|  |  |
| --- | --- |
| **e** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

**ОТЧЁТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*фамилия, имя, отчество*

Группа\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тип практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Название предприятия\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

Рекомендуемая оценка: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Руководитель практики   
от предприятия: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

Руководитель практики **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

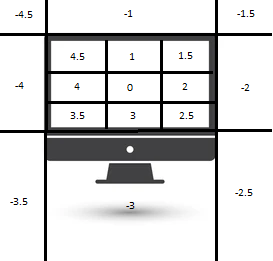
*2023г.*

**Оглавление**

1. Введение
2. О предприятии
3. Реализация
4. Запись окулограмм
5. Заключение
6. Список использованных источников

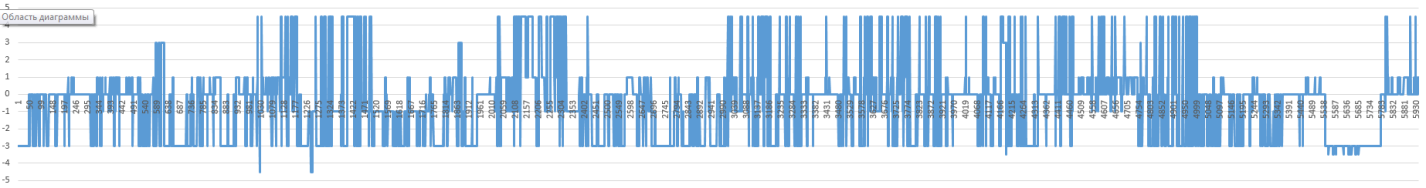
**Введение**

Задача состоит в том, чтобы при помощи API написать программу, которая будет сохранять получаемые от айтрекера GazePointer данные в виде временного ряда (пар время-значение). Временем будет сама та временная метка, которую устанавливает сам GazePointer (он делает замеры раз в 100 мс), а значением – некое условное число, обозначающее сектор экрана, в который в данный момент смотрит человек. Это число однозначно определяется из координат, вытаскиваемых API из самого GazePointer-а. Пример разметки экрана на сектора на рисунке ниже.



Нулевая точка системы координат находится в верхнем левом углу экрана, при этом противоположный имеет координаты [X; Y], где XxY–текущее разрешение экрана. Таким образом можно однозначно определять в том числе и тот факт, смотрит ли человек за пределы экрана (если одна из координат не принадлежит отрезку [0; X] или [0; Y]).

После написания программы необходимо собрать с помощью неё статистические данные: запустить на компьютере/ноутбуке одновременно программу с API и симулятор вождения, приближённый к реальности (камера в нём обязательно должна находиться внутри салона, на водительском месте) и записать не меньше 6 окулограмм (временных рядов с парами время–сектор экрана) длительностью не менее 5 минут, из которых в трёх нужно водить осторожно и внимательно следить за дорогой, а в трёх – часто отвлекаться на посторонние вещи и задерживать на них взгляд. Окулограммы необходимо представить в виде .csv-файлов и графиками как на рисунке ниже.



Дополнительно:

* Изучение языка программирования C#
* Изучение C# .NET Windows Forms

**О предприятии**

Государственный научный центр Российской Федерации ФГУП «НАМИ» основан 16 октября 1918 года как первый научно-исследовательский институт в области автомобильной теории и технологии. За почти вековую историю ГНЦ РФ ФГУП «НАМИ» прошел огромный путь в области исследования автомобильной науки и техники России.

История создания и развития «НАМИ» неразрывно связана со становлением и развитием отечественного автомобилестроения. За годы своей деятельности коллектив института внёс значительный вклад в развитие отечественного автостроения, тракторостроения, автомобильного и авиационного двигателестроения и оказал большое влияние на развитие науки в этих областях.

