

|  |  |
| --- | --- |
|  | Допустить к защите  Заместитель директора  \_\_\_\_\_\_\_\_\_\_\_\_\_/Т.П. Антонова/  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2024г. |

**ДИПЛОМНЫЙ ПРОЕКТ**

Специальность 09.02.07 Информационные системы и программирование

Тема дипломной работы (проекта): «Разработка дистанционной виртуальной клавиатуры с использованием Bluetooth»

Выполнил:

Студент(ка) 3 курса группы ИСП-3А-21

Егоров Вячеслав Сергеевич

(Ф.И.О.)

(подпись)

Руководитель дипломного проекта:

Преподаватель (должность)

Иванников А.А. (Ф.И.О.)

(подпись)

« » 2024года

Москва 2024 г.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| СОДЕРЖАНИЕ  ВВЕДЕНИЕ   1. Введение 2. Расчётно-пояснительная часть   2.1 Общая часть   1. . Технологическая часть   4. Экономическая часть  4.1 Расчет заработной платы исполнителя работ по созданию программного продукта  4.2 Расчет страховой выплаты  4.3 Расчет расходов на содержание и эксплуатацию ПЭВМ  4.4 Расчёт себестоимости программного продукта  4.5 Расчет цены программного продукта  5. Охрана труда  ЗАКЛЮЧЕНИЕ  СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ  ПРИЛОЖЕНИЕ А  ПРИЛОЖЕНИЕ Б  ПРИЛОЖЕНИЕ В | | | | | | | | | | |
|  |  |  |  |  | ДП 09.02.07.ИСП.6.2024ПЗ | | | | | |
|  |  |  |  |  |
| *Изм.* | *Лист* | *№ док.* | *Подпись* | *Дата* |
| *Разраб.* | | *Егоров В.С.* |  |  | *Пояснительная записка* | *Лит.* | | | *Лист* | *Листов* |
| *Пров.* | | *Иванников А.А.* |  |  |  | *Д* |  |  | *52* |
|  | |  |  |  | *ГБПОУ МИК, гр. ИСП-3А-21* | | | | |
| *Н.контр.* | | *Иванников А.А.* |  |  |
| *Утв.* | | *Иванников А.А.* |  |  |

**ВВЕДЕНИЕ**

В настоящее время компьютеры играют важнейшую роль в жизни каждого человека, невозможно представить современное предприятие без компьютеров персональных, компьютеров мобильных или серверов, любой компьютер подразумевает ввод-вывод информации, вычисления, хранение данных и ни один аспект компьютера не может быть убран, в противном случае компьютер станет неработоспособным, таким образом, сформировался рынок компьютерной периферии и компьютерных комплектующих, весомую нишу рынка компьютерной периферии занимают устройства ввода-вывода и в современных реалиях устройства ввода-вывода стараются делать беспроводными, работающими на ультра высоких радиочастотах, самые популярные из технологий использующих такие частоты – IEEE 802.11 или как эту технологию чаще называют “WI-FI”, а также IEEE 802.15 “Bluetooth”. Вышеуказанные технологии беспроводной связи несколько меняются, улучшается дальность действия и увеличивается объем передаваемых данных за единицу времени, таким образом, предприятиями разрабатывается компьютерная периферия, которая подходит под новый стандарт, удешевляется производство, так, если предыдущее поколение поддерживало передачу графической информации в разрешении 1920 на 1080 пикселей с частотой 60 Герц и глубиной цвета в 8 бит, то нынешнее поколение может передавать графическую информацию в разрешении 3840 на 2160 пикселей с частотой 60 Герц и глубиной цвета в 8 бит, из-за удешевления производства и общего улучшения техники, растёт спрос на такую технику, имеется спрос и на дальнейшее улучшение стандартов беспроводной связи, поскольку с каждым последующим поколением стандарты беспроводной связи находят новую нишу применения, зачастую одно устройство может поддерживать несколько различных стандартов беспроводной связи, например мобильный компьютер, такой как ноутбук, планшет или смартфон имеет возможность параллельно использовать как технологию “Bluetooth” так и технологию “WI-FI”, таким образом появился рынок программных эмуляторов физической компьютерной периферии, так, например, пользователь смартфона имеет возможность скачать приложение эмулятор клавиатуры, или монитора, заместо покупки физического экземпляра периферии, например в случае выхода из строя вышеуказанных устройств, в случае необходимости использования компьютера.

Актуальность данной темы заключается в растущей потребности в удобных способах ввода текста на мобильных устройствах и компьютерах. В мире, где мобильные устройства становятся все более распространенными, разработка дистанционной виртуальной клавиатуры с использованием “Bluetooth” представляет собой решение, которое обеспечивает удобство при вводе информации и комфорт при эксплуатации. Такая клавиатура может использоваться для ввода команд на устройства пользователя. Разработка инновационных методов ввода и обработки текста с помощью технологии “Bluetooth” также имеет потенциал для применения в различных областях, включая медицину, образование, производство. Объектом исследования дипломного проекта является создание программного эмулятора физической клавиатуры. Предметом исследования дипломного проекта является изучение доступности технологий устройств в средах разработки для дальнейшего создания программного эмулятора клавиатуры. Целью исследования дипломного проекта является изучение и применение технологии “Bluetooth” в создании программного эмулятора дистанционной клавиатуры. Задачами исследования являются:

* Выбор аппаратной платформы для реализации клавиатуры;
* Разработка программного обеспечения для работы клавиатуры;
* Разработка алгоритма распознавания нажатий на виртуальные клавиши;
* Разработка интерфейса клавиатуры

**РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЧАСТЬ**

**Общая часть**

При изучении различных интернет-магазинов электроники, было выявлено, что на сегодняшний день на рынке беспроводных клавиатур преобладает технология “Bluetooth”, данные из крупного интернет-магазина “DNS” (см.Рис.0).

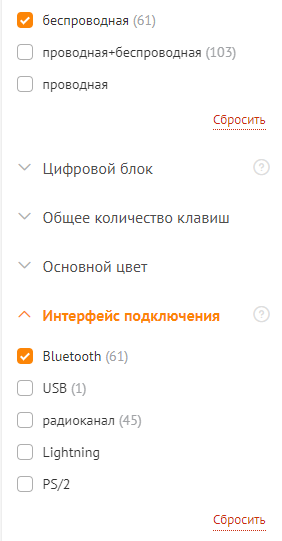


Рисунок 0 данные из интернет-магазина “DNS”

было принято решение разрабатывать эмулятор, использующий именно эту технологию, с точки зрения популярности и простоты подключения, упор был сделан именно на простоту подключения и общее удобство использования эмулятора клавиатуры.

Вышеуказанная технология развивается, разрабатывается, поддерживается с 1999 года, с версии технологии 1.0, данной технологией пользуются всевозможные электронные устройства от мобильных телефонов и ноутбуков до наушников, принтеров, клавиатур, мышей, игровых приставок и умных часов. Технология “Bluetooth” применяется во всевозможных устройствах, в том числе устройствах интернета вещей, данная технология работает, используя так называемую пико-сеть, радиус которой ограничен 50 метрами в случае использования технологии “Bluetooth” версии 5.2, на текущий момент данная версия является передовой. Радиус действия пико-сети может быть расширен благодаря использования подчинённого узла, представляющего собой устройство в пико-сети, функция которого – ретрансляция данных на серверный узел или вторичным подчиненным узлом пико-сети, такой узел создает пересечение сетей, а совокупность связанных пико-сетей образуют рассеянную сеть (см.Рис.1).

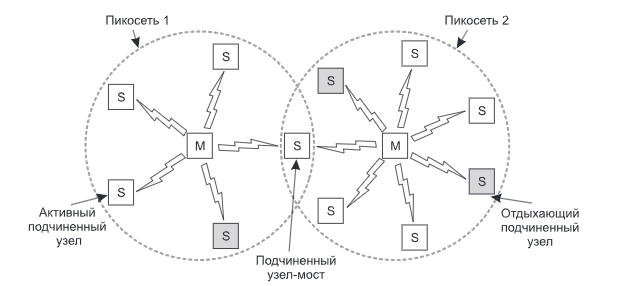


Рисунок 1 – Рассеянная сеть

На рисунке 1 присутствуют так называемые подчиненные узлы, они являются устройствами-пользователями пико-сети, например, устройствами-пользователями пико-сети могут быть беспроводные наушники, использующие технологию “Bluetooth”, в центре пико-сети, находится серверный узел, этим узлом может являться компьютер мобильный или стационарный, а также микроконтроллер, поддерживающий технологию “Bluetooth”, данная конфигурация сети представляет собой разновидность клиент-серверной архитектуры, суммарное количество узлов, находящихся в одной пико-сети может быть 255. Активный подчинённый узел является устройством спаренным, подключённым, например, беспроводная клавиатура, вводящая информацию. Отдыхающий узел является спаренным, но не подключённым. Следует обратить внимание, что подчинённый узел мост является узлом активным, обсуживающемся одновременно двумя серверными узлами, таким образом, теоретически, возможно использование беспроводной клавиатуры для обслуживания вплоть до 255 серверных узлов, данная возможность может позволить дублировать команды клавиатуры, посылаемые серверам, для их дальнейшей обработки, в таком случае, сервера выполняют одни и те-же команды, подобная технология может применятся в промышленности, медицине и в других отраслях.

Технология Bluetooth обладает технологиями защиты от перехвата информации, передаваемой по радиочастотным каналам, за защиту отвечает протокол канального уровня Logical Link Control and Adaptation Protocol (L2CAP), защита производится путём шифрования информации, данный способ напоминает защиту информации в сетях “WI-FI”.

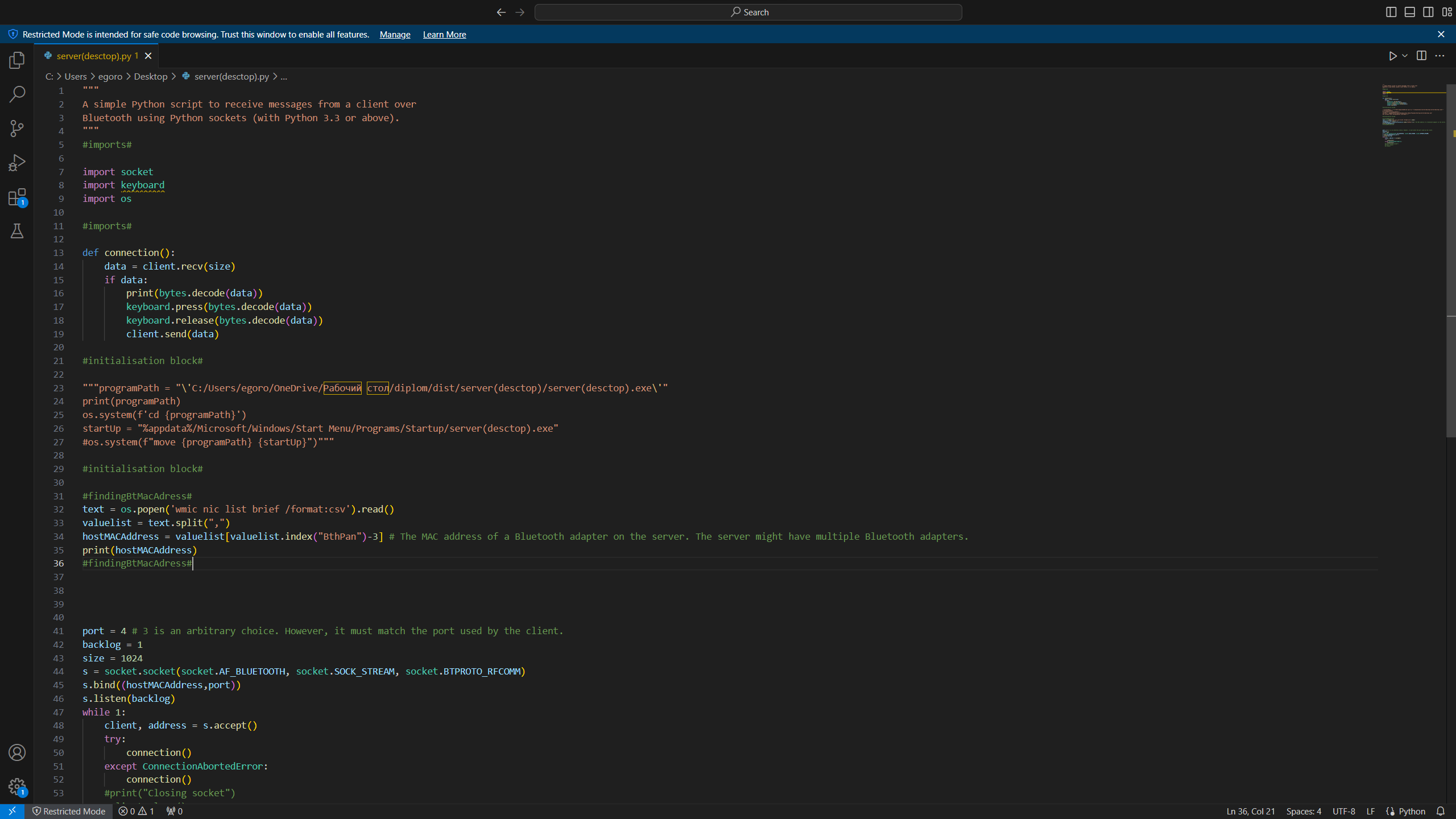
В качестве среды разработки программного продукта “дистанционная виртуальная клавиатура” было принято использовать “Microsoft Visual Studio Code”, (см.Рис.2). 

Рисунок 2 – среда разработки Microsoft Visual Studio Code

Данная среда разработки обладает удобным интерфейсом, гибкими возможностями сборки проекта, а также имеет набор утилит, благотворно способствующий ведению разработки, так, утилита “IntelliSence” помогает в поисках библиотек, функций и методов, необходимых для разработки, а также выделяет функциональные элементы кода различными цветами, так зелёным цветом обозначены библиотеки и классы, жёлтым обозначены функции и методы, голубым обозначены переменные, а розовым цветом обозначены операторы языка.

