

Проект на тему “Разработка графического калькулятора”

Выполнил: Васильев Тимофей Константинович 11 “М”
Принял: Поликарпов Александр Дмитриевич

Введение

На сегодняшний день среди как учеников школ, так и преподавателей становится все более востребованной потребность в простых, удобных и эффективных вычислительных программах. Одним из видов таких программ являются графические калькуляторы - визуализаторы математических и геометрических данных. В данном проекте мы научимся работать с новыми библиотеками в языке программирования Python, которые позволят визуализировать такие данные, а также помогут в разработке интерфейса.

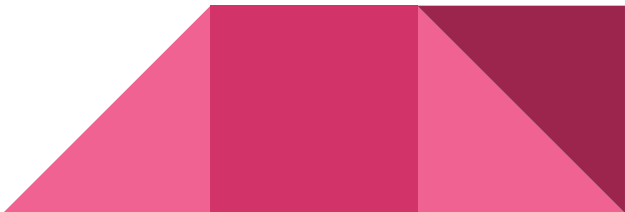


Цели и задачи проекта

Цель:


- разработать приложение со встроенным графическим калькулятором, у которого будет присутствовать возможность анимации графиков

Задачи:

- изучение существующих конкурентов в данной области
 - изучение различных алгоритмов для реализации проекта
 - разработка интерфейса программы
 - тестирование программы на наличие ошибок и неисправностей
 - документация и презентация проекта
- 

Актуальность, новизна и практическая применимость проекта

Можно с уверенностью сказать, что проект на сегодняшний день сможет пользоваться спросом среди как учеников, так и преподавателей, так как и тем и другим необходим удобный и простой инструмент для решения тех или иных задач, в основном связанных с алгеброй, матанализом, геометрией и т.д. Если кому то из учеников или преподавателей будет необходим графический калькулятор, работающий с анимациями и не требующий подключения к интернету, то данная программа должна быть одним из самых наилучших решений.



Анализ предметной области

Анализируем конкурентов в данной области:

- Desmos - самый популярный и самый удобный в применении графический онлайн - калькулятор, имеющий в своем составе огромное множество функций, которые подойдут для любого человека или организации.
- Mathway - менее популярный но все еще широко используемый веб инструмент для мат. вычислений и построения графиков
- GeoGebra - еще один инструмент, который имеет в себе больше функций с более узким применением



Техническое задание

Выделим некоторые тех. требования для приложения.

Функциональные требования:

- распознавать введенные символы с виртуальной клавиатуры
- распознавать введенные математические формулы
- заносить в отдельный текстовой файл введенные формулы для дальнейшей их обработки
- обрабатывать формулы для создания анимаций и вывода их на второй экран (после введения формулы - нажатие кнопки "Render")