Результатом деятельности института в содружестве с предприятиями отрасли является тот неоспоримый факт, что жизнедеятельность и оборона страны была обеспечена автомобильной техникой преимущественно отечественного производства. При непосредственном участии ученых и конструкторов института были созданы, испытаны и поставлены на производство почти все отечественные автомобили, многие двигатели и тракторы — от первых легковых автомобилей и до мощных высокопроизводительных грузовых автомобилей КАМАЗ, МАЗ, КрАЗ, БелАЗ и многих других.

«НАМИ» сегодня — это современная научно-исследовательская, опытно-производственная база, которая позволяет решить любую задачу в области исследований, проектирования, построения, разработки и испытаний автотранспортных средств.

За почти вековую историю ГНЦ РФ ФГУП «НАМИ» прошел огромный путь развития ключевых индустриальных основ и планирует сохранить лидирующие позиции в качестве основного российского индустриального инженерного и регулятивного центра, применяя 95-летний опыт в области научных исследований и развития автомобилестроения.

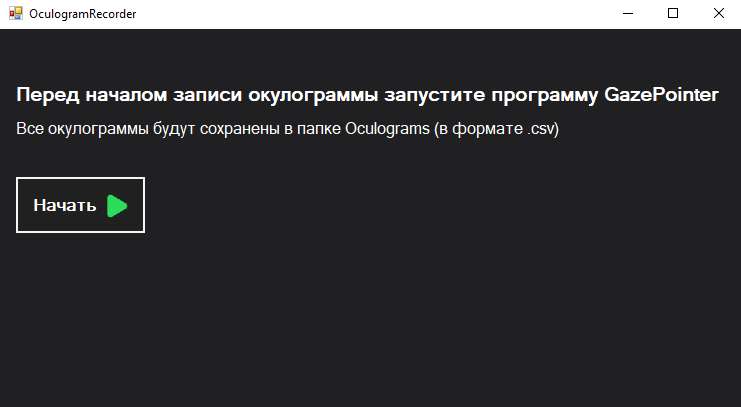
Сегодня ГНЦ РФ ФГУП «НАМИ» обладает большим потенциалом для роста, являясь современным научно- исследовательским экспериментальным центром развития производства для проектирования, конструирования и испытаний автомобильных платформ.

В 1994 году институт получил статус Государственного научного центра Российской Федерации, а в 2003 году Федеральное агентство по техническому регулированию и метрологии России аккредитовало ГНЦ РФ ФГУП «НАМИ» в качестве Испытательного центра продукции автомобилестроения (ИЦАИ), который становится технической службой с правом проведения испытаний по Правилам ЕЭК ООН.

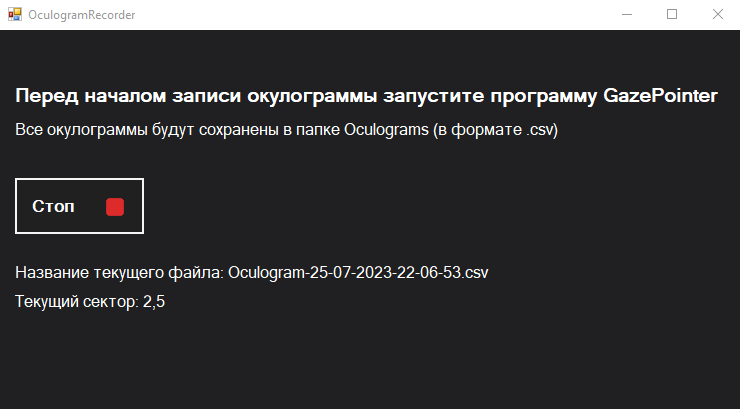
Указом Президента РФ от 4 августа 2004 года № 1009 НАМИ включен в Перечень стратегических предприятий, имеющих значение для обеспечения обороноспособности и безопасности государства, защиты нравственности, здоровья, прав и законных интересов граждан РФ. Также ГНЦ РФ ФГУП «НАМИ» является представителем Российской Федерации в Техническом комитете 22 «Дорожный транспорт» Международной организации по стандартизации.