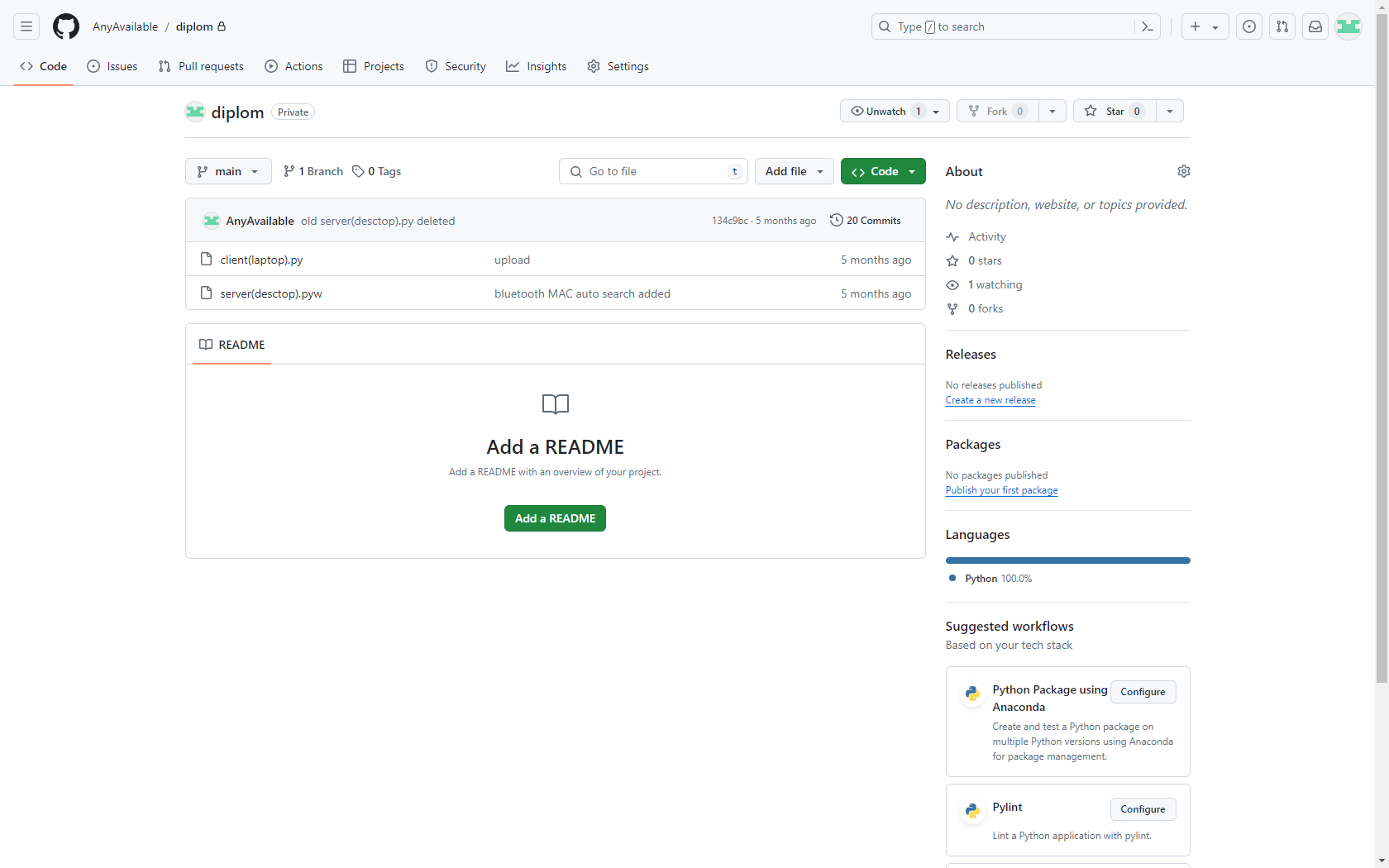
Учитывая важность и трудоёмкость работы было принято решение использовать Систему Контроля Версий (СКВ) “GIT HUB”, был создан репозиторий, (см.Рис.3) для хранения, учёта контроля версий и надёжности хранения кода и проекта в общем.

Рисунок 3 – Репозиторий проекта СКВ “GIT HUB”

Разработка и написание дипломного проекта велась на Операционной Системе (ОС) “Windows 11”, такое решение было принято в связи с массовостью, простотой эксплуатации, простотой установки программ, удобством файловой системы и обилием функций, в том числе с применением нейро-сетевых технологий.

В качестве сетевого навигатора был использован “Яндекс браузер”, преимуществами использования данного Программного Продукта, (ПП) являются:

* Красивый интерфейс
* Удобный интерфейс
* Продвинутая система поиска
* Поддержка обработки документов
* Поддержка всех форматов сайтов
* Встроенная нейросеть

Данный ПП был использован для поиска теоретических сведений о различных технологиях, применяемых в ПП «дистанционная виртуальная клавиатура с использованием Bluetooth»

Для создания макета интерфейса ПП «дистанционная виртуальная клавиатура с использованием Bluetooth» был использован ПП “Figma”(см.Рис.4)

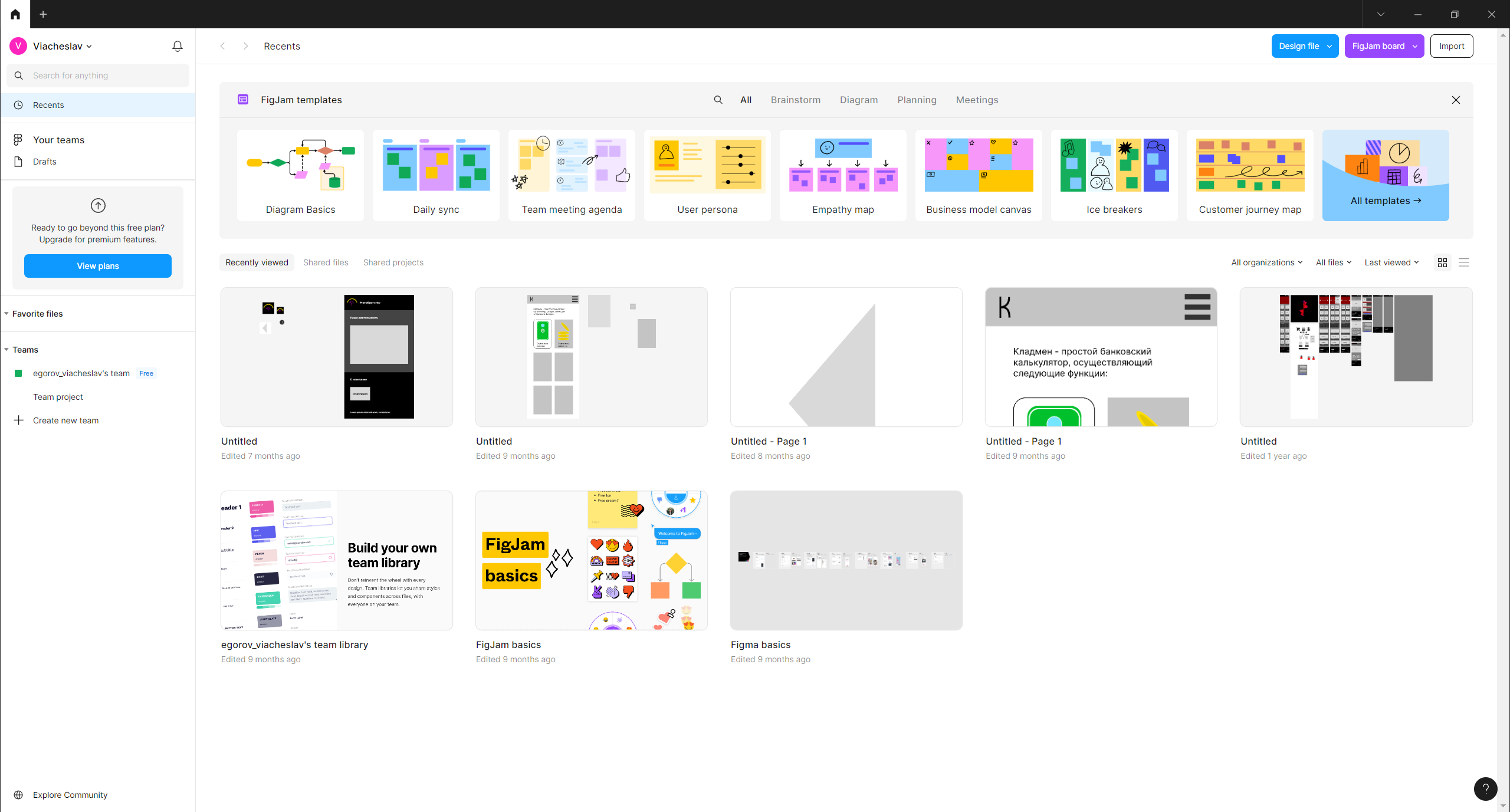


Рисунок 4 – ПП “Figma”

“Figma” является бесплатным инструментом для создания макета интерфейса, данный ПП полезен при разработке сайтов, мобильных приложений, а также программ с графическим пользовательским интерфейсом

Для отправки команд к ОС “Windows 11” была использована встроенная в ОС утилита, “Windows PowerShell”. (см.Рис.5)

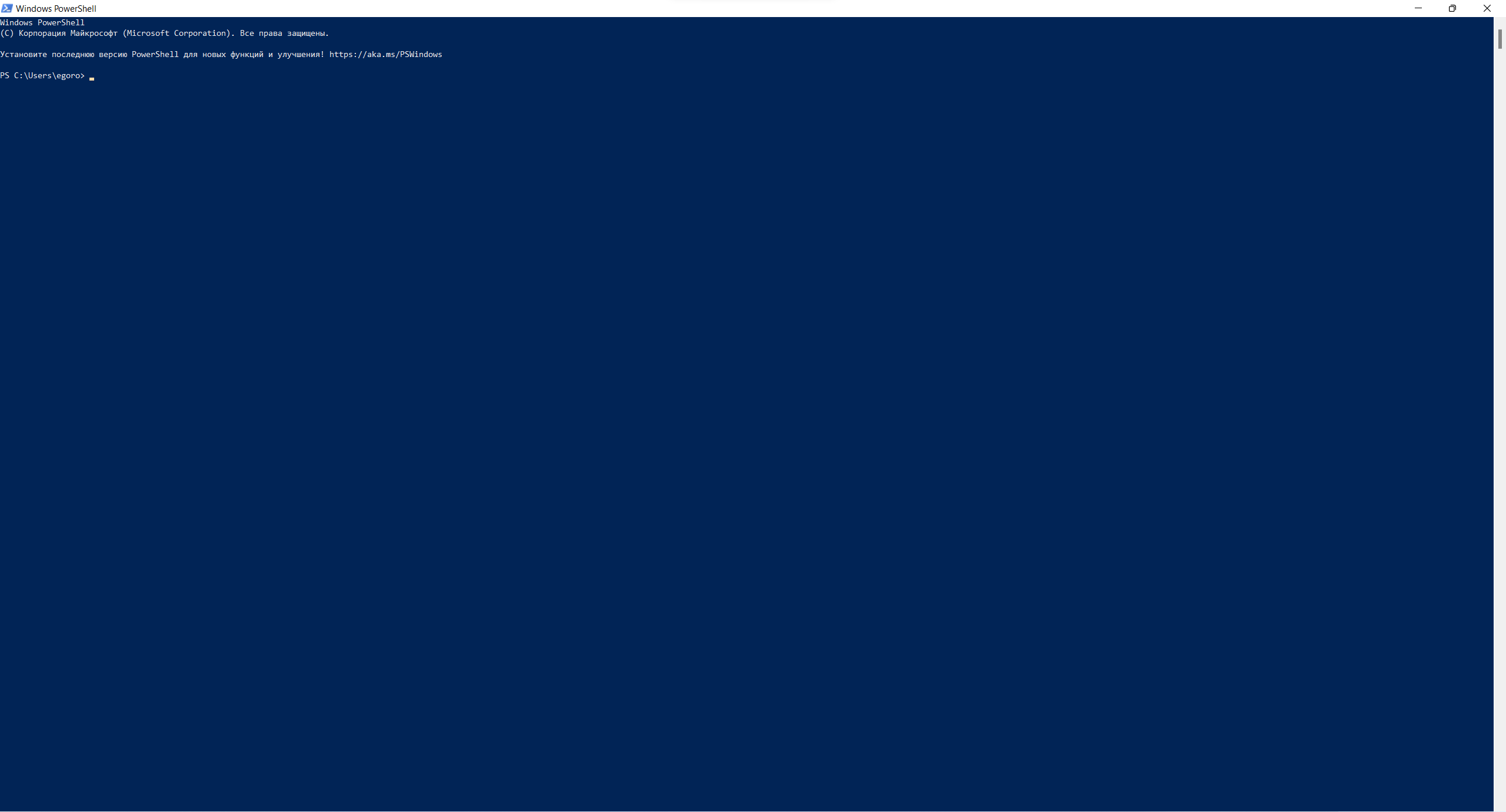


Рисунок 5 – Утилита “Windows PowerShell”

Использование данной утилиты было обосновано, необходимостью подбора и проверки команд, необходимых для автоматизации процесса инициализации ПП «дистанционная виртуальная клавиатура с использованием Bluetooth». Проверка команд обоснована сокращением времени тестирования отдельных элементов блока инициализации.

