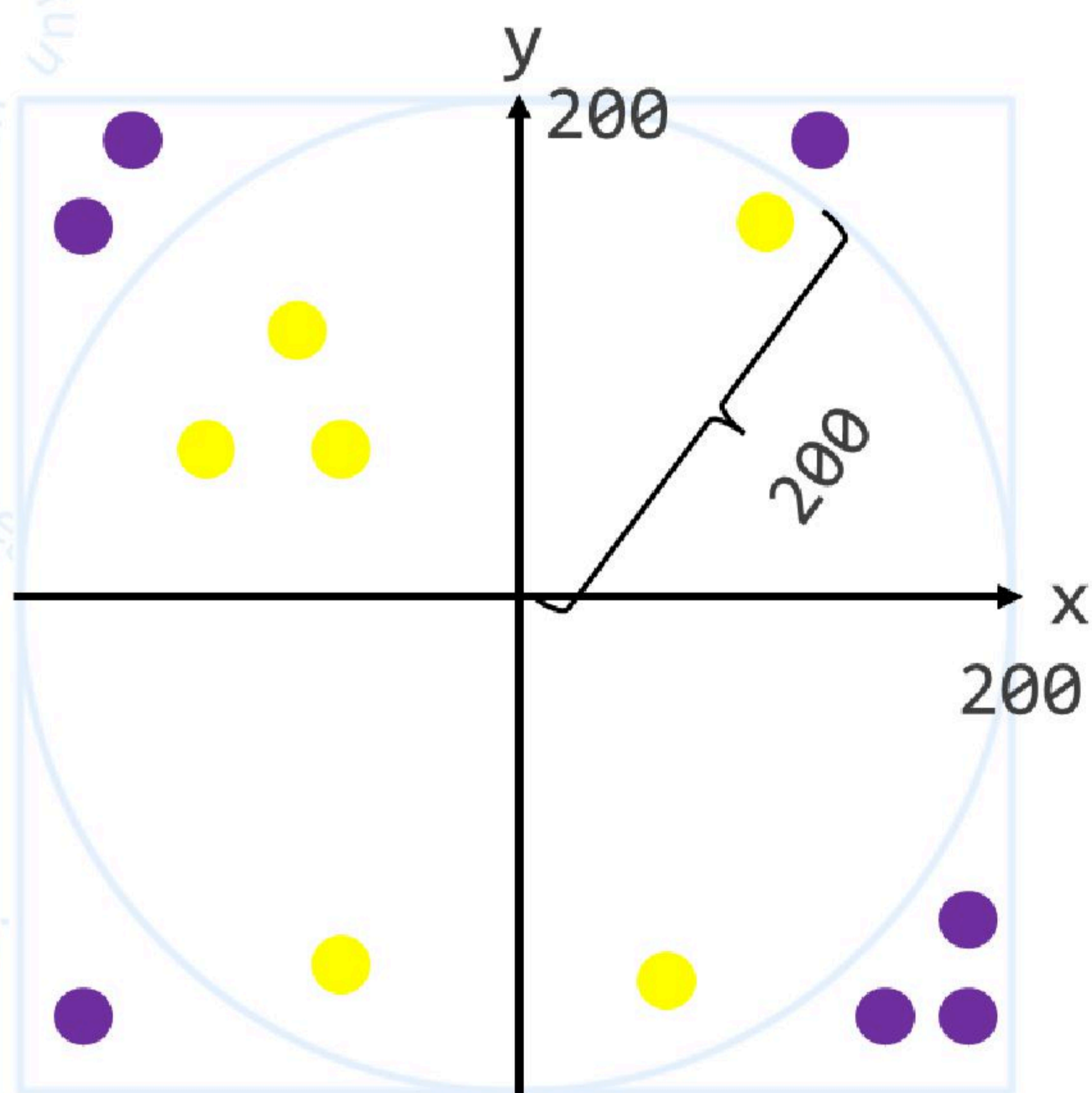
1. 探索新知

6. 点的区分

为了便于区分圆内点和圆外的点，我们给它们设置不同的颜色

如何用程序区分一个点在圆内还是圆外呢？

离中心点的距离



```
tracer(False)
```

```
.....
```

```
while True:
```

```
    x=random.randint(-200,200)
```

```
    y=random.randint(-200,200)
```

```
    t.goto(x,y)
```