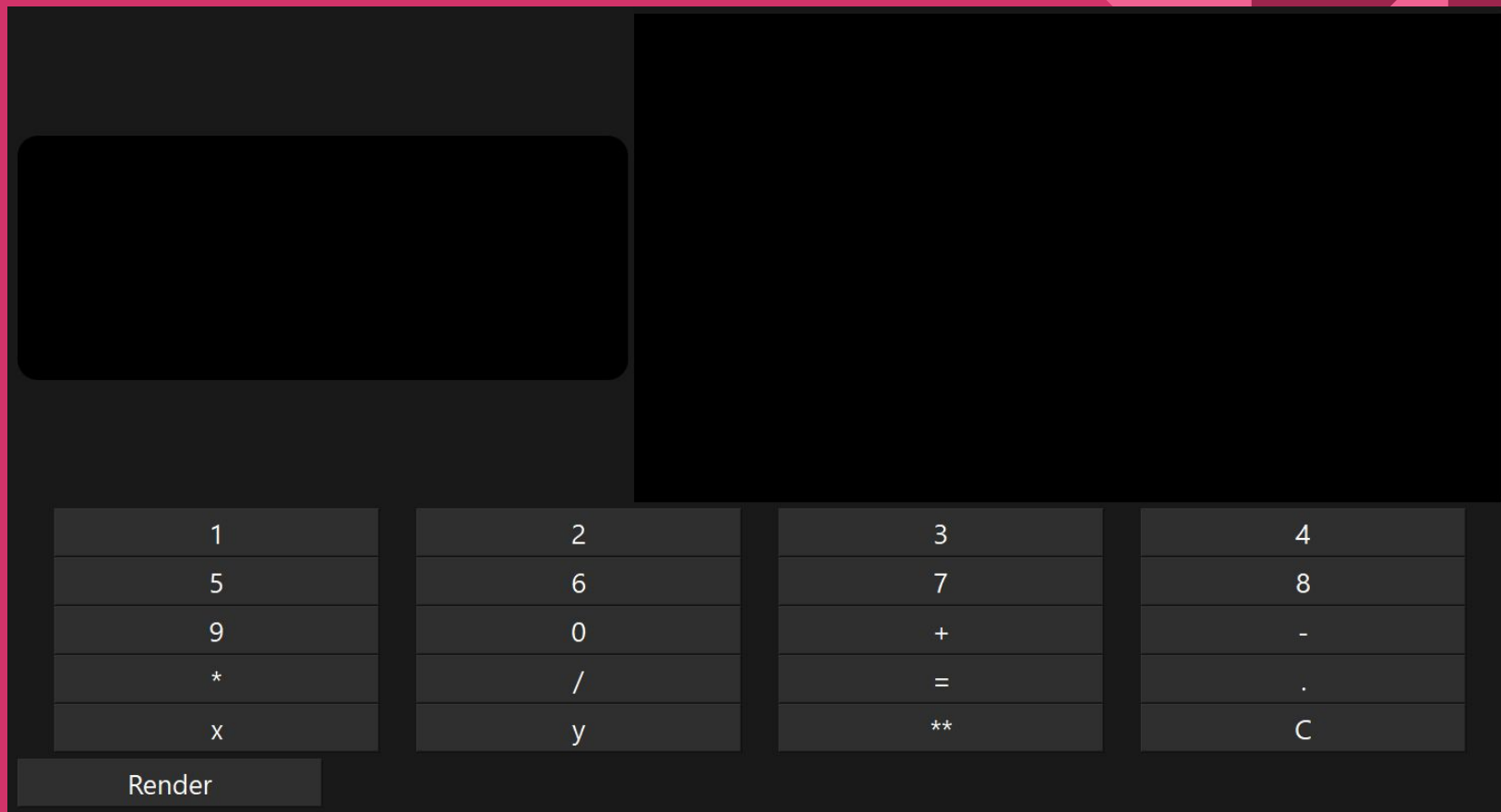


Техническое задание

Нефункциональные требования:

- интуитивно понятный интерфейс (цвет фона - темно серый, цвета окон ввода и вывода - черные)
- не слишком громоздкий шрифт и цвет текста (был выбран Segoe UI с размером в 20 пунктов, цвет текста - белый)
- окно ввода и окно вывода - ввод формулы и вывод анимации графика соответственно (цвет фона обоих окон - черный)
- при нажатии кнопки "Render" на окно вывода в кратчайшее время должна выводиться анимация графика





Общий внешний вид программы

Технологии разработки

Основной язык программирования - Python. Выбор обусловлен простотой изучения, количеством документации и библиотек, а также опытом, который был приобретен за долгое время работы с данным языком.

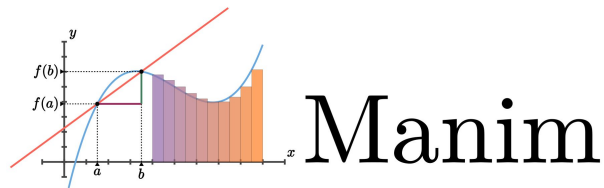
Для самого функционала были выбраны две библиотеки - PySide6 и Manim.



Технологии разработки

PySide6 - аналог PyQt6 в котором присутствует больше инструментов и удобства для работы с интерфейсом, собственно для разработки интерфейса он и был выбран.

Manim - библиотека с возможностью обработки графиков функций, геометрических объектов, статистических данных и т.д. Изначально задумывался как инструмент для работы с образовательными видео, но в силу удобства и простоты использования мы будем использовать его только в целях обработки графиков.



Логико - математический аппарат. Свойства ввода

Так как целью является разработка графического калькулятора, очевидно, что будет большая опора именно на точные вычисления и математические формулы. Как уже было описано, вместо обычной клавиатуры в приложение встроена виртуальная. Сделано это для того, чтобы пользователь не запутался в символах, необходимых для введения. Так, среди всех цифр от 0 до 9 вместе с +, -, /, * и = были добавлены кнопки возведения в степень ("**") и ввода переменных X и Y.



