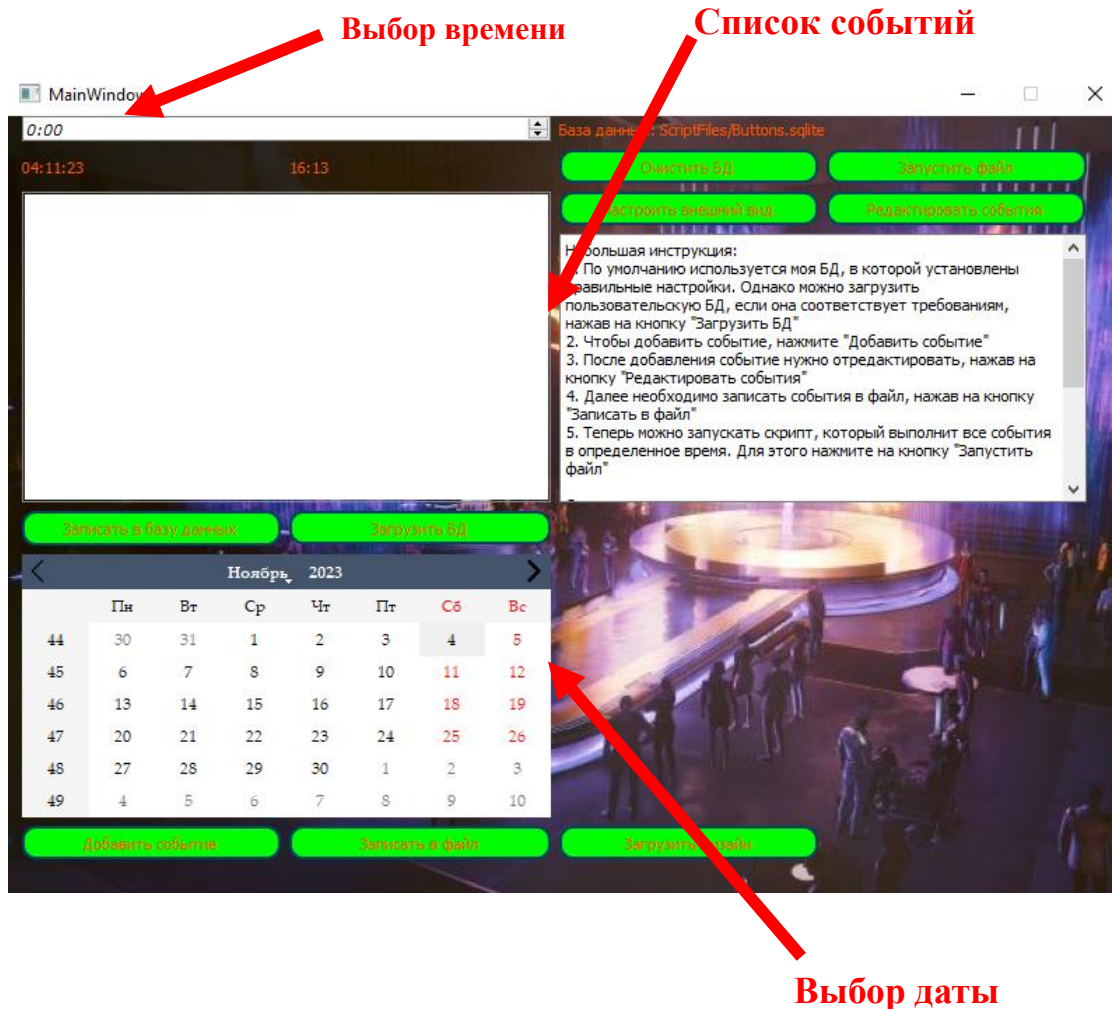


# Проект PyQt

Идея: *AutoHotkey*

Выполнил: *Клейзер Даниил*

# Введение

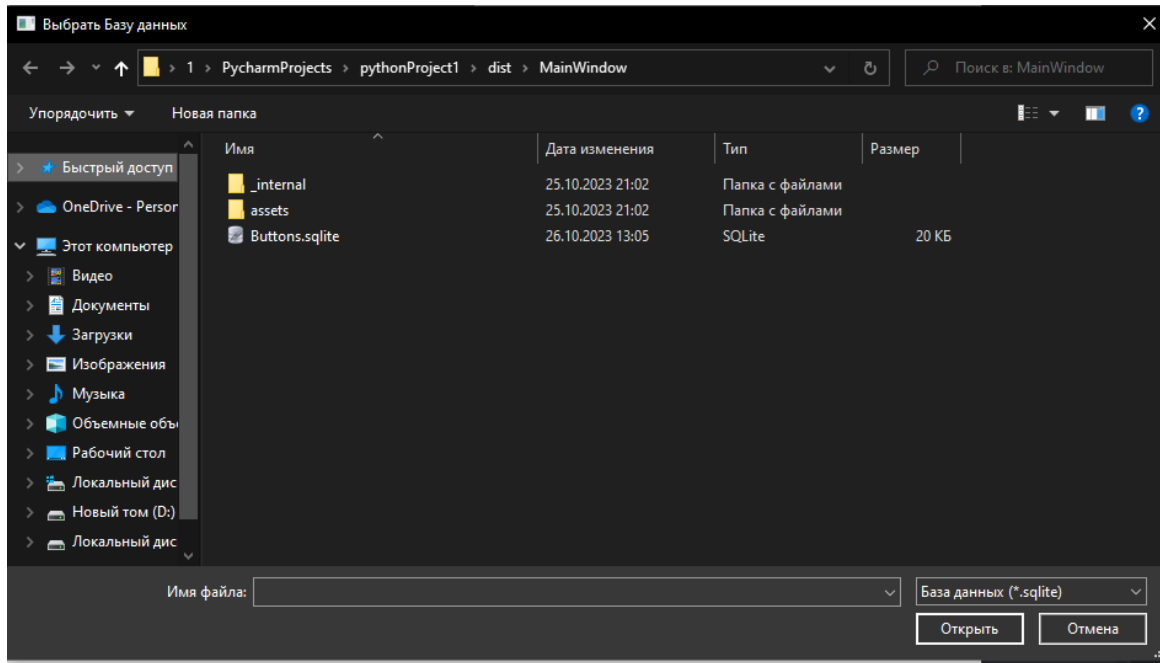


При включении приложения открывается главное окно с несколькими кнопками и календарем.

Кнопки:

- Загрузить БД
- Записать в БД – вносит событие в БД
- Добавить событие
- Записать в файл – записать все события в файл, из которого будут браться дата, время и действие
- Загрузить дизайн – загружает файл с настройкой внешнего вида
- Очистить БД – удаление всех записей из бд
- Запустить файл – запуск скрипта с событиями
- Настроить внешний вид
- Редактировать события

# Загрузка БД



При нажатии на кнопку загрузки БД откроется диалоговое окно для выбора файла. Настройки предусмотрены таким образом, чтобы не дать пользователю выбрать файл с неправильным форматом. В папке приложения уже находится база данных в нужном формате, однако можно загрузить любую.

