

СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ.....	3
ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	4
ВВЕДЕНИЕ.....	5
1 ТЕОРЕТИЧЕСКИЙ ОБЗОР	7
1.1 Актуальность	7
1.2 Обзор аналогов	8
1.3 Технологический стек.....	12
1.3.1 Языки программирования	12
1.3.2 Обработка график и распознавание рук	13
1.3.3 Определение жестов	14
1.3.4 Управление компьютером	16
1.3.5 Сохранение пользовательской настроек.....	16
1.4 Вывод.....	18
2 ТРЕБОВАНИЯ	20
2.1 Функциональные требования	21
2.2 Нефункциональные требования	22
2.2.1 Безопасность.....	22
2.2.2 Портативность	22
2.2.3 Удобство	23
2.3 Вывод.....	23
3 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ.....	24
3.1 Проектирование ПО.....	25
3.1.1 Варианты использования	25

3.1.2 Даталогическая модель.....	25
3.1.3 Диаграммы последовательность	26
3.2 Реализация ПО	29
3.2.1 Распознавание рук и получение информации о местонахождении опорных точек кистей.....	29
3.2.2 Определение направления движения точки	32
3.2.3 Составление и сравнение жестов	36
3.2.4 Управление компьютером	38
3.2.5 Пользовательские настройки	39
3.2.6 Создание пользовательского интерфейса	42
3.2.7 Повышение производительности	44
3.2.7.1 Выбор ключевых точек.....	44
3.2.7.2 Многопоточность	45
3.3 Вывод.....	47
4 ТЕСТИРОВАНИЕ	49
4.1 Автоматизированное модульное тестирование.....	49
4.2 Тестирование в виде опроса у студентов.....	49
4.3 Вывод.....	52
ЗАКЛЮЧЕНИЕ	53
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	55

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

БД - База Данных

ОС - Операционная Система

ПО - Программное обеспечение

HCI - Human - Computer Interaction.

CLI - Command Line Interface, интерфейс командной строки.

GUI - Graphical user interface, графический интерфейс пользователя

NUI - Natural user interface

СУБД - Система управления базами данных

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Жест – Набор направлений движений выбранных точек по осям.

Режим распознавания – Процесс распознавания рук для достижения своего цели.

Рабочий режим – Процесс управления компьютером в разных сценариях.

Модульное тестирование - Разновидность тестирования в программной разработке, которое заключается в проверке работоспособности отдельных функциональных модулей, процессов или частей кода приложения.

Даталогическая модель - Структурированное описание данных, которое определяет, как информация будет храниться и организована в базе данных.

Диаграмма последовательности - UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие акторов (действующих лиц) информационной системы в рамках прецедента.

Вариант использования - Описание поведения системы, когда она взаимодействует с кем-то (или чем-то) из внешней среды.

RGB - Формат цветовых каналов, при котором цвета передаются в следующем порядке: красный-зелёный-синий.

BGR - Формат цветовых каналов, при котором цвета передаются в следующем порядке: синий-зелёный-красный.

ВВЕДЕНИЕ

В 1970-е годы, с появлением персональных компьютеров, компьютеры стали доступны обычным пользователям. Вопрос о том, как проектировать интерфейсы связи человека с компьютером с точки зрения обычных пользователей, был поставлен на повестку дня, а также возникла концепция человеко-машинного взаимодействия. Человеко-машинное взаимодействие направлено на оптимизацию пользовательского опыта. Это может самым непосредственным образом повлиять на работу пользователя на компьютере.

При развитии технологий метод взаимодействия человека и компьютера тоже развивается. От первоначального взаимодействия с командной строкой до последующего появления пользовательского графического интерфейса. Теперь есть распознавание лиц для оплаты, голосовой помощник и так далее. Сегодня люди гонятся не только за красотой пользовательских интерфейсов, но и за более удобными и естественными методами взаимодействия человека с компьютером.

Наше ПО является реакцией на данную тенденцию в контексте развития взаимодействия между человеком и компьютером. Оно будет получать видеопоток с камеры и затем использовать нейронную сеть для распознавания движений рук. Затем ПО записывает и анализирует движения пальцев, которые интерпретируются как жесты. Приложение сопоставляет эти жесты с заранее определёнными действиями для управления персональным компьютером. Помимо этого, ПО предоставляет возможность пользователям настраивать жесты и связанные с ними операции в соответствии с их предпочтениями.

Наша конечная цель - упрощение взаимодействия человека с компьютером, а также обеспечение метода бесконтактного взаимодействия человека с компьютером в качестве дополнения к традиционным методами взаимодействия.

Для достижения наших целей нам необходимо решить следующие задачи:

- 1) Обзор предметной области;
- 2) Выбор технологического стека для захвата изображений;
- 3) Разработка алгоритма определения направления движения точек кистей рук;
- 4) Разработка алгоритма оформления жестов;
- 5) Выбор технологического стека для хранения жестов и проектирование метода хранения;
- 6) Выбор технологического стека для реализации управления компьютером при помощи жестов и проектирование способа управления;
- 7) Проектирование и разработка ПО;
- 8) Улучшение производительности в реальном времени.

Предметом исследования является процесс человеко-машинного взаимодействия с помощью жестовой.

Работа включает в себя введение, четыре главы, заключение и список использованных источников.

1 ТЕОРЕТИЧЕСКИЙ ОБЗОР

В данной главе представлен анализ существующих на сегодняшний день способов человеко-машинного взаимодействия и аналоги ПО. Также будет анализирован технологический стек разработки.

1.1 Актуальность

HCI - изучение, планирование и разработка взаимодействия между людьми (пользователями) и компьютерами. При развитии технология HCI в основном прошёл три стадии:

- 1) CLI - Взаимодействие с компьютером при помощи команд.
- 2) GUI - Взаимодействие с компьютером посредством мыши, клавиатуры и графического интерфейса пользователя.
- 3) NUI - Взаимодействие с компьютером максимально естественным образом (голосом, мимикой, жестами), избавляясь таким образом от клавиатуры и мыши.

Основной недостаток CLI заключается в том, что требуется от пользователей запоминания различных командных строк, что может быть нагрузкой на память пользователя. Появление графических интерфейсов с меньшей сложностью на обучение привело компьютеры в повседневную жизнь. В настоящее время большинство методов взаимодействия человека с компьютером, таких как мобильные приложения и веб-страницы, представляют собой графические интерфейсы. Однако теперь постепенно в жизни людей становятся популярными NUI, например, голосовых помощников. Преимущество NUI в том, что он имеет метод взаимодействия, более близкий к естественному поведению человека, чем GUI, и проще на обучение.

NUI может стать дополнением к существующим методам взаимодействия за счёт более естественного метода взаимодействия и более низкой стоимости обучения. Он может упрощать взаимодействие между человеком и компьютером. Взаимодействие при помощи жестов относится к NUI.

Взаимодействие при помощи жестов предлагает множество преимуществ с точки зрения доступности, простоты использования и погружения. Оно может существенно повлиять на то, как люди будут взаимодействовать с компьютерами в будущем. И ожидается, что в ближайшие годы рынок управления жестами будет расти.

В то же время пандемия covid-19 в предыдущие годы также открыла нам этот факт. Метод взаимодействия при помощи жеста. имеет ещё одно преимущество перед традиционными GUI и CLI - он позволяет бесконтактное взаимодействие с компьютером. Взаимодействие при помощи жестов позволяет пользователям взаимодействовать без физического контакта с каким-либо устройством в течение всего процесса взаимодействия. Это может быть полезно в особых обстоятельствах, таких как эпидемия.

1.2 Обзор аналогов

На рынке существует множество аналогов нашей ПО. Отличие аналогов от нашего ПО в основном заключается в подходе к реализации распознавания жестов. Обсудим два репрезентативных аналога. Два обсуждаемых аналога делают больше, чем просто управлять компьютерами. Их также можно использовать для управления роботами или использовать в виртуальной реальности. Но так как цель нашего

проекта только касается с взаимодействием с компьютером при помощи жестов. Поэтому обсуждение двух аналогов и сравнение с нашим ПО ограничиваются частью компьютерного управления.

Первый аналог – это продукт, использующий перчатки в качестве устройств распознавания жестов. Перчатки Manus VR Gloves представляют типичное решение этого типа. Продукт использует перчатки с рядом датчиков для получения информации о местонахождении руки в пространстве для определения жестов. Датчики передают информацию в ПО, которое управляет компьютером на основе этой информации. Преимуществами этих продукт являются их высокая степень точности и не подвержены влиянию факторов окружающей среды. На работы продуктов практически не влияют физические факторы, такие как интенсивность окружающего освещения и другие факты. Но у них есть и следующие недостатки:

- 1) Проблема рабочего времени от батареи: сами перчатки необходимо подзарядить. Так что будут проблемы с рабочего времени от батареи.
- 2) Высокая стоимость: цены этих продукт не низки, поскольку цена производства различных типов датчиков, не низкая. Возьмём, к примеру, перчатки Manus VR Gloves. Цена обычно составляет несколько сотен долларов. Некоторые из них могут стоить несколько тысяч долларов.
- 3) Меньшее удобство и комфорт использования: пользователю необходимо носить перчатки. Может повлиять на свободу и комфорт пользователя.

- 4) Низкая совместимость: программное обеспечение для этого типа продуктов обычно реализует специальный API. Другими словами, программное и аппаратное обеспечение данного типа продуктов разных компаний несовместимо друг с другом.

Второй аналог - продукт, который использует специальную камеру для захвата изображений, а затем использует нейронную сеть для распознавания жестов. Этот аналог ближе к нашему ПО, чем предыдущий, поскольку также использует нейронную сеть. Но для более точного распознавания обычно предназначена специальная камера. Камеры Leap Motion Controller типичны этого типа. В продуктах Leap Motion Controller для получения изображений используется специальная инфракрасная камера. Преимуществами этих продукт являются высокая точность, отсутствие необходимости учитывать время непрерывной работы и комфорт пользователя без ношения дополнительного оборудования. Но есть у него и следующие недостатки:

- 1) Высокая стоимость: эти продукты также имеют специальное аппаратное обеспечение, поэтому цена не низкая. Но цена немного ниже, чем прошлый аналог. Например, цена продукта Leap Motion Controller обычно составляет менее 500 долларов.
- 2) Восприимчивость к воздействию окружающей среды: этот тип продукта чувствителен к освещению и условиям окружающей среды, и на него могут влиять внешние воздействия окружающей среды, влияющие на точность распознавания жестов. Если пользователь носит украшения на руках, это также может повлиять на точность распознавания.