Логико - математический аппарат. Обработка данных

Основной модуль обработки формул, Manim, использует концепцию сцен для создания анимаций. По своей сути, это класс, в котором с помощью методов описывается расположение, анимация и передвижение тех или иных объектов. Как и было указано раньше, программа берет введенные данные из отдельного текстового файла и импортирует их в сцену, после чего обрабатывает ее и загружает в окно вывода данных. Сама анимация представляет собой файл формата .gif или .mp4, сгенерированного заданными в коде параметрами.



Ход разработки. Изучение документации

Первым делом были изучены все необходимые методы как из PySide6, так и модуля Manim. Часть основных использованных методов PySide6 включают в себя:

- QLabel() - создание окон для текста и медиа
- QVideoWidget(), QMediaPlayer() - сами окна медиа
- QPushButton() - создание кнопок



Ход разработки. Изучение документации

Модули Manim включают в себя:

- Axes() - создание оси координат
- FunctionGraph() - принимает на вход lambda функцию для прорисовки графика
- Write() - метод анимации объектов



Ход разработки. Написание кода

Весь код по итогу состоит из трех файлов - *main.py* (интерфейс, обработка текста), *anim_render.py* (обработка функций и визуализация графиков) и *rendering_queue.txt* (для хранения введенной информации). Разберем каждый из них.

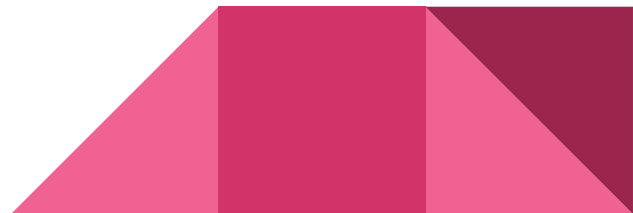
- *main.py* - в начале создается интерфейс, после чего вводятся методы `add_text`, `cancel_text` и `render_text`. Названия говорят сами за себя. После введения формулы данные переносятся в *rendering_queue.txt*.
- *rendering_queue.txt* - сюда копируется мат. формула вида $y=f(x)$, которая принимается на вход в файле *anim_render.py*.
- *anim_render.py* - данные из *rendering_queue.txt* преобразуются в `lambda` функцию, после чего отправляются в сцену для визуализации графика. После окончания рендеринга файл *main.py* инициализирует медиаплеер в основном окне.

Ход разработки. Разработка интерфейса

Разработка интерфейса упиралась в создание максимально простой и интуитивно понятной рабочей среды. Для этого было решено уподобить интерфейс виду стандартного калькулятора, но с добавлением окна вывода. Основным элементом цветовой палитры являются темные и серые цвета, которые не сильно мешают глазам и помогают сильнее выделяться белому шрифту текста. Как было сказано ранее, вся разработка интерфейса упиралась в библиотеку PySide6. Лишь малая часть (визуализация графиков) была использована из библиотеки Manim.

1	2	3	4
5	6	7	8
9	0	+	-
*	/	=	.
x	y	**	C

Render



Ход разработки. Тестирование и отладка.

Проблемы

В ходе первичного тестирования были выявлены следующие проблемы:

- разрешение обработанного видео не соответствовало окну приложения
- проигрыватель .gif изображений не работал
- видео rendering.mp4 не удавалось сохраняться в нужную директорию
- из-за ошибок создания сетки текст был слишком маленьким



Ход разработки. Тестирование и отладка. Решения

Соответственные решения:

- разрешение окна вывода было изменено на 854x480, чтобы соответствовать стандартам рендеринга модуля Manim
- проигрыватель изменен на зацикленный медиаплеер видео .mp4
- была изменена структура обработки видео, чтобы подстроиться под путь сохранения видео
- размер шрифта текста был изменен собственноручно на каждом QLabel и кнопке



Развитие и продвижение проекта

На данный момент проект находится в виде, который имеет практическую применимость только для студентов и учеников, так как функционал еще не расширен до конца. Планируется добавление обработки фигур, статистических данных, графиков НЕ функций, визуализация интегральных и дифференциальных уравнений, а также спектр инструментов для создателей контента и преподавателей. Также возможна и реклама на различных сайтах для учеников и учителей, демонстрация продукта на различных образовательных платформах и соцсетях.

