HEX Editor

Автор: Никифоров Сергей.

Проект предназначен специально для разработчиков, чтобы они могли проверять корректность данных в файле, или поменять в нём значение.

# Реализация:

Всего в проекте один Python файл hex.py, 2 файла, содержащие дизайн программы: hex.ui и add\_type.ui и файл базы данных file\_types.sqlite. В файле .py 2 класса: HEXEditor() и AddTypeForm().

HEXEditor() — основная программа. Её интерфейс берётся из файла hex.ui. В ней расписаны методы для кнопок и метод обновления. Данный класс читает данные из БД для вставки подсказки.

AddTypeForm() — форма для добавления типа файла в БД. Её интерфейс — add\_type.ui. В ней, как и в HEXEditor(), расписаны методы кнопок и метод обновления. Данный класс умеет записывать в БД.

В HEXEditor() при нажатии кнопки «Добавить тип файла в БД» вызывается функция открытия окна AddTypeForm(). В свою очередь, благодаря второму классу первый может ставить подсказки в таблицу, так как AddTypeForm() записывает типы файлов в БД.

# Методы класса HEXEditor():

1. \_\_init\_\_() — выполняет \_\_init\_\_() класса QWidget(), выполняет метод initUI() и ставит переменную-защиту от лишних обновлений.
2. initUI() — загружает интерфейс и подключает виджеты к соответствующим им методам.
3. open\_file() — загрузка данных из файла в бинарном виде в таблицу (уже в виде шестнадцатиричных чисел). Привязан к кнопке «Загрузить из файла».
4. save\_file() — сохранение данных из таблицы в файл в бинарном виде. Привязан к кнопке «Сохранить файл».
5. add\_byte() — добавление строки в таблицу в её конец. Привязан к кнопке «Добавить байт».
6. remove\_byte() — удаление строки из таблицы с конца.
7. update\_data() — обновление таблицы и виджет-списка. Используется виджетами и методами add\_byte() и remove\_byte(). Использует переменную-защиту.
8. clear\_data() — возвращение программы в изначальное состояние. Привязан к кнопке «Новый файл».
9. add\_type() — создаёт окно класса AddTypeForm().

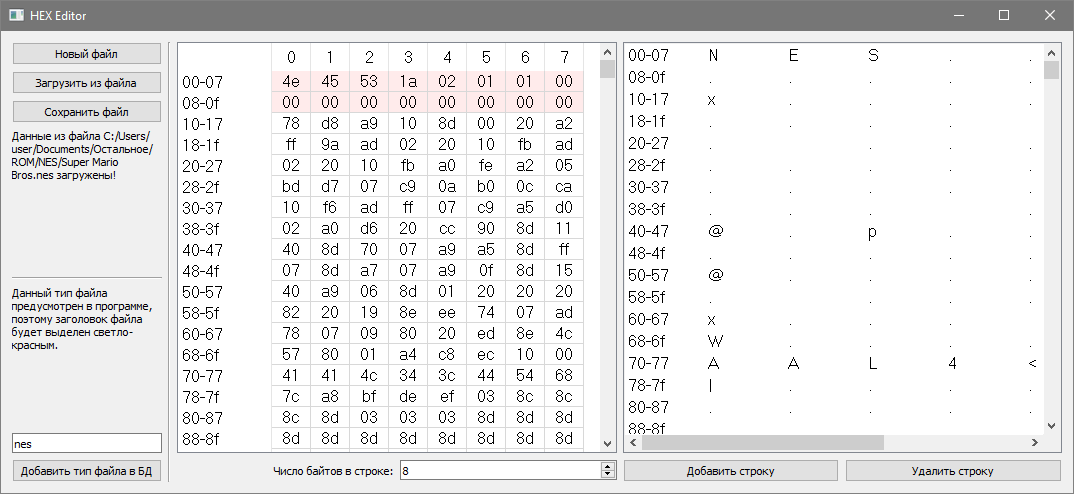
# Методы класса AddTypeForm():

1. \_\_init\_\_() — выполняет \_\_init\_\_() класса QWidget(), выполняет метод initUI().
2. initUI() — загружает интерфейс и подключает виджеты к соответствующим им методам.
3. add\_type() — записывает введённые в виджетах данные в БД. В случае существования типа файла, введённого в текстовой линии, в этой базе, он спрашивает об изменении значения конечного байта заголовка и выдаёт диалоговые кнопки. Привязан к кнопке «Добавить».
4. update\_data() — очищает оповещения и меняет в текстовой линии большие буквы на маленькие. Вызывается виджетами.
5. dialogue() — перезаписывает значение в зависимости от нажатой диалоговой кнопки. Привязан к кнопкам «Поменять значение» и «Отменить действие».

Используется язык программирования Python и библиотеки sys, PyQT5 (интерфейс) и sqlite3 (база данных). Перед использованием следует установить эти библиотеки.

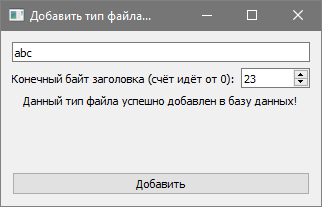
# Скриншоты:

1)



Это скриншот этой программы, в котором открыт файл игры на консоль NES “Super Mario Bros.”.

2)



Это скриншот привязанного к программе окна, в котором можно добавить или изменить существующую запись в БД. Например, тут был добавлен формат .abc со значением конечного байта заголовка 23.