3) Низкая совместимость: программное обеспечение для этого типа продуктов обычно реализует специальный API. Другими словами, программное и аппаратное обеспечение данного типа продуктов разных компаний несовместимо друг с другом.

По сравнению с этими аналогами преимуществом нашего ПО является то, что оно представляет собой чисто программное решение, не требующее специального оборудования. В таблице 1.1 подробно показано сравнение нашего ПО с аналогами.

Таблица 1.1 – Сравнение ПО с аналогами

Продукт	Невосприимчивость	Рабочего времени от батареи	Стоимость	Совместимость
Наше ПО	Подвержен влиянию окружающей среды	Не нужно заряжать	Бесплатно	Может использовать любую веб-камеру
Leap motion controller	Подвержен влиянию окружающей среды	Не нужно заряжать	Несколько сот долларов	Нужно использовать особенный аппарат
Manus VR Gloves	Не подвержен влиянию окружающей среды	Нужно заряжать	От несколько сот до несколько тысяч долларов	Нужно использовать особенный аппарат

Каждый вариант может достичь точности на уровне миллиметра.

1.3 Технологический стек

1.3.1 Языки программирования

При выборе языка программирования мы должны учитывать сложность разработки и наличие на языке полных инструментов, которые помогут достичь наших целей.

Язык программирования, который будет выбран, должен обладать следующими характеристиками:

- 1) Язык должен позволять нам легко манипулировать и обрабатывать изображения.
- 2) Язык должен быть максимально простым. Это позволяет нам больше сосредоточиться на обработке данных, а не на таких проблемах, как предотвращение утечек памяти.
- 3) Язык должен иметь множество инструментов и документации, которые помогут нам с обработкой изображений и распознаванием жестов. Например, в следующем обсуждении упоминается mediapipe. Google не предоставляет официальных руководств и демонстраций на java, но предоставляет их на python.
- 4) Написанный код может работать на разных операционных системах, и нет необходимости рассматривать вопросы как портирования.

В результате python был выбран в качестве языка программирования.

1.3.2 Обработка график и распознавание рук

Для реализации распознавания жест, необходимо выполнить две задачи:

- 1) получение изображения и выполнение предварительную обработку изображения;
- 2) распознавание жестов на обработанных изображениях.

Для получения изображения и предварительной обработки применяется библиотеку `opencv`. Причины выбора именно его следующие:

- 1) Он может легко захватывать изображения с любой веб-камеры.
- 2) Существует множество операций с изображениями, которые могут помочь выполнить предварительную обработку изображения.

После завершения серии предварительной обработки изображения начинается распознавание жестов. Для распознавания жестов используется библиотеку `mediapipe`. Причины использования этой библиотеки следующие:

- 1) Он поставляется с моделью с открытым исходным кодом. Нам не нужно самим разрабатывать новую модель для распознавания жестов.
- 2) Он может читать координаты точек в пространстве. Полученные данные мы можем обработать напрямую.
- 3) Существует довольно обширная документация, которая поможет нам использовать библиотеку.

Mediapipe определяет 21 опорных точек кистей рук. Распределение 21 точек показано на рисунке 1.1.

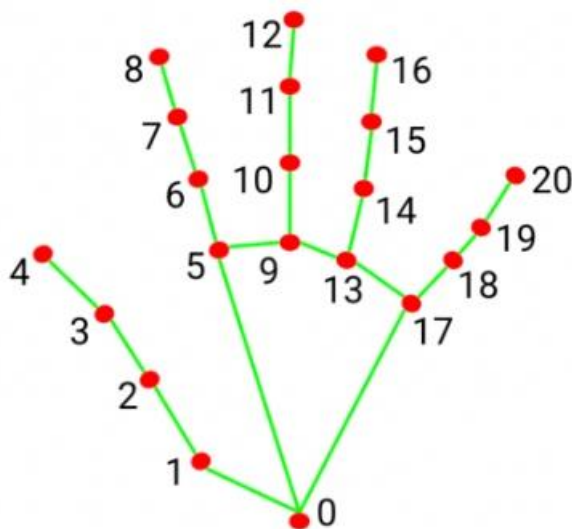


Рисунок 1.1 – Опорные точки кистей рук

1.3.3 Определение жестов

В нашем контексте под жестом понимается набор направлений движения точек руки в пространстве или на плоскости в течение определённого периода времени.

Определение направления движения точки осуществляется не просто по смещению точки. Нестабильность распознавания изображений приведёт к тому, что распознанные точки будут колебаться, даже если рука пользователя не будет двигаться. Составляется нечёткая логика, основанную на смещении, для определения направления движения каждой точки руки. Таким образом, погрешность, вызванная джиттером, снижается.

При сопоставлении жестов направления движения всех полученных точек будут сравниваться с направлениями движения каждой точки в заданном жесте. Преимущество этого метода заключается в том, что он позволяет лучше справляться с неопределённостью и двусмысленностью при распознавании жестов. Например, пользователь может двигаться не точно в соответствии с заранее заданными жестами или на жесты могут влиять шум и помехи. Более того, обработка данных относительно проста, а нагрузка на компьютер будет относительно низкой. Это позволяет ПО может работать в реальном времени. В то же время этот метод сравнительно прост в реализации и не требует создания модели, подобно методу с использованием нейронную сеть. По сравнению с такими методами, как нейронные сети, точность распознавания этого метода относительно низка, и сложно проектировать сложные жесты. Однако, учитывая, что наш сценарий использования — это управление персональными компьютерами, а не сценарии применения, такие как управление промышленными машинами или медицина, которые требуют более высокой точности, этот недостаток приемлем. Что касается недостатка, заключающегося в сложности разработки сложных жестов, то, учитывая наши сценарии использования, дизайн жестов должен быть максимально простым, чтобы пользователи могли быстро его запомнить и выполнить, поэтому этот недостаток также приемлем.

На Python мы можем использовать библиотеку `skfuzzy` для построения нечёткой логики.

1.3.4 Управление компьютером

Управление компьютером может быть достигнуто при помощи библиотеки `pyautogui`. Она предоставляет богатые функции и простой в использовании API, который может имитировать ввод с мыши и клавиатуры и реализовывать автоматизированные операции.

Следует отметить, что при управлении компьютерами следует учитывать проблемы безопасности. Должно разумно спроектировать привязку жестов и операций, чтобы предотвратить возможные непредвиденные ситуации.

1.3.5 Сохранение пользовательской настроек

Необходимо реализовать функцию пользовательской настройки жестов. Другими словами, для хранения пользовательских настроек необходим метод постоянного хранения. Обычно существует два способа хранения пользовательских настроек. Первый - сохранить настройки в виде файла, а затем прочитать и записать её. То есть реализовано в виде сохранения в файле. Другой - сохранить её в базе данных.

Резервное копирование упрощается с помощью файловой хранилища. Этот метод очень портативен, и пользователи могут напрямую копировать файлы для копирования настроек и резервного копирования настроек. При этом методе не требуется дополнительной СУБД, поэтому нагрузка на компьютер будет меньше. Однако согласованность данных и целостность трудно поддерживать. Из-за прямых манипуляций обслуживание данных становится более трудным,

особенно для данных со сложной структурой. А по мере увеличения объёма данных производительность может снизиться. Кроме того, параллелизм низкий, файловый ввод-вывод неэффективен при обработке нескольких одновременных запросов, а также часто возникают конфликты чтения и записи.

При хранении данные в БД обеспечиваются согласованность и целостность данных. Система БД оптимизирована для быстрого поиска и запроса данных. Но СУБД требуют дополнительных затрат. Будет дополнительная нагрузка на компьютер. В то же время резервное копирование базы данных сложнее, чем непосредственное использование файлов.

Что касается с сохранением настройки в нашем ПО, мы надеемся, что оно может:

- 1) обеспечивать целостность и согласованность данных,
- 2) иметь хорошую переносимость,
- 3) легко создавать резервные копии настроек,
- 4) иметь как можно меньше накладных расходов на компьютере,
- 5) гарантировать хорошую производительность, при увеличении объём данных.

Наше ПО не предъявляет высоких требований к высокой степени параллелизма, поскольку чтение и запись данных происходят нечасто.

В итоге выбрана SQLite в качестве метода хранения пользовательских настроек. Она обладает характеристиками БД, гарантирующими целостность и согласованность данных. Независимый серверный процесс отсутствует, поэтому нагрузка на компьютер невелика. Он хранит данные в файле с суффиксом db. Нужно только

скопировать файл базы данных для резервного копирования и переноса настроек. SQLite не требует сложной настройки, как большинство СУБД, и очень удобен в использовании. Его недостаток в том, что из-за простоты конструкции он не подходит для сложных структур данных и сложных запросов. Хотя параллелизм хуже, чем у больших баз данных, таких как postgresql, учитывая, что наша структура данных не сложна и не будет одновременно выполняться большое количество операций чтения и записи данных, SQLite очень подходит к нашему ПО.

1.4 Вывод

Теоретический обзор раскрывает актуальность проблемы, обоснованность и недостатки существующих решений. Были рассмотрены существующие технологии, необходимые для реализации этого ПО, и в итоге был выбран следующий стек технологий:

- Python 3
- Opencv
- SQLite
- Mediapipe
- Skfuzzy
- Pyautogui

В конце концов было решено непрерывно отслеживать движение каждой точки руки. Будет определено направление движения каждой точки по трём осям пространства. Набор направлений движения точек является жестом.

Наконец, определяются возможности и функции, которыми должно обладать ПО, а также определяется рациональность выбранного технологического стека.

2 ТРЕБОВАНИЯ

При разработке ПО жестового управления есть два варианта: создать универсальный набор жестов или дать возможность пользователям самостоятельно определять этот набор. Они имеют свои преимущества и недостатки. В ПО оба они будут реализованы. То есть в ПО есть базовый набор жестов для управления и позволяет пользователям определять новый набор жестов или изменять старый набор.

Некоторые жесты могут быть не уникальными в рамках всей системы. Это в зависимости от контекста. Один жест, может быть, используется в разных контекстах. Но в одном контексте один жест только может соответствовать одной операций.

В результате получено, что наше ПО должно позволяет пользователям управлять компьютером с помощью жестов и позволяет пользователям настраивать жесты и привязывать к ним соответствующие операции. Поэтому ПО будет иметь два режима распознавания. Это режим управления и режим настройки соответственно.

