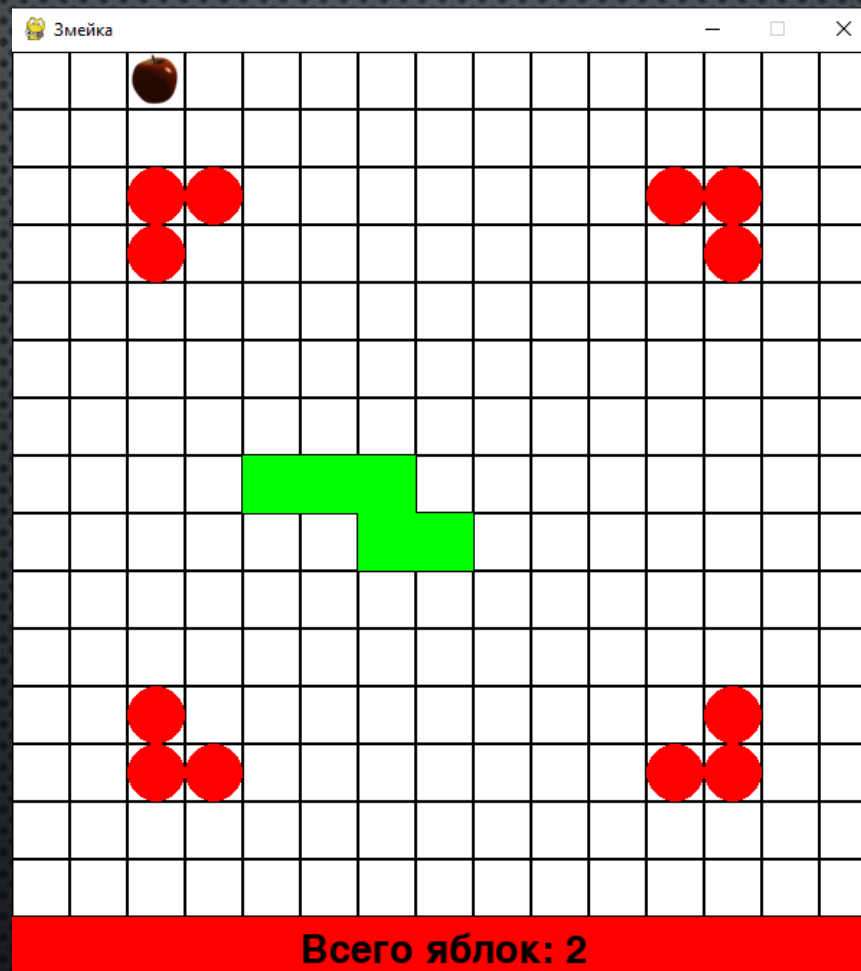


ЗМЕЙКА



ПОДГОТОВИЛ ОВЧАР СЕМЁН

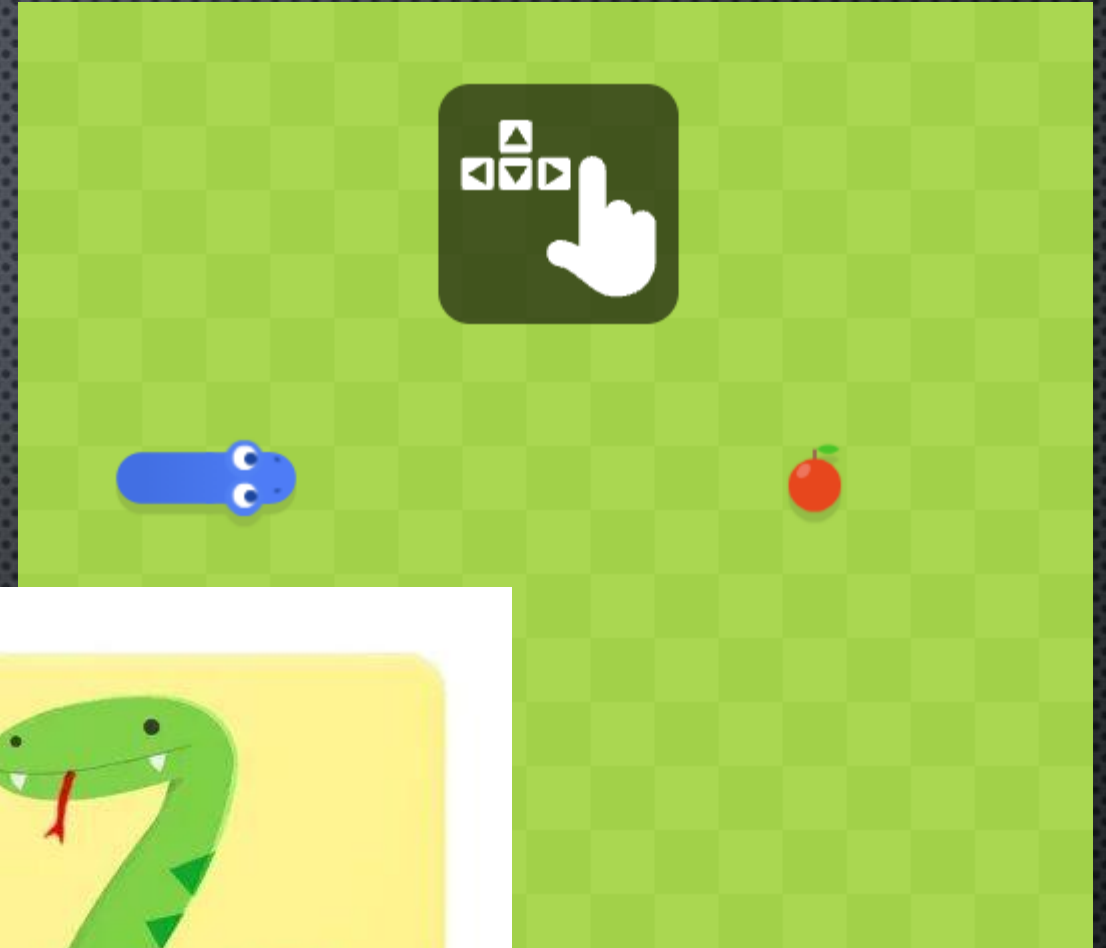
ПРОБЛЕМА

- ИГРА ЗМЕЙКА ТЕРЯЕТ СВОЮ ПОПУЛЯРНОСТЬ. В СВЯЗИ С ЭТИМ Я РЕШИЛ СОЗДАТЬ ПРИЛОЖЕНИЕ С ИНТЕРЕСНЫМИ ПАТЧАМИ И РАЗНЫМИ УРОВНЯМИ, КОТОРЫЕ ПОМОГУТ ОСВЕЖИТЬ ЛЕГЕНДАРНУЮ ИГРУ И СДЕЛАТЬ ЕЕ БОЛЕЕ СОВРЕМЕННОЙ И ИНТЕРЕСНОЙ ДЛЯ МОЛОДЁЖИ