Для создания интеллектуальной карты ПП «дистанционная виртуальная клавиатура с использованием Bluetooth» был использован условно бесплатный ПП “Xmind”. (см. Рис.6)

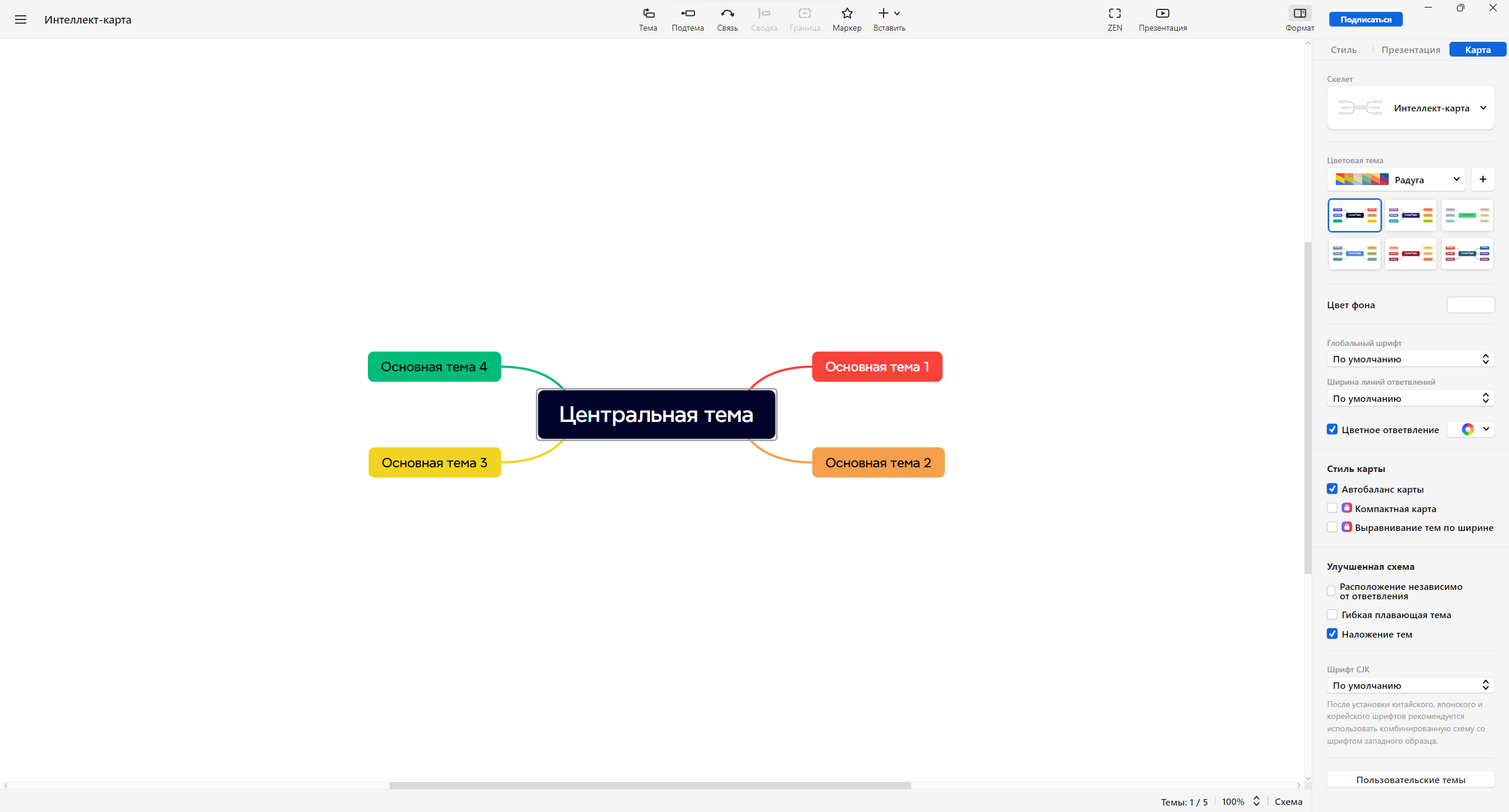


Рисунок 6 – ПП “Xmind”

В связи с общей сложностью проекта, в частности сложного блока автоматизации подразумевающего разную инициализацию для разных ОС, было принято решение использовать вышеуказанный ПП. Также использование подобного ПП для составления интеллектуальной карты упростит разработку, так-как все концептуальные решения будут видны разработчику, что также благотворно влияет на концептуальную целостность ПП «дистанционная виртуальная клавиатура с использованием Bluetooth».

После составления макета интерфейса в ПП “Figma”, было-бы необходимо перенести макет интерфейса в программную оболочку, для этих целей была выбрана условно – бесплатная программа – конструктор графического пользовательского интерфейса “Qt Designer”. (см. Рис.7)

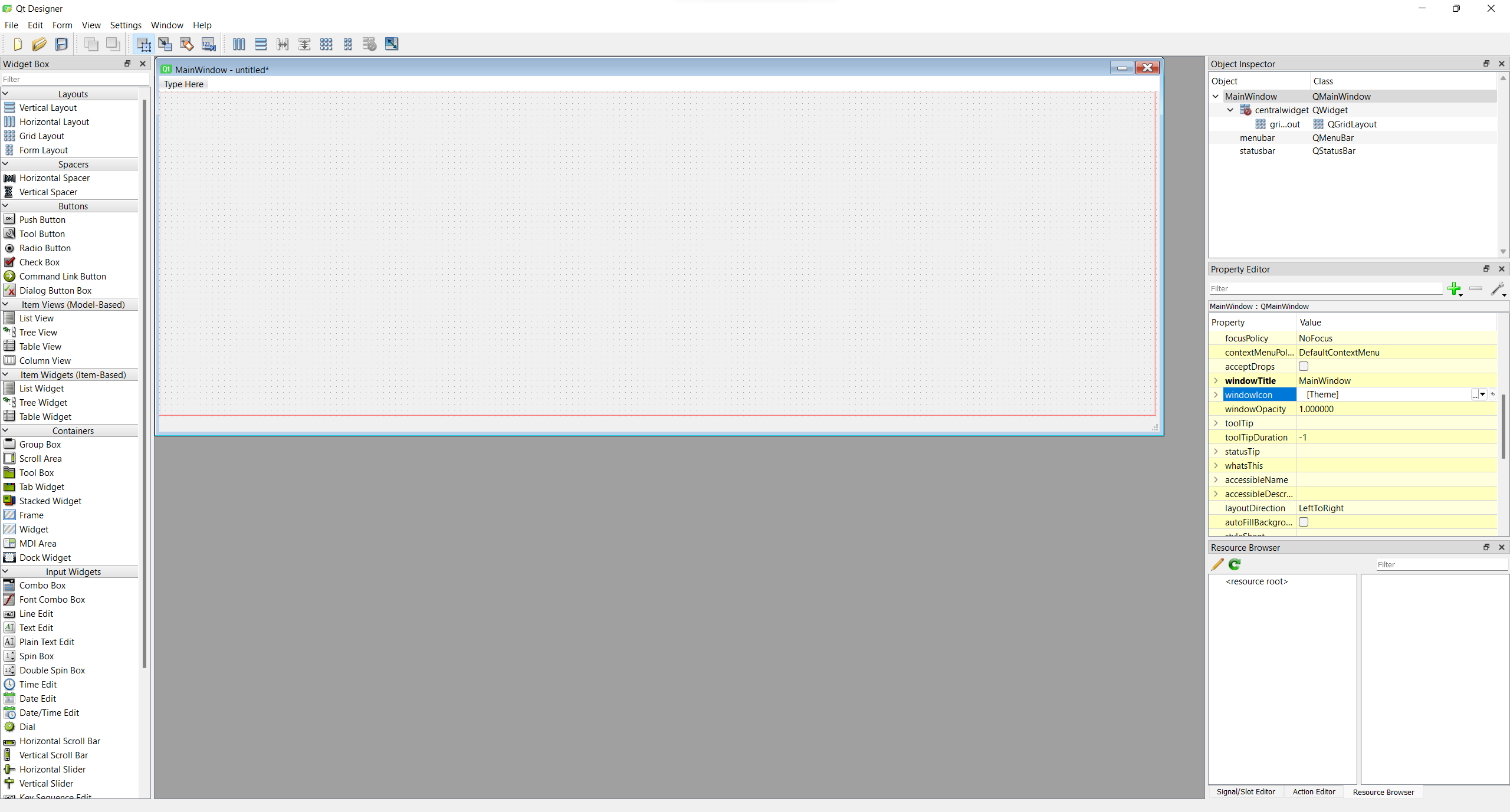


Рисунок 7 – программа – конструктор графического пользовательского интерфейса

В качестве Языка Программирования (ЯП) был выбран “Python” версии 3.10 в связи с его высоким уровнем разработки, таким образом, сокращается количество строк кода, при сохранении функциональности, данного ЯП более чем достаточно для выполнения поставленных перед дипломной работой целей, также стоит отметить, что часть методов, классов ЯП “Python” уже были разработаны программистами со всего мира, это также сокращает расходы на разработку ПП «дистанционная виртуальная клавиатура с использованием Bluetooth».

Учитывая стоимость аппаратной платформы на которой велась разработка было принято решение использовать виртуальную машину “Oracle VM VirtualBox”, таким образом, команды, используемые для проверки системных файлов и поиска необходимой для работы информации методов автоматизации инициализации проверялись бы на ошибки, это позволило бы избежать физических проблем с аппаратной системой на которой велась разработка, например, в случае необратимого повреждения операционной системы, в вышеуказанном ПП предусмотрен так называемый откат к снимку состояния, это обозначает, что в любой момент пользователь “Oracle VM VirtualBox” имеет возможность вернуться к состоянию ОС, в момент создания снимка. Также следует отметить, что виртуальные машины позволяют установить любую ОС, что также позволяет определить, будет ли ПП «дистанционная виртуальная клавиатура с использованием Bluetooth» работоспособен под управлением различными ОС такими как “Windows 11” и “Linux”. (см. Рис. 8)

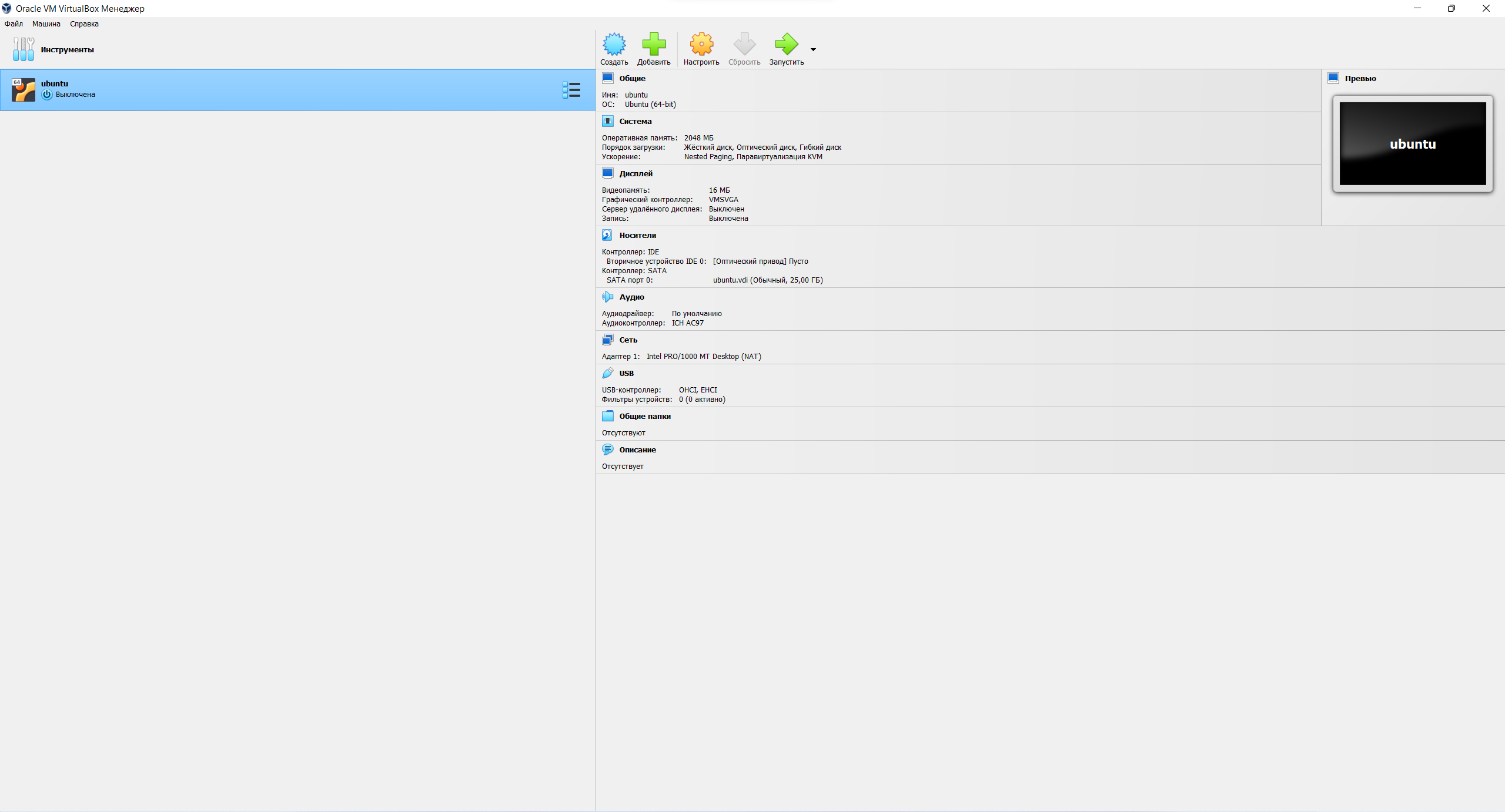


Рисунок 8 – Виртуальная машина “ Oracle VM VirtualBox”

В качестве методологии разработки ПО была выбрана итеративная методология разработки программного обеспечения. (см.Рис.9)

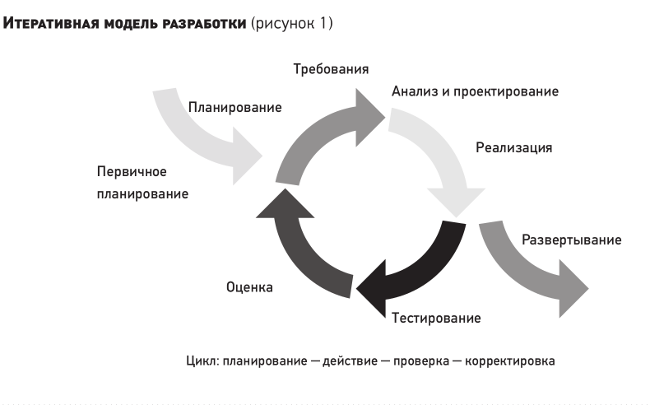


Рисунок 9 – итеративная методология разработки ПО

Выбор выше указанной методологии был обоснован тем, что методология достаточно гибкая и позволяет разному количеству разработчиков получать качественный продукт, а также методология содержит в себе такие важные аспекты разработки как тестирование и анализ, это позволяет гибко подстроится под новые требования разработки, все этапы разработки поделены на блоке, а самое главное трудозатраты намного меньше при использовании данной методологии, по наблюдению писателя и главного разработчика ПО компании “Construx Software” Стива Макконнелла (см. Рис. 10, 11)

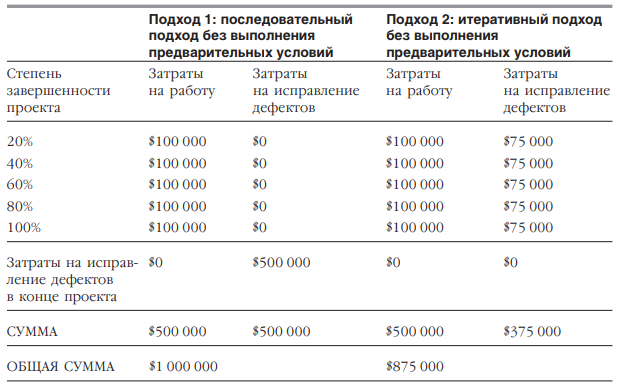


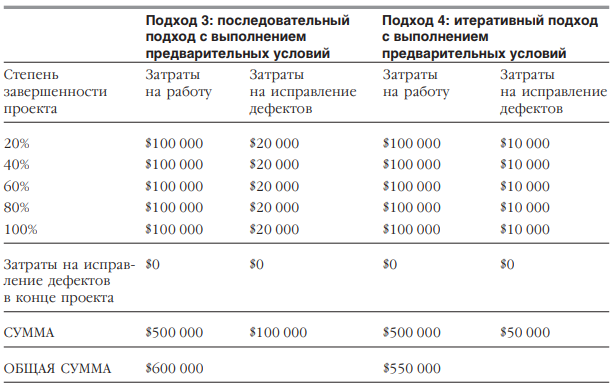
Рисунок *1*0 – трудозатраты при использовании различных методологий

Рисунок 11 – трудозатраты при использовании различных методологий

Как можно наблюдать выше, мы используем подход итеративный, также стоит сравнить общую стоимость затрат на исправление дефектов ПО, при использовании итеративной методологии достигается сокращение трудозатрат практически в два раза

В качестве разделения времени каждого блока цикла разработки было принято взять систему разделения Фредерика Брукса Младшего из его книги “мифический человеко-месяц”, эта система распределяет время этапов цикла следующим образом:

* 1/3 планирование
* 1/6 реализация
* 1/4 тестирование
* 1/4 оценка

При изучении рынка подобных эмуляторов, было выявлено что совокупность приложений – эмуляторов клавиатуры можно условно поделить на два вида, первый вид использует клиент и сервер написанные вручную, а второй вид использует клиент написанный вручную, а в качестве сервера используются существующие в ОС “Windows 11”интерфейсы взаимодействия с модулем “Bluetooth”, было принято начать разработку по первому виду программного обеспечения, это обоснованно тем, что требования к дипломному проекту, разработке ПП «дистанционная виртуальная клавиатура с использованием Bluetooth» подразумевает поддержку всех современных ОС.

