**Пояснительная записка.**

**ВНИМАНИЕ: для запуска необходимо перейти в папку login&register и запустить файл start\_this.py**

1.Регистрация и вход

При запуске приложения встречает окно входа, в котором, с помощью отдельной кнопки, можно и зарегистрироваться. Оба окна выполнены на PyQt5.  
Для регистрации необходимо ввести логин, e-mail(для, чтобы не создавалась множество аккаунтов на одну почту), пароль и его повтор.  
Требования к логину: отсутствие кириллических символов и уникальность

E-mail проходить проверку на валидность

Требования к паролю: длина не менее 8 символов, отсутствие трёхбуквенных клавиатурных сочетаний, содержание и букв, и цифр.  
  
Пароль хэшируется перед записью в базу данных путём перевода в бинарное представление и затем записи бинарного представление справа налево.

2. Меню

При вводе правильного логина и пароля пользователя встречают меню.Программно оно реализовано с помощью классов Menu, Button и Car(animated\_background.py).  
Оно представляет из себя:

* Анимированный задний фон
* 4 кнопки с выделением при наведении на них мыши  
  Кнопки и их функции:

1. Кнопка «Играть». При нажатии запускает процесс игры с выбранной в прошлый раз машиной
2. Кнопка «Магазин». Место, где можно потратить заработанные монеты на новые автомобили
3. Кнопка «Ваш гараж». Позволяет выбрать одну машину из купленных для игры на этом автомобиле.
4. Кнопка «Выход». При нажатии отключает игру.

3. Магазин

Представляет из себя изображения из 9 *кликабельных* картинок и кнопок под ними. Программно реализован с помощью классов Shop, Button и ClickableImage + InfoWindow

Кнопки в магазине бывают двух видов:

1. Либо кнопка с текстом, состоящим из двузначного числа(цены). При клике на неё при достаточном количестве монет транспортное средство будет куплено, а при недостаточном будет вызвано всплывающее окно, которое скажет пользователю, что у него недостаточно средств.
2. Кнопка с текстом “Куплено”. При клике на неё всплывёт окно с текстом: «Вы уже купили эту машину»

При покупке автомобиля в базе данных user\_info.db в папке game\_data обновляется строка, принадлежащая текущему пользователю. Изначально строка имеет вид:

#0000\_1\_0\_0\_0\_0\_0\_0\_0\_0\_1  
Где первые четыре цифры - количество монет пользователя, часть из единицы и восьми нулей(1\_0\_0\_0\_0\_0\_0\_0\_0) – это купленные пользователем машины(изначально есть только первая) и последняя цифра – это выбранная в данный момент машина.

Также, при покупке обновляется список машин в самой игре, что даёт пользователю возможность выбрать новую машину без перезапуска игры.

9 *кликабельных* картинок – представляют из себя изображения машин, которые можно купить. При клике на ограничивающий картинку прямоугольник откроется окно, сделанное на PyQt5, в котором отобразится название машины, её цена и скорость.

**4. Гараж**

Очень похожее на магазин место. Представляет собой изображения имеющихся машин с кнопками под ними. Программно реализован с помощью классов Garage, Button и ClickableImage + InfoWindow.  
  
Текст на кнопках бывает двух видов:  
“Выбрано” и “Выбрать”.  
  
Соответственно, при нажатии на кнопку с текстом «Выбрать», машина будет выбрана и в базе данных по нужному логину будет изменена последняя цифра в хэшированной строке(#0000\_1\_0\_0\_0\_0\_0\_0\_0\_0\_1).  
  
В гараже, также как и в магазине, при клике на картинку отобразится информация об автомобиле.

**5. Игра и её элементы**

Сама игра реализована с помощью классов Game, Coin, Npc, Nitro и Player.

Давайте рассмотрим каждый элемент по отдельности:

1. **Класс Game.**

Исходя из названия, основной класс игры, который обеспечивает спавн(отображение на экране) игровых объектов, просчитывание их пересечения и то, какие действия после этого пересечения должны происходить.

1. **Класс Player.**  
   Класс, который отвечает за игрока и его движение. При объявлении класса в аргументы передаётся номер картинки, по которому из json-файла определяется скорость машины и сама картинка.  
     
   Метод “update” отвечает за движения игрока и анимацию при движении(картинка наклоняется на 30 градусов в сторону движения)  
   Также, класс содержит метод “check”, который проверяет нахождение игрока в границах дороги(экрана) и при несоблюдении этого условия с помощью класса Menu будет вызываться экран проигрыша.
2. **Класс Coin**Просто класс для хранения спрайта монетки, содержит в себе картинку и скорость, с которой монетка едет вниз по экрану.
3. **Класс Nitro**

Место для хранения спрайта нитро(временного ускорения) и того, как он перемещается вниз по экрану.

1. **Класс Npc**Содержит в себе метод ***load\_images***, отвечающий за выбор случайной картинки врага из 9 существующих, и перемещение вражеской машины ниже или выше по экрану.
2. **Настройки**

Представляет собой небольшое меню настроек громкости музыки в меню и звуков в игре. Реализовано с помощью классов settings и button. Позволяет отдельно настроить громкость музыки в меню и звуков в игре. Уровень громкости измеряется в числах от 0 до 10 и присвается нажатием на кнопку с необходимой цифрой.

Запуск игры производится с помощью открытия файла **start\_this.py(папка login&register),** и затем подпроцессом вызова файла game\_startup.py.

Не были рассмотрены классы Car, InfoWindow и ClickableImage, так как их принцип работы понятен из названия и затрагивается в ходе описания классов.