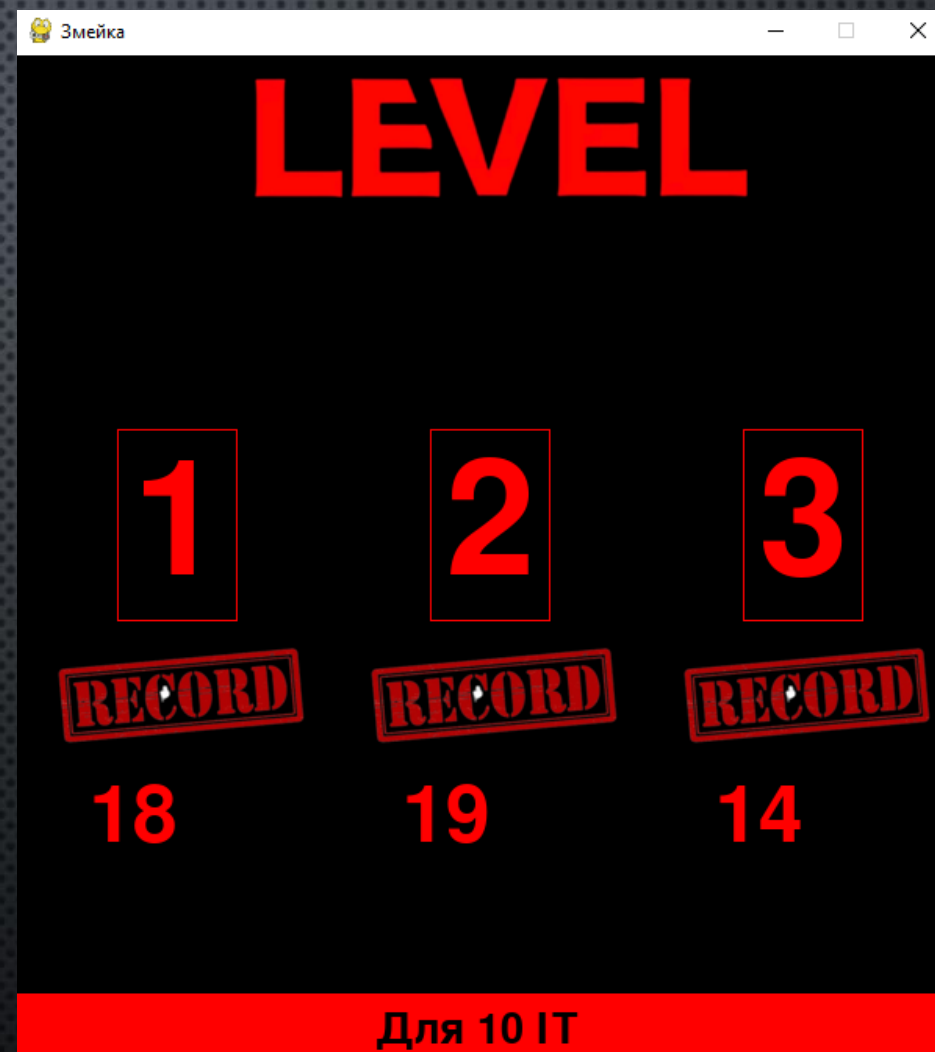
КОНКУРЕНТЫ

- НА ДАННЫЙ МОМЕНТ МОИМИ КОНКУРЕНТАМИ ЯВЛЯЮТСЯ ЛЕГКОВЕСНЫЕ ПРИЛОЖЕНИЯ ЗМЕЙКИ ДЛЯ ТЕЛЕФОНОВ В GOOGLEPLAY И APPSTORE, А ТАКЖЕ САЙТЫ, ПРЕДЛАГАЮЩИЕ СЫГРАТЬ В ИГРУ ОНЛАЙН