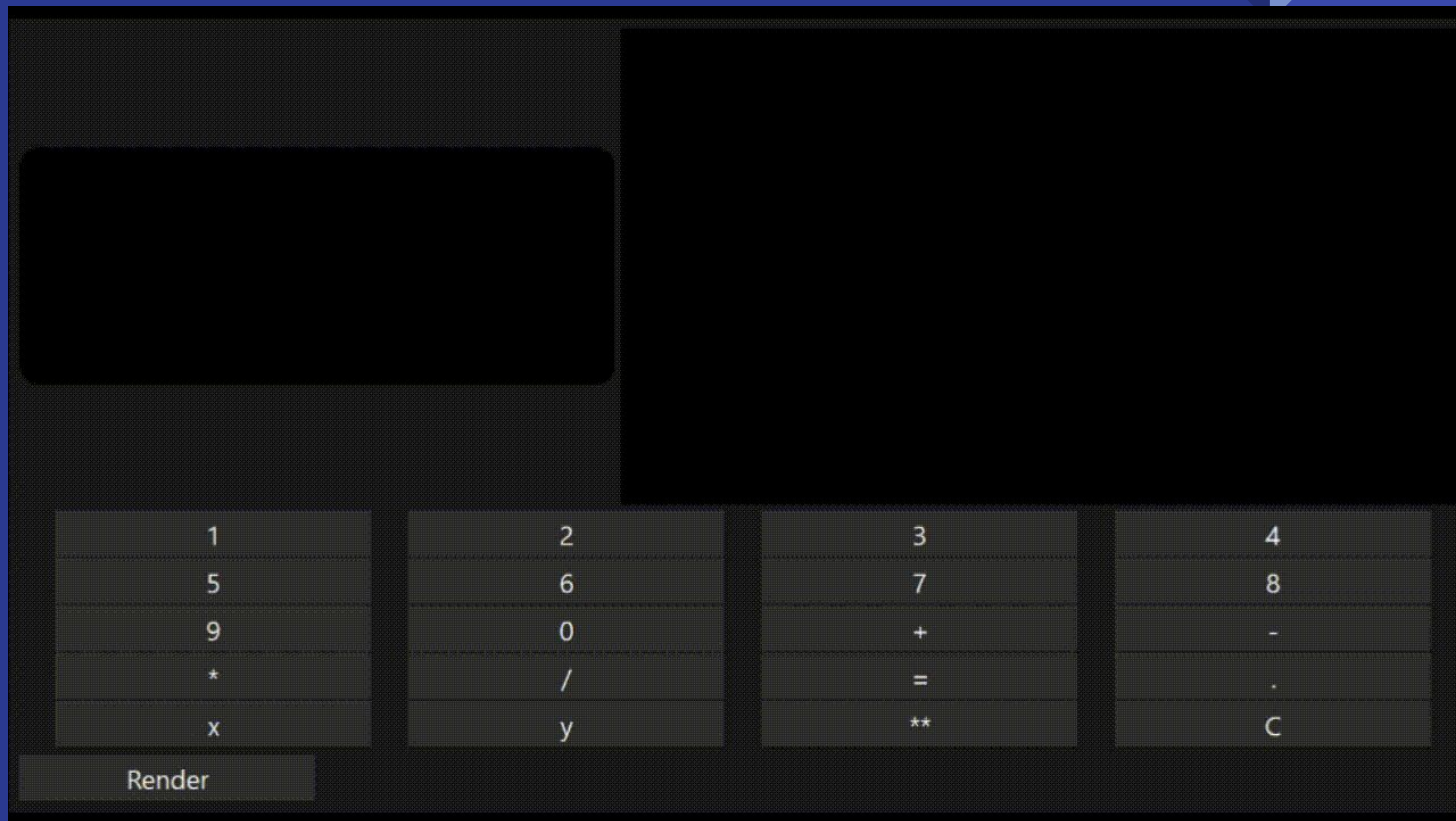


Заключение

В заключение можно сказать, что все поставленные цели и задачи были выполнены. Я научился работать с библиотеками разработки интерфейса и визуализации математических данных. Данный проект может приоткрыть мне в будущем возможности разработки программного обеспечения для образовательных целей.



Демонстрация конечного продукта



Спасибо за
внимание