От дуэта роботов-музыкантов до Eyetracking, разработки алгоритма определения зрачка

Меня зовут Гордеев Иван Евгеньевич, мне 16 лет. Я ученик 10-го класса МБОУ «СОШ №36»г. Чита, член конструкторского бюро ЦМИТ «ZabLab». Занимаюсь программированием, робототехникой с 2017 года. За это время:

- Стал трехкратным победителем муниципального этапа Всероссийской олимпиады школьников по информатике, призером краевого этапа Всероссийской олимпиады школьников по информатике.
- Окончил ЧУДО «Компьютерный колледж» с красным дипломом.
- Стал участником летней школы инженерных проектов НГУ (Новосибирский государственный университет), «Стартап-экспедиции», НТТМ 2019.
- Стал победителем форума технологического предпринимательства «Краштест стартапов»; регионального этапа Всероссийской научно-практической конференции «Шаг в будущее» в симпозиуме «Математика и информационные технологии» и в секции «Информатика, вычислительная техника, телекоммуникации»; фестиваля «TechfestChita 2019»; краевого конкурса «Информационная безопасность» в номинации «Веб-сайт».

В 2018г., после того какокончил подготовительную образовательную программу по робототехнике в центре молодёжного инновационного творчества «ZabLab», вместе со своей группой приступил к нашему первому проекту – роботу-гитаристу. Вдохновились мыработой Бена Рирдона - http://arduino-projects.ru/projects/robot-gitarist/. Сначала был собран весь необходимый инструментарий и материал, а именно:

- Гитара с нейлоновыми струнами
- Arduino Uno
- Сервоприводы SG90
- Светодиодные матрицы
- Макетная плата
- Картонные коробки
- Провода
- Профиль металлический
- Труба ПВХ.

Из профиля мы изготовили раму для гитары. (Рама выделена красным цветом). Из картона



трубы ПВХ собрали туловище гитариста. светодиодных матриц и картона сделали голову. Затем мы начали думать над механизмом из сервоприводов и пришли к следующему: Далее мы разобрались со всей электроникой

приступили программированию.

Посмотреть, что мы имели

текущему К моменту онжом здесь https://www.instagram.com/p/BughNBzFAlo/

После началась отладка всей системы. В очередной раз, подключая гитариста к блоку питания, мы допустили ошибку, это привело к тому, что вся проводка сгорела. Быстро всё исправив, мы решили, что, так как времени еще много, можно сделать еще одного музыканта,



создание робота-драммера. К счастью, над туловищем и механизмом заморачиваться не пришлось, так как в ЦМИТ е был готовый скелет со всей электроникой и прочим, наша задача – восстановить.

В итоге получилось так: https://www.instagram.com/p/Bv_nuNnlH6E/



После восстановления Я занялся разработкой программного обеспечения, которое координировало бы работу роботов. Ha техно-фестивале «TechFestChita» наш дуэт занял первое место, а также попал на телевидение http://gtrkchita.ru/news/?id=24735.

В сентябре наша группа разделилась. Мне нужно было выбрать проект. Инструктор по робототехнике предложил мне почитать в интернете про EyeTracking'a. Система EyeTracking`а – технология, определяющая направление взора. Область применения EyeTracking a огромна! Небольшой список самых интересных сфер, где требуется система EyeTracking`a:

Когнитивные, медицинские, маркетинговые исследования.

- Управление транспортными средствами.
- Виртуальная реальность.
- Коммуникационные системы для полностью парализованных людей.
- Управление компьютером.

Началась работа над проектом с изучения труда Барабанщикова Владимира Александровича «Методы окулографии в исследовании познавательных процессов и деятельности». Данная книга познакомила нас с самыми популярными методами окулографии. Среди них:

- Электроокулография.
- Кинорегистрация.
- Видеорегистрация.
- Фотооптический метод.

Мы выбрали видеорегистрацию по следующим причинам:

- Доступные компоненты.
- Основа метода компьютерная программа.
- Проект легко масштабировать.

Суть данного метода заключается в поиске зрачка на изображении. Для того чтобы приводить изображения глаз к какому-то единому формату (у всех пользователей переменно окружение, освещение, отражение в зрачке, цвет глаз, длина ресниц), используется инфракрасная подсветка.

Разобравшись с видеорегистрацией, мы приступили к созданию очков, оснащённых веб-камерой и инфракрасной подсветкой. Оправа была взята из очков, купленных на рынке. Веб-камерой послужил Defender C-110. Подсветка была сделана из 5 миллиметровых инфракрасных светодиодов.

Когда аппаратная часть была готова, мы занялись разработкой алгоритма на высокоуровневом объектно-ориентированном языке программирования С#.

Начали мы с малого. Наш первый алгоритм просто искал барицентр всех тёмных точек. Главными её минусами являлись долгое время выполнения (200 миллисекунд, процессор IntelCore i3-6100, изображение 426х320) и требовательность к расположению камеры, она должна была снимать только глаз. Её реализация представлена в листинге I.

Далее мы немного модифицировали код. Уменьшили количество перебираемых пикселей в 2 раза. Изменили способ получения цвета пикселя. Изначально мы использовали метод класса Bitmap – GetPixel(int x, int y) (пространство имен: System.Drawing), который достаточно медленный. Вместо этого мы решили получать

данные напрямую из оперативной памяти, используя маршалинг. Это помогло нам увеличить время работы в 25 раз! 8 миллисекунд, при той же конфигурации. Реализация в листинге II.

После этого мы решили, что нужно разработать более универсальный алгоритм, который будет менее требователен к положению камеры относительно глаза.

Мы начали с того, что изучили то, как устроено растровое изображение. В этом нам помог труд Д.В. Иванова «Алгоритмические основы растровой машинной графики». Растровое изображение — это изображение, представляющее собой сетку пикселей — цветных точек (обычно прямоугольных). Каждый пиксель имеет свой цвет. Для работы с цветами мы будем использовать модель RGB. В ней цвет характеризуется интенсивностью красного, зелёного и синего. Обычно каждая компонента представляется числом от 0 до 255. Используются эти значения, так как они ровно помещаются в один байт. Чем меньше значения компонент, тем темнее изображение, соответственно, чем больше значения, тем светлее изображение.

Что мы вынесли из этого? Нам необходимо найти пиксели, чьи интенсивности красного, зелёного и синего цветов ниже определённого порога. При этом они должны образовывать круг. При выполнении этих условий можно сделать вывод, что это зрачок.

Работу такого алгоритма можно разделить на несколько этапов:

- 1. Вычисление множеств темных пикселей.
- 2. Нахождение множества, образующего круг.
- 3. Возвращение геометрического центра круглого множества в качестве результата.

После разработки алгоритма, мы приступили к проектированию динамической библиотеки классов, для того чтобы можно было легко создавать программы, связанные с окулографией. Библиотека получила название «ZabLabEyeTrackingAPI».