**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

**Лабораторная работа № 1**

Тема: построение изображения 2D кривых

Студент: Пищик Евгений Сергеевич

Группа: 80-306

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2021

1. Постановка задачи

Написать и отладить программу, строящую изображение заданной замечательной кривой. Вариант 20: y^2 = x^3 / (a - x), 0<=x<=b; a,b > 0. Обеспечить автоматическое масштабирование и центрирование кривой при изменении размеров окна.

1. Описание программы

Программа состоит из кода, создающего GUI при помощи библиотеки tkinter для ввода параметров a и b, кода который при помощи библиотеки PyOpenGL отрисовывает заданную кривую при помощи линий соединяющих пары точек из массива координат.

1. Набор тестов

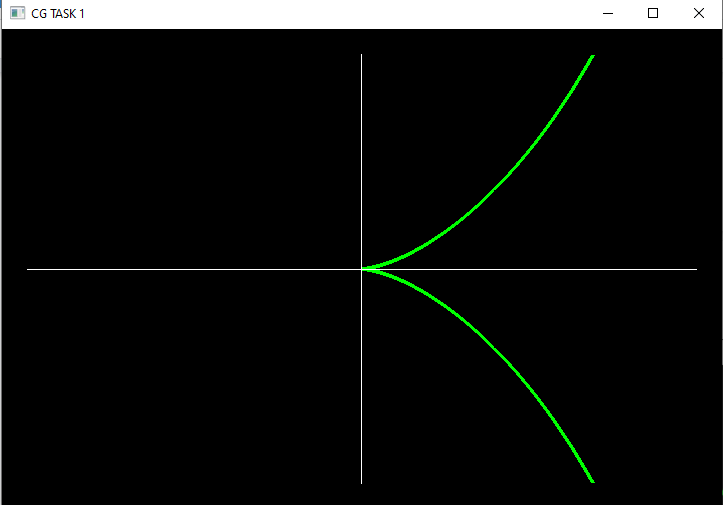
3.1 Запуск с дефолтными значениями a и b.

3.2 Изменение параметра a.

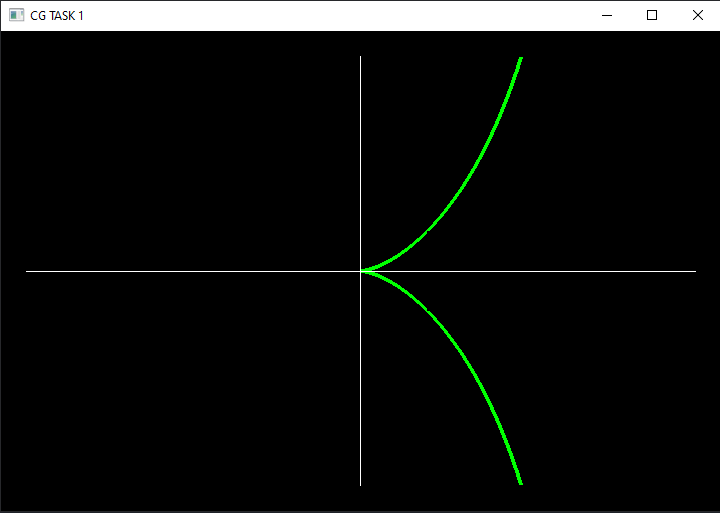
3.3 Изменение параметра b.

1. Результаты выполнения тестов

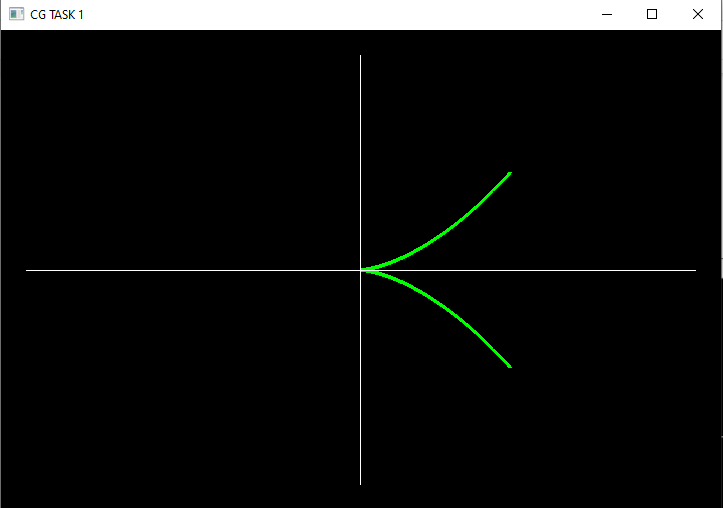
3.1 a = 1000, b = 700



3.2 a = 500, b = 700



3.3 a = 1000, b = 300



1. Листинг программы

5.1 main.py

import gui

def main() -> None:

gui.main()

if \_\_name\_\_ == '\_\_main\_\_':

main()