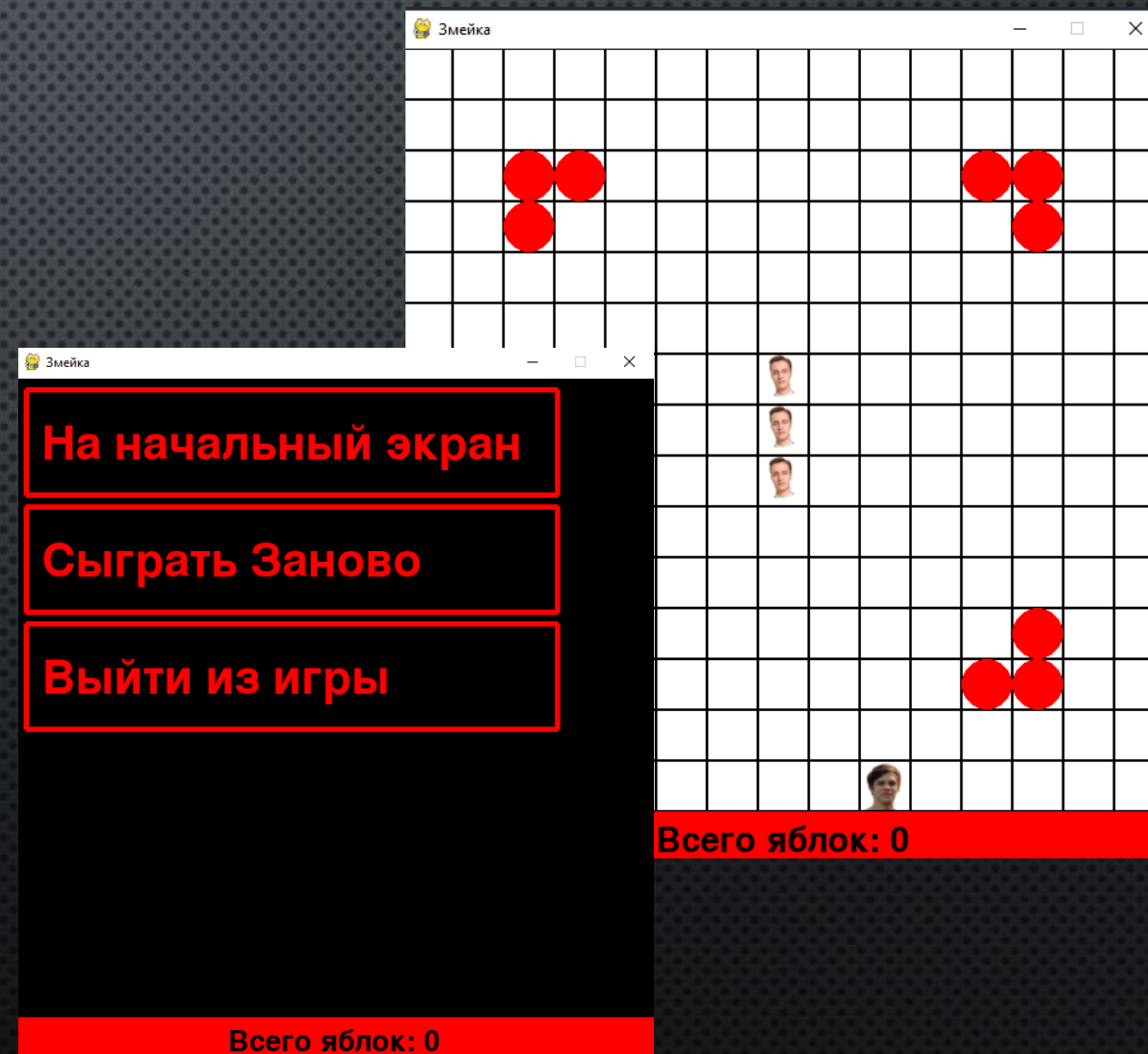
РЕШЕНИЕ

- СДЕЛАТЬ ПРИЛОЖЕНИЕ ЗМЕЙКА С РАЗНЫМИ УРОВНЯМИ СЛОЖНОСТЯМИ И ПАТЧАМИ, ПОЗВОЛЯЮЩИМИ ОСВЕЖИТЬ ИГРУ.



