ГБОУ Гимназия №116

**Тема**: «Mp3-плейер на PyQt5»

**Проект сделал:**

Куценко Дмитрий Вадимович

**Руководитель:**

Сысоев Иван Михайлович

Санкт-Петербург, 2020

Исходя из темы проекта мой руководитель согласовал со мной следующий набор функций, которые должны быть реализованы в проекте:

1. Прослушивание треков с возможностью ставить их на паузу
2. Создание новых плейлистов
3. Добавление новых треков в плейлисты
4. Изменение громкости играющего трека непосредственно в самом приложении
5. Перематывание содержимого трека
6. Удаление треков из плейлистов

**РАЗБОР КОДА**

ИСПОЛЬЗОВАННЫЕ БИБЛИОТЕКИ

****

mutagen, eyed3 – библиотеки для работы с mp3-файлами

pygame – python библиотека, используемая для создания игр, в которой мне понадобился mixer для проигрывания mp3

**\_\_init\_\_**

На этом этапе происходит инициализация основных элементов программы, присваивание им методов, выполняемых при посылании сигналов.

**self.play\_pause**

Метод, вызываемый при нажатии на кнопку play снизу по центру, изменяющий состояние нынешнего трека (проигрывание/пауза), а также останавливающий, действующие таймеры и заново их запускающий.

**self.change\_content**

Метод, вызываемый при взаимодействии пользователя с верхним слайдером, изменяющий положение, секнуду, на которой играет трек. Вызывается только, когда пользователь вручную передвинет ползунок с зажатой лкм.

**self.change\_volume**

Метод, отвечающий за изменение громкости играющего трека, которой первоначально присваивается значение 50%.

Вызывается при взаимодействии с нижним ползунком.

**self.add\_music**

Метод, используемый для выбора нового трека и занесения данных о нем в БД и вызываемый при нажатии на кнопку ADD MUSIC

**self.update\_scrollarea**

Метод, отвечающий за отображение данных о треках, добавленных в выбранный плейлист, в QScrollArea слева.

Вызывается при выполнении других методов.

**self.delete\_music**

Метод, главная и единственная задача которого – удаление треков и данных о них из БД. Вызывается при нажатии на кнопку delete (с иконкой мусорного ведра) в QScrollArea слева.

**self.start\_music**

Метод, отвечающий за начало проигрывания трека, выбранного слева в QScrollArea.

**self.show\_info**

Метод, применяемый для отображения информации о приложении по нажатию кнопки info в левом нижнем углу.

**self.add\_playlist**

Метод, созданный для добавления плейлистов и данных о них в БД и являющийся следствием нажатия кнопки ADD PLAYLIST.

**self.update\_combobox**

Метод, используемый для отображения списка всех плейлистов, добавленных пользователем, в QComboBox в левом верхнем углу, автоматически вызывается при выполнении программы.

**self.change\_playlist**

Метод, используемый для изменения данных о выбранном пользователем плейлисте, вызывается программой.

**self.next\_track**

Метод, применяемый для начала воспроизведения следующего после играющего в плейлисте трека. Вызывается при нажатии кнопки next\_track справа от кнопки play и самой программой автоматически при выполнении.

**self.previous\_track**

Аналогично с методом next\_track используется для начала воспроизведения предыдущего трека.

**self.move\_content\_slider**

Автоматически вызываемый программой метод для реализации автоматического движения ползунка во время проигрывания трека.

