Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

**Практическая работа № 1**

**«Графические примитивы. Фракталы. Рисунки и видео в Python»**

**Выполнил**:

ст. гр. ПРИ-120

Д. А. Грачев

**Принял**:

Преподаватель

О. Н. Шамышева

Владимир, 2021

ЦЕЛЬ РАБОТЫ

На языке Python научиться создавать фото и видео с помощью библиотек Pillow, numpy и opencv-python.

ВЫПОЛНЕНИЕ РАБОТЫ

**Задание 1**

Рисование графических примитивов на языке Python.

Листинг программы:

from PIL import Image,ImageDraw

img = Image.new("RGB",(500,500),"white")

draw = ImageDraw.Draw(img)

draw.line([(0,0),(100,100),(200,100)],"red",5)

draw.arc([(200,100),(300,200)],0,90,"blue",2)

draw.chord([(250,150),(350,250)],0,90,None,"blue",2)

draw.ellipse([(50,150),(200,200)],"green",None,0)

draw.point([(250,250)],"magenta")

draw.rectangle([(50,300),(100,350)],"magenta","green",2)

draw.polygon([(400,400),(400,500),(500,500)],"magenta",None)

img.show()

Скриншот работы программы представлен на Рис. 1.

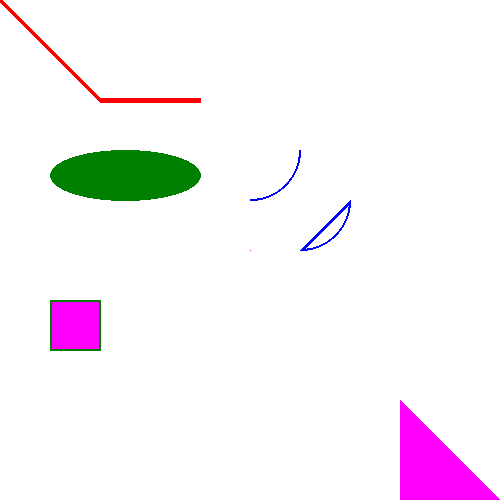


Рисунок 1. Графические примитивы

**Задание 2**

Рисование фракталов

Листинг программы:

from PIL import Image, ImageDraw

*def* sq\_fr(*A*,*B*,*C*,*D*,*deep*=10,*alpha*=0.1):

    '''

    Построение фрактала квадрата по 4 точкам\n

    deep - глубина\n

    alpha - коэффициент погружения(>1 - расширяется, <1 - уменьшается)

    '''

    if deep < 1:

        return

    draw.polygon([A,B,C,D], None, 3)

    A1 = (A[0]\*(1-alpha)+B[0]\*alpha, A[1]\*(1-alpha)+B[1]\*alpha)

    B1 = (B[0]\*(1-alpha)+C[0]\*alpha, B[1]\*(1-alpha)+C[1]\*alpha)

    C1 = (C[0]\*(1-alpha)+D[0]\*alpha, C[1]\*(1-alpha)+D[1]\*alpha)

    D1 = (D[0]\*(1-alpha)+A[0]\*alpha, D[1]\*(1-alpha)+A[1]\*alpha)

    sq\_fr(A1,B1,C1,D1,deep-1,alpha)

img = Image.new("RGB", (700,700), "white")

draw = ImageDraw.Draw(img)

A = (5,5)

B = (695,5)

C = (695,695)

D = (5,695)

sq\_fr(A,B,C,D,*deep*=994,*alpha*=0.01)

img.show()

Скриншот работы программы представлен на Рис. 2.



Рисунок 2. Фрактал

**Задание 3**

Передвижение круга по горизонтали

Листинг программы:

import numpy as np

from PIL import Image, ImageDraw

import cv2

videoDimensions = (1280,720)

videoFps = 75

videoCodec = "avc1"

videoFilename = "output.mp4"

videoLength = 3

videoFourcc = cv2.VideoWriter\_fourcc(\*videoCodec)

video = cv2.VideoWriter(videoFilename, videoFourcc, videoFps, videoDimensions)

img = Image.new("RGB", videoDimensions, *color* = "#FFBD9B")

imgDrawer = ImageDraw.Draw(img)

imgDrawer.line([(0,360), (1280,360)], *fill* = "#0A1D37", *width* = 1)

totalPercent = 0

for frameNumber in range(0, videoFps\*videoLength):

    frame = img.copy()

    frameDrawer = ImageDraw.Draw(frame)

    shift = (frameNumber / videoFps) \* (1330 / videoLength)

    frameDrawer.ellipse([(-50 + shift, 310), (shift, 360)], *fill* = "#0A1D37")

    video.write(cv2.cvtColor(np.array(frame), cv2.COLOR\_RGB2BGR))

    currentPercent = round(frameNumber / (videoFps\*videoLength)\*100)

    if currentPercent > totalPercent:

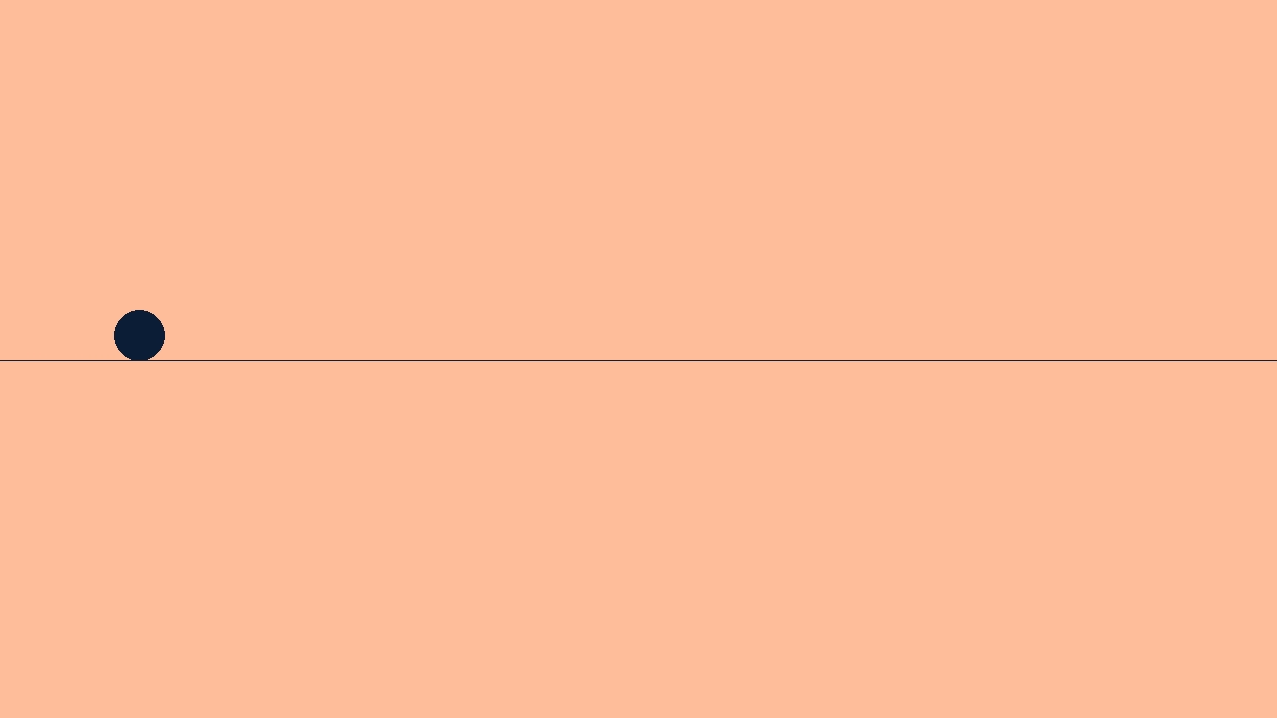
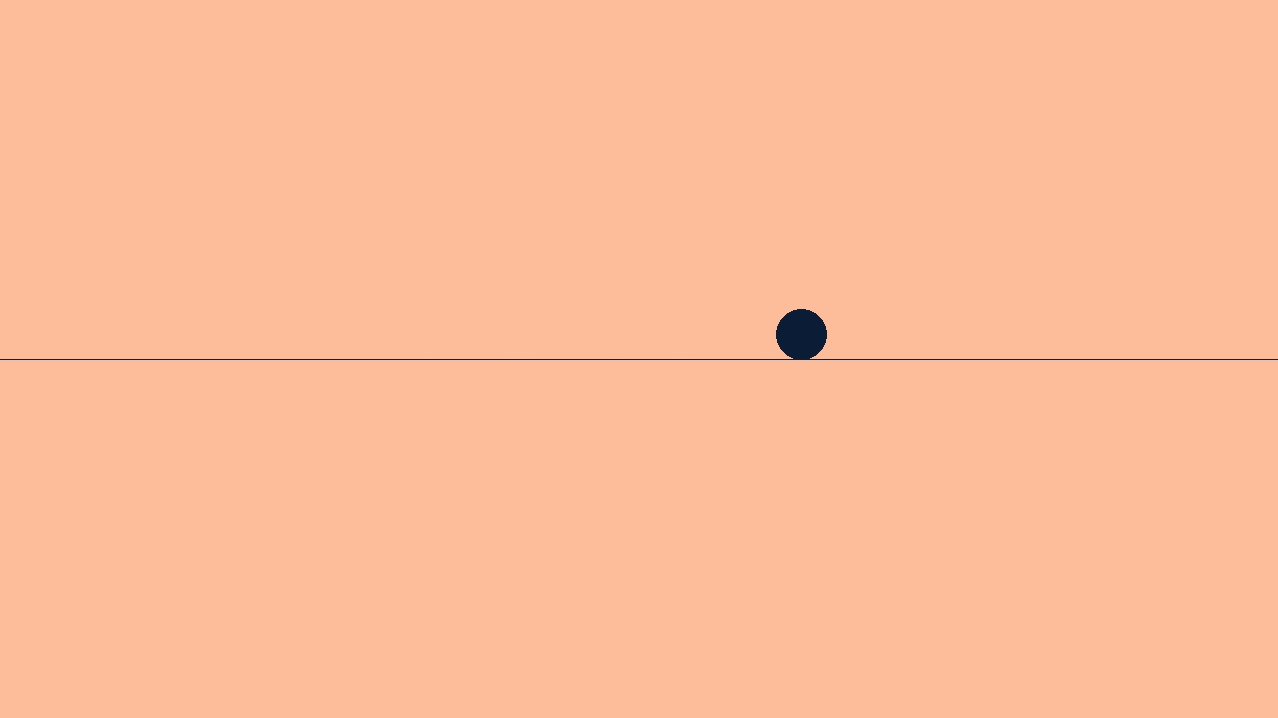
        totalPercent = currentPercent

