МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | |  | |  |  |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | |  | | ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  на лабораторную работу №11  по дисциплине «Алгоритмы и структуры данных»  Тема «Разработка компьютерной симуляции Светофор» | | | | | | |
|  | | |  | |  | | | |
|  | | Исполнители  студент гр. ИСТбд-21  Васильева А., Грунин Н., Казаров Д., Костенко Д., Мишин А., Саранцев С., Тимощук М., Фоменков В.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |
|  | | 2024 | | | | | | |

**Введение**

Разрабатываемое приложение представляет собой симуляцию светофора и дорожного движения, разработанное с использованием языка программирования Python и библиотек Tkinter для графического интерфейса и Pygame для работы со звуком. Целью приложения является моделирование работы светофора, включая взаимодействие пешеходов и автомобилей на перекрестке. Программа демонстрирует навыки программирования и создания пользовательских интерфейсов для симуляции реальных процессов.

**Основания для разработки:**

Задание для лабораторной работы по учебному плану направления «Информационные системы и технологии».

**Постановка задачи**

Цель разработки заключается в создании симуляции работы светофора, которая управляет движением автомобилей и пешеходов, демонстрируя работу сигналов светофора и алгоритмов движения объектов.

**Структура программы**

**Классы:**

1. Car: класс для создания и управления машинами.

2. Pedestrian: класс для создания и управления пешеходами.

**Основные функции:**

1. draw\_road - отрисовка дороги.

2. draw\_crosswalk - отрисовка пешеходного перехода.

3. draw\_traffic\_lights - отрисовка светофоров.

4. update\_canvas - обновление холста при изменении размера окна.

**Функциональное назначение:**

1. start\_simulation - запуск симуляции.

2. pause\_simulation - приостановка симуляции.

3. resume\_simulation - возобновление симуляции.

4. stop\_simulation - остановка симуляции.

5. open\_settings - открытие окна настроек.

6. update\_lights - обновление состояния светофоров.

7. spawn\_cars - создание новых машин.

8. spawn\_pedestrians - создание новых пешеходов.

9. move\_cars - перемещение машин.

**Проектная часть**

**1.1 Математические методы**

Приложение использует систему координат для управления перемещением объектов (автомобилей и пешеходов) по экрану. Для расчета взаимодействий между автомобилями и пешеходами применяются простые геометрические методы: проверка пересечения координат объектов и расчет времени остановки перед линией светофора.

**1.2 Архитектура и алгоритмы**

**1.2.1 Архитектура**

Архитектура приложения включает несколько основных компонентов:

* **Поле симуляции** – область экрана, где происходит движение автомобилей и пешеходов.
* **Автомобили** – объекты, движущиеся по двум направлениям (вправо и влево), которые могут останавливаться на светофоре и реагировать на пешеходов.
* **Пешеходы** – объекты, перемещающиеся по пешеходному переходу в зависимости от сигналов светофора.
* **Светофор** – объект, управляющий состояниями (красный, желтый, зеленый) для автомобилей и пешеходов.
* **Звуковое оповещение** – звуковой сигнал включается при переходе светофора в зеленый для пешеходов.

**1.2.2 Алгоритмы**

* **Алгоритм работы светофора**: Светофор переключает сигналы между красным, желтым и зеленым в зависимости от таймера, управляя автомобилями и пешеходами.
* **Алгоритм движения автомобилей**: Автомобили останавливаются перед светофором и пешеходами, двигаются при зеленом свете и избегают столкновений друг с другом.
* **Алгоритм движения пешеходов**: Пешеходы начинают движение по переходу при зеленом сигнале и ускоряются при приближении к окончанию времени перехода.

**Стадии и этапы разработки**

1. Определение требований: постановка задач и определение функциональности программы.

2. Проектирование: разработка структуры классов и функций.

3. Реализация: написание кода программы.

4. Тестирование: проверка работоспособности всех функций.

5. Оптимизация: улучшение производительности и пользовательского интерфейса.

6. Документирование: создание пояснительной записки и комментариев к коду.

**Тестирование**

**Цель тестирования**

Цель тестирования заключается в проверке корректности работы светофора и взаимодействия объектов (автомобилей и пешеходов) на перекрестке.

**Методика тестирования**

Тестирование проводилось методом "черного ящика", где проверялись функциональные возможности светофора, автомобилей и пешеходов при различных сценариях.

**Тестирование**

1. Модульное тестирование: проверка работы отдельных классов и функций.

2. Интеграционное тестирование: проверка взаимодействия между различными компонентами программы.

3. Функциональное тестирование: проверка соответствия программы заданным требованиям.

4. Тестирование пользовательского интерфейса: проверка удобства использования и отзывчивости интерфейса.

5. Стресс-тестирование: проверка работы программы при большом количестве объектов на экране.

* 1. Проверка переключения сигналов светофора.
  2. Проверка остановки автомобилей на стоп-линии.
  3. Проверка движения пешеходов на зеленый сигнал.
  4. Проверка остановки автомобилей перед пешеходами.

**Результаты тестирования**

Были выявлены и исправлены ошибки в логике движения автомобилей и пешеходов, а также в отображении сигналов светофора.

**Заключение**

Разработанная программа полностью соответствует поставленным задачам и обеспечивает визуализацию работы светофора и движения транспорта и пешеходов. Программа имеет удобный пользовательский интерфейс и возможность настройки параметров симуляции.

**Источники, использованные при разработке**

1. Пирогов, А.А. Python для начинающих. Изучаем программирование с нуля. – М.: Издательство "БХВ-Петербург", 2020.
2. "Документация Pygame" [Электронный ресурс] // Официальный сайт библиотеки Pygame: https://www.pygame.org/docs/ – Режим доступа: свободный.
3. ГОСТ 19.701-90 «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения». – М.: Издательство стандартов, 1990.
4. В.В. Шишкин, Д.С. Афонин РАЗРАБОТКА ЛОГИЧЕСКИХ КОМПЬЮТЕРНЫХ ИГР С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ В СРЕДЕ ПИТОН