**Реализация**

Для написания программы был выбран язык программирования C#. В качестве API для получения данных от GazePointer был взят GazeFlowAPI, написанный на этом языке. Изначально планировалось представить программу в виде консольного приложения, однако в конечном итоге было принято решение создать на основе C# .NET Windows Forms, чтобы предоставить пользователю удобный GUI. Программа при запуске выглядит следующим образом:

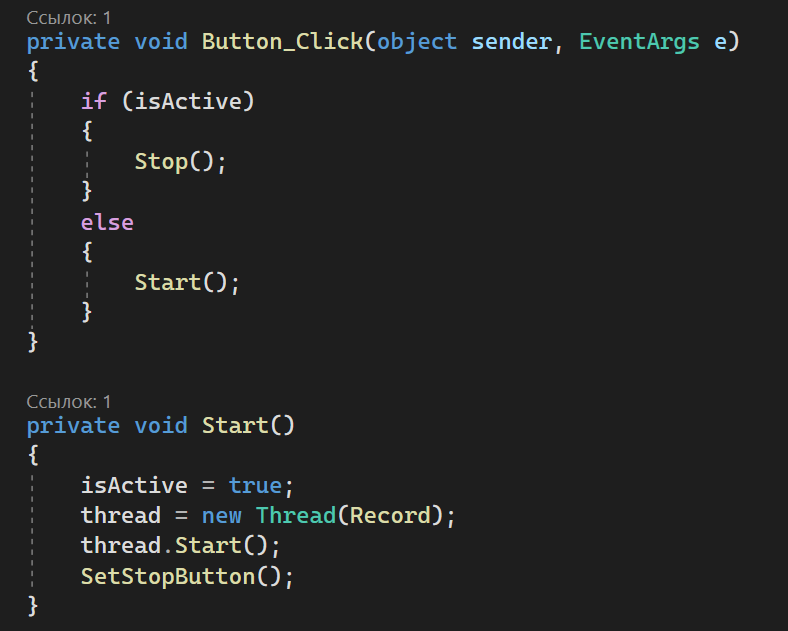


После нажатия на кнопку “Начать”, предварительно запустив программу GazePointer, начнется генерация окулограммы, и программа примет следующий вид:



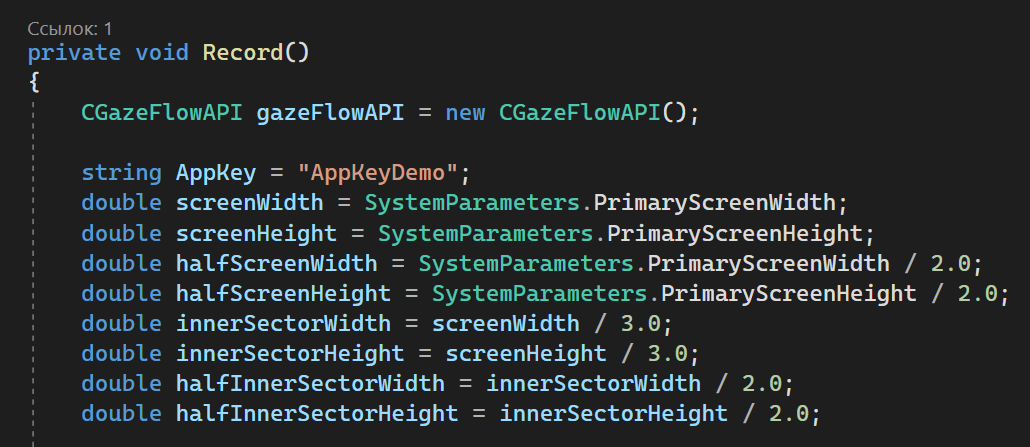
При записи окулограммы снизу от кнопки будет отображаться ее название и сектор, на который в текущий момент смотрит пользователь. По завершении генерации необходимо нажать на кнопку “Стоп”. При этом окулограмма сохраниться в автоматически создаваемой папке Oculograms, которая будет находиться вместе с исполняемым файлом программы. Эта окулограмма будет представлена в виде файла в формате CSV, состоящий из пар Time (время в миллисекундах) и Sector (номер сектора).

