**Государственное бюджетное общеобразовательное учреждение "Президентский физико-математический лицей № 239"**

**«Исследование площадей множества окружностей»**

Годовой проект по информатике

Автор: Амиров Рамиль, 10-7 класс

Санкт-Петербург

2021

**Постановка задачи**

Дано множество окружностей, требуется исследовать их на пересечение, а именно распределить их пересечения, в зависимости от количества окружностей, которые пересекаются. Также найти долю занятой окружностями площади от размера всего холста.

**Уточнение исходных и выходных данных**

**Входные данные**

* Множество окружностей Задается точкой левой нижней вершины квадрата, в которую вписана окружность радиусом окружности.

**Выходные данные**

* Найти и распределить их пересечения
* Посчитать площадь объединения и пересечения

**Математическая модель**

1. **Пересечение прямых**

Метод Монте-Карло:

Сin / Call = Sin / Sout   
Отношение точек внутри круга ко всем точкам есть отношение площади этого круга к площади всего холста.

**2. Расстояние между точками**

Найдя точки пересечения, которые задают отрезок, чтобы найти его длину, нужно воспользоваться формулой:

**D = √((x1 - x2)2 + (y1 - y2)2)**

**Анализ используемой структуры данных**

Имеется массив **d,** хранящий координаты левого угла квадрата, в который вписана окружность и ее радиус.

**Выбор метода решения**

Для решения задачи я перебираю очень много точек и смотрю, в какие окружности они попадают. В зависимости от количества окружностей окрашиваю точки в разные цвета.

**Комментированный листинг**

import tkinter as tk

import time

from tkinter import \*

import random # импортирую нужные мне библиотеки

x = 0

d = [0] \* 101

XY = []

def sel():

selection = "Кол-во = " + str((var.get()))

label.config(text = selection)

x = int(var.get())

for i in range(x):

x11 = random.random() \* 850

y11 = random.random() \* 500

r = random.random() \* 100

XY.append([x11 + (r), y11 + (r), r])

c.create\_oval(x11, y11, x11 + r \* 2, y11 + r \* 2)

button.config(state='disabled') # сколько окружностей надо поставить(ползунок)

root = Tk()

var = DoubleVar()

scale = Scale(root, variable = var, orient = HORIZONTAL, bg="white", cursor = "dot", font = "georgia", fg = "red", troughcolor = "red")

scale.pack(anchor = CENTER)

button = Button(root, text="Кол-во окружностей", command=sel, height=1, bg="white")

button.pack(anchor = CENTER)

label = Label(root, width=10, height=1, bg="white")

label.pack(expand = 1, fill = Y)

c\_1 = c\_2 = -1

def show():

s = f'{var1.get()}, ' \

f'{var2.get()}'

#lab['text'] = s

def show2():

# print(var1.get(), var2.get())

if var1.get() and var2.get() == 0:

#button3.config(state='disabled')

showper()

z = 0

qq = 700 \* 1000

for i in range(len(d)):

if d[i] != 0:

z = i

lab['text'] = "Максимальное количество \n пересечений:",z, "\n", "Площадь пересечения:", (sum(d) - d[1]) / 1000, "%", "\n", "Площадь объединения:", (sum(d)) / 1000, "%"

#lab['text'] = "hooray"

elif not var1.get() and var2.get() == 1:

check()

z = 0

qq = 700 \* 1000

for i in range(len(d)):

if d[i] != 0:

z = i

lab['text'] = "Максимальное количество \n пересечений:",z, "\n", "Площадь пересечения:", (sum(d) - d[1]) / 1000, "%", "\n", "Площадь объединения:", (sum(d)) / 1000, "%"

# lab['text'] = "hooray2"

elif var1.get() and var2.get() == 1:

lab['text'] = "Выберите что-то одно"

else:

lab['text'] = "Выберите что-то"

def check():

# d = [0] \* 101

ans = 0

lab['text'] = "aaa"

for i in range(100000):

lab['text'] = str(i)

z = random.random() \* 1000

w = random.random() \* 700

res = 0

for [x1, y1, r1] in XY:

if(((z - x1) \* (z - x1) + (w - y1) \* (w - y1)) < r1 \* r1):

#print(x1, y1, r1, z, w)

res += 1

d[res] += 1

if res >= 8:

c.create\_line(z, w, z+1, w, fill="orange")

elif res >= 7:

c.create\_line(z, w, z+1, w, fill="yellow")

elif res >= 6:

c.create\_line(z, w, z+1, w, fill="lightblue")

elif res >= 5:

c.create\_line(z, w, z+1, w, fill="purple")

elif res >= 4:

c.create\_line(z, w, z+1, w, fill="blue")

elif res >= 3:

c.create\_line(z, w, z+1, w, fill="green")

elif res >= 2:

c.create\_line(z, w, z+1, w, fill="red") # окраска точек

d[0] = 0

def showper():

ans = 0

lab['text'] = "aaa"

for i in range(100000):

lab['text'] = str(i)

z = random.random() \* 1000

w = random.random() \* 700

res = 0

for [x1, y1, r1] in XY:

if(((z - x1) \* (z - x1) + (w - y1) \* (w - y1)) < r1 \* r1):