В режиме управления ПО непрерывно распознавает жесты и выполняем соответствующие операции на компьютере по жесту. И в режиме управления ещё есть несколько рабочих режимов для взаимодействия с разными приложениями. И в одном рабочем режиме один жест только может привязан к одной операции.

Режим настройки отвечает за распознавание жестов человека и привязку жестов к соответствующим операциям.

2.1 Функциональные требования

Функциональные требования ПО показаны в таблице 2.1 и таблице 2.2.

Таблица 2.1 Функциональные требования ПО режима управления

ID	Описание
1	Пользователь может выходить из ПО
2	Пользователь может перейти в режим настройки
3	ПО должно распознавать жесты и выполнять соответствующие операции на основе жестов
4	Пользователь может приостановить операции. Пользователь не выходит из ПО во время паузы. Работа камеры прекращается, и распознавание изображения не производится. Работы можно перезапустить после паузы.
5	Пользователь может выбирать рабочие режимы.
6	ПО может читать настройки.

Таблица 2.2 Функциональные требования ПО режима настройки

ID	Описание
7	ПО может распознавать и записывать жесты и устанавливать соответствующие операции, привязанные к жестам. Настройки сохраняются в БД.
8	Пользователь может войти в режим управления.
9	Пользователь может приостановит. При остановке пользователь не выходит из ПО. Работа камеры

	прекращается, и распознавание изображения не производится. Запись жеста прервана.
10	Пользователь может удалять существующие настройки.
11	Жест должен не был записан, если он раньше был привязан к операции в одном рабочем режиме.
12	Пользователь может выбирать рабочий режим, в который будет записана настройка.
13	Если при настройке операция уже была привязана к жесту раньше, то жест обновляется.

2.2 Нефункциональные требования

2.2.1 Безопасность

Операции на компьютер должна быть спроектирована с учётом безопасности. То есть ПО должно не на компьютере опасные операции. И целостность операций должна быть гарантирована для предотвращения непредвиденных последствий.

2.2.2 Портативность

Пользовательские настройки должны легко переноситься на другой компьютер.

2.2.3 Удобство

Пользователь должен иметь возможность видеть результат распознавания в пользовательском интерфейсе. При записи жестов должны быть чёткие инструкции, чтобы пользователи могли понимать статус записи жестов.

2.3 Вывод

В этой главе были определены функциональные и нефункциональные требования. Предварительно определены возможности ПО.

Мы определили, что ПО должно иметь два режима распознавания. Режим управления отвечает за управление компьютером. Режим настройки отвечает за сохранение настроек. И в режиме управления ещё есть несколько рабочих режимов, которые применяются в разных сценарий – то есть для управления разными приложениями.

И определено, что ПО не должно выполнять опасные операции на компьютере. Необходимо обеспечить удобство пользовательского интерфейса и портативность пользовательских настроек.

3 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

В предыдущей главе мы обсудили технологический стек, который будет использоваться для реализации ПО, проанализировали аналоги ПО и выяснили актуальность ПО. В этой главе мы подробно обсудим проектирование и конкретную реализацию ПО. Мы будем решать следующие задачи:

- распознавание рук и получение информации о местонахождении опорных точек кистей;
- определение направление движения опорных точек кистей в течение определённого периода времени;
- управление компьютером;
- сохранения пользовательских настроек;
- создание пользовательского интерфейса
- оптимизация производительности ПО.

3.1 Проектирование ПО

3.1.1 Варианты использования

Спроектированы варианты использования по требованиям. Результат показан в рисунке 3.1.

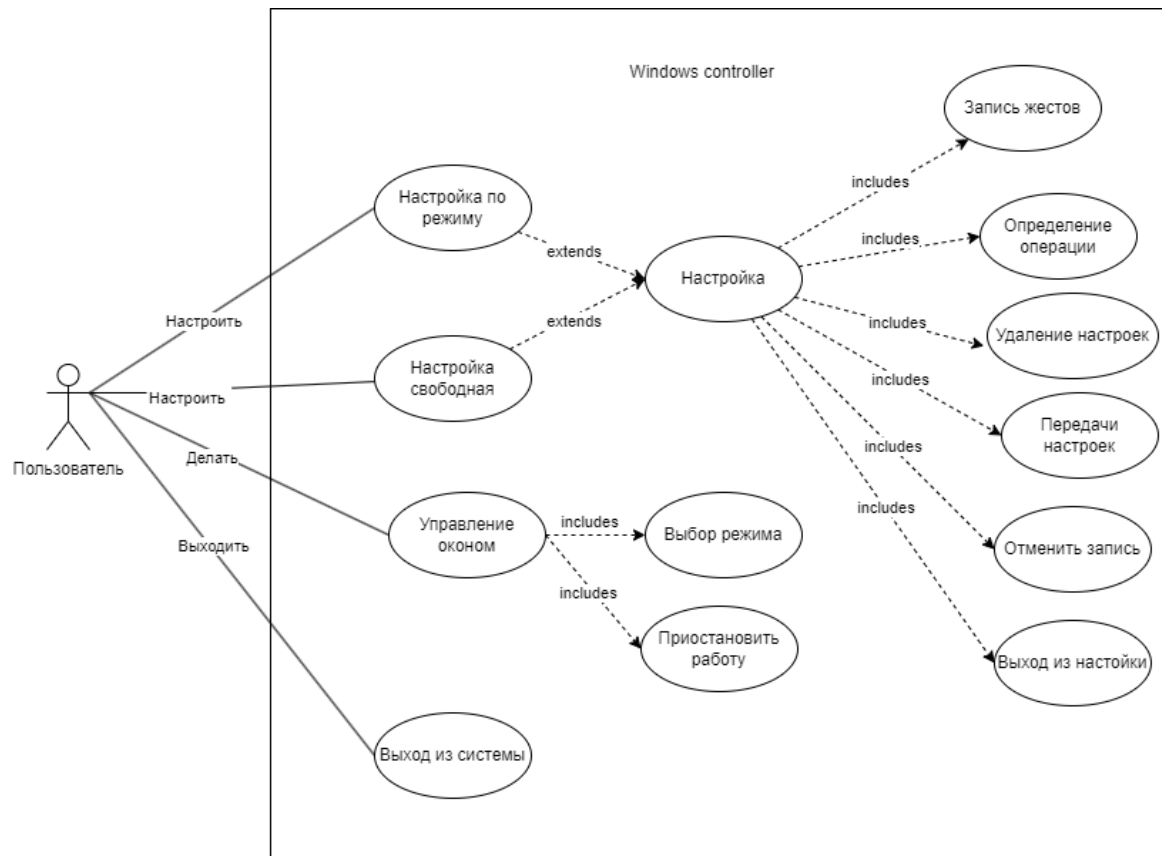


Рисунок 3.1 - Варианты использования

3.1.2 Даталогическая модель

Проектированная БД показана в рисунке 3.2 в виде даталогической модели.

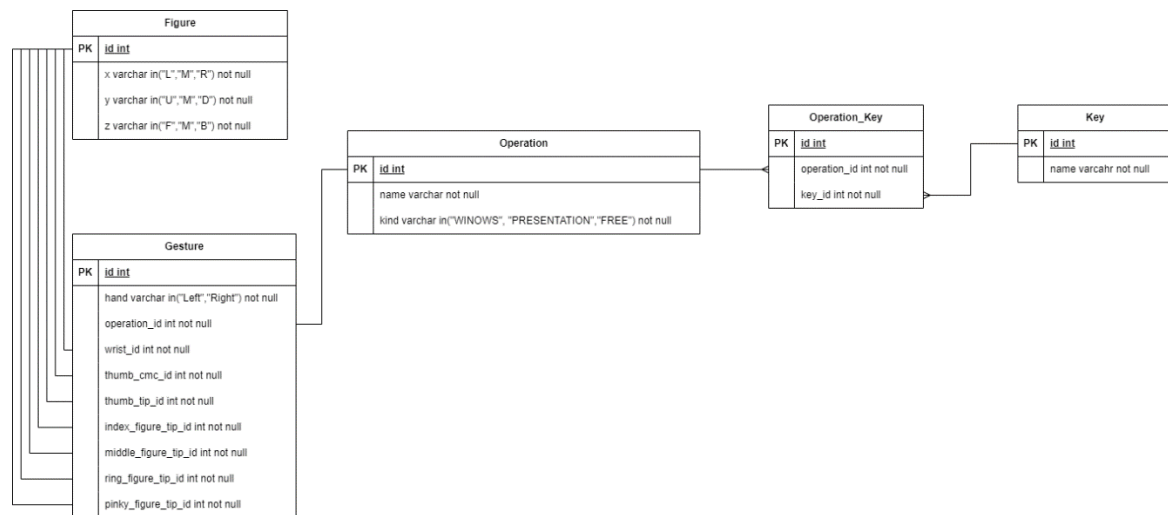


Рисунок 3.2 – Дatalogическая модель БД

3.1.3 Диаграммы последовательность

ПО имеет два режима. Поэтому мы получим две диаграммы. Это диаграмма управления в рисунке 3.3 и диаграмма настройки в рисунке 3.4.