Выбирая аппаратную платформу для разработки было принято решение использовать компьютер с процессорной архитектурой “x86\_64”, с поддержкой технологии “Bluetooth” версии 5.2.

**Версии технологии “Bluetooth”**

**Bluetooth 1.0**

Устройства версий 1.0 (1998) и 1.0B имели плохую совместимость между продуктами различных производителей. В версиях 1.0 и 1.0B была обязательной передача адреса устройства (BD\_ADDR) на этапе установления связи, что делало невозможной реализацию анонимности соединения на протокольном уровне и было основным недостатком данной спецификации.

**Bluetooth 1.1**

В Bluetooth 1.1 было исправлено множество ошибок, найденных в версии 1.0B, добавлена поддержка для нешифрованных каналов, индикация уровня мощности принимаемого сигнала (RSSI).

**Bluetooth 1.2**

Главные улучшения:

Быстрое подключение и обнаружение.

Адаптивная перестройка частоты с расширенным спектром (AFH), которая повышает стойкость к радиопомехам.

Более высокие, чем в версии 1.1, скорости передачи данных, практически до 1 Мбит/с.

Расширенные синхронные подключения (eSCO), которые улучшают качество передачи голоса в аудиопотоке, позволяя повторно передавать повреждённые пакеты, и при необходимости могут увеличить задержку аудио для лучшей поддержки параллельной передачи данных.

В Host Controller Interface (HCI) добавлена поддержка трёхпроводного интерфейса UART.

Утверждён как стандарт IEEE Standard 802.15.1-2005.

Введены режимы управления потоком данных (Flow Control) и повторной передачи (Retransmission Modes) для L2CAP.

**Bluetooth 2.0 + EDR**

Bluetooth версии 2.0 был выпущен 10 ноября 2004 г. Имеет обратную совместимость с предыдущими версиями 1.x. Основным нововведением стала поддержка Enhanced Data Rate (EDR) для ускорения передачи данных. Номинальная скорость EDR — около 3 Мбит/с, однако на практике это позволило повысить скорость передачи данных только до 2,1 Мбит/с. Дополнительная производительность достигается с помощью различных радиотехнологий передачи данных.

Стандартная (базовая) скорость передачи данных использует GFSK-модуляцию радиосигнала при скорости передачи 1 Мбит/с. EDR использует сочетание модуляций GFSK и PSK с двумя вариантами, π/4-DQPSK и 8DPSK. Они имеют бо́льшие скорости передачи данных по воздуху — 2 и 3 Мбит/с соответственно.

Bluetooth SIG издала спецификацию как «Технология Bluetooth 2.0 + EDR», которая подразумевает, что EDR является дополнительной функцией. Кроме EDR есть и другие незначительные усовершенствования к спецификации 2.0, и продукты могут соответствовать «Технологии Bluetooth 2.0», не поддерживая более высокую скорость передачи данных. По крайней мере одно коммерческое устройство— HTC TyTN Pocket PC — использует «Bluetooth 2.0 без EDR» в своих технических спецификациях.

Согласно спецификации 2.0 + EDR, у EDR имеются следующие преимущества:

В некоторых случаях увеличение скорости передачи в три раза (2,1 Мбит/с).

Уменьшение сложности нескольких одновременных подключений из-за дополнительной полосы пропускания.

Снижение потребления энергии благодаря уменьшению нагрузки.

**Bluetooth 2.1**

2007 год. Добавлена технология расширенного запроса характеристик устройства (для дополнительной фильтрации списка при сопряжении), энергосберегающая технология Sniff Subrating, которая позволяет увеличить продолжительность работы устройства от одного заряда аккумулятора в 3—10 раз. Кроме того, обновлённая спецификация существенно упрощает и ускоряет установление связи между двумя устройствами, позволяет производить обновление ключа шифрования без разрыва соединения, а также делает указанные соединения более защищёнными благодаря использованию технологии Near Field Communication.

**Bluetooth 2.1 + EDR**

В августе 2008 года Bluetooth SIG представила версию 2.1 + EDR. Новая редакция Bluetooth снижает потребление энергии в пять раз, повышает уровень защиты данных и облегчает распознавание и соединение Bluetooth-устройств благодаря уменьшению количества шагов, за которые оно выполняется.

**Bluetooth 3.0 + HS**

3.0 + HSбыла принята Bluetooth SIG 21 апреля 2009 года. Она поддерживает теоретическую скорость передачи данных до 24 Мбит/с. Её основной особенностью является добавление AMP (Alternate MAC/PHY), дополнение к 802.11 как высокоскоростное сообщение. Для AMP были предусмотрены две технологии: 802.11 и UWB, но UWB отсутствует в спецификации.

Модули с поддержкой новой спецификации соединяют в себе две радиосистемы: первая обеспечивает передачу данных в 3 Мбит/с (стандартная для Bluetooth 2.0) и имеет низкое энергопотребление; вторая совместима со стандартом 802.11 и обеспечивает возможность передачи данных со скоростью до 24 Мбит/с (сравнима со скоростью сетей Wi-Fi). Выбор радиосистемы для передачи данных зависит от размера передаваемого файла. Небольшие файлы передаются по медленному каналу, а большие — по высокоскоростному. Bluetooth 3.0 использует более общий стандарт 802.11 (без суффикса), то есть несовместим с такими спецификациями Wi-Fi, как 802.11b/g/n.

**Bluetooth 4.0**

30 июня 2010 года Bluetooth SIG утвердил спецификацию Bluetooth 4.0. Включает в себя протоколы:

Классический Bluetooth,

Высокоскоростной Bluetooth

Bluetooth с низким энергопотреблением.

Высокоскоростной Bluetooth основан на Wi-Fi, а классический Bluetooth состоит из протоколов предыдущих спецификаций Bluetooth.

Частоты работы системы Bluetooth (мощность не более 0,0025 Вт).

Полоса частот: 2 402 000 000 — 2 480 000 000 Гц (2,402 — 2,48 ГГц)

Протокол Bluetooth с низким энергопотреблением предназначен прежде всего для миниатюрных электронных датчиков (использующихся в спортивной обуви, тренажёрах, миниатюрных сенсорах, размещаемых на теле пациентов и т. д.). Низкое энергопотребление достигается за счёт использования особого алгоритма работы. Передатчик включается только на время отправки данных, что обеспечивает возможность работы от одной батарейки типа CR2032 в течение нескольких лет. Стандарт предоставляет скорость передачи данных 1 Мбит/с при размере пакета данных 8—27 байт. В новой версии два Bluetooth-устройства смогут устанавливать соединение менее чем за 5 мс и поддерживать его на расстоянии до 100 м. Для этого используется усовершенствованная коррекция ошибок, а необходимый уровень безопасности обеспечивает 128-битное AES-шифрование.

Датчики температуры, давления, влажности, скорости передвижения и т. д. на базе этого стандарта могут передавать информацию на различные устройства контроля: мобильные телефоны, КПК, ПК и т. п.

Первый чип с поддержкой Bluetooth 3.0 и Bluetooth 4.0 был выпущен компанией ST-Ericsson в конце 2009 года.

**Bluetooth 4.1**

В конце 2013 года Bluetooth Special Interest Group (SIG) представила спецификацию Bluetooth 4.1. Одно из улучшений, реализованных в спецификации Bluetooth 4.1, касается совместной работы Bluetooth и мобильной связи четвёртого поколения LTE. Стандарт предусматривает защиту от взаимных помех путём автоматического координирования передачи пакетов данных.

**Bluetooth 4.2**

3 декабря 2014 Bluetooth Special Interest Group (SIG) представила спецификацию Bluetooth 4.2. Основные улучшения — повышение конфиденциальности и увеличение скорости передачи данных.

**Bluetooth 5.0**

16 июня 2016 года Bluetooth Special Interest Group (SIG) представила спецификацию Bluetooth 5.0. Изменения коснулись в основном режима с низким потреблением и высокоскоростного режима. Радиус действия увеличен в четыре раза, скорость увеличена вдвое. Также версия Bluetooth 5.0 полностью совместима с предыдущими версиями Bluetooth.

В этом типе протокола были серьёзные качественные обновления, которые позволили назвать новую версию не 4.3, а именно 5.0. Bluetooth 5.0 — большое обновление «синезуба», но оно почти не затрагивает беспроводную передачу звука.

**Bluetooth 5.1**

От предыдущих версий Bluetooth 5.1 отличается тем, что у пользователей есть возможность определять местоположение и направление с максимальной точностью. Ещё лучше оптимизировалось энергопотребление и увеличилась надёжность соединения по Bluetooth Low Energy.

**Bluetooth 5.2**

Спецификация опубликована SIG 6 января 2020 года. Новые функции:

Улучшенная версия протокола атрибутов ATT — Enhanced Attribute protocol (EATT), который более безопасен, так как использует только шифрованное соединение. EATT поддерживает параллельные транзакции, а также позволяет изменять блок максимальной передачи ATT (MTU) во время соединения. В EATT добавлен новый L2CAP-режим безопасного управления потоком — Enhanced Credit Based Flow Control Mode.

Новый LE Power Control — позволяет устройствам динамически оптимизировать мощность для связи между подключенными устройствами. Приёмники Bluetooth LE теперь могут отслеживать уровень сигнала и запрашивать изменения уровня мощности передачи в подключенных устройствах, как правило, для поддержания оптимального уровня сигнала как с точки зрения качества сигнала, так и с точки зрения снижения энергопотребления.

LE Isochronous Channels — функция для поддержки нового стандарта передачи аудио LE Audio, следующего поколения Bluetooth-аудио. Позволяет передавать данные с привязкой ко времени на одно или несколько устройств для синхронизированной по времени обработки (пример: беспроводные наушники с раздельными приёмниками), а также для параллельной трансляции на неограниченное количество устройств.

Bluetooth LE Audio (с 2022 г.)

**Технологическая часть**

Разработка ПП «дистанционная виртуальная клавиатура с использованием Bluetooth» началась с установки программ, необходимых для дальнейшей разработки.

Для разработки вышеуказанного ПП было необходимо составить план разработки, включающий в себя следующие пункты:

* Выбор среды разработки и языка программирования
* Выбор технологии взаимодействия компьютеров
* Выбор архитектуры приложения
* Выбор методологии разработки ПО
* Выбор ПО необходимого для поддержания методологии разработки вышеуказанного ПП и сохранения концептуальной целостности приложения
* Создание плана разработки
* Разделение теоретического прототипа ПП на составные части
* Создание интеллектуальных карт в соответствии с составными частями теоретического прототипа
* Составление прототипов модулей вышеуказанного ПП
* Тестирование прототипов модулей вышеуказанного ПП
* Исправление ошибок в модулях
* Сборка прототипа версии программного продукта
* Тестирование прототипа программного продукта
* Исправление ошибок, появившихся в ходе сборки прототипа
* Создание контрольной точки в СКВ “GIT HUB”
* Создание макета графического пользовательского интерфейса
* Тестирование функционала макета
* Исправление ошибок функционала макета
* Создание прототипа макета графического пользовательского интерфейса
* Тестирование функционала прототипа макета графического пользовательского интерфейса
* Исправление ошибок функционала прототипа макета графического пользовательского интерфейса
* Внедрение графического пользовательского интерфейса в программную оболочку
* Тестирование прототипа дистанционной виртуальной клавиатуры с использованием “Bluetooth”
* Исправление различных ошибок, возникших в следствии внедрения нового функционала
* Создание контрольной точки в СКВ “GIT HUB”
* Создание прототипа работающего на дистрибутиве ОС “Linux” – ОС “Ubuntu”
* Тестирование прототипа работающего на ОС “Ubuntu”
* Исправление ошибок, обнаруженных в ходе тестирования
* Создание ответвления и контрольной точки в СКВ “GIT HUB”

Вышеуказанный план разработки ПП «дистанционная виртуальная клавиатура с использованием Bluetooth» допускает отступление от некоторых, не критичных пунктов

Для создания прототипа и последующей доработки было принято решение составить интелектуальную карту в ПП “Xmind”, (см. Рис. 12)