#print(x1, y1, r1, z, w)

res += 1

d[res] += 1

if res == 1:

c.create\_line(z, w, z+1, w, fill="black") # точки, принадлежащие только одной окружности

frame = Frame()

frame.pack(side=LEFT)

var1 = BooleanVar()

var1.set(0)

c1 = Checkbutton(frame, text="Объединение",

variable=var1,

onvalue=1, offvalue=0,

command=show)

c1.pack(anchor=W, padx=10)

var2 = IntVar()

var2.set(-1)

c2 = Checkbutton(frame, text="Пересечение",

variable=var2,

onvalue=1, offvalue=0,

command=show)

c2.pack(anchor=W, padx=10)

lab = Label(width=25, height=5, bg="lightblue")

lab.pack(side=RIGHT)

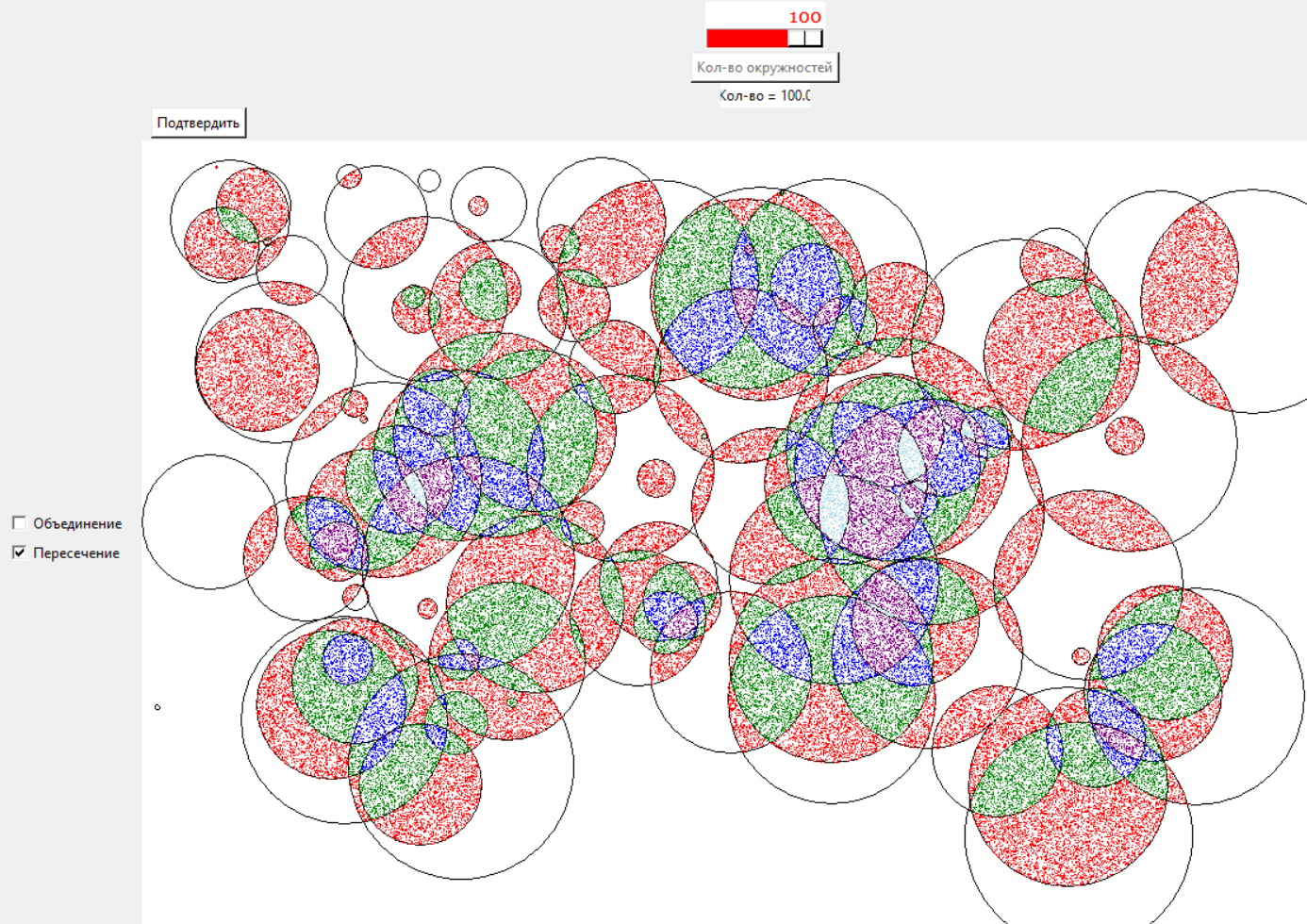
button3 = Button(root, text="Подтвердить", command=show2, height=1, bg="white")

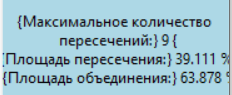
button3.pack(anchor = W, padx = 10)

c = Canvas(width = 1000, height = 700, bg = "white")

c.pack()

**Пример работы программы**





**Анализ правильности решения**

В основе решения задачи используются точные и простейшие математические модели: метод Монте-Карло, а так как точек очень много, то погрешность довольно мала.