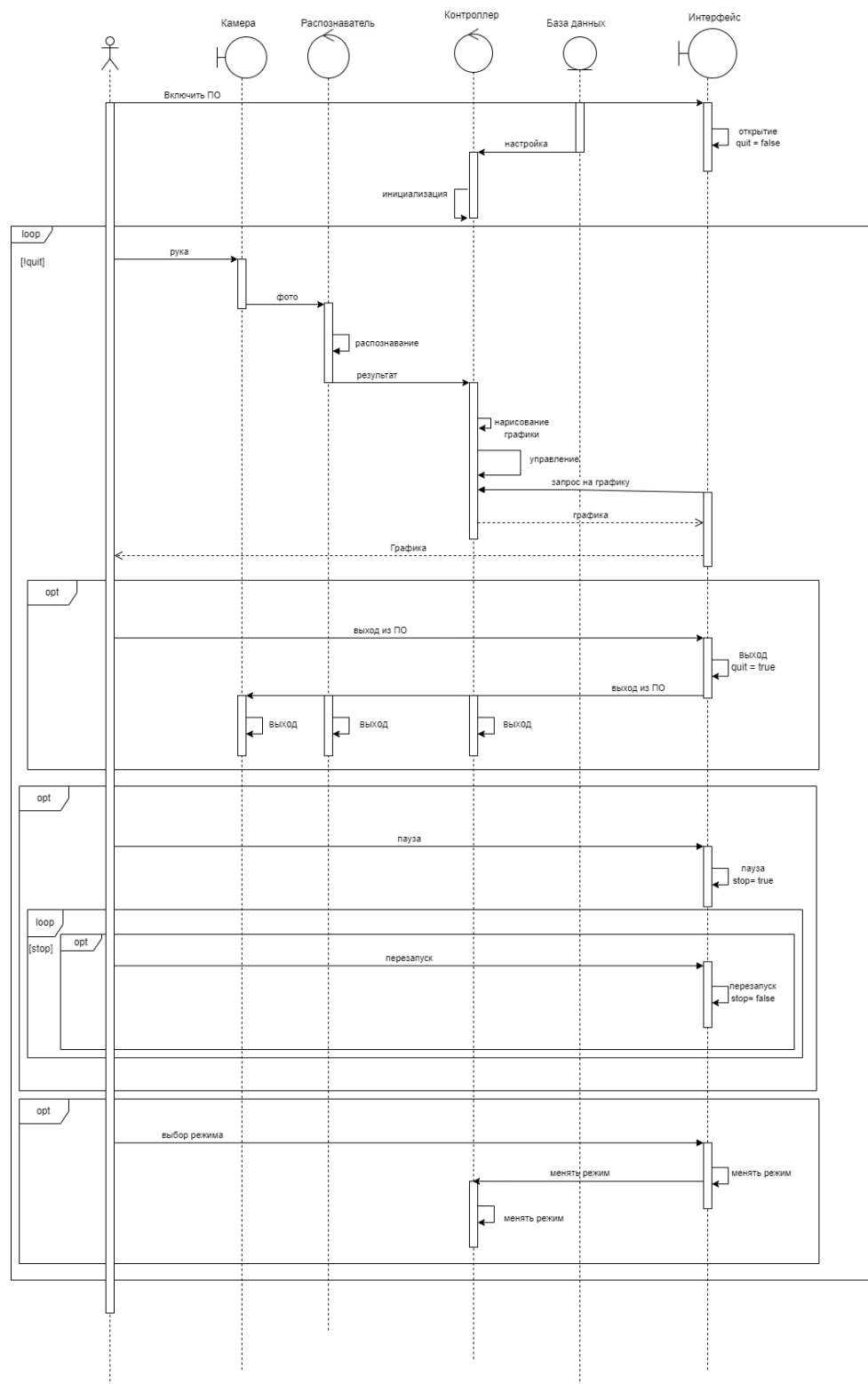


Рисунок 3.3 - Диаграмма последовательности режима управления

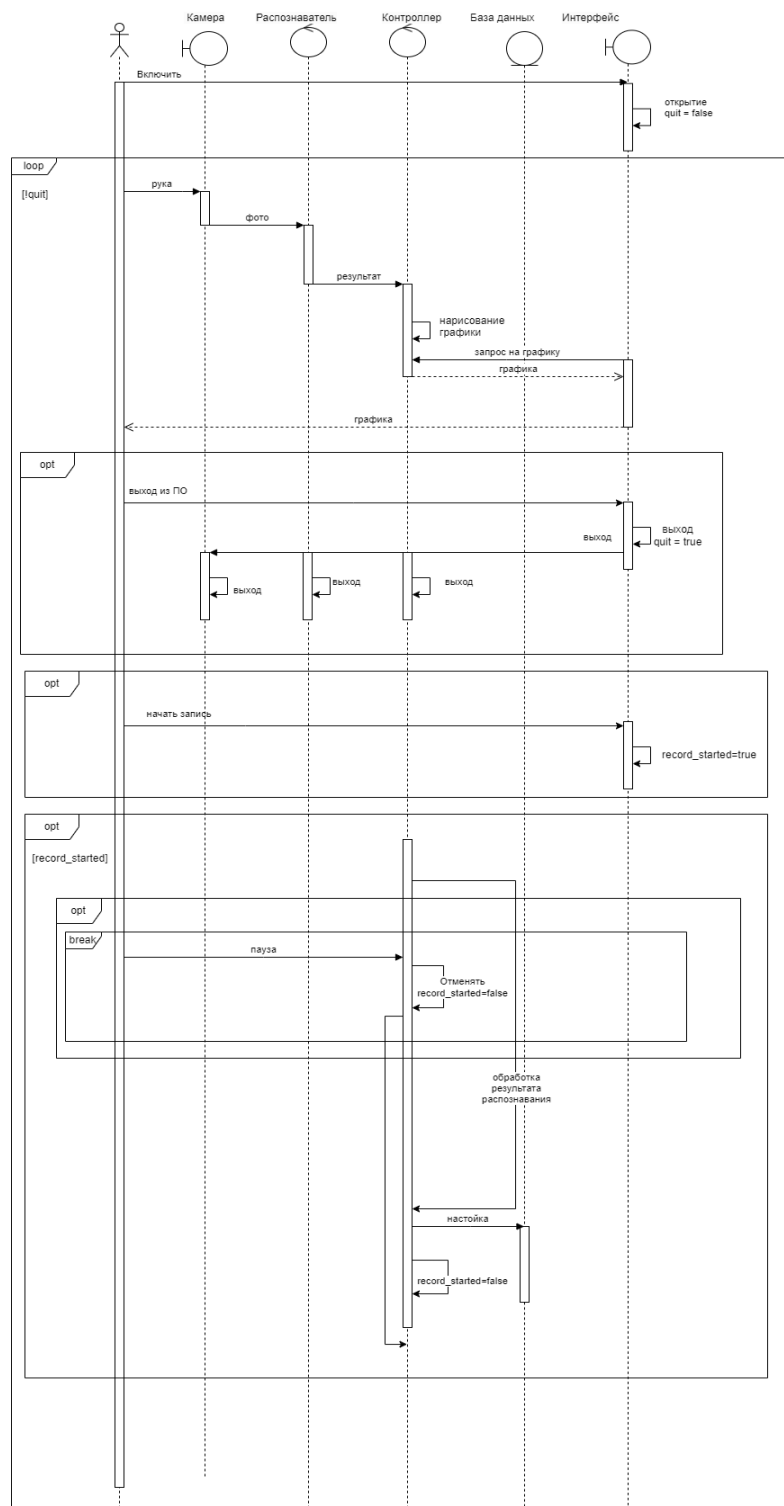


Рисунок 3.4 - Диаграмма последовательности режима настройки

3.2 Реализация ПО

3.2.1 Распознавание рук и получение информации о местонахождении опорных точек кистей

Как обсуждалось выше, будет использоваться `mediapipe` для распознавания рук и получения информации о местонахождении точек. И будет использоваться `opencv` для получения изображений. Полученные изображения необходимо предварительно обработать перед их распознаванием.

Во-первых, необходимо перевернуть изображение, чтобы пользователь мог видеть его в зеркальном отображении для удобства использования. Для этого можем использовать функцию `cv2.flip()`. Кроме того, следует отметить, что формат изображения, считываемого `opencv`, - BGR. Нам необходимо преобразовать изображение в формат RGB, чтобы улучшить качество распознавания и позволить пользователю видеть изображение с правильными цветами. Для преобразования формата достаточно использовать функцию `cv2.cvtColor()`.

Для использования нейронной сети для распознавания рук сначала необходимо инициализировать модель. Параметры модели приведены в таблице 3.1.

Таблица 3.1 Параметры модели

Параметр	Описание	Значение
model_complexity	Сложность модели. Тем сложность больше, чем точность выше.	1
min_detection_confidence	Минимальное значение достоверности для обнаружения рук, которое можно считать успешным.	0.5
min_tracking_confidence	Минимальное доверительное значение, позволяющее считать, что ориентиры в виде рук успешно отслеживаются.	0.5
max_num_hands	Максимальное количество рук для распознавания	1

Mediapipe отображает координаты 21 точки руки в трёхмерное пространство. Если установлена начальная точка пространственных координат в верхнем левом углу изображения, то при нормальных обстоятельствах значение оси Y после нормализации будет от 0 по -1, а

оси X от 0 по 1 как показано в рисунке 3.5. Если на изображении отсутствуют некоторых точек, mediapipe предполагает их расположения как показано в рисунке 3.6. Это предположение может быть неточным, поэтому из соображений безопасности, по крайней мере в режиме управления, ПО не должно ничего делать с компьютером, когда руки не полностью присутствуют на изображении. Эта нормализация выполняется относительно размеру изображения.

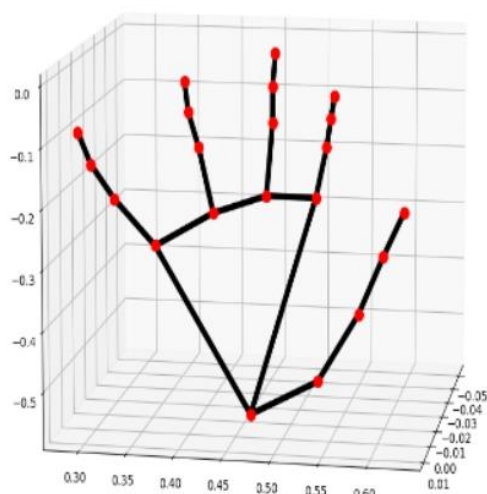


Рисунок 3.5 - Нормальная нормализация

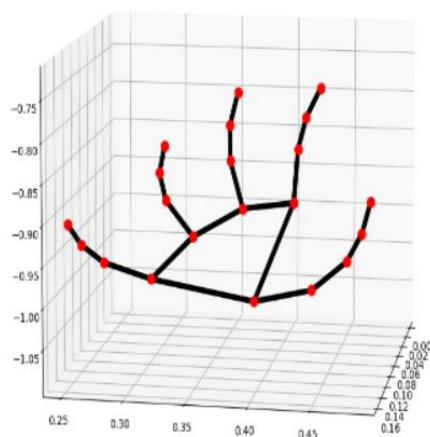


Рисунок 3.6 - Нормализация при отсутствии некоторых точек

Начальная точка нормализации `mediapire` находится в левом нижнем углу изображения. Поэтому для получения двумерного координата изображения, нужно умножить нормализованные координаты на ширину и высоту изображения, чтобы получить положение точки в пикселях.

Ось Z имеет особенность. В документации написано, что нормализация оси Z аналогична нормализации оси, и начальная точка координат для оси Z — это координаты точки 0. Поэтому для получения координата в пикселях нужно умножить значение оси Z на ширину изображения. Однако такой подход не гарантирует точность обработки значений оси Z , поскольку в официальной документации отсутствует подробное объяснение.