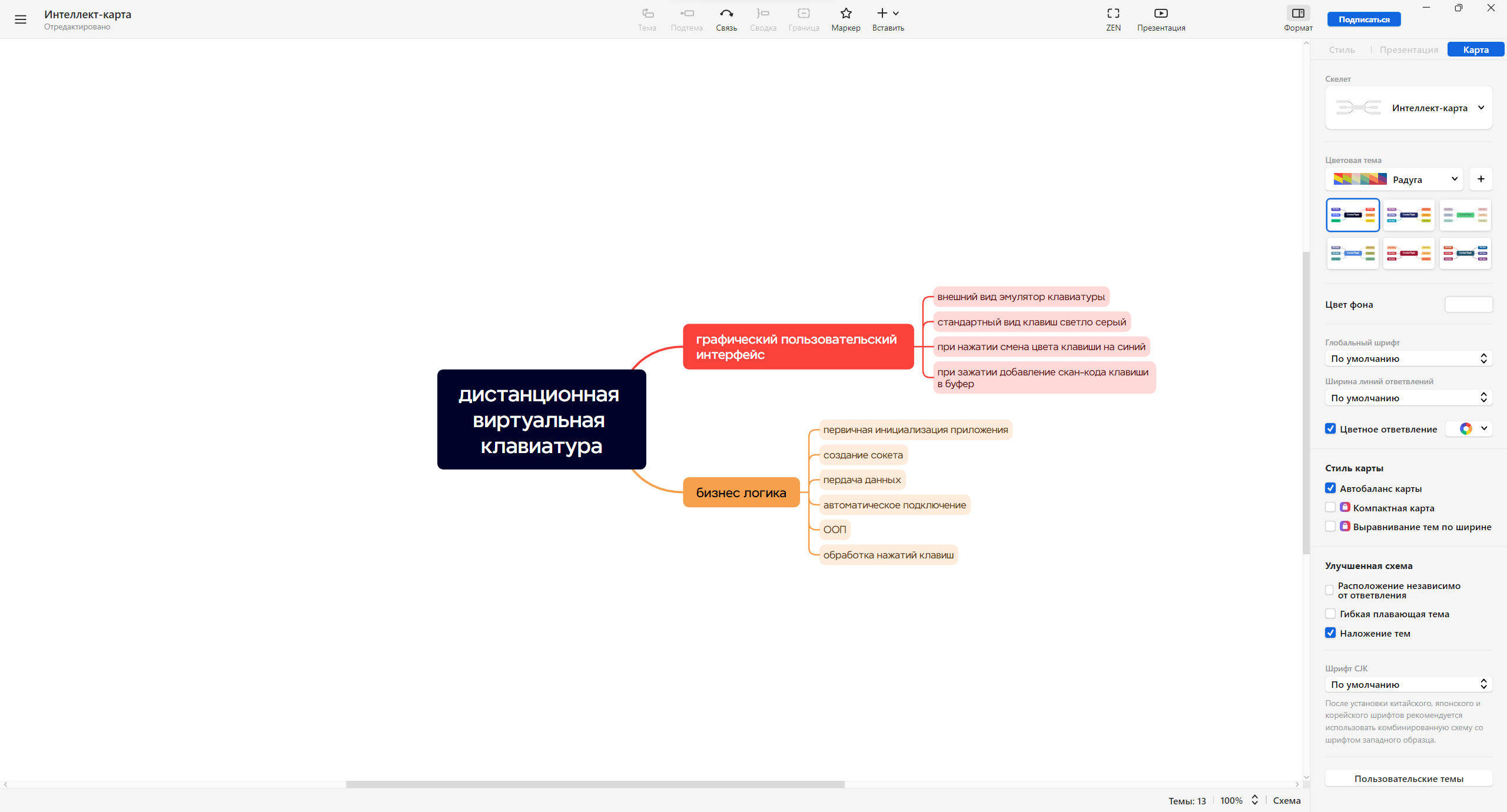


Рисунок 1*2* – интеллектуальная карта прототипа

Было принято решение поделить интеллектуальную карту на две части, бизнес логики и графического пользовательского интерфейса, блок бизнес логики описывал требования к физическому поведению прототипа, блок интерфейса описывал общие требования к интерфейсу и его взаимодействию с бизнес логикой

Вышеуказанная интеллектуальная карта является общим требованием к ПП для разработки архитектуры была построена интеллектуальная карта архитектуры ПП, где ПП разделён по функционалу и по роли в архитектуре на клиент и сервер, (см. Рис. 13)

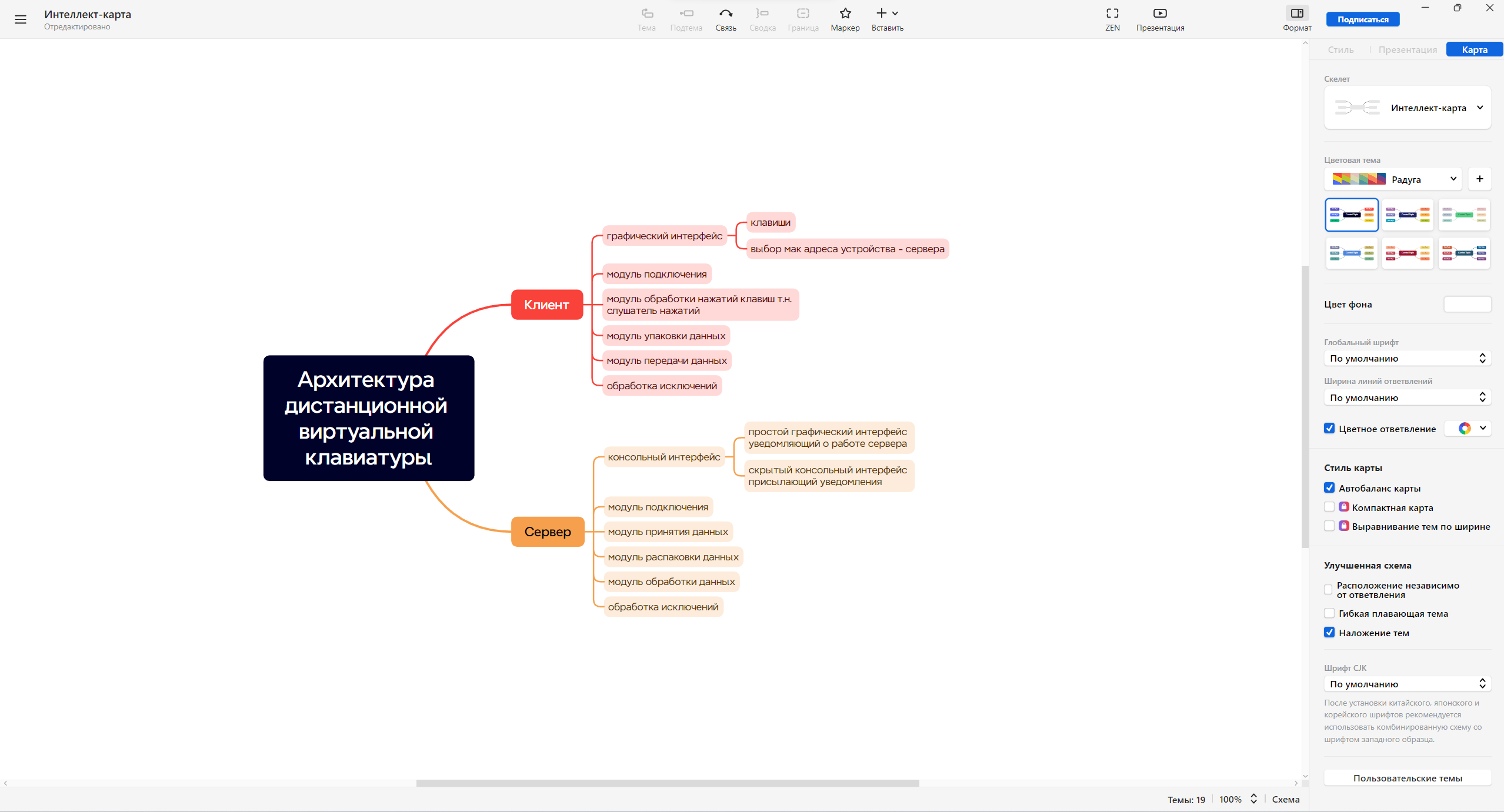


Рисунок 1*3* – интеллектуальная карта архитектуры ПП

Для разработки вышеуказанного ПП, было необходимо разработать простой прототип, обеспечивающий одностороннюю передачу данных одним методом с фиксированными входящими данными, этими данными был “MAC-адрес” компьютера сервера, фиксированный порт, массив данных для передачи, следует отметить, что передача данных шла в зашифрованном режиме, с использованием двоичного кода, это означает, что в метод передачи попадал тип данных массив строк, данный массив переводился в двоичный код, шифровался модулем “Bluetooth” и передавался на компьютер – сервер.

Вышеуказанные требования от прототипа были удовлетворены, было проведено тестирование передачи

Со стороны сервера проблем с получением не возникло кроме некоторых искажений данных, следствием этих искажений является недоработанность метода упаковки данных, не технологии передачи, (см. Рис. 14).

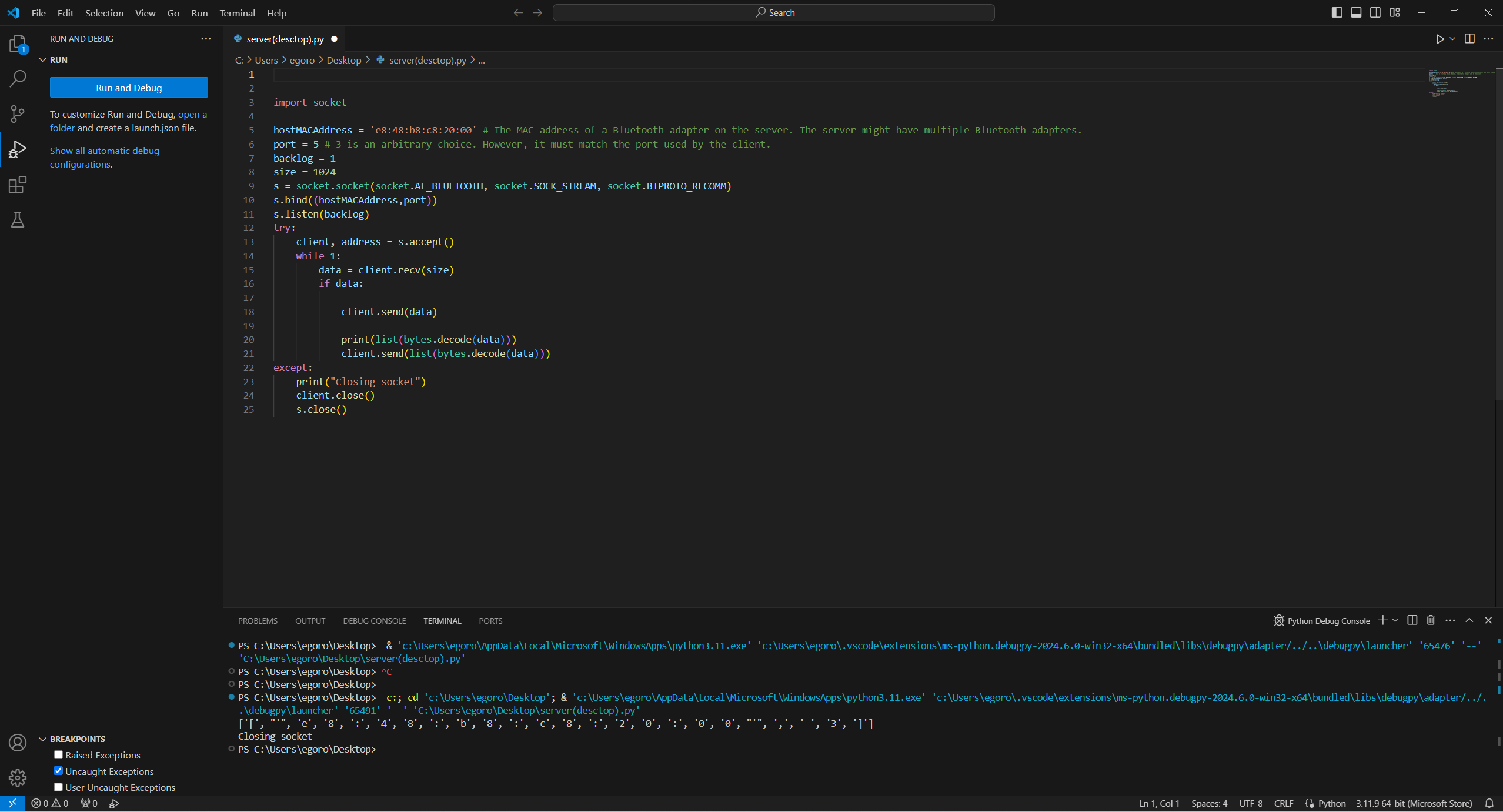


Рисунок 14 – успешно полученный массив данных со стороны сервера

Также следует отметить, что со стороны сервера возникла ошибка об использовании порта, используемого другим активным приложением, данная ошибка была устранена сменой активного порта и показала необходимость добавления функционала, обеспечивающего выбор порта, который не использовался бы другим приложением.

Помимо вышеуказанной ошибки в ходе тестирования прототипа было обнаружено что передача данных прекращается после получения первого пакета данных, данная ошибка также не критичная и указывает что проблема в теле цикла подключения и передачи, для исправления данной ошибки необходимо разделить цикл подключения и передачи на цикл подключения и цикл передачи в таком случае модуль передачи и подключения будет в рабочем состоянии.

Со стороны клиента не возникло проблем при подключении, однако в следствии вышеуказанной проблемы с телом цикла передачи и подключения сервера, было выявлено аварийное отключение модуля передачи клиента от сервера, данная проблема решится при исправлении ошибок со стороны сервера.

После исправления ошибок было проведено повторное тестирование прототипа, было выявлено, что вышеуказанные ошибки были устранены, передача данных более не искажается, подключение можно осуществлять и прерывать без последствий для работы сервера, (см. Рис. 15).