Программа работает в 2 потоках, один из которых используется для отображения пользовательского интерфейса и обработки событий, а второй - для генерации окулограммы. Происходит это следующим образом:

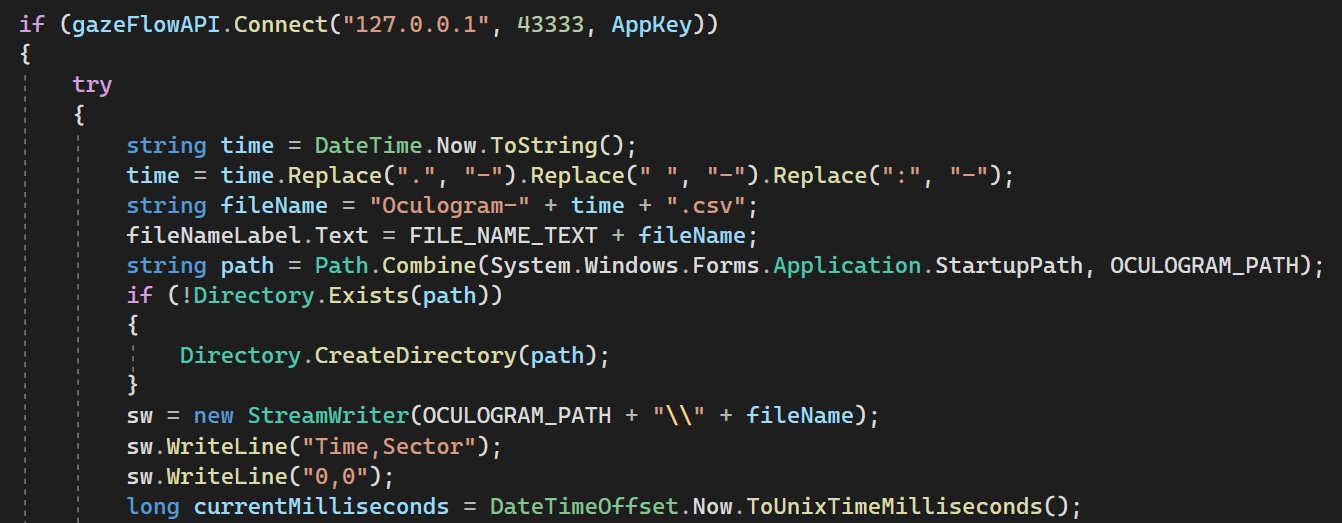


При нажатии на кнопку “Начать” срабатывает ее слушатель Button\_Click, который запускает генерацию, вызвав метод Start(), где переменная isActive сообщает о процессе генерации, первоначальное значение которой равно false. В этом методе создается и запускается новый поток с методом Record. В основном потоке после этого выполняется метод SetStopButton(), который изменяет вид кнопки на другой и включает вспомогательный интерфейс.

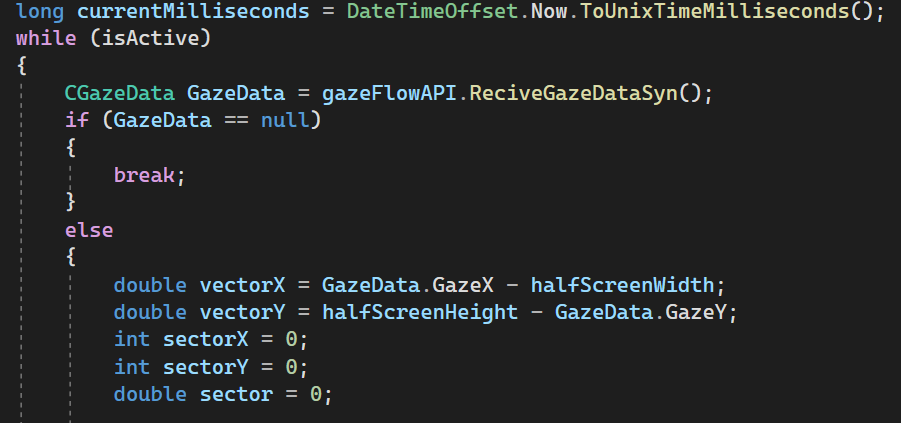
В методе Record сначала создается экземпляр класса GazeFlowAPI. Затем объявляются и инициализируются вспомогательные переменные: ключ приложения, ширина и высота экрана, половины ширины и высоты экрана, ширина и высота внутренних секторов, которые лежат в пределах экрана, а также их половины ширины и высоты.