3.2.2 Определение направления движения точки

Уже обсуждалась нечёткая логика при обсуждении технологического стека. Координаты точек и временные метки собираются непрерывно. Вычисляются перемещения всех точек между двумя последовательными точками времени по трём осям в пространстве. Затем на основе этих смещений составляется нечёткая логика для определения направления движения точки.

В течение определённого периода времени будут собраны все направления движения, оценённые по трём осям точки. То есть, если предположить, что направление движения точки внутри интервала времени t будет определено, то t_1 — время начала этого интервала

времени, а t_2 — время окончания этого интервала времени. Это не просто вычислить смещение от t_1 до t_2 и затем определить направление движения точки. ПО будет непрерывно собирать n сегментов смещения за интервал времени, а затем определять направление движения за интервал времени t путём интегрирования направлений всех n смещений. Через n смещений получится n направлений движения. Большинство направлений движения n направлений движения точки в интервал t будет рассматриваться как направления движения точки внутри интервала времени t . Этот метод может предотвратить ошибочное определённое направления движения, вызванное плохим распознаванием в момент времени t_2 в крайних случаях.

При проектировании нечёткой логики по каждой из трех осей будет три направления движения:

- 1) ось X: влево (L), вправо (R), неподвижно (M)
- 2) ось Y: вверх (U), вниз (D), неподвижно (M)
- 3) ось Z: вперёд (F), назад (B), неподвижно (M)

Определение направления движения не достигается простым вычислением смещения в единицах пикселей. Потому что есть недостаток в определении направления смещения в единицах пикселей, то есть прямое расстояние между рукой и камерой будет влиять на результаты оценки. Для того же движения в реальном мире смещение в пикселях будет больше, когда камера находится ближе к камере, и смещение в пикселях будет меньше, когда камера находится дальше. Набор нечёткой логики в этом случае будет работать только в пределах определённого интервала расстояний.

Поэтому чтобы нечёткая логика могла определять направление движения точки независимо от погрешности, вызванной расстоянием между рукой и камерой, необходима новая единица, описывающая перемещение точки на разных расстояниях. Мы решили использовать расстояние между точкой 0 и точкой 1 в качестве единицы описания смещения. Уравнение для нормализации всех смещений по осям в этой единице за интервал времени t показано в уравнении (1), уравнении (2) и уравнении (3).

$$\Delta x = \frac{x_{t'_2} - x_{t'_1}}{\bar{l}} * 100, \quad (1)$$

$$\Delta y = \frac{y_{t'_2} - y_{t'_1}}{\bar{l}} * 100, \quad (2)$$

$$\Delta z = \frac{z_{t'_2} - z_{t'_1}}{\bar{l}} * 100, \quad (3)$$

Где \bar{l} – среднее расстояние между точкой 0 и точкой 1 за интервал времени t , единица в пикселе. t'_1 - начало времени одного сегмента в интервал t . t'_2 - окончание времени одного сегмента в интервал t

Помимо устранения ошибки, вызванной расстоянием между камерой и рукой, это также может в определённой степени решить проблему, вызванную размером человеческой руки - то есть, чем меньше рука, тем меньше будет смещение.

Причина выбора точки 0 и точки 1 заключается в том, что в большинстве жестов, по сравнению с другими точками, эти две точки не будут заблокированы. Это обеспечивает достоверность измерений расстояния между точкой 0 и точкой 1.

Замечено, что даже если рука не движется, координаты распознанной точки могут быть нестабильными. Поэтому необходимо

устранить ошибку, вызванную дрожанием результата распознавания модели. Решение о перемещении точки вдоль определённой оси принимается только в том случае, если амплитуда её движения вдоль этой оси достигает определённого порога. В противном случае состояние движения этой точки вдоль этой оси считается равным М, то есть никакого движения не происходит.

В результате получены отношения между направлением и смещением по трём осям, и они показаны в рисунке 3.7, рисунке 3.8 и рисунке 3.9.

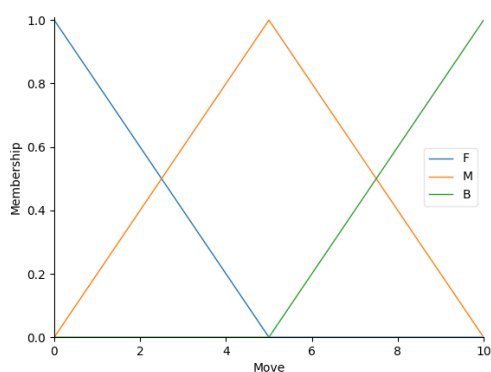


Рисунок 3.7 - Направление движения по оси Z

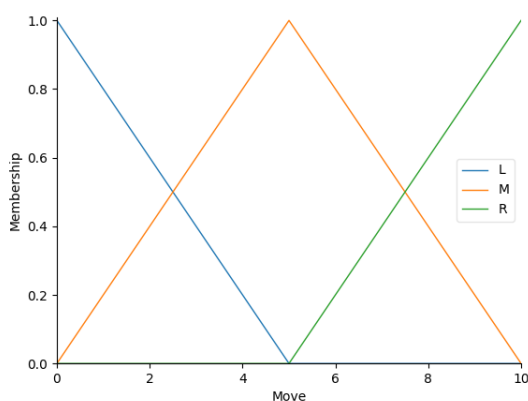


Рисунок 3.8 - Направление движения по оси X

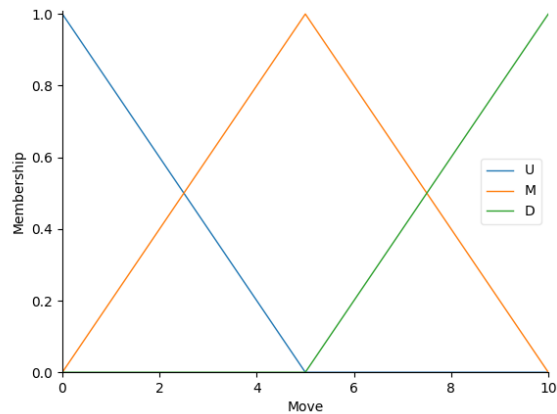


Рисунок 3.9 - Направление движения по оси Y

3.2.3 Составление и сопоставление жестов

ПО воспринимает набор направлений движения всех точек по трём осям пространства как жест. Поскольку mediapipe использует точку 0 в качестве начала координат по оси Z, ПО не может определять движение всей руки по оси Z. Она может только определять движение других точек по оси Z относительно точке 0. Чтобы облегчить распознавание жестов и предотвратить случайное управление компьютером пользователями при бессознательном движении рук, мы оговариваем, что состояние движения точки 0 всех определённых жестов в трех направлениях должно быть отсутствием движения. Таким образом, работать с компьютером можно только тогда, когда точка 0 неподвижна.

Сопоставление жестов осуществляется путём сопоставления направлений движения всех точек по трём осям. Только когда направления движения всех точек по трём осям совпадают с настройкой, жест считается распознанным.

Но поскольку точность полученных смещений по оси Z не гарантируется. Поэтому в ПО реализовано две возможности. Первая возможность заключается в определении направления движения точек в двумерном направлении. Жесты также состоят из движений точек в двумерном направлении. Другая возможность заключается в определении направления движения точек в трёхмерном направлении. Жесты также состоят из движений точек в трёхмерном направлении.

Программист может сам настроить при нахождении операции по жесту, именно направления каких точек по каким осям будет сравниваться через настроить объект класса operation, который показан в рисунке 3.10, при создании объекта.

```
class Operation:
    # LiaoYihong-1
    def __init__(self, fun=None, hand=None, name=None):
        self.__name = name
        self.__operation = fun
        self.__hand = hand
        self.__x = [const.UNDEFINED for _ in range(const.size)]
        self.__y = [const.UNDEFINED for _ in range(const.size)]
        self.__z = [const.UNDEFINED for _ in range(const.size)]
        self.__check_x = [const.WRIST, const.THUMB_CMC, const.THUMB_TIP, const.INDEX_FINGER_TIP,
                           const.MIDDLE_FINGER_TIP, const.RING_FINGER_TIP,
                           const.PINKY_FINGER_TIP]
        self.__check_y = [const.WRIST, const.THUMB_CMC, const.THUMB_TIP, const.INDEX_FINGER_TIP,
                           const.MIDDLE_FINGER_TIP, const.RING_FINGER_TIP,
                           const.PINKY_FINGER_TIP]
        self.__check_z = [const.WRIST, const.THUMB_CMC, const.THUMB_TIP, const.INDEX_FINGER_TIP,
                           const.MIDDLE_FINGER_TIP, const.RING_FINGER_TIP,
                           const.PINKY_FINGER_TIP]
        self.__keys = []
```

Рисунок 3.10 – Класс operation

И функция compare, которая показана в [рисунке 3.11](#), сравнивает полученный жест с настройкой.

```
def compare(x: list, y: list, z: list, hand: str, o: Operation) -> bool:
    if hand != o.get_hand():
        return False
    ox = o.get_x()
    oy = o.get_y()
    oz = o.get_z()
    for i in o.get_check_x():
        if ox[i] != x[i]:
            return False
    for i in o.get_check_y():
        if oy[i] != y[i]:
            return False
    for i in o.get_check_z():
        if oz[i] != z[i]:
            return False
    return True
```

Рисунок 3.11 – Функция compare

3.2.4 Управление компьютером

Управление осуществляется путём имитации ввода с клавиатуры с помощью pyautogui. Каждая операция привязана к жесту, и в рабочем режиме связь между жестами и операциями является один к одному.

Каждый раз, когда жест распознается, настройки сравниваются, чтобы определить, привязан ли жест к операции. Если жест привязан к какой-то операции, то выполняется операция. После выполнения операции вся сохранённая информация о местонахождении точек будет удалена, чтобы начать следующее распознавание. Также после каждой операции будет небольшой период охлаждения.

По умолчанию ПО имеет два рабочего режима в режиме управления: режим Windows и режим презентации. Они могут управлять окнами компьютера и различным программным обеспечением для

слайд-шоу соответственно. Существует также режим customization, который полностью настраивается пользователем.

Важно отметить, что эксплуатационная целостность должна быть обеспечена по соображениям безопасности, чтобы предотвратить непредвиденные последствия. Клавишу на клавиатуре необходимо поднять после нажатия независимо от того, завершена ли операция.