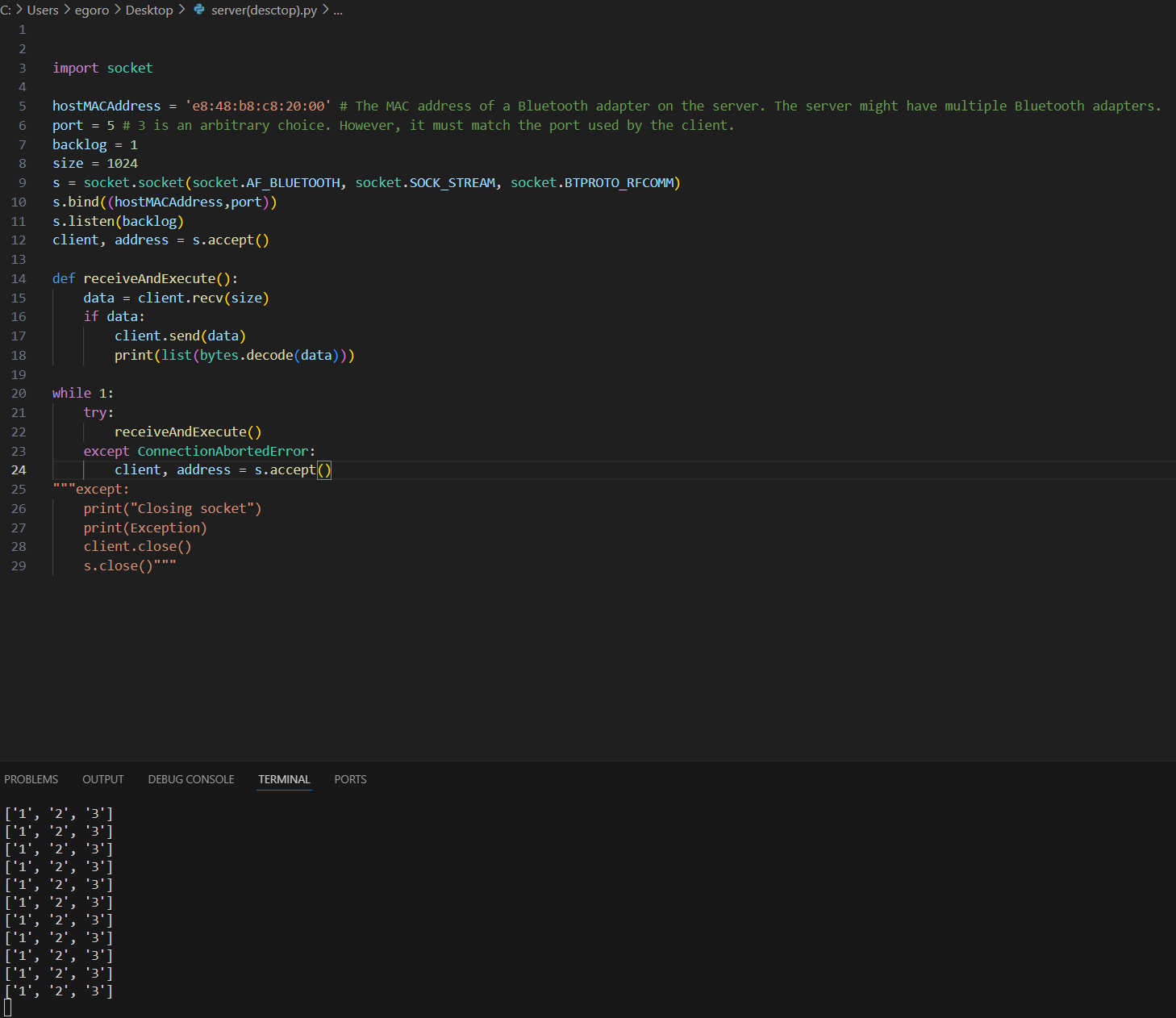


Рисунок 15 – рабочий прототип серверной части приложения

Для реализации модуля двусторонней связи была использованная сторонняя библиотека “Socket” её преимущество в простоте использования, надёжности подключения, а также данная библиотека совместима со всеми современными ОС

Необходимо было зафиксировать текущий прогресс разработки в СКВ “GIT HUB” для последующего упрощения тестирования ПП на разных компьютерах в различных виртуальных машинах, под управлением различных ОС. Все последующие изменения будут фиксироваться подобным образом. (см. Рис. 16)

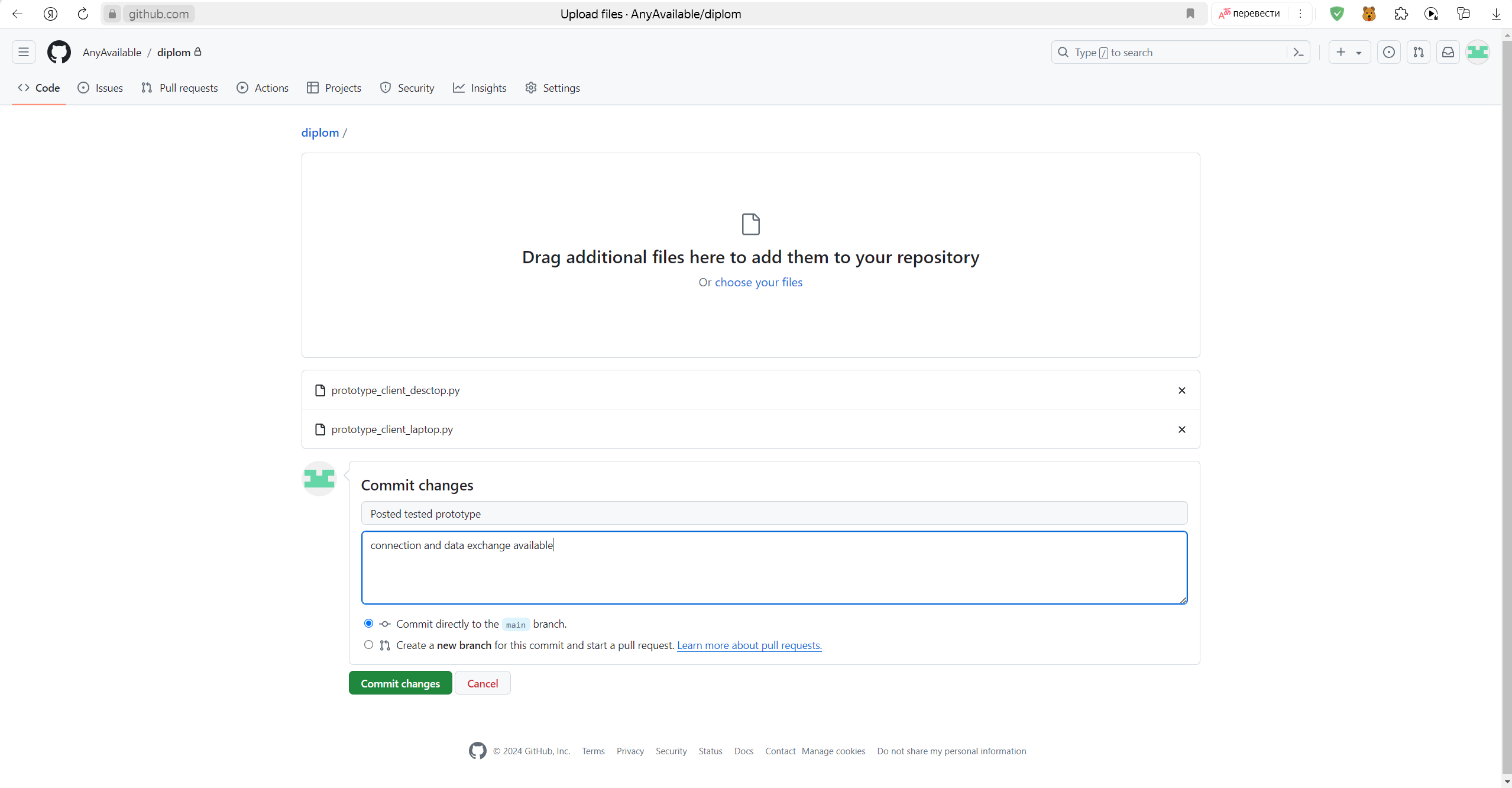


Рисунок 16 – фиксация изменений прототипа

Для дальнейшей реализации обработки нажатий со стороны сервера было принято решение использовать стороннюю библиотеку “keyboard” данная библиотека позволит аппаратно обработать нажатие клавиши, как если бы было произведено нажатие клавиши на физической клавиатуре, а также данная библиотека совместима со всеми современными ОС.

После добавления обработчика нажатий и передачи кода нажатой с клиента на сервер, сервер обработал нажатие в штатном режиме, без ошибок.

Убедившись в работоспособности прототипа, было принято решение создать графический пользовательский интерфейс, реализующий возможность нажимать клавиши и выбирать устройство для подключения и порт.

Для создания макета графического пользовательского интерфейса был применён ПП “Figma” (см. Рис. 17)

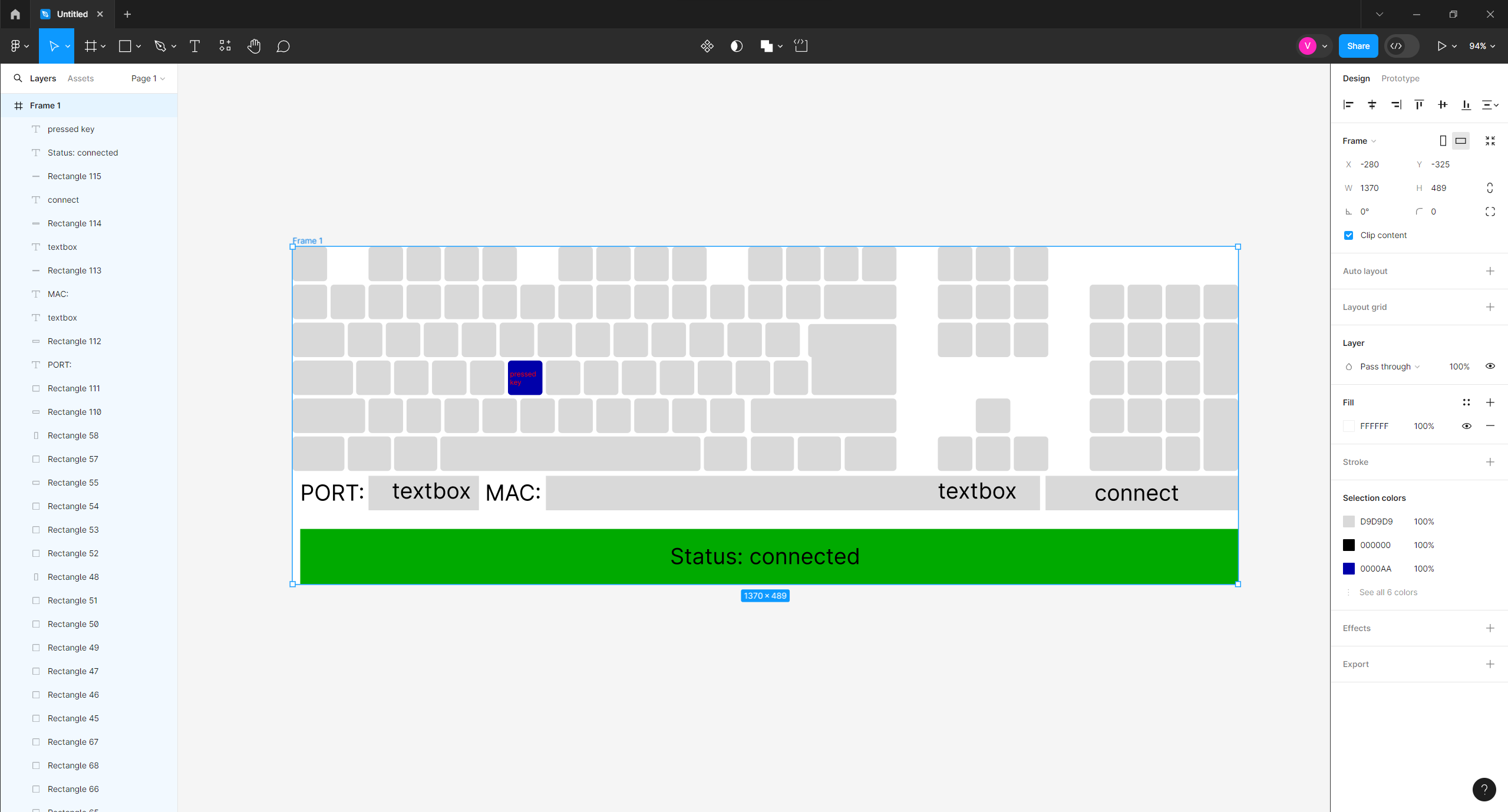
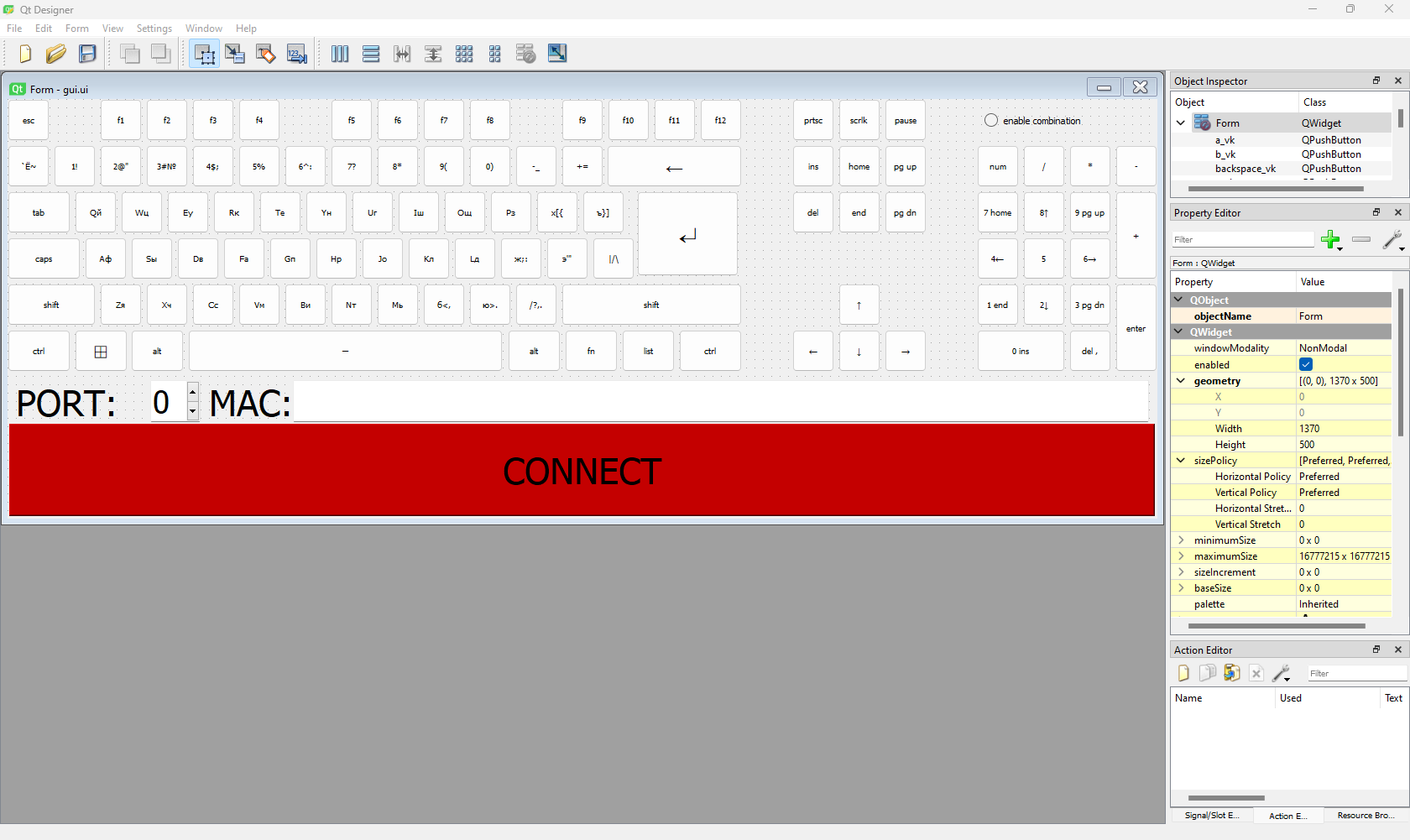


Рисунок 17 – Общий макет графического пользовательского интерфейса

Использование вышеуказанного ПП было обоснованно необходимостью проверки размера окна интерфейса на различных устройствах, а также подбором необходимых размеров клавиш для дальнейшего переноса размеров в программную оболочку графического пользовательского интерфейса.