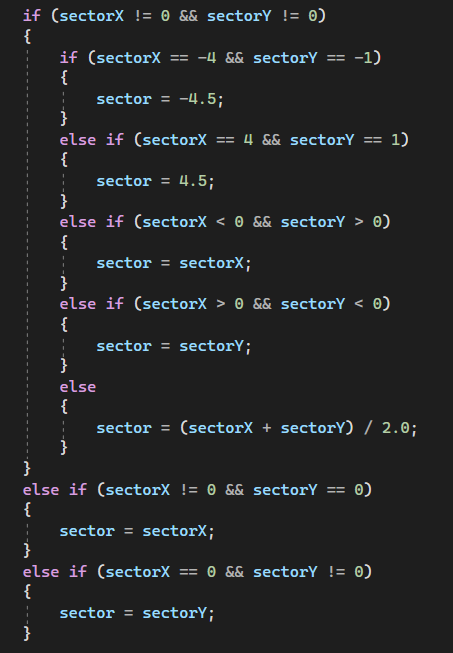
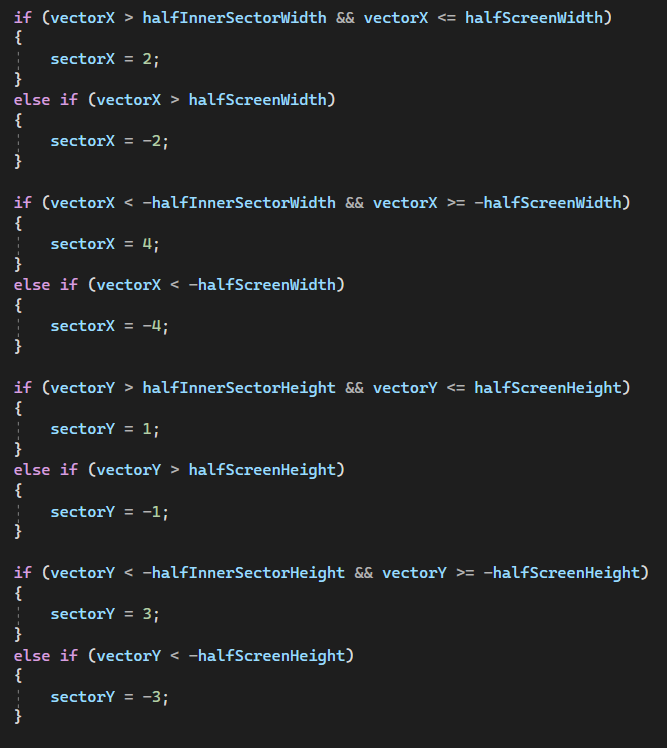


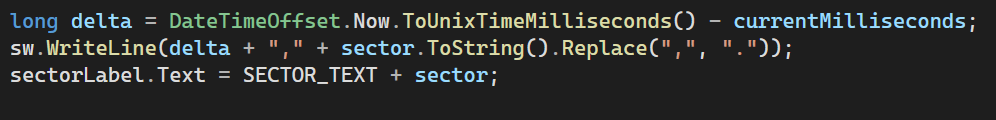
Далее идет подключение к GazePointer. Если оно прошло неудачно, то метод Record завершается присваиванием переменной isActive значения false и вызовом метода SetStartButton(), который возвращает программу в исходное состояние, изменив вид кнопки и отключив дополнительный интерфейс. В случае успеха создается конструкция try-catch-finally. В блоке try происходит получение текущего времени в виде строки, создание имени файла в формате CSV на основе данного времени, который является окулограммой. Затем изменяется элемент пользовательского интерфейса, отображающий текущее название окулограммы. После этого создается путь, куда этот файл будет сохранен. В том случае, если директория не существует по этому пути, то программа ее создает автоматически. Далее создается экземпляр класса StreamWriter, благодаря которому программа записывает этот файл согласно формату CSV, вызывая у него метод WriteLine().