**self.change\_color\_theme**

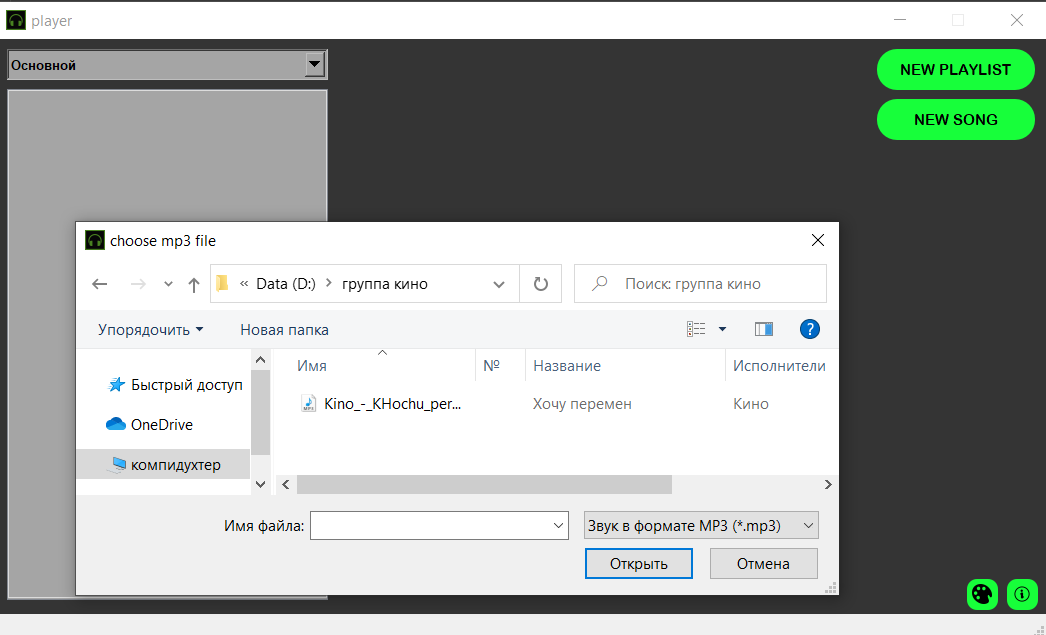
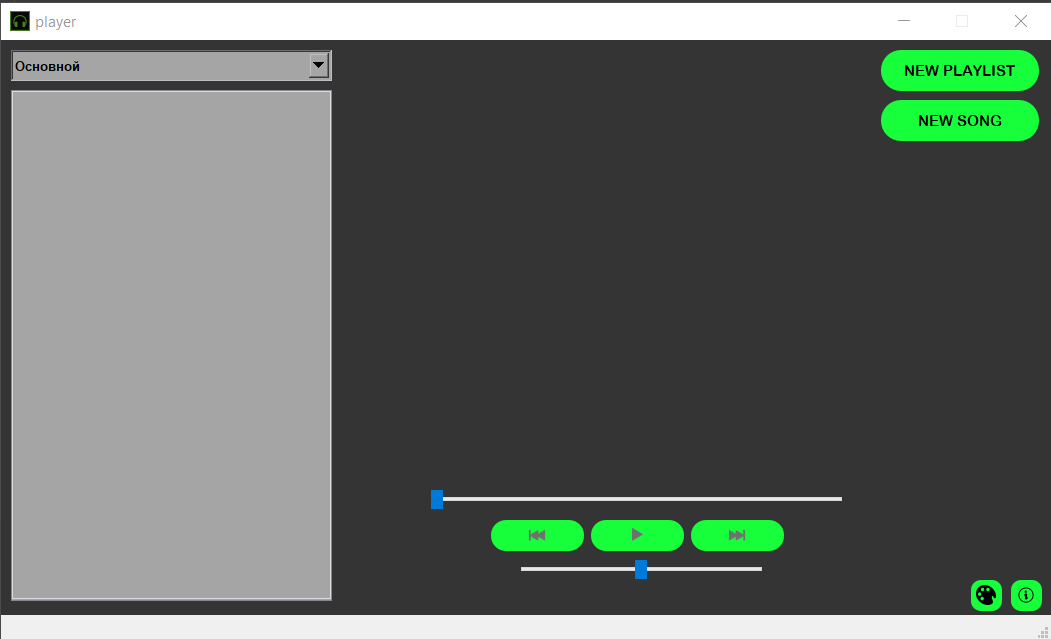
Метод, использующийся для смены цветовой темы приложения и вызываемый при нажатии кнопки change\_color в левом нижнем углу.

**self.closeEvent**

Обработка события закрытия окна пользователем, используемая для генерации окна с вопросом, подтверждает ли пользователь свое действие (закрытие окна).

**self.choose\_random\_music**

Метод, нужный для выбора рандомного трека из играющего в данный момент плейлиста.

**Скриншоты**