Во-вторых, когда руки не полностью появляются на изображении, результаты распознавания менее надёжны, поэтому в том случае ПО не управляет компьютером. И при записи жестов, когда рука не полностью наблюдается, запись не будет завершена. ПО кидает исключение в таком случае.

3.2.5 Пользовательские настройки

При обсуждении технологического стека была выбрана SQLite для сохранения пользовательских настроек в связи с тем, что она обладает небольшим потреблением ресурсов, простотой использования и высокой переносимостью.

Каждый раз при запуске ПО, если БД не существует, то создаётся БД. Если база данных существует, ПО загружает пользовательские настройки.

Когда пользователь вносит пользовательские настройки, настройки в БД обновляются так же, как и настройки в программе, загруженные в память. Чтобы предотвратить несогласованность данных между памятью и базой данных, настройки сначала сохраняются в БД, а затем обновляются настройки, загруженные в память. Чтобы сократить

чтение и запись в БД, существуют специальные методы обновления настроек в памяти.

Когда нужно удалить существующие настройки, только нужно удалить только соответствующий настройки в БД. Если нужно скопировать пользовательские настройки, только нужно скопировать только файл db.

Под процессом настройки понимаются запись жест и привязка к операции. При записи в начале ПО наблюдает руку. Когда рука не двигается за определённое время, запись может начинаться. В том случае пользователь видит красный квадрат как показано в рисунке 3.12, который значит, что запись может начинаться, но ещё не начался. Потом рука может двигаться. В процессе движения ПО записывает движение и в том случае пользователь видит зелёный квадрат как показано в рисунке 3.13, который значит, что запись идёт. После окончания жеста рука должна не двигаться определённый период. ПО наблюдает это и понимает, что запись должна заканчиваться. После завершения записи, зелёный квадрат исчезает.

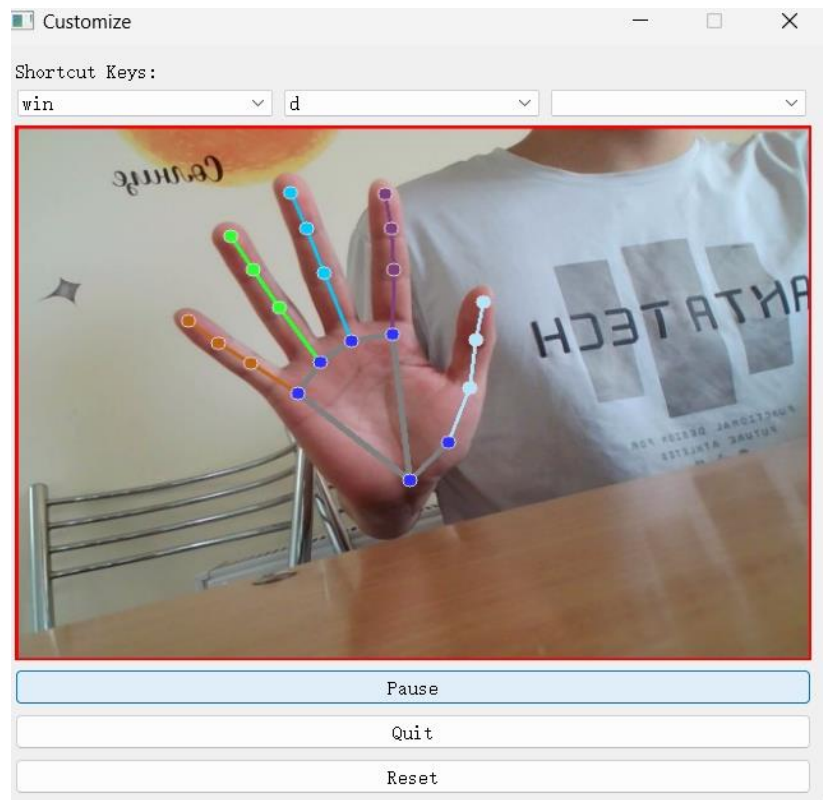


Рисунок 3.12 – Красный квадрат

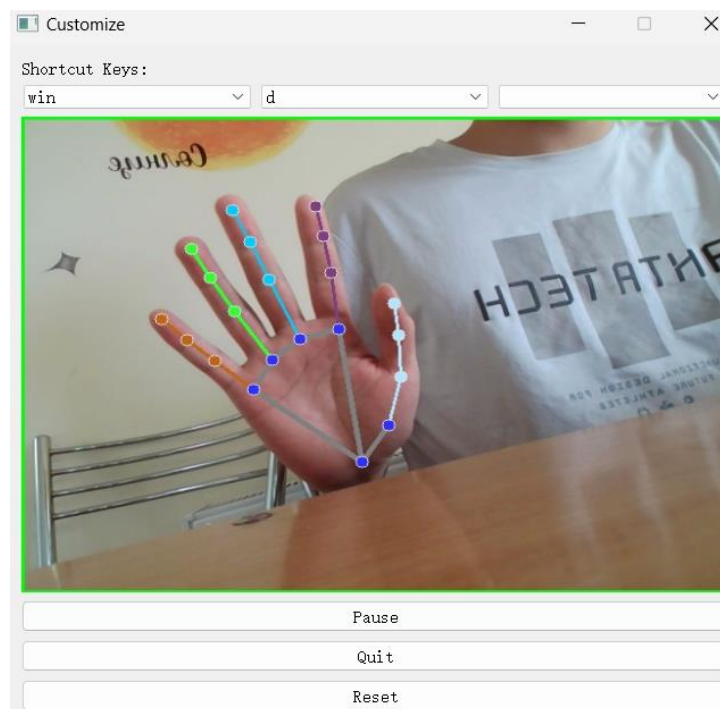


Рисунок 3.13 – Зелёный квадрат

3.2.6 Создание пользовательского интерфейса

В ПО есть три страницы. Главная страница в рисунке 3.14 распознает жест и управляет компьютером. Страница настройки в рисунке 3.15 записывает жест и привязывает его к определённой операции. Страница пользовательской настройки в рисунке 3.16 записывает жест и привязывает его к операции, которая настроена пользователем.