        print(*f*"{currentPercent}% выполнено ")

video.release()

print(*f*"\nВидео с названием {videoFilename} готово!")

Скриншот работы программы представлен на Рис. 3.1 и Рис. 3.2.

Рисунки 3.1 и 3.2. Движение круга по горизонтали

**Задание 4**

Визуализация фрактала

Листинг программы:

from PIL import Image, ImageDraw

import numpy as np

import cv2

videoDimensions = (1280,720)

videoFps = 75

videoCodec = "avc1"

videoFilename = "frac.mp4"

videoFourcc = cv2.VideoWriter\_fourcc(\*videoCodec)

video = cv2.VideoWriter(videoFilename, videoFourcc, videoFps, videoDimensions)

img = Image.new("RGB", videoDimensions, *color* = "#FFBD9B")

imgDrawer = ImageDraw.Draw(img)

frame = img.copy()

frameDrawer = ImageDraw.Draw(frame)

*def* sq\_fr(*A*,*B*,*C*,*D*,*deep*=10,*alpha*=0.1,*k*=0,*totalPercent*=0):

    '''

    Построение фрактала квадрата по 4 точкам\n

    deep - глубина\n

    alpha - коэффициент погружения(>1 - расширяется, <1 - уменьшается)

    '''

    if deep < 1:

        return

    draw.polygon([A,B,C,D], None, 3)

    A1 = (A[0]\*(1-alpha)+B[0]\*alpha, A[1]\*(1-alpha)+B[1]\*alpha)

    B1 = (B[0]\*(1-alpha)+C[0]\*alpha, B[1]\*(1-alpha)+C[1]\*alpha)

    C1 = (C[0]\*(1-alpha)+D[0]\*alpha, C[1]\*(1-alpha)+D[1]\*alpha)