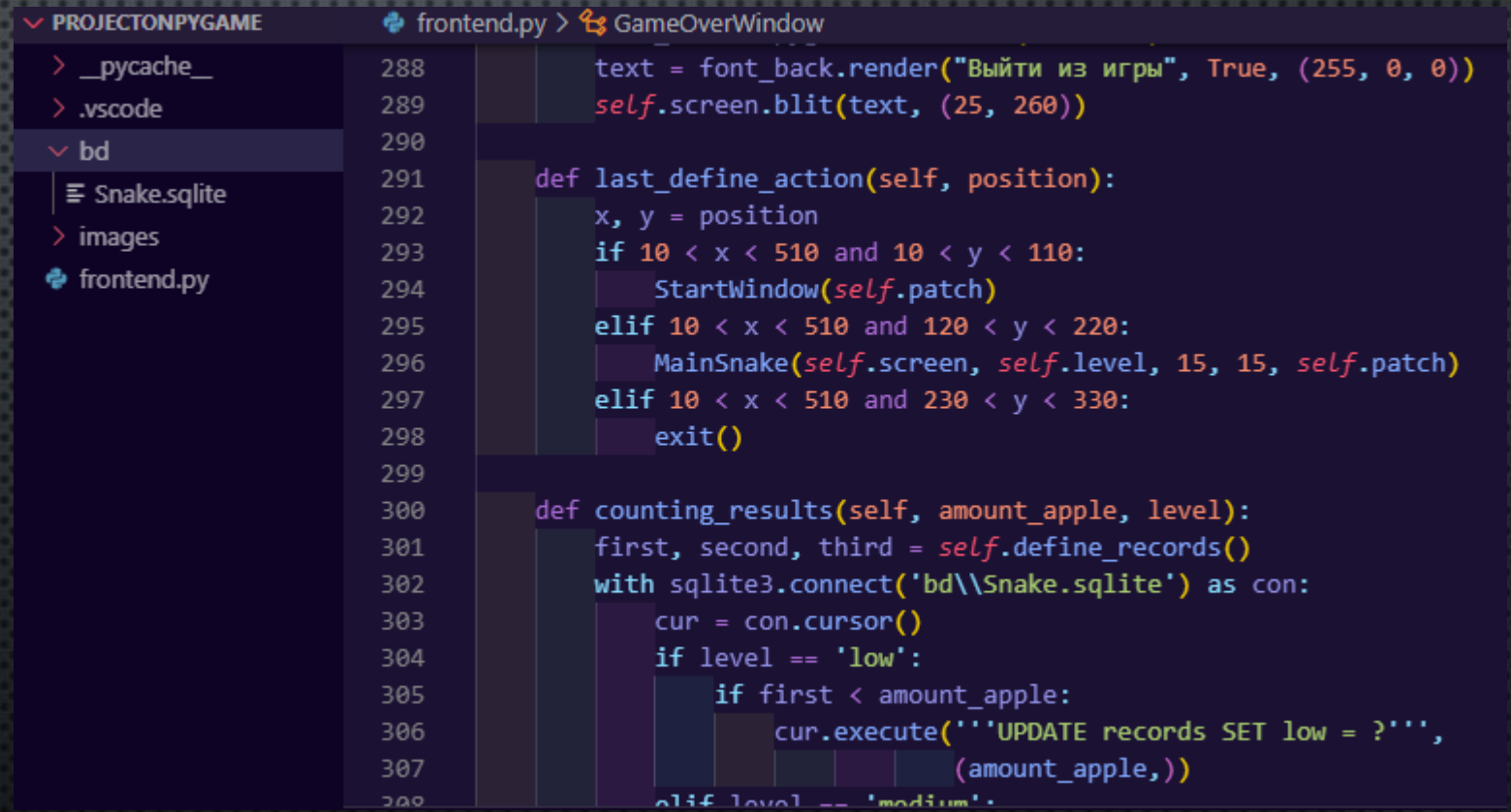
УНИКАЛЬНАЯ ЦЕННОСТЬ РЕШЕНИЯ

- В МОЕМ ПРИЛОЖЕНИИ БУДУТ 3 УРОВНЯ СЛОЖНОСТИ, А ТАКЖЕ ПЕРЕКЛЮЧАТЕЛЬ ВЕРСИИ, КОТОРЫЙ ПОМОЖЕТ ВЫБРАТЬ, ЧТО У ВАС БУДЕТ ЯВЛЯТЬСЯ ЧАСТЬЮ ЗМЕЙКИ, А ЧТО ЯБЛОКОМ