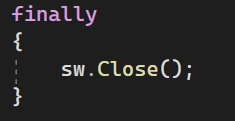
Затем запускается основной цикл while с условием, что переменная isActive равна true. В начале этого цикла выполняется получение данных с GazePointer путем вызова метода ReciveGazeDataSyn() у объекта класса GazeFlowAPI. Эти данные затем записываются в экземпляр класса CGazeData. Далее выполняется проверка на null: если данные не были получены, то цикл завершает свою работу, в противном случае вычисляет сектор, на который в данный момент смотрит пользователь, на основе полученных данных, а в конце записывает в файл пару время-сектор.



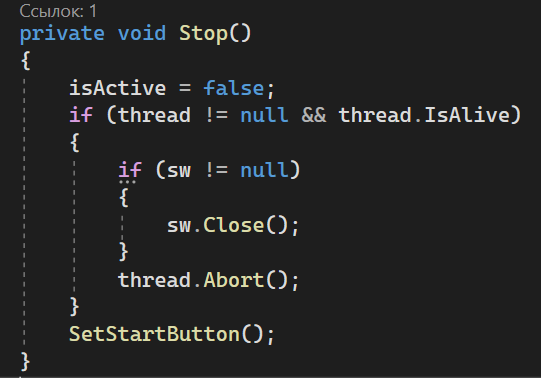




В конструкции try-catch-finally последний блок закроет запись файла.



Когда пользователь нажмет на кнопку “Стоп”, то будет выполнен метод Stop(), который завершит поток с генерацией окулограммы, сохранив ее.



**Запись окулограмм**

Для сбора статистических данных был выбран симулятор вождения City Car Driving, приближенный к реальности. Помимо основной программы, записывающей окулограммы, айтрекера и симулятора было задействовано следующее:

1. Утилита NVIDIA GeForce Experience, использованная для записи экрана.
2. Программа DroidCam, позволяющая использовать камеру смартфона в качестве веб-камеры для компьютера.

Итого было записано 6 окулограмм.

Осторожное вождение с внимательным слежением за дорогой:

Вождение с частым отвлечением на посторонние вещи и задержкой на них взглядом:

Проанализировав данные окулограммы, можно отметить, что при первом варианте вождения чаще всего встречается сектор 0, который сообщает о том, что водитель сконцентрирован на дороге, а также преимущественно встречаются сектора 1, 2, 3, 3.5, 4, которые говорят, что водитель следит за дорожными знаками, приближающимися транспортными средствами и другими участниками движения, поглядывая в зеркала заднего вида. При втором варианте наблюдается частое изменение секторов, свидетельствующее о том, что водитель часто отвлекается на посторонние вещи.

**Заключение**

Проведенная практика позволила успешно разработать программу на языке C# с использованием C# .NET Windows Forms, предназначенную для записи окулограмм c полученными данными от айтрекера GazePointer. Дальнейшие исследования включали создание шести окулограмм с помощью симулятора вождения City Car Driving в двух различных вариантах. Проведенный анализ полученных данных дал ценные выводы относительно визуальной активности водителя в различных ситуациях. Эти результаты имеют практическую ценность для улучшения безопасности и эффективности вождения. В ходе практики были получены ценные знания и навыки, которые смогут быть использованы в будущем при дальнейших исследованиях и разработках в области визуальной аналитики и управления транспортными системами.

**Список использованных источников**

1. <https://nami.ru/ob-institute>
2. <https://sourceforge.net/projects/gazepointer>
3. <https://github.com/szydej/GazeFlowAPI>