Для создания рабочей оболочки графического пользовательского интерфейса, реализаций связей графических сущностей и методов, которые вызывают данные сущности был использован ПП “Qt Designer” После создания макета графического интерфейса в ПП “Figma” было необходимо создать разметку пользовательского интерфейса, для этого был использован ПП “Qt Designer” это решение было обосновано экономией времени и удобством в разработке, из-за того что ранее все размеры были выставлены в макете графического пользовательского интерфейса, размеры, расположение кнопок и общая концепция графического пользовательского интерфейса была перенесена в ПП “Qt Designer” за основу форм фактора была взята клавиатура полноразмерная клавиатура “Logitech G513” (См. Рис. 18)



*Рисунок 18 – Внешний вид графического пользовательского интерфейса клиента*

Для создания логики нажатия программу клиента пришлось полностью переписать, было принято решение создать универсальную логику нажатия на каждую клавишу, за исключением служебных клавиш и прочих элементов, для упрощения кода обработки нажатия клавиш была добавлена служебная клавиша обработки комбинаций клавиш, все блоки, возвращающие ошибки в исключительных ситуациях были помещены в блоки обработки исключительных ситуаций.

Добавление каждого элемента сопровождалось тестированием клиента, тестированием сервера, сервер также пришлось переработать, это решение было обосновано тем, что при отключении клиента программа сервера имело тенденцию прекращать свою работу.

Для запуска программы на компьютере, не имеющем программ и библиотек, необходимых для работы было принято решение создать программу в расширении “.EXE” в процессе сборки было обнаружено что программа опознается ОС как вирусная или потенциально опасная, это обоснованно логикой программы, она вмешивается в работу аппаратной части компьютера с использованием внешнего источника связи, что по определению делает ПП “Дистанционная виртуальная клавиатура с использованием Bluetooth” трояном, для решения этой проблемы было принято решение отключить антивирус, а программу поместить в зашифрованный архив, защищённый паролем.

После тестирования собранной программы на компьютере сервере и компьютере клиенте, после тестовой сессии использования данной клавиатуры было принято решение все файлы связанные с разработкой выгрузить в СКВ “GIT HUB”

**Экономическая часть**

**4.1 Расчет заработной платы исполнителя работ по созданию программного продукта**

Определяем общую заработную плату исполнителя работ:

ЗПобщ.=Змес.+П, руб. (4.1)

где Змес. - заработная плата за месяц, руб;

П - сумма премии, руб.

Для этого определим сумму премии:

П=Змес.\*%П/100 (4.2)

П=100 000\*10/100=10 000 рублей

Теперь определим основную заработную плату исполнителя работ:

ЗПобщ. =100 000+10 000руб.

**4.2 Расчет страховой выплаты**

Производим расчет страховой выплаты, которая начисляется на заработную плату и составляет 30%.

 ,руб. (4.3)

33 000руб.

**4.3 Расчет расходов на содержание и эксплуатацию ПЭВМ**

Определяем амортизационные отчисления, которые составляют 25% от балансовой стоимости ПЭВМ.

,руб. (4.4)

где Кб - балансовая стоимость одной ПЭВМ с периферией, руб.

Таблица 4.1 - Балансовая стоимость ПЭВМ с периферией

|  |  |
| --- | --- |
| Наименование | Цена в рублях |
| Монитор | 15 000 |
| Материнская плата | 25 000 |
| Процессор | 11 000 |
| Видео карта | 16 000 |
| HDD | 7 500 |
| FDD | 5 500 |
| DWD-RW | 2 700 |
| Клавиатура | 1 300 |
| Принтер | 25 000 |
| Мышь | 1 000 |
| BOX | 10 000 |
| Итого: | 120 000 |

24 000 руб.

Затраты на электрическую энергию складываются из:

- затрат на силовую электрическую энергию,

- затрат на электрическую энергию, которая идет на освящение.

Определяем затраты на силовую электрическую энергию:

Зс.эн = Т \* Цэ \* Р ,руб. (4.5)

где Т - часы работы за месяц исполнителя работ, час

Цэ - стоимость 1 кВт / часа, руб.

Зс. эн =168\*6,42\*0,9 = 970,704 руб.

Определяем затраты на электрическую энергию, которая идет на освещение:

Зосв. = Т \* Цэ \* Росв ,руб. (4.6)

где Росв. - суммарная мощность, которая идет на освещение, кВт /час. Росв = 500 Вт / час.

Зосв =168\*6,42\*0,5= 539,28 рублей

Определяем общие затраты на эл. энергию:

Зэн = Зс.эн + Зосв ,руб. (4.7)

Зэн = 970,704+539,28=1 509,984=1510

Таблица 4.2

Затраты на расходные материалы

|  |  |  |
| --- | --- | --- |
| Наименование | Кол-во | Цена в рублях |
| Картридж | 1 шт. | 2 500 |
| Flesh-card | 1 шт. | 600 |
| Бумага | 1 пачка | 500 |
| Итого: |  | 3 600 |

Затраты на расходные материалы берутся по факту и составляют:

Зр. = 3600 руб.

Определяем общие расходы на содержание и эксплуатацию 1-ой ПЭВМ.

Р с.э.=А+Зэн.+ Зр., руб. (4.8)

Рс.э. = 120 000+1510+3600=125 110 рублей

**4.4 Расчёт себестоимости программного продукта**

В себестоимость программного продукта входят следующие элементы:

1) Общая заработная плата исполнителя работ по созданию программного продукта.

2) Начисления на заработную плату.

3) Расходы на содержание и эксплуатацию ПЭВМ, относящихся к данному программному продукту.

4)Прочие расходы (составляют 10% от суммы первых трех элементов).

Пр.р.= (ЗПобщ. + Нзп + Рсэ) \* 0,10 ,руб. (4.9)

Пр.р. = (110000+33 000+125 110) \* 0,10 =26 811

Определяем себестоимость программного продукта:

Сп.п.=ЗПобщ.+ Нзп + Рсэ + Пр.р., руб . (4.10)

Сп.п. = 110 000+ 33 000+120 510+26 811=290 321 рублей

Отразим структуру себестоимости программного продукта в таблице 4.3

Таблица 4.3 - Себестоимость программного продукта

|  |  |  |
| --- | --- | --- |
| Элементы себестоимости | Сумма  в рублях | % в общей сумме себестоимости |
| Общая заработная плата исполнителя | 110 000 | 37,89 |
| Начисления на заработную плату | 33 000 | 11,37 |
| Расходы на содержание и эксплуатацию ПЭВМ | 120 510 | 41,51 |
| Прочие расходы | 26 811 | 9,23 |
| Итого: | 290 321 | 100 |

**4.5 Расчет цены программного продукта**

Определяем цену программного продукта:

Ц = С+П+НДС ,руб. (4.11)

где С - себестоимость программного продукта, руб.

П - прибыль, которую берем в размере 20% от себестоимости.

НДС – налог на добавленную стоимость, берется в размере 20% от суммы себестоимости и прибыли.

П= 290 321\*20%= 58 064,2

НДС=(279 861+58 064,2)\*20%=67 585,04

Подставляя значения в формулу (4.11), определяем цену программного продукта:

Ц= 279 861+58 064,2+67 585,04= 405 510,24

Сводная таблица показателей

Исходя из расчетов цены программного продукта была составлена таблица 2.4, а также был составлен график статьи затрат, которые представлен на рисунке

Таблица 4.4 - Результаты расчетов

|  |  |
| --- | --- |
| Наименование показателя | Сумма |
| Себестоимость программного продукта | 279 861 |
| Прибыль | 58 064,2 |
| НДС | 67 585,04 |
| Цена программного продукта | 405 510,24 |

Рисунок 19 – Статьи затрат

В ходе анализа были выявлены следующие положения. Наибольшая часть расходов приходится на содержание и эксплуатацию ПЭВМ, составляя 41,51% от общих затрат. Это указывает на значительную зависимость деятельности исполнителя от компьютерной техники и ее обслуживания. Общая заработная плата исполнителя составляет 37,89% от всех расходов, что также является значительной частью бюджета. Это подчеркивает важность человеческого ресурса в общей структуре затрат. Дополнительные начисления на заработную плату составляют 11,37%. Это отражает обязательства по социальным отчислениям и другим начислениям, связанным с оплатой труда. Прочие расходы составляют 9,23% от общих затрат. Хотя эта часть не самая крупная, она все же существенна и включает в себя различные дополнительные расходы, которые могут варьироваться. Суммарные расходы исполнителя составляют 290 321 рубль. Основные статьи расходов четко распределены, с наибольшими затратами на содержание и эксплуатацию ПЭВМ и на заработную плату, что в сумме составляет 79,4% всех расходов.

Наибольшая часть бюджета направлена на содержание и эксплуатацию ПЭВМ и заработную плату, что подчеркивает важность технического и человеческого ресурсов в деятельности исполнителя. Для оптимизации расходов можно рассмотреть возможные способы снижения затрат на содержание ПЭВМ и начисления на заработную плату без ущерба для качества работы. Прочие расходы, хотя и составляют меньшую долю, также могут быть проанализированы для выявления возможных точек сокращения затрат.

**Охрана труда**

1. Прежде, чем приступать к работе следует проверить состояние рабочего места; если оно не убрано или загромождено, необходимо принять меры к наведению порядка.

2. Перед началом работы работник должен осмотреть инструмент, контрольно-измерительные приборы, переходные шнуры, переносной светильник (при необходимости), которые будут использоваться в работе, и убедиться в их исправности.

3. Рабочий инструмент (отвертки, плоскогубцы, кусачки и др.) должен иметь электрическую изоляцию на рукоятках, не имеющую механических повреждений; изоляционное покрытие рукояток инструмента должно плотно прилегать к металлическим частям инструмента и полностью изолировать ту его часть, которая во время работы находится в руке.

4. Предотвращение ошибочного включения напряжения может быть обеспечено такими мерами, как снятие предохранителей, установка между контактами изолирующих прокладок и т.п.

5. Перед началом работы нужно убедиться в достаточности и равномерности освещения рабочего места.

6. Перед началом работ работник должен ознакомиться с эксплуатационно-технической документацией по обслуживанию ПК.

7. Работник должен лично убедиться в том, что все меры, необходимые для обеспечения безопасности выполнены.

8. Работник не должен приступать к работе, если у него имеются сомнения в обеспечении безопасности при выполнении предстоящей работы.

9. При обслуживании ПК работнику следует пользоваться только исправным инструментом.

10. Инструмент на рабочем месте должен быть расположен так, чтобы исключалась возможность его скатывания и падения.

11. Отвертка, в том числе длинная, диэлектрическая с металлическим жалом для обслуживания ПК, должна выбираться по ширине рабочей части (лопатки или звездочки), зависящей от размера шлица в головке винта.

12. До начала работы по обслуживанию ПК работнику необходимо убедиться в соответствии номиналов предохранителей с напряжением питающей сети и в отсутствии замыкания в шнуре питания и его вилке.

13. Перед снятием защитного кожуха с ПК следует обесточить его и принять меры против ошибочного включения.

14. Обслуживать ПК под напряжением разрешается только в тех случаях, когда иначе выполнить работу невозможно; при этом работнику необходимо быть особенно внимательным во избежание прикосновения к токоведущим частям, работать следует одной рукой.

15. При обслуживании ПК необходимо установить его таким образом, чтобы смещение центра тяжести не привело к падению.

16. При обслуживании ПК работнику необходимо выполнять следующие ограничения:

— нельзя включать ПК с помощью оголенных проводов без вилки в розетку;

— запрещается производить обслуживание включенного в сеть ПК в

непосредственной близости от заземленных конструкций, в помещениях с

повышенной влажностью и в помещениях с токопроводящими полами;

— запрещается иметь на одном рабочем столе два включенных в сеть ПК

со снятыми защитными кожухами;

— запрещается производить прозвонку и другие операции, выполняемые

двумя руками в ПК без предварительного отключения его от

электрической сети;

— не следует пользоваться защитными средствами, срок испытания

которых истек.

17. Кабель в месте ввода в электроинструмент должен быть защищен от

истирания и перегибов эластичной трубкой из изоляционного материала;

трубка должна быть закреплена в корпусных деталях электроинструмента,

и выступать из них на длину не менее пяти диаметров кабеля.

**ЗАКЛЮЧЕНИЕ**

В процессе создания дипломного проекта, реализуя цели и задачи работы было выявлено множество ошибок в программном коде, несколько концептуальных противоречий, при соблюдении плана разработки были допущены отклонения от некоторых пунктов, итеративная модель разработки ПП также не соответствовала теоретическому прототипу в реальности, однако ПП “Дистанционная виртуальная клавиатура с использованием Bluetooth” оправдал возложенные на него цели и ожидания, основной функционал программы был составлен, протестирован, выгружен в СКВ, графический интерфейс был создан, подключён, была реализована логика обработки нажатий, в процессе тестирования было обнаружено, что программа способна работать во всех современных ОС, а также имеет интуитивно понятный графический пользовательский интерфейс, в процессе разработки были обсуждены все вопросы, обсуждение которых предполагалось, в процессе разработки также возникли дополнительные вопросы, например, был обнаружен вопрос, как систематизировать разработку и поставить её на поток, найти определённый алгоритм разработки и успешно его внедрить в текущую разработку, этот вопрос был поднят и был решён, также в процессе разработки была изучена сама технология “Bluetooth”, был получен опыт по работе с различным программным обеспечением, а также опыт по организации и менеджменту рабочего времени, а в частности опыт использования итеративной методологии разработки программного обеспечения поставленные задачи перед дипломным проектом было принято считать успешно выполненными, а программный продукт конкурентоспособным.