АРХИТЕКТУРА

- КОД МОЕГО ПРИЛОЖЕНИЯ НАХОДИТСЯ В ФАЙЛЕ FRONTEND.PY, КАРТИНКИ В ПАПКЕ IMAGES, А БАЗА ДАННЫХ В ПАПКЕ BD. В КОДЕ КАЖДОЕ ОКНО ИМЕЕТ СВОЙ КЛАСС, КОТОРЫЙ ПОДРАЗДЕЛЯЕТСЯ НА ФУНКЦИИ ДЛЯ УСКОРЕНИЯ ИГРОВОГО ПРОЦЕССА



```
PROJECTONPYGAME frontend.py > GameOverWindow
288 text = font_back.render("Выйти из игры", True, (255, 0, 0))
289 self.screen.blit(text, (25, 260))
290
291 def last_define_action(self, position):
292     x, y = position
293     if 10 < x < 510 and 10 < y < 110:
294         StartWindow(self.patch)
295     elif 10 < x < 510 and 120 < y < 220:
296         MainSnake(self.screen, self.level, 15, 15, self.patch)
297     elif 10 < x < 510 and 230 < y < 330:
298         exit()
299
300 def counting_results(self, amount_apple, level):
301     first, second, third = self.define_records()
302     with sqlite3.connect('bd\\Snake.sqlite') as con:
303         cur = con.cursor()
304         if level == 'low':
305             if first < amount_apple:
306                 cur.execute('UPDATE records SET low = ?',
307                             (amount_apple,))
308             elif level == 'medium':
```


ЭТАПЫ РАЗРАБОТКИ(ГЛАВНЫЕ ЗАДАЧИ)

- 1) СДЕЛАТЬ ПРИЛОЖЕНИЕ С УДОБНЫМ ПОЛЬЗОВАТЕЛЬСКИМ ИНТЕРФЕЙСОМ
- 2) СДЕЛАТЬ В ПРИЛОЖЕНИИ 3 РАЗНЫХ УРОВНЯ СЛОЖНОСТИ
- 3) СДЕЛАТЬ НАЧАЛЬНОЕ, ОСНОВНОЕ-ИГРОВОЕ И ФИНАЛЬНОЕ ОКНО ПРИЛОЖЕНИЯ
- 4) ПОСТАВИТЬ ПАТЧ, ОТВЕЧАЮЩИЙ ЗА ИНТЕРФЕЙС ИГРОВОГО ПРОЦЕССА

ЭТАПЫ РАЗРАБОТКИ(ПОБОЧНЫЕ ЗАДАЧИ)

- 1) РАЗГРАНИЧИТЬ КОД, ИЗОБРАЖЕНИЯ И БАЗУ ДАННЫХ НА ПАПКИ
- 3) СДЕЛАТЬ ДЛЯ КАЖДОГО ОКНА СОБСТВЕННЫЙ КЛАСС
- 3) РАЗБИТЬ КЛАССЫ НА ФУНКЦИИ, УСКОРЯЮЩИЕ РАБОТУ ПРИЛОЖЕНИЯ
- 4) СДЕЛАТЬ ФАЙЛ REQUIREMENTS.TXT, В КОТОРОМ БУДУТ НАХОДИТЬСЯ ВСЕ ИСПОЛЬЗУЕМЫЕ МОДУЛИ
- 5) СДЕЛАТЬ ОПИСАНИЕ, ПОЯСНИТЕЛЬНУЮ ЗАПИСКУ И ТЕХНИЧЕСКУЮ ЗАПИСКУ К ПРИЛОЖЕНИЮ

ДЕМОНСТРАЦИЯ РЕЗУЛЬТАТОВ(ОПИСАНИЕ И ТЗ)

Змейка

Данное приложение совершенствует всемирно известную игру змейка, наполняя ее уровнями различной сложности и патчами на оформление игры.