## Добавить событие

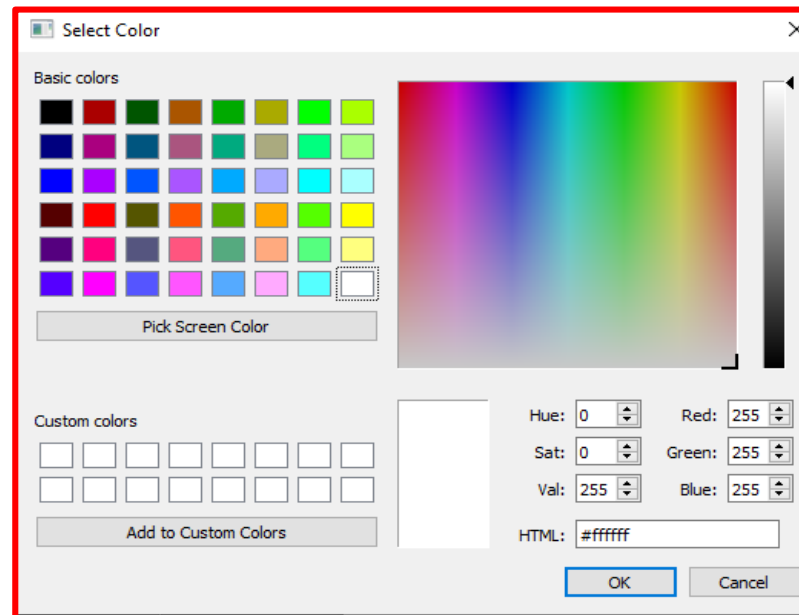
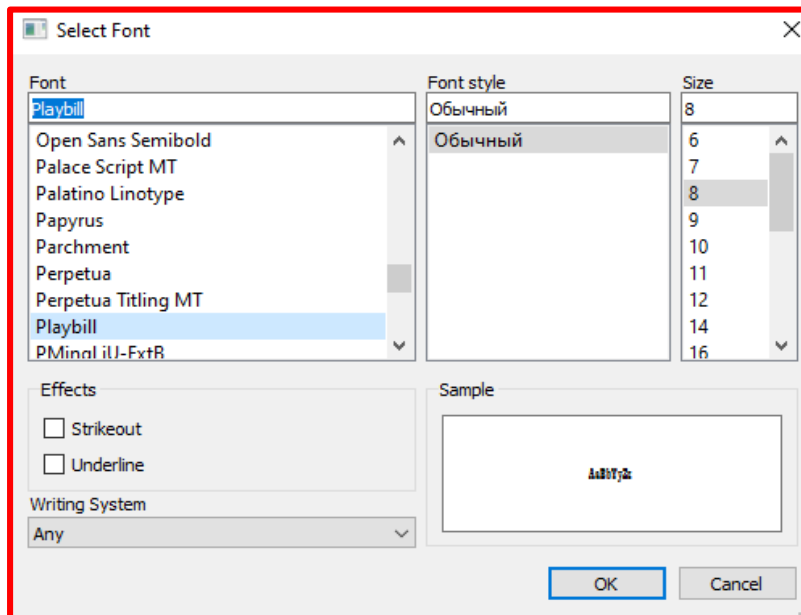
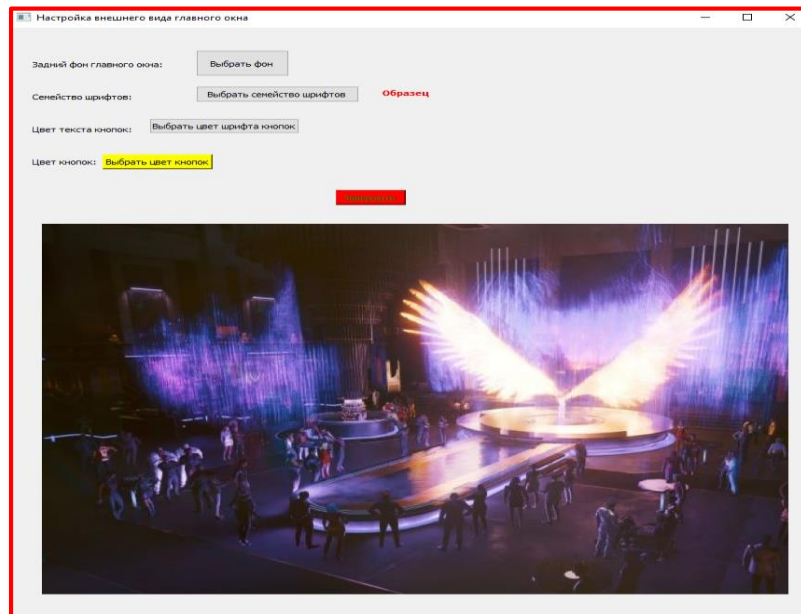


При нажатии на кнопку «Добавить событие» откроется окно, имитирующее раскладку клавиатуры. В этом окне нужно кликнуть на нужные кнопки. В конце нужно нажать на кнопку «**Завершить**».

# Настроить внешний вид

В этом окне можно настроить внешний вид главного окна. Можно изменить:

- Фоновое изображение (при помощи диалогового окна)
- Шрифт надписей на кнопках
- Цвет шрифта надписей на кнопках
- Цвет кнопок



# Редактировать события

Form

Загрузить id

Обновить Действие

кликнуть

Длительность (в секундах)

В этом окне происходит редактирование события.

Необходимо загрузить БД, в которой хранится дата и время. Далее в выпадающем списке «Действие» нужно выбрать тип взаимодействия:

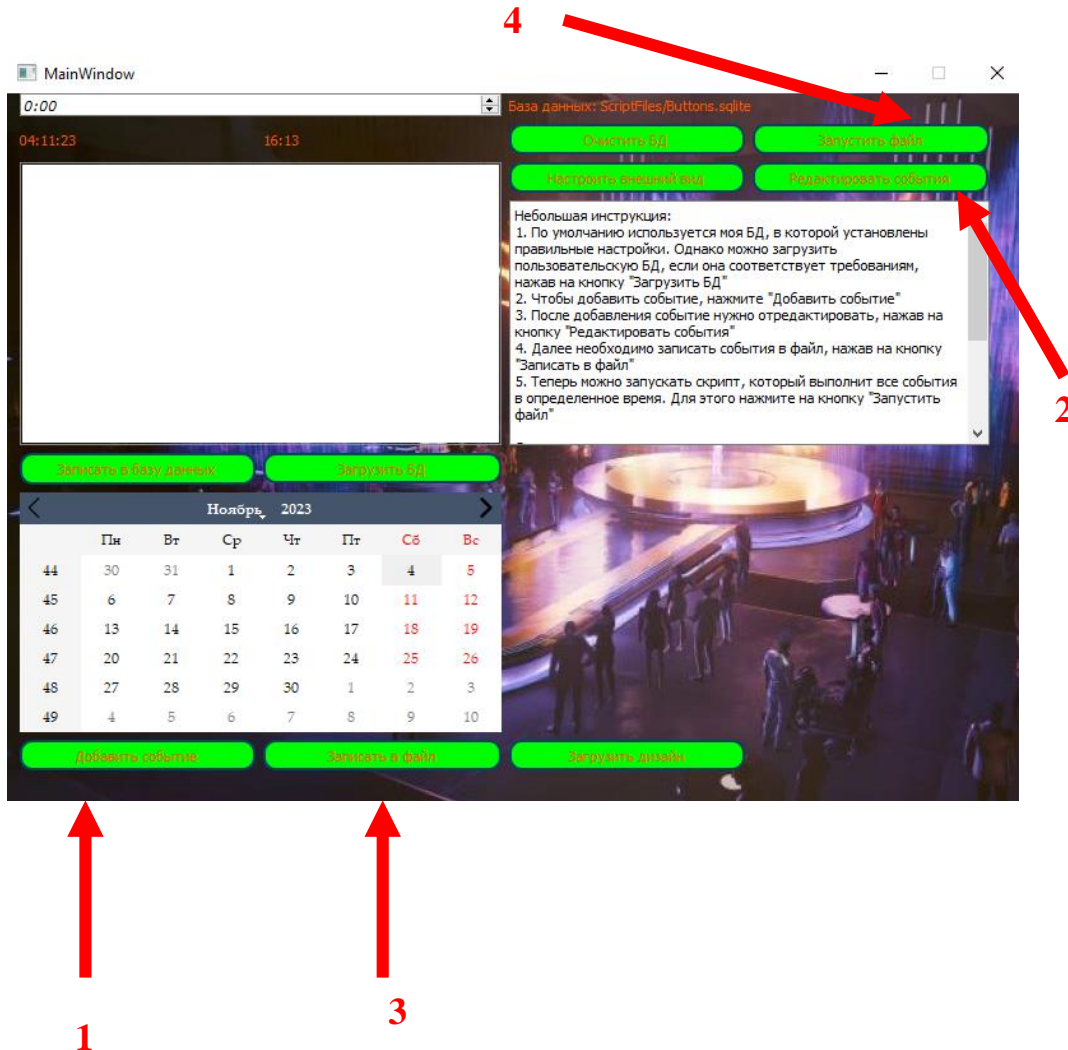
- Кликнуть, если нужно просто кликнуть на клавиши
- Удерживать. Если выбрали удерживать, то необходимо указать длительность удержания в поле ввода

# Инструкция по использованию приложения

По умолчанию в приложение загружена моя собственная БД с уже правильно настроенной таблицей, однако пользователь может добавить и свою БД.

Порядок действий:

1. Необходимо выбрать время, дату, затем нажать на кнопку «Добавить событие»
2. Далее необходимо установить тип взаимодействия в окне редактирования, нажав на кнопку «Редактировать события»
3. Далее необходимо записать весь список событий в файл, нажав на кнопку «записать в файл»
4. В самом конце нужно запустить исполняемый файл нажатием на кнопку «Запустить файл»



# Технологии

```
1 certifi==2023.7.22
2 charset-normalizer==3.3.0
3 idna==3.4
4 PyQt5==5.15.9
5 PyQt5-Qt5==5.15.2
6 PyQt5-sip==12.12.2
7 urllib3==2.0.7
8 APScheduler~=3.10.4
9 pandas~=2.1.1
10 ctypes-callable==1.0.4
```

В проекте использовались следующие библиотеки:

- PyQt5 – для реализации приложения с графическим интерфейсом
- Pandas – для преобразования данных из sqlite в формат csv
- Ctypes-callable – для взаимодействия с клавиатурой
- APScheduler – для возможности создать расписание событий
- Time – Для создания объектов времени
- Csv – для взаимодействия с csv-файлами
- SQLite3 – для взаимодействия с SQLite-файлами



## Заключение

В ходе работы над проектом я закрепил знания, полученные на лекциях, на практике.

Основные функции приложения были проверены на работоспособность. Однако все еще можно улучшить несколько деталей:

- Внешний вид программы. Можно поменять оформление второстепенных окон, добавить больше функций для редактирования главного окна
- Добавить возможность задавать действие для каждой отдельной клавиши в определенном событии
- Сделать интерфейс более дружелюбным для пользователя