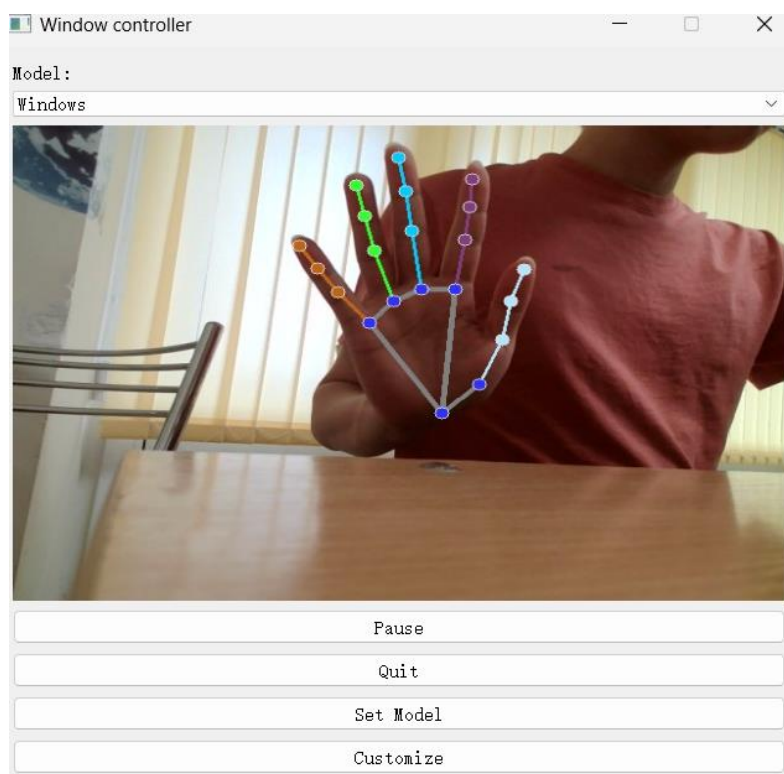


Рисунок 3.14 – Главная страница

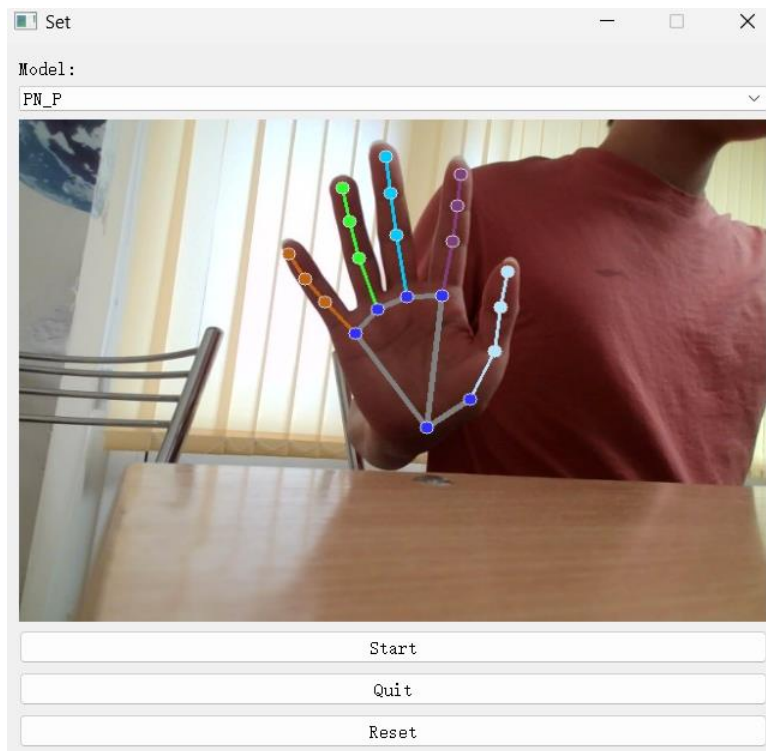


Рисунок 3.15 - Страница настройки режима работы

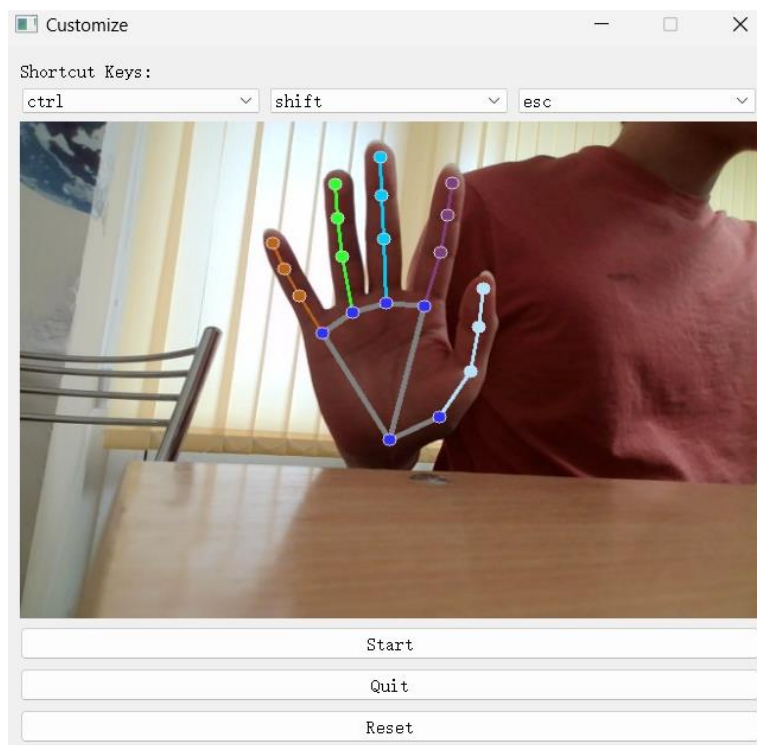


Рисунок 3.16 - Страница пользовательской настройки

3.2.7 Повышение производительности

При первоначальной доработке ПО наблюдаются явные задержки. Необходимо оптимизировать производительность ПО. В конце концов, есть два метода оптимизации производительности:

- выбор ключевых точек, чтобы уменьшить объем данных, которые необходимо обработать,
- создание потоков для обработки данных.

3.2.7.1 Выбор ключевых точек

При распознавании жестов, поскольку точка 0 фиксирована, перемещения некоторых точек очень слабы, и результатом распознавания почти всегда является отсутствие движения. Представителями этого типа точек являются точка 2, точка 5 и точка 9. Некоторые точки движутся в том же направлении, что и другая точка, но в то же время их движения не так сильны, как у другой точки. Представителем этого типа точек является точка 7. Движение и направление точки 7 соответствуют точке 8, но движения слабее, чем у точки 8. Мы можем выбирать в качестве ключевых точек только точки, не относящиеся к этим двум типам, и лишь определять направление движения ключевых точек. Это позволяет уменьшать объем данных, которые необходимо обработать. В то же время мы также должны помнить о роли точки 0 и точки 1 в определении направлении движений, поэтому они также являются ключевыми точками, которые необходимо выбрать.

В итоге в качестве ключевых точек были выбраны следующие семь точек: точка 0, точка 1, точка 4, точка 8, точка 12, точка 16 и точка 20. В результате нам остаётся обработать только одну треть данных.

3.2.7.2 Многопоточность

В исходной версии ПО распознавание изображений и сбор данных были реализованы в графическом интерфейсе. В `pygame` есть таймер, который можно использовать для периодического выполнения функции. В этом случае процесс управления ПО показан в рисунке 3.17 и процесс пользовательских настроек в рисунке 3.18.

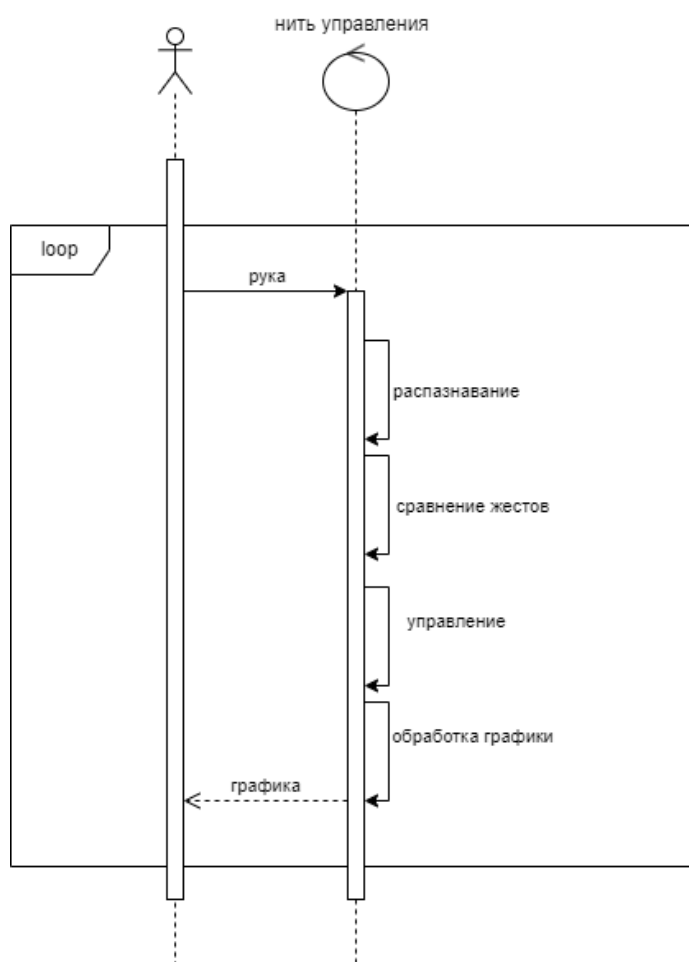


Рисунок 3.17 - Диаграмма последовательности процесса управления

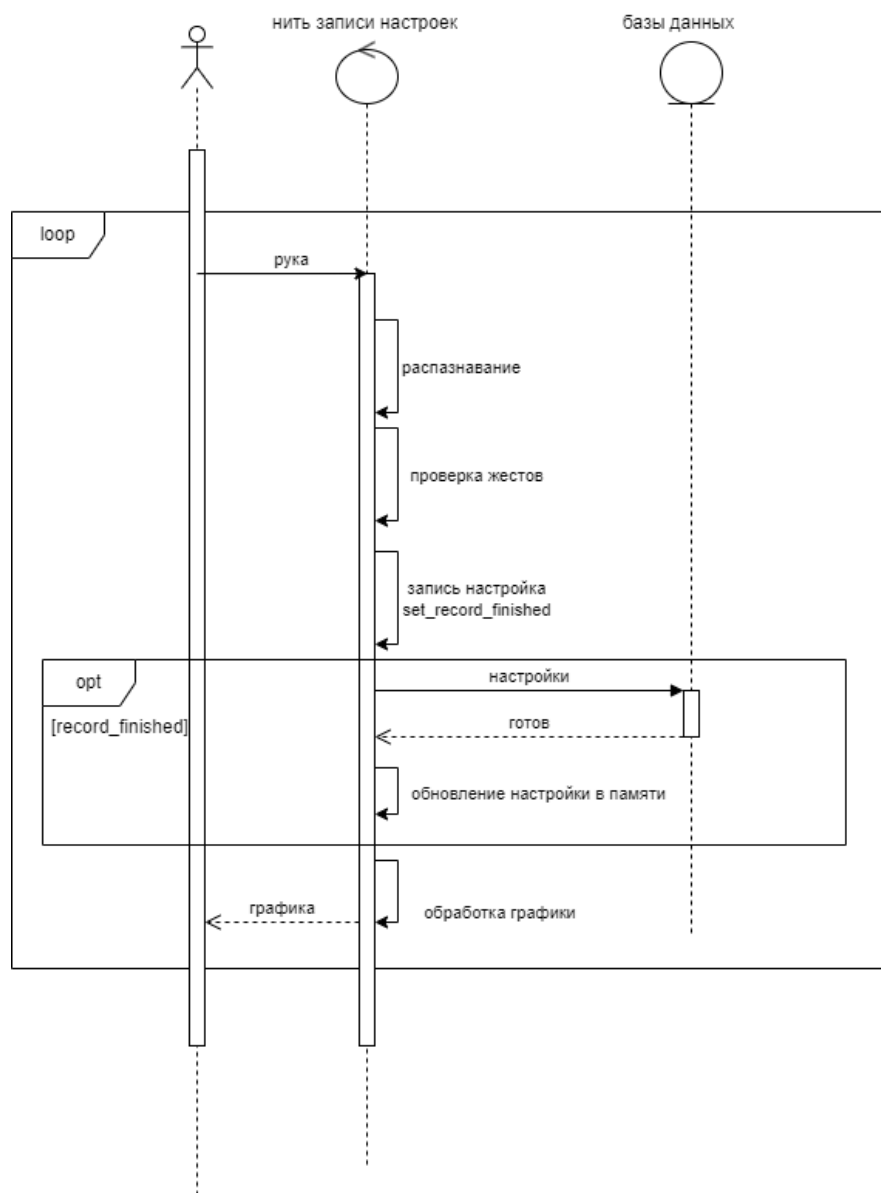


Рисунок 3.18 – Диаграмма последовательности процесса настройки

Цикл в этом процессе выполняется регулярно. В этом процессе есть проблема. Установленный интервал времени выполнения запланированной задачи фактически определяет частоту кадров увиденного пользователем видео и частоту распознавания. И у распознавания есть ещё другой параметр для установки частоты

распознавания. Это приводит к двум временным интервалам для распознавания. Следовательно, необходимо делать разъединение и улучшать производительность программного обеспечения вовремя разъединения.

QTheard предоставляется в PyQt5, чтобы помочь реализовать многопоточное программирование. В конце пользовательский интерфейс отвечает только за отображение изображений пользователям, а все остальные функции передаются QTheard. Кроме того, создание и уничтожение QTheard приведёт к накладным расходам. Поэтому будет создан только один QTheard, когда ПО начнёт отвечать за распознавание, обработку данных и взаимодействие с базой данных. Когда пользователь переключается между разными страницами ПО, QTheard переключается для выполнения разных задач. Вместо создания нового потока для выполнения задачи, когда пользователь переходит на другую страницу, а затем уничтожения потока при выходе пользователя. Хотя если каждая страница будет снабжена фоновым потоком, сложность программирования снизится. В ходе практики было замечено, что потребление ресурсов станет высоким, а переходы между разными страницами ПО станут крайне медленными. Без сомнений, что это неприемлемо как с точки зрения производительности ПО, так и с точки зрения удобства.

3.3 Вывод

Проектирование было выполнено со соблюдением функциональных и нефункциональных требований. В ходе

проектирования ПО определены варианты использования ПО. Была спроектирована БД, где сохраняются настройки пользователя, и в результате была создана даталогическая модель БД. Была спроектированы процессы работы в разных режимах и в результате были получены диаграммы последовательности.