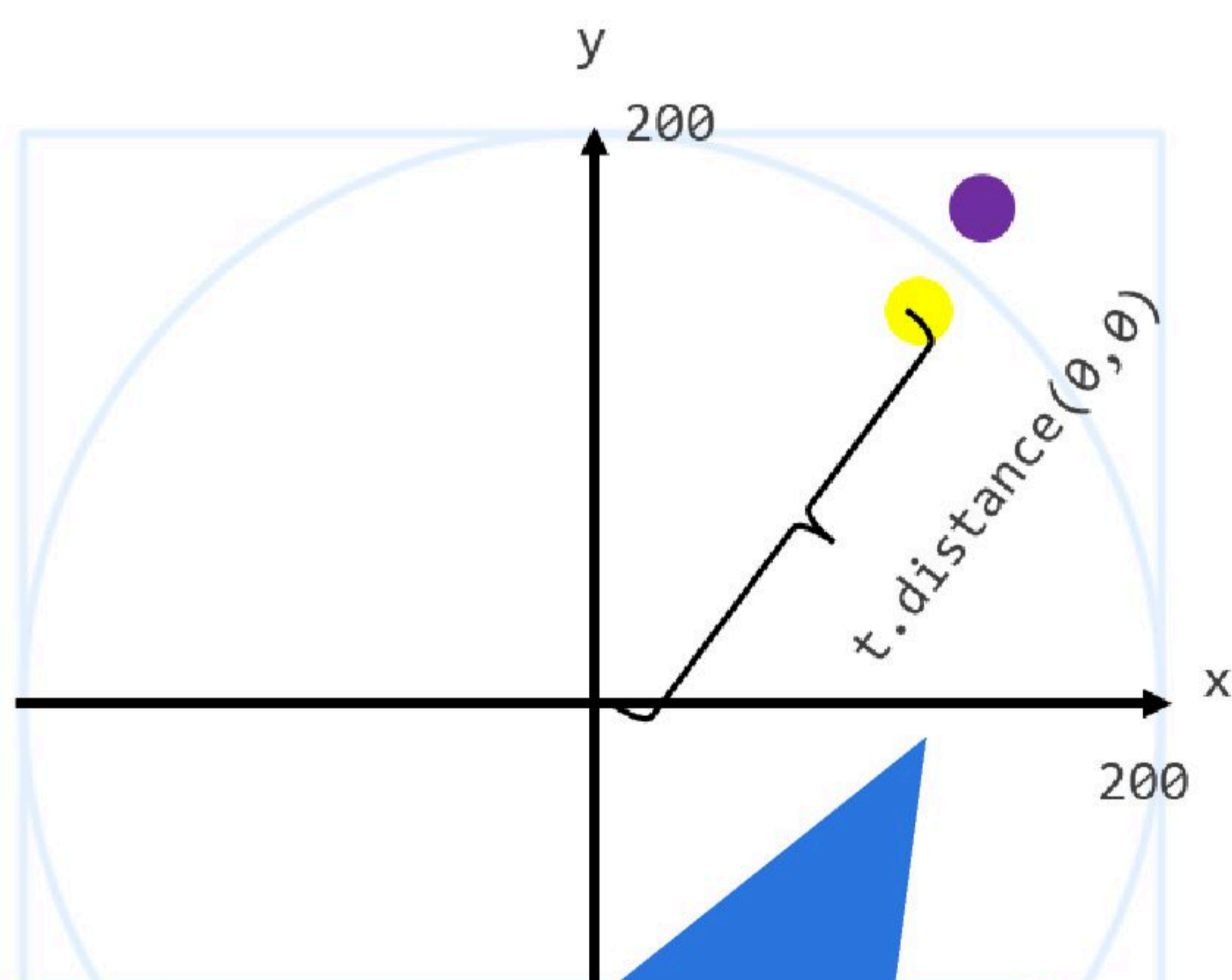
```
    if t.distance(0, 0) < 200:
```

```
        t.dot(5, 'yellow')
```

```
    else:
```

```
        t.dot(5, 'magenta')
```

```
    update()
```