Всего в игре 3 уровня. С увеличением уровня растет количество препятствий в нем и скорость змейки.

Особенно вам понравится игра, если вы являетесь учеником 10 IT класса и 2 года Яндекс Лицея, так как в нее добавлен специальный плагин.

Ставьте рекорды прохождения змейки и наслаждайтесь игрой.

В этом проекте код приложения находится в одном файле под названием frontend. Картинки должны находиться и доставаться из в папке images.

В главном файле каждый класс отвечает за одно окно. Всего 3 окна – стартовое, основное-игровое и окно окончания игры.

1) StartWindow

Данный класс запускается из конструкции `if __name__ == '__main__':`. В данном классе должно рисоваться основное окно.

Сверху должна располагаться надпись Level.

После нее должны быть 3 цифры – 1, 2, 3. Каждая из них должна быть с границей. После нажатия на пространство в границе или на цифру должен запускаться игровое окно, но об этом позже.

Далее под цифрами должны располагаться надписи Record, под каждой из которой должен цифрами отображаться максимальное количество яблок, собранное при прохождении уровня. Количество яблок должно браться из базы данных `snake.sqlite`. Последнее, что в данном классе надо осуществить это кнопку в конце экрана, с [текстом](#) Для 10 IT. При нажатии кнопка становится зеленой и патч для 10 IT должен начинать работать

2) MainSnake

Главный класс приложения. В нем должна осуществляться игра. Поле 15 на 15 клеток. Размером 640 на 600 пикселей. Каждая клетка размером 40 на 40.

В конце поля должна располагаться надпись: всего яблок ". Значение, для которой меняется в зависимости от съеденных змейкой яблок

Чем выше уровень, тем сложнее для него карта. Изначально, на 1 первом уровне должны располагаться по 1 препятствию на 4 частях карты. Далее, на 2 и 3 уровне должно быть 3 и 5 препятствий, сложенные в уголки.

Змейка должна располагаться по вертикали на 3 центральных клетках, а именно на центральной клетке и на 2 соседних по вертикали.

Патчи действуют так. Если патч включен, то вместо яблока едой становятся ученики 10 IT и Яндекс Лицея, а частями змейки вместо кубиков становятся головы преподавателя Яндекс Лицея (KE)

При нажатии на стрелочки или на кнопки WASD змейка должна начинать двигаться в определенном направлении со скоростью 40 пикселей в секунду. Fps змейки начинается с

ДЕМОНСТРАЦИЯ РЕЗУЛЬТАТОВ(ПОЯСНИТЕЛЬНАЯ ЗАПИСКА)