    D1 = (D[0]\*(1-alpha)+A[0]\*alpha, D[1]\*(1-alpha)+A[1]\*alpha)

    frameDrawer.polygon([A,B,C,D], None, 3)

    video.write(cv2.cvtColor(np.array(frame), cv2.COLOR\_RGB2BGR))

    currentPercent = round(k / 900 \* 100)

    if currentPercent > totalPercent:

        totalPercent = currentPercent

        print(*f*"{currentPercent}% выполнено ")

    sq\_fr(A1,B1,C1,D1,deep-1,alpha,k+1,totalPercent)

img = Image.new("RGB", (700,700), "white")

draw = ImageDraw.Draw(img)

A = (5,5)

B = (695,5)

C = (695,695)

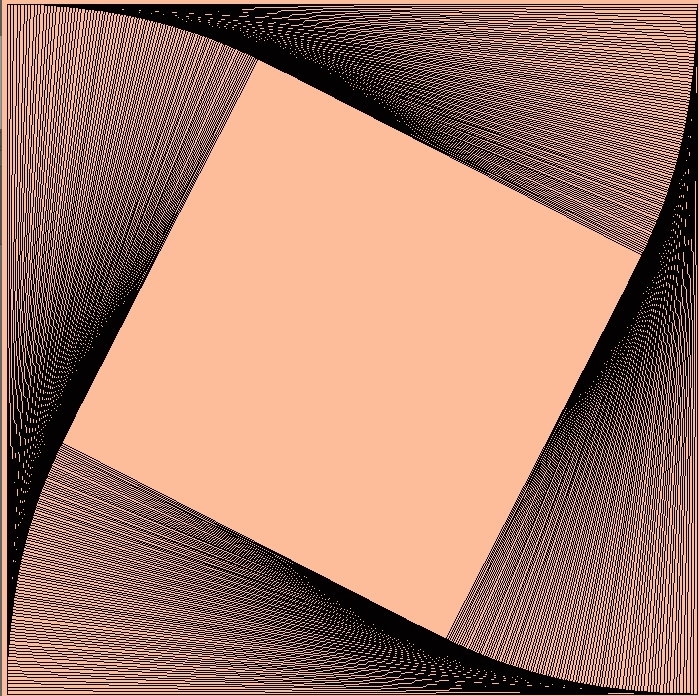
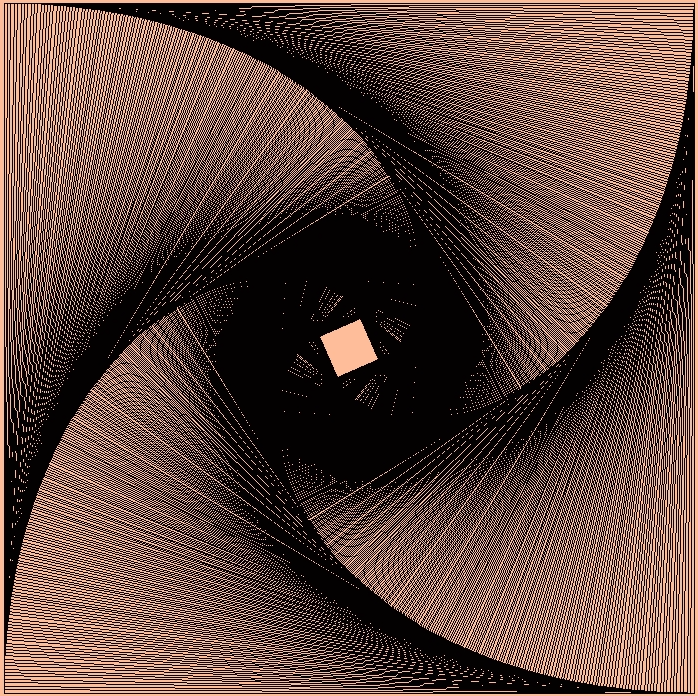
D = (5,695)

sq\_fr(A,B,C,D,*deep*=900,*alpha*=0.005)

video.release()

print(*f*"\nВидео с названием {videoFilename} готово!")

Скриншот работы программы представлен на Рис. 4.1 и Рис. 4.2.

Рисунки 4.1 и 4.2. Визуализация фрактала

ВЫВОД

В ходе выполнения работы были получены практические навыки по созданию фото и видео с помощью библиотек Pillow, numpy и opencv-python.