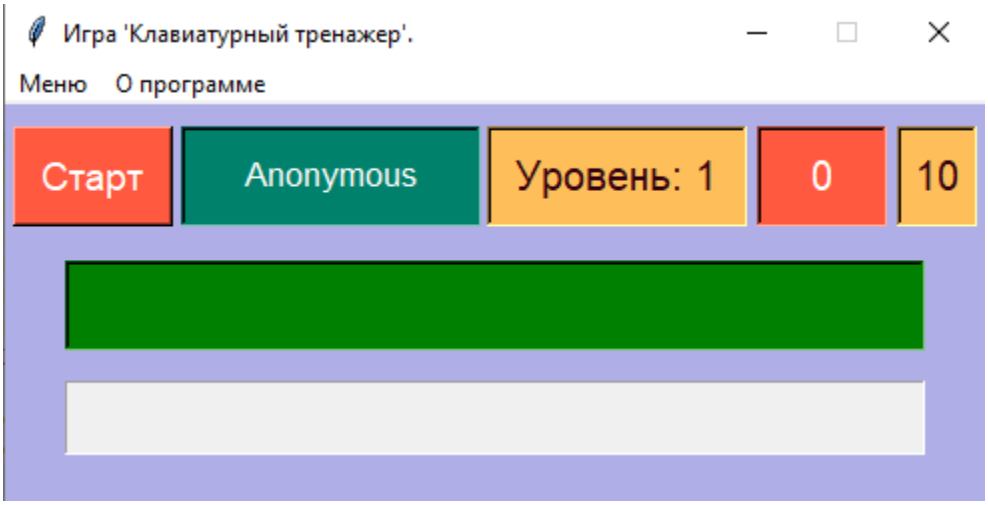
Доброго времени суток, уважаемые коллеги.

В рамках "МиниХакатона" наша команда, "**ПНЗКН Хакатон'21**", в составе:

Дмитрий Бурса (@Prospero2005) (Идея, логика, сборка)
Сергей (@Posegrey) (База данных)
Lenz (@Lenz_Gardfild) (Подбор данных, тесты)
Denis (@quqpup) (Интерфейс, отладка)

представляет проект **"Игра 'Клавиатурный тренажер'."**



Немного об описании игры:

Игра 'Клавиатурный тренажер' предназначена для развития навыков набора текста в различных раскладках.
Подразумевается, что пользователь уже умеет ставить пальцы в правильное положение, а если не умеет, одним пальцем тоже можно справиться…☺

Игра содержит 8 уровней:

- с 1 по 5 уровень - случайный набор символов,
- 6, 7 уровни - случайный выбор слова из списка слов,
- 8 уровень - предложение.

На каждом уровне предлагается по 10 наборов символов / слов / предложений.
Уровни предполагают как регистрозависимый ввод (фиолетовый фон поля), так и игнорирование регистра при вводе (зеленый фон поля).
Доступна регистрация с последующим сохранением результатов в базе и авторизация.
Зарегистрированным пользователям также доступны настройки:

- выбор уровня (1-8);
- языка набора символов / слов / предложений (английский (EN), русский (RU));
- ориентация окна (горизонтально (H), вертикально (V));
- выбор темы (светлая (light), темная (dark));
- отображение предупреждений при новой игре и регистрозависимых уровнях.

В игре реализован редактор тем оформления, где пользователь может изменить существующие темы оформления, и на их основе создать свою, особенную, для последующего использования.
Зарегистрированный пользователь может сохранить темы в файл .json и при следующем запуске файл подгрузится в список тем. Обычный пользователь может пользоваться темами и изменять существующие, но только в рамках текущего запуска.

ВНИМАНИЕ! Если вы создаете свою тему - обязательно меняйте название и псевдоним темы.

Пользователь **Anonymous** - зарезервирован.

После прохождения каждого уровня предлагается продолжить игру.

Если пользователь не продолжает - ставится пауза и при следующем старте в пределах одного запуска игра продолжится с этого уровня. У зарегистрированного пользователя текущий уровень сохраняется в базе данных.

При проигрыше у зарегистрированного пользователя игра возобновляется с текущего уровня, незарегистрированный пользователь начинает с 1 уровня.

Ограничение на длину логина - 12 символов.
Ограничение на длину пароля - 32 символа.
Ограничение на длину названия темы / псевдонима темы - 25 символов.