Название проекта: Змейка

Автор проекта: Овчар Семён

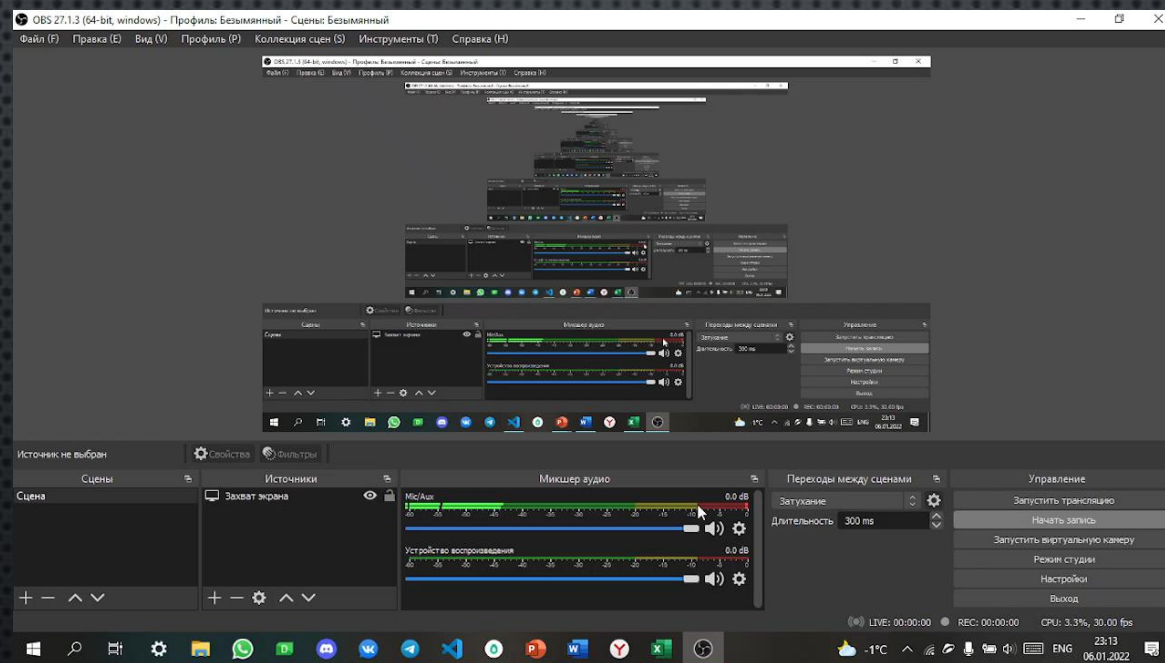
class StartWindow

- 1) def __init__
Рисует экран 640 на 600 пикселей, передается значение false для патча и оглашаются функции.
 - 2) define_records
Достает из базы данных под названием snake.sqlite данные о рекордах разных уровней и отображает их на экран
 - 3) image_on_screen
Переносит картинки на экран
 - 4) special
Определяет цвет кнопки для установки патча и саму установку патча, а именно будет ли змейка в виде кубиков или голов KE и будут ли яблоки учениками 10 IT
 - 5) levels
Рисует 3 цифры, обозначающие уровни и границы вокруг них
 - 6) define_action
Определяет куда нажал пользователь и в зависимости от клика, открывает новое окно, изменяет данное или вообще ничего не делает
- +) if __name__ == '__main__' – главная строка приложения при нажатии на экран запускает define_action, а при закрытии окна закрывает его.

class MainSnake

- 1) def __init__
Принимает главные параметры приложения. Определяет отсутствие или наличие патча, а также содержит главный игровой класс, который при нажатии на стрелки или кнопки wads

ДЕМОНСТРАЦИЯ РЕЗУЛЬТАТОВ(100%)



ПЕРСПЕКТИВЫ РАЗВИТИЯ

- 1) СДЕЛАТЬ ВОЗМОЖНОСТЬ, ЧТОБЫ ПОЛЬЗОВАТЕЛЬ САМ ДОБАВЛЯЛ ПАТЧИ, ИЗМЕНЯЮЩИЕ ИНТЕРФЕЙС ИГРЫ
- 2) ДОБАВИТЬ ЕЩЕ НЕСКОЛЬКО УРОВНЕЙ СЛОЖНОСТИ
- 3) ВЫПУСТИТЬ ПРИЛОЖЕНИЕ В ОБЩЕСТВЕННОЕ ИНТЕРНЕТ-ПОЛЬЗОВАНИЕ



ИТОГИ

- Я ОСУЩЕСТВИЛ ВСЕ ЗАДУМАННЫЕ ГЛАВНЫЕ И ПОБОЧНЫЕ ЗАДАЧИ, СДЕЛАЛ ПРЕЗЕНТАЦИЮ, ОПИСАНИЕ И ТЕХНИЧЕСКОЕ ЗАДАНИЕ С ПОЯСНИТЕЛЬНОЙ ЗАПИСКОЙ. ТАКЖЕ Я НАМЕТИЛ СТРАТЕГИЮ РАЗВИТИЮ ПРИЛОЖЕНИЯ НА БУДУЩЕЕ