distance() 获取画笔离某一点的距离



1. 探索新知

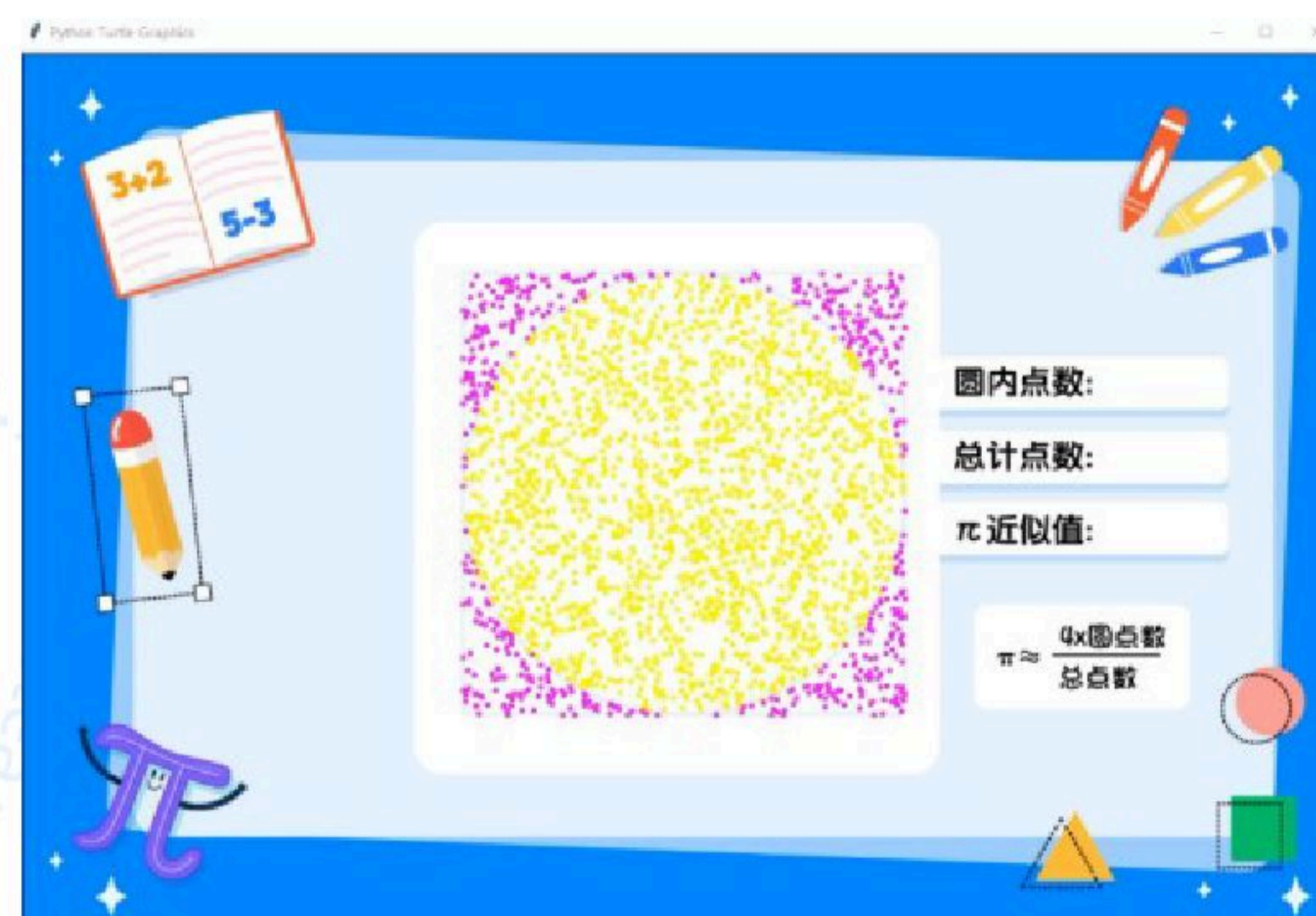
7. 数据统计

```

n=0
total=0
while True:
    x=random.randint(-200,200)
    y=random.randint(-200,200)
    t.goto(x,y)
    if t.distance(0, 0) < 200:
        t.dot(5, 'yellow')
        n+=1
    else:
        t.dot(5, 'magenta')
    total+=1
    pai=4*n/total
    update()

```

$$\pi \approx \frac{4 \times \text{圆内点数}}{\text{总点数}}$$



8. 显示结果

```

t2=Pen()
t2.ht()
t2.penup()
t2.goto(380, -50)

```

创建第2个画笔，显示数据

```

n=0
total=0
while True:

```

```

.....
.....
t2.clear()
t2.write("{}\n\n{}\n\n{}".format(n, total, round(pai, 4)),
font=("宋体", 25, 'bold'))
update()

```

在这里显示3行数据信息





2. 强化练习

1. 蒙特卡洛法求 π 的最终推导公式为 ()

A $\pi \approx \frac{4 \times \text{总点数}}{\text{圆内点数}}$

B $\pi \approx \frac{\text{总点数}}{\text{圆内点数}}$

C $\pi \approx \frac{4 \times \text{圆内点数}}{\text{总点数}}$

D $\pi \approx \frac{\text{圆内点数}}{\text{总点数}}$

2. 哪个能够判断点在圆内的是 ()

A `if t.distance(0, 0) < 200:`

B `if t.distance(200, 200) < 0:`

C `if t.distance(0, 0) > 200:`

D `if t.distance(200, 200) > 0:`

3. 绘制一个绿色的半径20的点, 正确选项是 ()

A `t.dot(20, 'green')`

B `t.dot(40, 'green')`

C `t.dot('green', 40)`

D `t.dot('green', 40,)`



2. 强化练习

4. 正确能输出Hello,world!的程序是 ()

A `print("Hello, {}!":format("world"))`

B `print("Hello, {}!".format)`

C `print("Hello, {}!".format["world"])`

D `print("Hello, {}!".format("world"))`

5. Python表达式 `round(12.56, 0)` 的值。 ()

A 13

B 12.6

C 12.5

D 13.0

3. 术语箱

| | |
|----------|----|
| distance | 距离 |
| total | 总计 |
| magenta | 洋红 |
| format | 格式 |

4. 课后挑战

修改程序，实现圆内圆外打点形状不同的效果，结合shape()和stamp()实现

shape()参数有：
arrow（箭头形状）
turtle（海龟形状）
circle（圆形）
square（方形）
triangle（三角形）
classic（经典形状）