ПО реализовано. При запуске ПО будет загружена пользовательская настройка. OpenCV применяется для получения изображений и обработки изображений. В обработке формат изображений преобразован с BGR в RGB и изображение зеркально отражено и перевёрнуто. После обработки изображения применяются для распознавания руки. При распознавании получают местонахождение всех опорных точек кистей руки. Но только местонахождение выбранных ключевых точек обрабатывается. После распознавания смещения ключевых точек рассчитаны. Смещения выбранных ключевых точек в определённый период собираются. Направления движений ключевых точек определяются по составленной нечёткой логике на основе смещений. Набор направления движений ключевых точек является жестом. В режиме управления, при сравнении полученного жеста с загруженной настройкой с БД ПО будет выполняет соответствующую операцию. В режиме настройки после проверки законность жест будет сохраняться в БД, привязываясь к выбранной пользователем операции.

4 ТЕСТИРОВАНИЕ

Для тестирования были проведены опрос у студентов и модульное тестирование для ключевых функций. Для модульного тестирования было использована unittest.

4.1 Автоматизированное модульное тестирование

Написаны автоматизированного тестирования для функций:

- 1) для сохранения настройки;
- 2) для нахождения операции по жестам;
- 3) для загрузки настройки;

Не только протестировано при правильном вводе, но и при неожиданном вводе, при котором ПО кидает исключение.

4.2 Тестирование в виде опроса у студентов

Для проведения тестирования было проведён опрос у студентов. Есть две сценарии для тестирования:

- 1) Студент управляет приложениями простой операцией.
- 2) Студент управляет приложениями сложной операцией.

Сложные операции — это операции, требующие использования мыши и нажатия клавиш клавиатуры, которые расположенные далеко друг от друга.

После тестирования студент будет отвечать на следующие вопросы:

- 1) Считаете ли вы что ПО упрощает взаимодействие с приложениями?
- 2) В какой ситуаций вы считаете, что ПО может упрощает управление или помочь? Задайте только один ответ.

3) Есть ли у вас совет для улучшения ПО? Например, с точки зрения пользовательского интерфейса и процесса работы ПО.

Первый и второй вопросы считаются обязательными. Студенты могут не отвечать на третий вопрос.

В итоге тестирование было тестировано 17 студентов. Ответ на первый вопрос показан в рисунке 4.1. Ответ на второй вопрос показан в рисунке 4.2.

Были собраны следующие советы:

- 1) Надо присвоить раскрывающемуся меню метку, чтобы пользователи могли понять, что раскрывающиеся списки делают.
- 2) Надо добавить инструкции по использованию. Лучше добавить инструкции по использованию для каждой страницы.
- 3) Будет лучше если можем одновременно использовать две руки.
- 4) Ещё надо улучшать распознавание жестов.

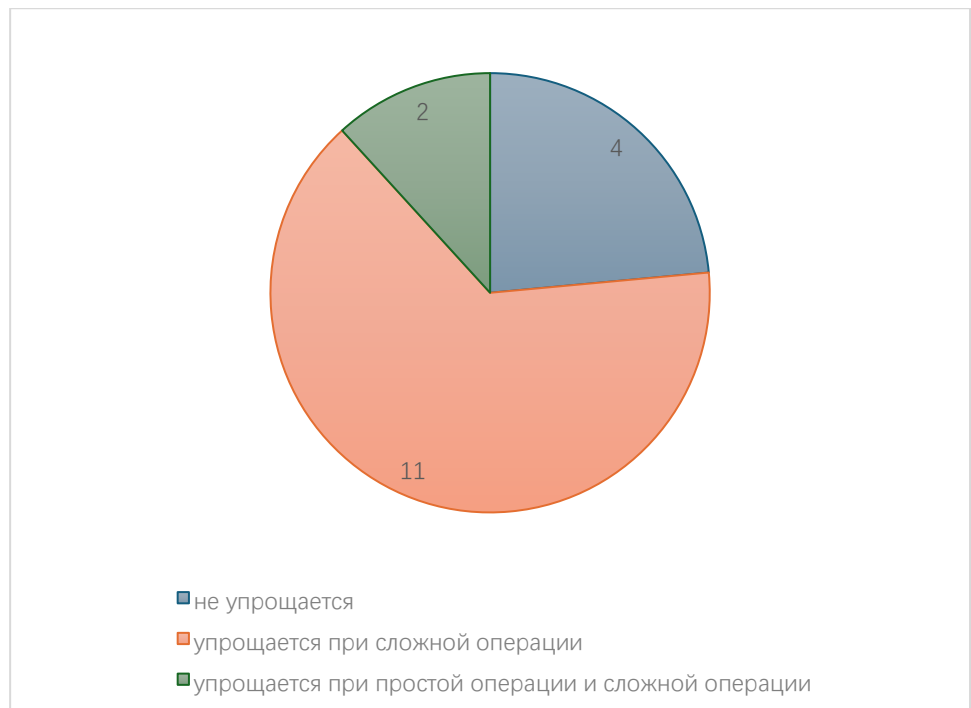


Рисунок 4.1 – Ответы на вопрос об упрощении



Рисунок 4.2 – Ответы на вопрос о применении ПО

Ответы от студентов различны. Объединены похожие ответы в одну категорию.

4.3 Вывод

Были проведены автоматизированное модульное тестирование и тестирование в виде опроса у студентов.

Автоматизированное модульное тестирование утверждает, что наше ПО действует по нашим требованиям. Ключевые функции были протестированы.

Тестирование в виде опроса у студентов утверждает, что цель достигнута. И тоже утверждает, что ПО может нормально выполнять все процессы. В результате получен вывод, что упрощает взаимодействие в ситуации, когда операция через клавиатуру и мыши сложна. И в ходе выполнения опроса получили отзыв что большинство студентов по-прежнему предпочитают взаимодействие с помощью клавиатуры и мыши. Возможно, это связано с тем, что они лучше знакомы с взаимодействием клавиатуры и мыши. Однако большинство студентов согласны с тем, что в некоторых ситуациях взаимодействие при помощи жестов может быть полезным или упростить взаимодействие.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы было реализовано ПО, которое обеспечивает для управление компьютером при помощи жестов. Получили следующие результаты:

- 1) Была изучена актуальность темы. Тенденция развития метода взаимодействия с компьютером была проанализирована. И в конце получен вывод, что управление компьютером при помощи жестов может быть дополнением к существующим методам взаимодействия с компьютером.
- 2) Были проанализированы аналоги нашего ПО. При сравнении с аналогами было утверждено, что преимущество ПО обеспечения в том, что это чисто программное решение.
- 3) Нечёткая логика была выбрана для рассуждения направлений движений опорных точек кистей.
- 4) Был проанализирован технологический стек, который применяется для реализации ПО. OpenCV была выбрана для получения изображений. Mediapipe была выбрана для распознавания рук. PyautoGui был выбрана для управления компьютером. SQLite была выбрана для сохранения настроек. PyQt 5 была выбрана для создания пользовательского интерфейса. Skfuzzy была выбрана для составления нечёткой логики.
- 5) Функциональные требования и нефункциональные требования ПО были оформлены.

- 6) Была спроектирована структура БД, где будет сохраняться пользовательские настройки. В результате получена даталогическая модель БД.
- 7) Были проектировали процессы работы режимов распознавания. В результате получены две диаграммы последовательности.
- 8) Было реализовано ПО для управления компьютером при помощи жестов.
- 9) Было тестировано ПО. В результате получен вывод, что упрощает взаимодействие в ситуации, когда операция через клавиатуру и мыши сложна.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- 1) Документация google mediapipe [Электронный ресурс] – URL: <https://developers.google.com/mediapipe/api/solutions> (дата обращения: 10.05.2024)
- 2) Документация pyautogui [Электронный ресурс] – URL: <https://pyautogui.readthedocs.io/en/latest> (дата обращения: 10.05.2024)
- 3) Документация skfuzzy [Электронный ресурс] – URL: <https://pythonhosted.org/scikit-fuzzy> (дата обращения: 10.05.2024)
- 4) Документация PyQt5 [Электронный ресурс] – URL: <https://www.riverbankcomputing.com/static/Docs/PyQt5> (дата обращения: 10.05.2024)
- 5) Документация open-cv [Электронный ресурс] – URL: https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html (дата обращения: 10.05.2024)
- 6) The Future of Hand Gestures as Digital Controls [Электронный ресурс] – URL: <https://www.linkedin.com/pulse/future-hand-gestures-digital-controls-patrick-mutabazi> (дата обращения: 10.05.2024)
- 7) Различные сценарии использования mediapipe - URL: <https://mediapipe.readthedocs.io/en/latest/solutions/hands.html> (дата обращения: 10.05.2024)
- 8) Формирование базового словаря жестов для естественного компьютерного бесконтактного интерфейса [Электронный ресурс] -URL: <https://cyberleninka.ru/article/n/formirovanie-bazovogo-slovary-a-zhestov->

- dlya-estestvennogo-kompyuternogo-beskontaktnogo-interfeysa/viewer (дата обращение: 10.05.2024)
- 9) Fuzzy Control Systems: The Tipping Problem [Электронный ресурс] -URL: https://scikit-fuzzy.readthedocs.io/en/latest/auto_examples/plot_tipping_problem_newapi.html (дата обращение: 10.05.2024)
- 10) An Introduction to Fuzzy Logic Applications in Intelligent Systems - Ronald R. Yager, Lotfi A. Zadeh, 1992 г
- 11) Human-Computer Interaction. Multimodal and Natural Interaction - Masaaki Kurosu, 2020 г
- 12) The Potential of Gesture-Based Interaction [Электронный ресурс] -URL: https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2733261/The_Potential_of_Gesture-based_Interaction_HCI2020.pdf?sequence=1&isAllowed=y (дата обращение: 10.05.2024)
- 13) Hand tracking for clinical applications: validation of the Google MediaPipe Hand (GMH) and the depth-enhanced GMH-D frameworks [Электронный ресурс] -URL: https://www.researchgate.net/publication/372858609_Hand_tracking_for_clinical_applications_validation_of_the_Google_MediaPipe_Hand_GMH_and_the_depth-enhanced_GMH-D_frameworks (дата обращение: 10.05.2024)
- 14) Gesture Control Market Snapshot (2023 to 2033) [Электронный ресурс] - URL: <https://www.futuremarketinsights.com/reports/gesture-control-market> (дата обращение: 10.05.2024)