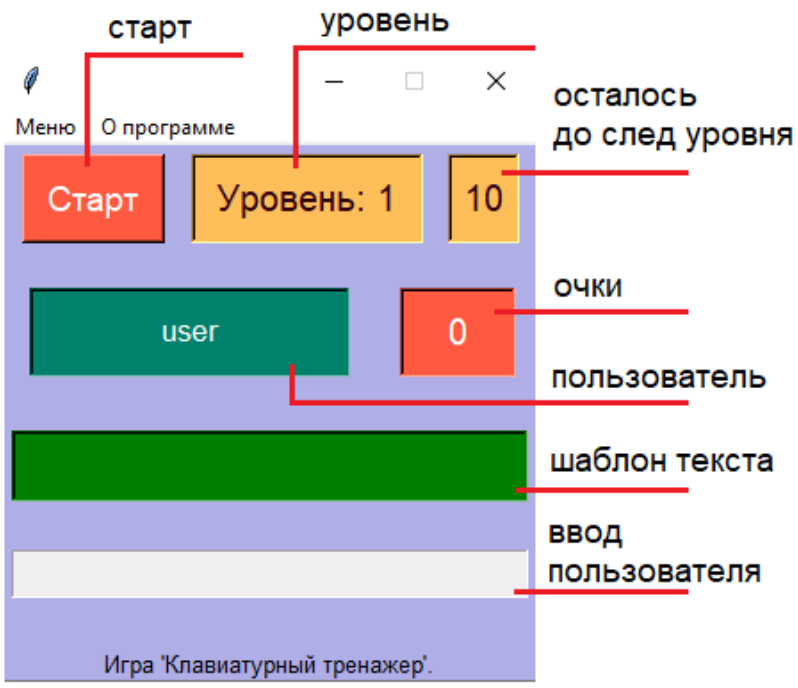
Идея была предложена **@Prospero2005** и активно обсуждалась всеми участниками до начала хакатона. Предлагались различные варианты оформления **(@quqpup**), реализация базы данных (**@Posegrey**), различные уровни **(@Lenz_Gardfild**). Позже была предложена, и в процессе разработки реализована, идея создания редактора тем оформления.

К началу хакатона были определены направления разработки и команда приступила к написанию кода.
В качестве оболочки GUI был выбран модуль **tkinter**, как наиболее простой в реализации и не требующий установки дополнительных внешних библиотек.
Для базы данных был выбран модуль **sqlite3**, также входящий в стандартную библиотеку Python.
Игра реализована в стиле ООП.
Окна авторизации / регистрации, настроек, редактора тем было решено реализовать отдельными классами.
Обработка таймера реализована на основе модуля **time** отдельным процессом с использованием модуля **threading**, случайный выбор символов сделан на основе **random**, для чтение, записи и обработки файлов использовались модули **json** и **os**, для генерации пароля использовался модуль **hashlib**, также входящий в состав стандартной библиотеки.

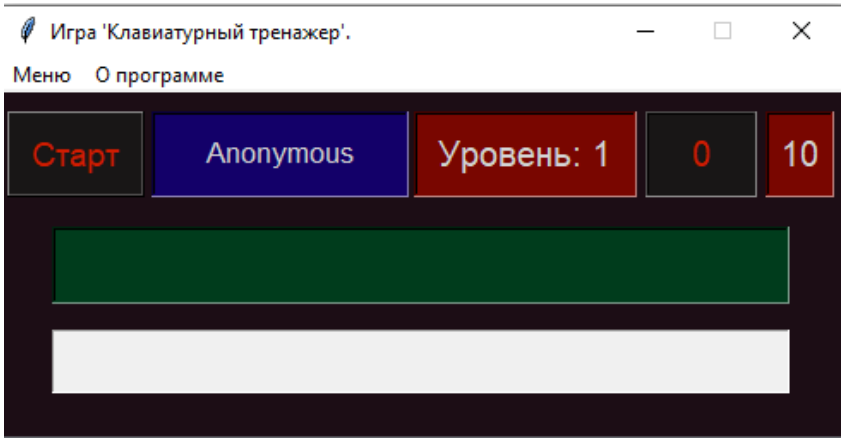
В результате сборки проекта получилась следующая структура:
game.py – основной файл игры.
logindlg.py – реализация класса окна регистрации / авторизации.
settingsdlg. py – реализация класса окна настроек.
schemaedit.py – реализация класса окна редактора тем.
dbfuncs.py – реализация класса работы с базой данных.
config.py – настройки игры (классы для уровней и конфигурации).

Немного скринов игры:

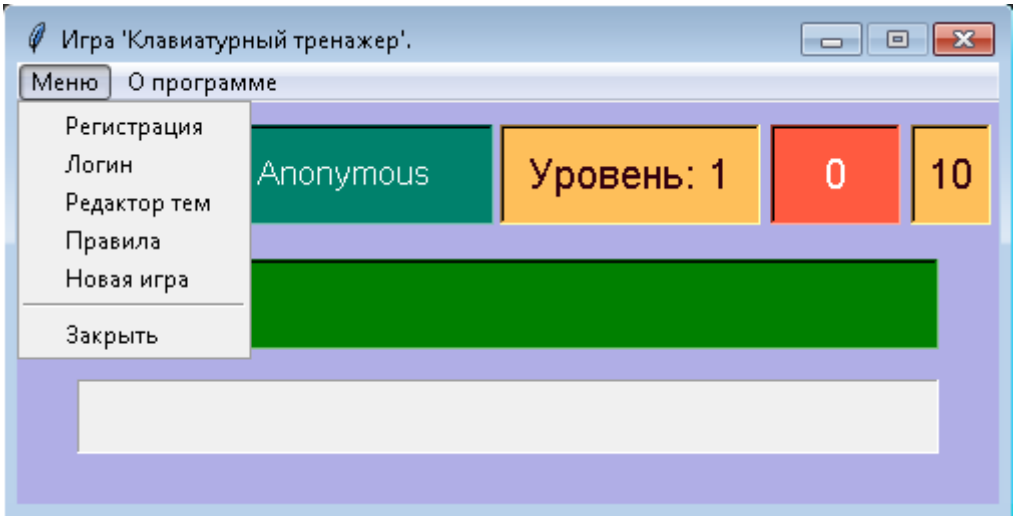
Вертикально:



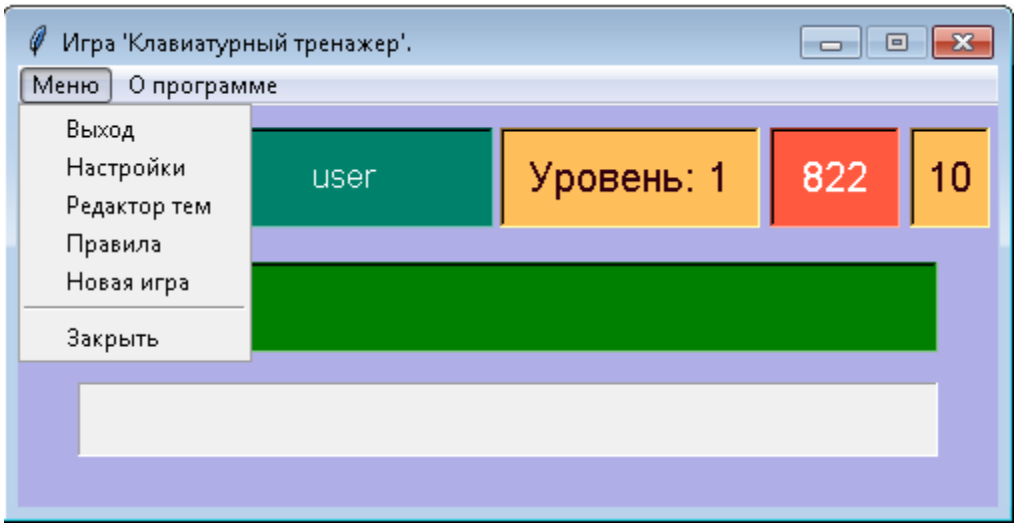
Темная тема:



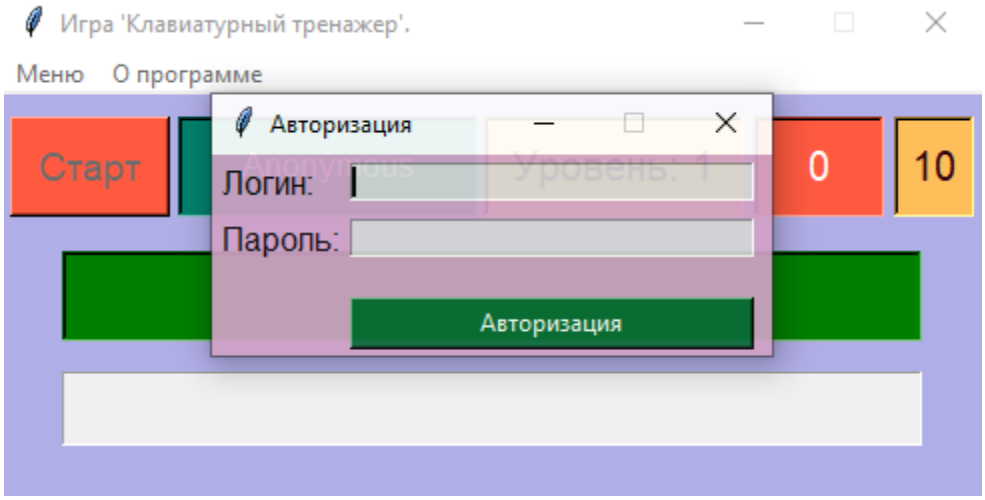
Меню обычное:



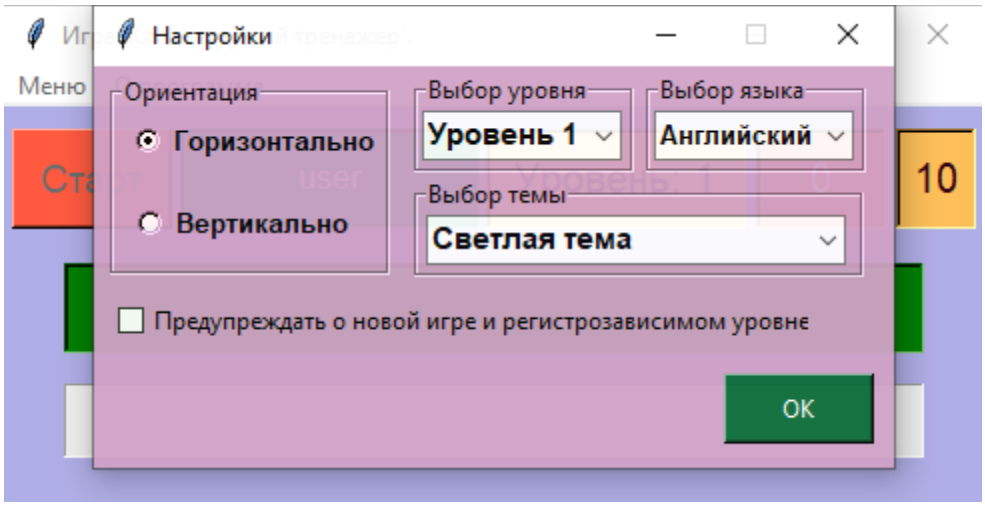
Меню зарегистрированного пользователя:



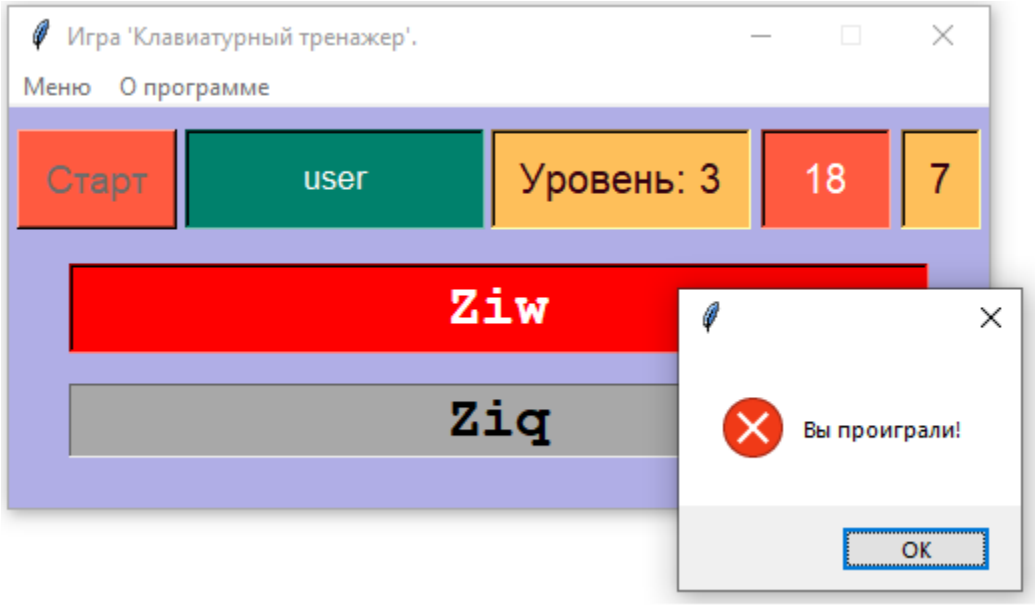
Окно авторизации:



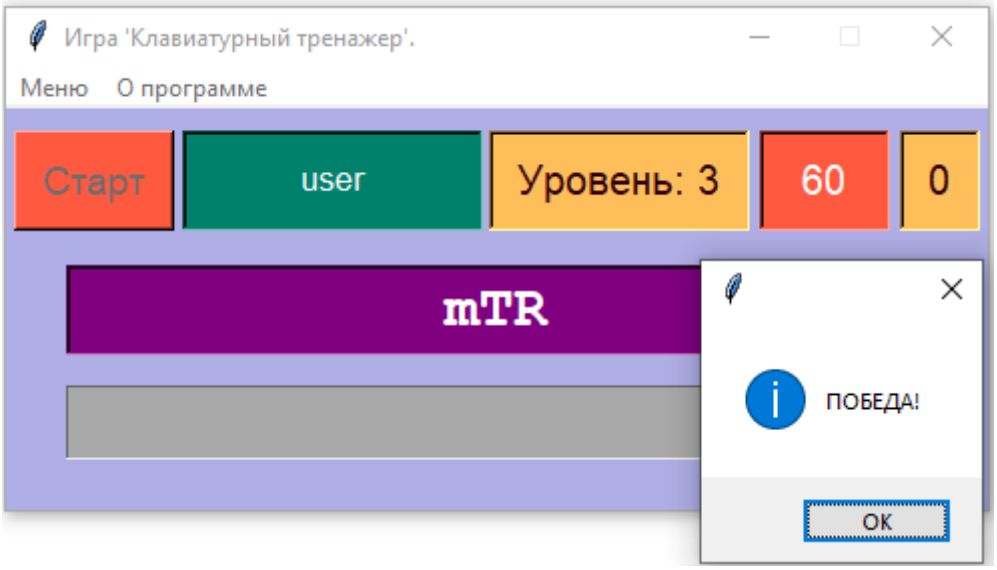
Окно настроек:



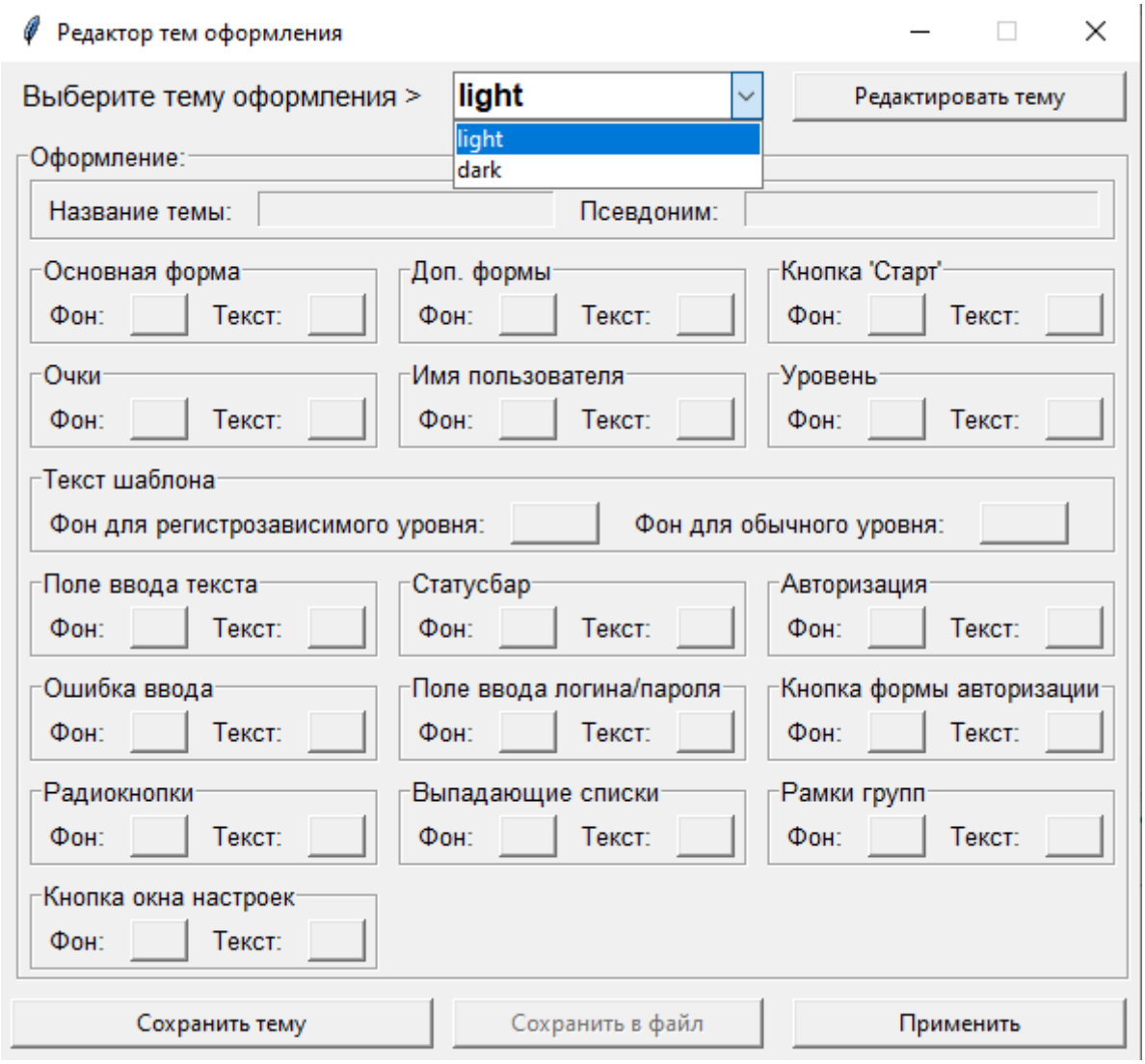
Проиграл ☺:



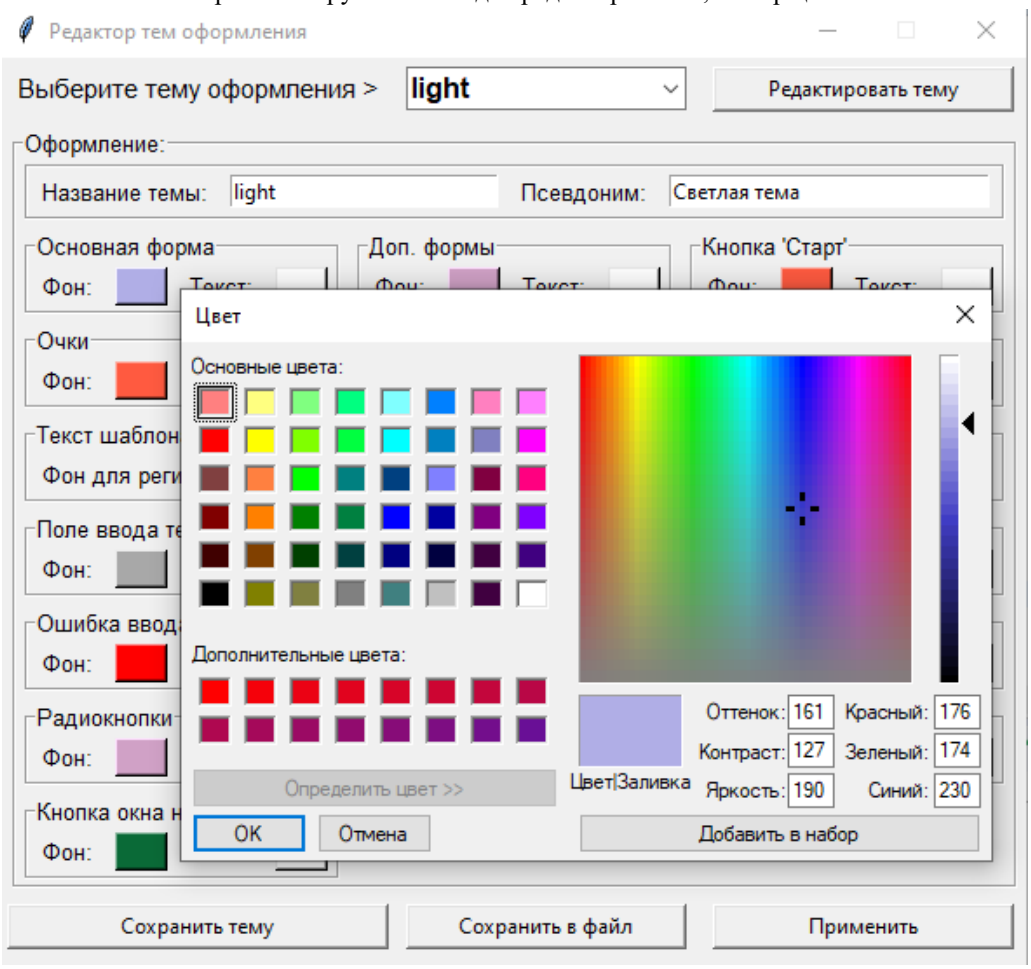
Пользователь прошел уровень:



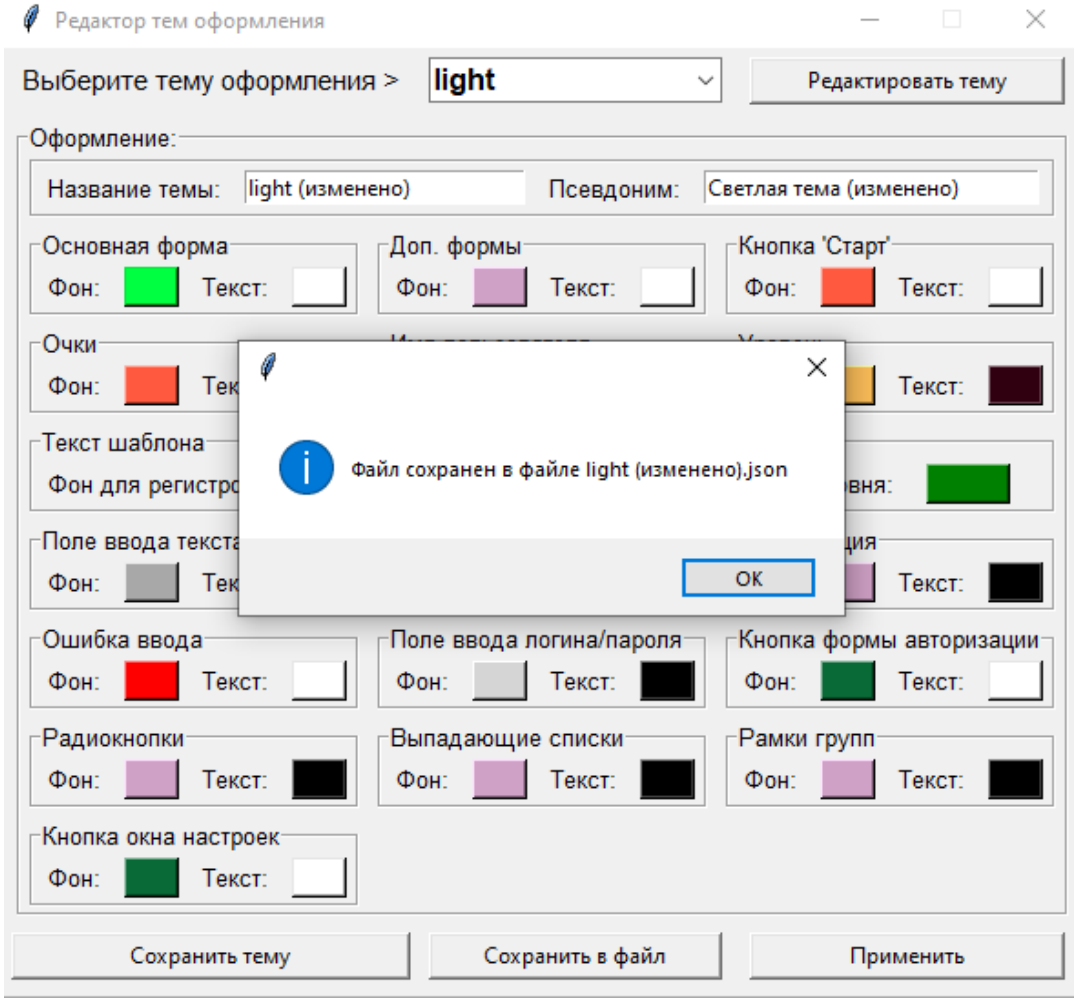
Редактор тем оформления:



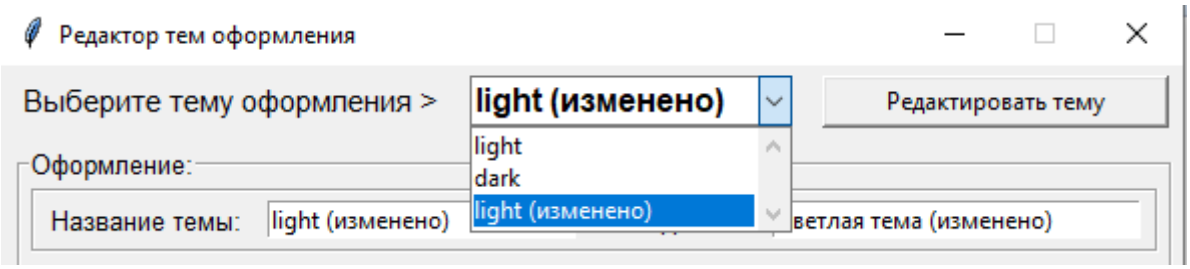
Выбрана и загружена тема для редактирования, выбор цвета:



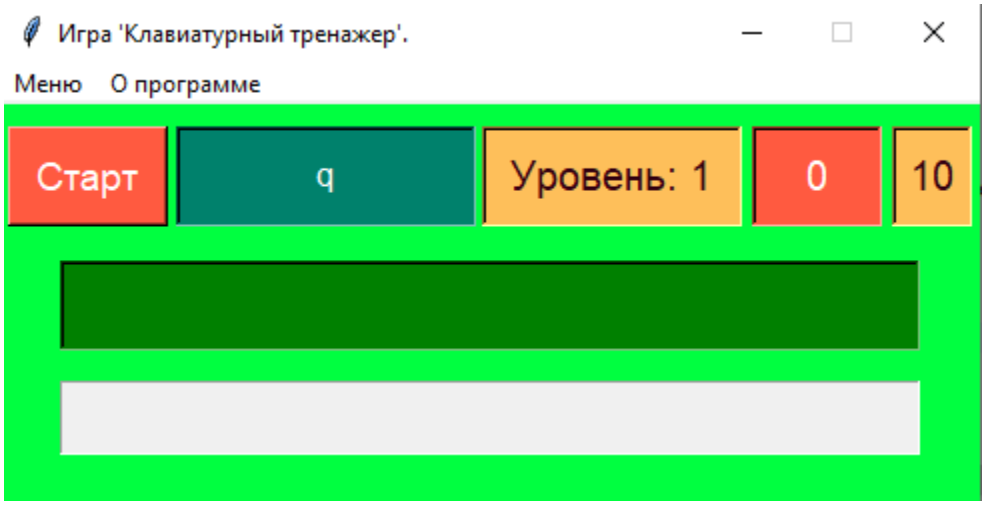
Изменили цвет и сохранили в файл (для зарегистрированных):



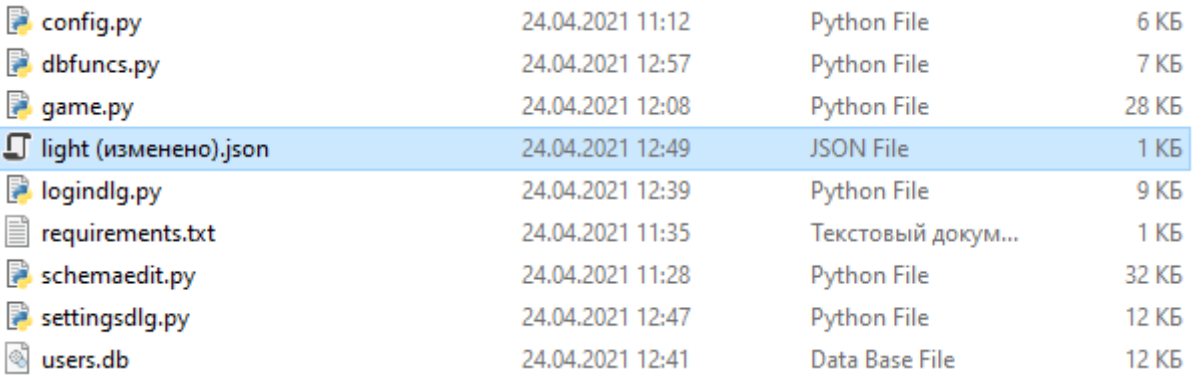
Тема в списке:



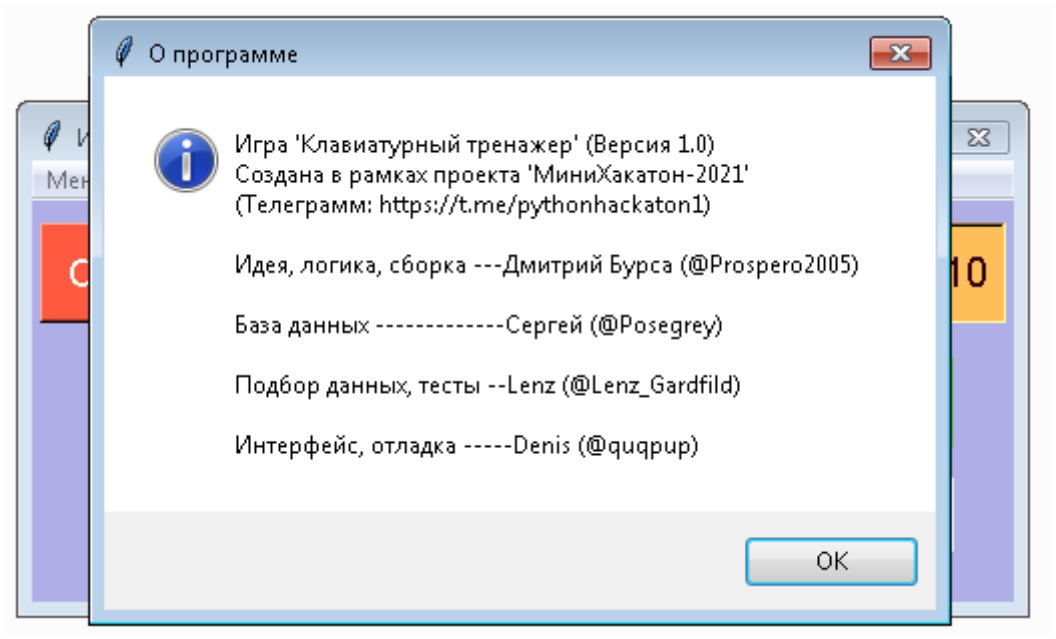
Применили тему:



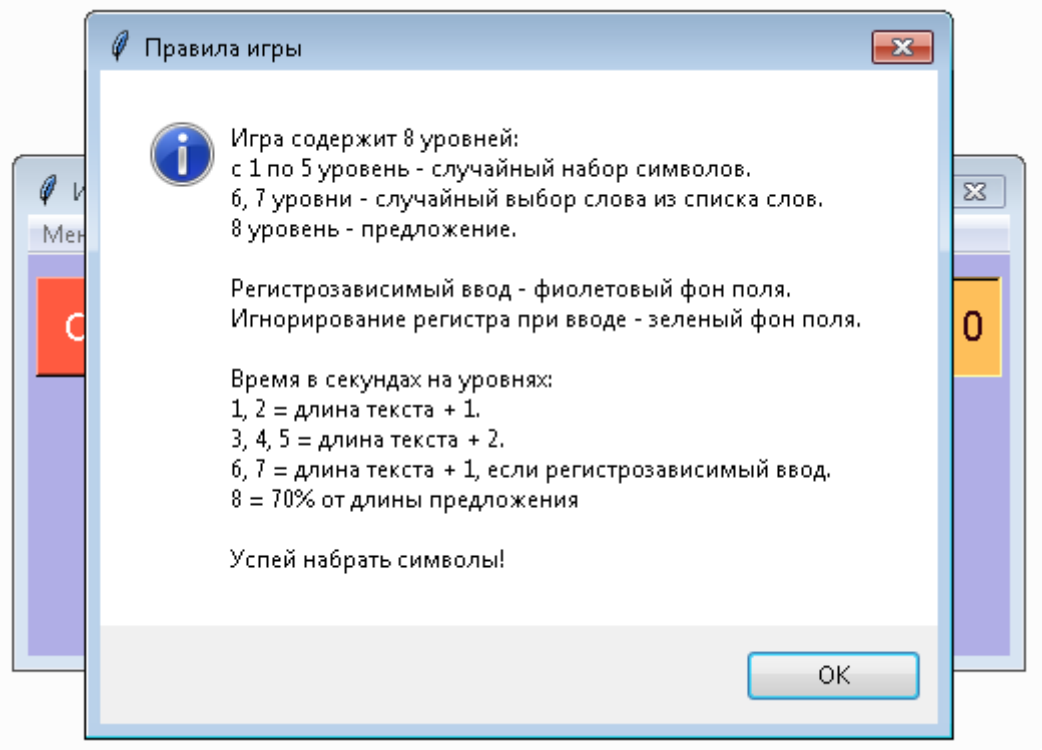
Файл создан:



Меню "О программе":



Правила



Резюмируя:
Несмотря на кажущийся огромный объем кода, фактически кода было написано немного – если посмотреть внутри – большая часть – это создание окна и элементов и позиционирование элементов на форме, центрирование окна, применение тем, по сути – копипаст с изменением пары-тройки параметров.
Остальное немногое – это обработка выбора (форма настроек), проверка пользователей (форма авторизации), выбор цвета и сохранение в файл (редактор).

У нас в команде, как и в других, есть новички, которые с удовольствием разбирали код, задавали море вопросов, предлагали идеи. Несмотря на это, после того, как им объяснялось, как реализован тот или иной метод, как задача раскладывалась на подзадачи для последующей реализации, они соглашались с тем, что это не так уж и сложно ☺.

Главное – правильно подойти к решению задачи, уметь разбить ее на подзадачи, реализовать их и собрать потом воедино. Собственно, это основа декомпозиции.

Как и любая программа, наша не лишена ошибок, поэтому с удовольствием будем ждать от вас их описание.