5.2 gui.py

import tkinter

import subprocess

from typing import NewType, Dict, Any, Tuple, List, Union, Set

def update(ent1: Any = None, ent2: Any = None) -> None:

try:

subprocess.run(f'C:/Users/jenja/anaconda3/python.exe C:\\Users\\jenja\\Downloads\\VSC\\computer\_graphics\\task\_1\\draw.py {ent1.get()} {ent2.get()}')

except Exception as \_:

subprocess.run(f'C:\\Users\\SuperPC\\anaconda3\\envs\\p39\\python.exe C:\\Users\\SuperPC\\Downloads\\VSC\\CG\\cg\_exercise\_01\\draw.py {ent1.get()} {ent2.get()}')

def main() -> None:

window = tkinter.Tk()

window.title("task 1")

window.geometry('200x200')

panel = tkinter.Frame(window, width=200, height=200)

panel.place(x=0, y=0, width=200, height=200)

lab1 = tkinter.Label(panel, text='a')

lab1.place(x=10, y=30, width=5)

lab2 = tkinter.Label(panel, text='b')

lab2.place(x=10, y=60, width=5)

lab3 = tkinter.Label(panel, text='y^2 = x^3 / (a-x), 0<=x<=b')

lab3.place(x=10, y=85, width=150)

ent1 = tkinter.Entry(panel, bd=2)

ent1.place(x=20, y=30, width=100)

ent2 = tkinter.Entry(panel, bd=2)

ent2.place(x=20, y=60, width=100)

but = tkinter.Button(panel, text="отобразить", command=lambda: update(ent1=ent1, ent2=ent2))

but.place(x=5, y=120, width=100, height=30)

window.mainloop()

if \_\_name\_\_ == '\_\_main\_\_':

main()

5.3 draw.py

from typing import Tuple

from OpenGL.GL import \*

from OpenGL.GLU import \*

from OpenGL.GLUT import \*

from typing import NewType, Dict, Any, Tuple, List, Union, Set

import math

import sys

def math\_func(a: float = 3.0, b: float = 2.0, eps: float = 0.1) -> Tuple[List[float], List[float], List[float]]:

'''y^2 = x^3 / (a - x), 0<=x<=b; a,b > 0 - function parameters; eps - delta'''

x\_lst = []

y\_lst1 = []

y\_lst2 = []

init\_x = 0.0

while init\_x <= b:

if init\_x < a:

x\_lst.append(init\_x)

init\_x += eps

for x in x\_lst:

y = math.sqrt(x \* x \* x / (a - x))

y\_lst1.append(y)

y\_lst2.append(-y)

return x\_lst, y\_lst1, y\_lst2

def draw(MARGIN: float = 50.0, EPS: float = 1.0) -> None:

'''draw own function by pair of points'''

curr\_w = glutGet(GLUT\_WINDOW\_WIDTH)

curr\_h = glutGet(GLUT\_WINDOW\_HEIGHT)

margin\_w = MARGIN / curr\_w

margin\_h = MARGIN / curr\_h

x\_lst, y\_lst1, y\_lst2 = math\_func(a=A, b=B, eps=EPS)

x\_lst = [(i / curr\_w) for i in x\_lst if (((i / curr\_w) > (-1.0 + margin\_w)) and ((i / curr\_w) < (1.0 - margin\_w)))]

y\_lst1 = [(i / curr\_h) for i in y\_lst1 if (((i / curr\_h) > (-1.0 + margin\_h)) and ((i / curr\_h) < (1.0 - margin\_h)))]

y\_lst2 = [(i / curr\_h) for i in y\_lst2 if (((i / curr\_h) > (-1.0 + margin\_h)) and ((i / curr\_h) < (1.0 - margin\_h)))]

glColor4f(0, 1, 0, 1)

glLineWidth(4)

glBegin(GL\_LINE\_STRIP)

for x, y in zip(x\_lst, y\_lst1):

glVertex2fv((x, y))

glEnd()

glBegin(GL\_LINE\_STRIP)

for x, y in zip(x\_lst, y\_lst2):

glVertex2fv((x, y))

glEnd()

glColor4f(255, 255, 255, 1)

glLineWidth(1)

glBegin(GL\_LINE\_STRIP)

for x, y in zip([-1.0 + margin\_w, 1.0 - margin\_w], [0.0, 0.0]):

glVertex2fv((x, y))

glEnd()

glBegin(GL\_LINE\_STRIP)

for x, y in zip([0.0, 0.0], [-1.0 + margin\_h, 1.0 - margin\_h]):

glVertex2fv((x, y))

glEnd()

def display() -> None:

'''display screen function'''

MARGIN = 50.0

EPS = 1.0

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT) # clear screen

glClearColor(0.0, 0.0, 0.0, 1.0) # rgba parametrs for screen

glLoadIdentity() # replace current matrix with the identity matrix

draw(MARGIN=MARGIN, EPS=EPS)

glutSwapBuffers() # double buffering

def main(a: float = None, b: float = None) -> None:

'''main program function'''

global A, B

A = a

B = b

W, H = 720, 480

OUR\_W, OUR\_H = 1366, 768

glutInit() # init glut

glutInitDisplayMode(GLUT\_RGBA) # set display mode to rgba

glutInitWindowSize(W, H) # set w,h for current window

glutInitWindowPosition(int((OUR\_W - W) / 2), int((OUR\_H - H) / 2)) # set the window start position

glutCreateWindow('CG TASK 1') # set a window title

glutDisplayFunc(display) # set display callback for current window

glutIdleFunc(display) # set global idle callback

glutMainLoop() # main loop

if \_\_name\_\_ == '\_\_main\_\_':

try:

main(a=float(sys.argv[1]), b=float(sys.argv[2]))

except Exception as e:

print(str(e))

1. Выводы

Выполнив данную лабораторную работу, я изучил работу с библиотеками PyOpenGL и tkinter, а также научился отрисовывать графические 2D объекты.

ЛИТЕРАТУРА

1. Документация PyOpenGL: [PyOpenGL documentation](http://pyopengl.sourceforge.net/documentation/index.html)
2. Документация tkinter: [tkinter documentation](https://docs.python.org/3/library/tkinter.html)