* Код клиентской части ПП – ПРИЛОЖЕНИЕ А
* Код серверной части ПП – ПРИЛОЖЕНИЕ Б
* Код разметки графического пользовательского интерфейса ПП – ПРИЛОЖЕНИЕ В

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[https://www.dns-shop.ru/catalog/17a8950d16404e77/klaviatury/?stock=now-today-tomorrow-later&f[54i]=1oqp&f[54j]=1oqs](https://www.dns-shop.ru/catalog/17a8950d16404e77/klaviatury/?stock=now-today-tomorrow-later&f%5b54i%5d=1oqp&f%5b54j%5d=1oqs)

<https://library-it.com/os/linux/sovremennye-operacionnye-sistemy_4e-izd-2015/>

<https://www.lants.ru/books/IT/Таненбаум.Э%20-%20Компьютерные%20сети(Классика%20Computer%20Science)-2012.PDF>

<https://ru.wikipedia.org/wiki/Bluetooth>

<https://learn.microsoft.com/ru-ru/windows/win32/winsock/windows-sockets-error-codes-2>

<https://superuser.com/questions/1247382/is-there-a-unicode-character-for-the-windows-key>

<https://ru.piliapp.com/symbols/enter/#:~:text=%D0%A1%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB%20Enter%2C%20%D0%BE%D0%B1%D0%BE%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9%20%D0%BA%D0%B0%D0%BA%20%E2%86%B5%2C,%D0%BD%D0%BE%D0%B2%D0%BE%D0%B9%20%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B8%20%D0%B2%20%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D1%85%20%D0%BE%D0%BA%D1%80%D1%83%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F%D1%85>

<https://stackoverflow.com/questions/66081534/pyqt5-store-time-of-keypressevent>

<https://stackoverflow.com/questions/24659239/how-to-change-qpushbutton-text-and-background-color>

<https://stackoverflow.com/questions/15478127/remove-final-character-from-string>

<https://github.com/boppreh/keyboard/blob/master/README.md>

<https://stackabuse.com/guide-to-pythons-keyboard-module/>

<https://programmersought.com/article/36897737565/>

<https://www.geeksforgeeks.org/how-to-access-textedit-fields-in-pyqt/>

<https://www.pythonguis.com/installation/install-pyqt-linux/>

<https://pythonhint.com/post/1083074656446618/bluetooth-server-with-python-33>

https://www.geeksforgeeks.org/iterative-vs-incremental-model-in-software-development/

**ПРИЛОЖЕНИЕ А**

import sys

from PyQt5 import uic

from PyQt5.QtWidgets import QApplication, QMainWindow

import socket

import time

class App(QMainWindow):

    def \_\_init\_\_(self):

        super().\_\_init\_\_()

        uic.loadUi('gui.ui', self)

        self.serverMACAddress = 'e8:48:b8:c8:20:00'

        self.port = 7

        self.s = socket.socket(socket.AF\_BLUETOOTH, socket.SOCK\_STREAM, socket.BTPROTO\_RFCOMM)

        self.connected = False

        self.memory = ""

        self.combo = self.comboenable\_rb

        self.port\_spinbox.textChanged.connect(self.portChange)

        self.mac\_textarea.textChanged.connect(self.macChange)

        self.esc\_vk.clicked.connect(self.esc)

        self.f1\_vk.clicked.connect(self.f1)

        self.f2\_vk.clicked.connect(self.f2)

        self.f3\_vk.clicked.connect(self.f3)

        self.f4\_vk.clicked.connect(self.f4)

        self.f5\_vk.clicked.connect(self.f5)

        self.f6\_vk.clicked.connect(self.f6)

        self.f7\_vk.clicked.connect(self.f7)

        self.f8\_vk.clicked.connect(self.f8)

        self.f9\_vk.clicked.connect(self.f9)

        self.f10\_vk.clicked.connect(self.f10)

        self.f11\_vk.clicked.connect(self.f11)

        self.f12\_vk.clicked.connect(self.f12)

        self.del\_vk.clicked.connect(self.delete)

        self.tilda\_vk.clicked.connect(self.tilda)

        self.one\_vk.clicked.connect(self.one)

        self.two\_vk.clicked.connect(self.two)

        self.three\_vk.clicked.connect(self.three)

        self.four\_vk.clicked.connect(self.four)

        self.five\_vk.clicked.connect(self.five)

        self.six\_vk.clicked.connect(self.six)

        self.seven\_vk.clicked.connect(self.seven)

        self.eight\_vk.clicked.connect(self.eight)

        self.nine\_vk.clicked.connect(self.nine)

        self.zero\_vk.clicked.connect(self.zero)

        self.minus\_vk.clicked.connect(self.minus)

        self.plus\_vk.clicked.connect(self.plus)

        self.backspace\_vk.clicked.connect(self.backspace)

        self.ins\_vk.clicked.connect(self.ins)

        self.home\_vk.clicked.connect(self.home)

        self.pgup\_vk.clicked.connect(self.pgup)

        self.del\_vk.clicked.connect(self.delete)

        self.end\_vk.clicked.connect(self.end)

        self.pgdn\_vk.clicked.connect(self.pgdn)

        self.tab\_vk.clicked.connect(self.tab)

        self.q\_vk.clicked.connect(self.q)

        self.w\_vk.clicked.connect(self.w)

        self.e\_vk.clicked.connect(self.e)

        self.r\_vk.clicked.connect(self.r)

        self.t\_vk.clicked.connect(self.t)

        self.y\_vk.clicked.connect(self.y)

        self.u\_vk.clicked.connect(self.u)

        self.i\_vk.clicked.connect(self.i)

        self.o\_vk.clicked.connect(self.o)

        self.p\_vk.clicked.connect(self.p)

        self.leftbracket\_vk.clicked.connect(self.leftbracket)

        self.rightbracket\_vk.clicked.connect(self.rightbracket)

        self.a\_vk.clicked.connect(self.a)

        self.s\_vk.clicked.connect(self.sl)

        self.d\_vk.clicked.connect(self.d)

        self.f\_vk.clicked.connect(self.f)

        self.g\_vk.clicked.connect(self.g)

        self.h\_vk.clicked.connect(self.h)

        self.j\_vk.clicked.connect(self.j)

        self.k\_vk.clicked.connect(self.k)

        self.l\_vk.clicked.connect(self.l)

        self.zh\_vk.clicked.connect(self.zh)

        self.commas\_vk.clicked.connect(self.commas) #

        self.slash\_vk.clicked.connect(self.slash)

        self.z\_vk.clicked.connect(self.z)

        self.x\_vk.clicked.connect(self.x)

        self.c\_vk.clicked.connect(self.c)

        self.v\_vk.clicked.connect(self.v)

        self.b\_vk.clicked.connect(self.b)

        self.n\_vk.clicked.connect(self.n)

        self.m\_vk.clicked.connect(self.m)

        self.leftarrow\_vk.clicked.connect(self.leftarrow)

        self.rightarrow\_vk.clicked.connect(self.rightarrow)

        self.dot\_vk.clicked.connect(self.dot)

        self.rightshift\_vk.clicked.connect(self.rightshift)

        self.enter\_vk.clicked.connect(self.enter)

        self.rightctrl\_vk.clicked.connect(self.rightctrl)

        #self.list\_\_vk.clicked.connect(self.list)

        #self.fn\_vk.clicked.connect(self.fn)

        self.rightalt\_vk.clicked.connect(self.rightalt)

        self.leftalt\_vk.clicked.connect(self.leftalt)

        self.space\_vk.clicked.connect(self.space)

        self.leftctrl\_vk.clicked.connect(self.leftctrl)

        self.up\_vk.clicked.connect(self.up)

        self.down\_vk.clicked.connect(self.down)

        self.left\_vk.clicked.connect(self.left)

        self.right\_vk.clicked.connect(self.right)

        self.capslock\_vk.clicked.connect(self.caps)

        self.capitalize = False

        self.win\_vk.clicked.connect(self.win)

        self.r\_vk.clicked.connect(self.r)

        self.connect\_vk.clicked.connect(self.connectKeyboard)

    def connectKeyboard(self):

        if self.connected:

            try:

                self.s.close()

                self.connect\_vk.setStyleSheet('#connect\_vk{background: rgba(196,0,0,255)}')

                self.connect\_vk.setText("CONNECT")

                self.connected = False

            except:

                pass

        else:

            try:

                self.s.connect((self.serverMACAddress, self.port))

                self.connect\_vk.setStyleSheet('#connect\_vk{background: rgba(0,196,0,255)}')

                self.connect\_vk.setText("DISCONNECT")

                self.connected = True

            except:

                pass

    def portChange(self):

        self.port = self.port\_spinbox.value()

    def macChange(self):

        self.serverMACAddress = self.mac\_textarea.toPlainText()

    def esc(self):

        if self.combo.isChecked():

            self.memory += "esc+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("esc"), "utf-8"))

            self.s.recv(1024)

    def f1(self):

        if self.combo.isChecked():

            self.memory += "f1+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("f1"), "utf-8"))

            self.s.recv(1024)

    def f2(self):

        if self.combo.isChecked():

            self.memory += "f2+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("f2"), "utf-8"))

            self.s.recv(1024)

    def f3(self):

        if self.combo.isChecked():

            self.memory += "f3+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("f3"), "utf-8"))

            self.s.recv(1024)

    def f4(self):

        if self.combo.isChecked():

            self.memory += "f4+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("f4"), "utf-8"))

            self.s.recv(1024)

    def f5(self):

        if self.combo.isChecked():

            self.memory += "f5+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("f5"), "utf-8"))

            self.s.recv(1024)

    def f6(self):

        if self.combo.isChecked():

            self.memory += "f6+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("f6"), "utf-8"))

            self.s.recv(1024)

    def f7(self):

        if self.combo.isChecked():

            self.memory += "f7+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("f7"), "utf-8"))

            self.s.recv(1024)

    def f8(self):

        if self.combo.isChecked():

            self.memory += "f8+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("f8"), "utf-8"))

            self.s.recv(1024)

    def f9(self):

        if self.combo.isChecked():

            self.memory += "f9+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("f9"), "utf-8"))

            self.s.recv(1024)

    def f10(self):

        if self.combo.isChecked():

            self.memory += "f10+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("f10"), "utf-8"))

            self.s.recv(1024)

    def f11(self):

        if self.combo.isChecked():

            self.memory += "f11+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("f11"), "utf-8"))

            self.s.recv(1024)

    def f12(self):

        if self.combo.isChecked():

            self.memory += "f12+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("rf12"), "utf-8"))

            self.s.recv(1024)

    def delete(self):

        if self.combo.isChecked():

            self.memory += "del+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("del"), "utf-8"))

            self.s.recv(1024)

    def tilda(self):

        if self.combo.isChecked():

            self.memory += "~+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("~"), "utf-8"))

            self.s.recv(1024)

    def one(self):

        if self.combo.isChecked():

            self.memory += "1+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("1"), "utf-8"))

            self.s.recv(1024)

    def two(self):

        if self.combo.isChecked():

            self.memory += "2+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("2"), "utf-8"))

            self.s.recv(1024)

    def three(self):

        if self.combo.isChecked():

            self.memory += "3+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("3"), "utf-8"))

            self.s.recv(1024)

    def four(self):

        if self.combo.isChecked():

            self.memory += "4+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("4"), "utf-8"))

            self.s.recv(1024)

    def five(self):

        if self.combo.isChecked():

            self.memory += "5+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("5"), "utf-8"))

            self.s.recv(1024)

    def six(self):

        if self.combo.isChecked():

            self.memory += "6+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("6"), "utf-8"))

            self.s.recv(1024)

    def seven(self):

        if self.combo.isChecked():

            self.memory += "7+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("7"), "utf-8"))

            self.s.recv(1024)

    def eight(self):

        if self.combo.isChecked():

            self.memory += "8+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("8"), "utf-8"))

            self.s.recv(1024)

    def nine(self):

        if self.combo.isChecked():

            self.memory += "9+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("9"), "utf-8"))

            self.s.recv(1024)

    def zero(self):

        if self.combo.isChecked():

            self.memory += "0+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("0"), "utf-8"))

            self.s.recv(1024)

    def minus(self):

        if self.combo.isChecked():

            self.memory += "minus+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("minus"), "utf-8"))

            self.s.recv(1024)

    def plus(self):

        if self.combo.isChecked():

            self.memory += "plus+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("plus"), "utf-8"))

            self.s.recv(1024)

    def backspace(self):

        if self.combo.isChecked():

            self.memory += "backspace+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("backspace"), "utf-8"))

            self.s.recv(1024)

    def tab(self):

        if self.combo.isChecked():

            self.memory += "tab+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("tab"), "utf-8"))

            self.s.recv(1024)

    def q(self):

        if self.combo.isChecked():

            self.memory += "q+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("q"), "utf-8"))

            self.s.recv(1024)

    def w(self):

        if self.combo.isChecked():

            self.memory += "w+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("w"), "utf-8"))

            self.s.recv(1024)

    def e(self):

        if self.combo.isChecked():

            self.memory += "e+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("e"), "utf-8"))

            self.s.recv(1024)

    def r(self):

        if self.combo.isChecked():

            self.memory += "r+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("r"), "utf-8"))

            self.s.recv(1024)

    def t(self):

        if self.combo.isChecked():

            self.memory += "t+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("t"), "utf-8"))

            self.s.recv(1024)

    def y(self):

        if self.combo.isChecked():

            self.memory += "y+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("y"), "utf-8"))

            self.s.recv(1024)

    def u(self):

        if self.combo.isChecked():

            self.memory += "u+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("u"), "utf-8"))

            self.s.recv(1024)

    def i(self):

        if self.combo.isChecked():

            self.memory += "i+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("i"), "utf-8"))

            self.s.recv(1024)

    def o(self):

        if self.combo.isChecked():

            self.memory += "o+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("o"), "utf-8"))

            self.s.recv(1024)

    def p(self):

        if self.combo.isChecked():

            self.memory += "p+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("p"), "utf-8"))

            self.s.recv(1024)

    def leftbracket(self):

        if self.combo.isChecked():

            self.memory += "[+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("["), "utf-8"))

            self.s.recv(1024)

    def rightbracket(self):

        if self.combo.isChecked():

            self.memory += "]+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("]"), "utf-8"))

            self.s.recv(1024)

    def enter(self):

        if self.combo.isChecked():

            self.memory += "enter+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("enter"), "utf-8"))

            self.s.recv(1024)

    def caps(self):

        self.s.send(bytes(str("capslock"), "utf-8"))

        self.s.recv(1024)

    def a(self):

        if self.combo.isChecked():

            self.memory += "a+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("a"), "utf-8"))

            self.s.recv(1024)

    def sl(self):

        if self.combo.isChecked():

            self.memory += "s+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("s"), "utf-8"))

            self.s.recv(1024)

    def d(self):

        if self.combo.isChecked():

            self.memory += "d+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("d"), "utf-8"))

            self.s.recv(1024)

    def f(self):

        if self.combo.isChecked():

            self.memory += "f+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("f"), "utf-8"))

            self.s.recv(1024)

    def g(self):

        if self.combo.isChecked():

            self.memory += "g+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("g"), "utf-8"))

            self.s.recv(1024)

    def h(self):

        if self.combo.isChecked():

            self.memory += "h+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("h"), "utf-8"))

            self.s.recv(1024)

    def j(self):

        if self.combo.isChecked():

            self.memory += "j+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("j"), "utf-8"))

            self.s.recv(1024)

    def k(self):

        if self.combo.isChecked():

            self.memory += "k+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("k"), "utf-8"))

            self.s.recv(1024)

    def l(self):

        if self.combo.isChecked():

            self.memory += "l+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("l"), "utf-8"))

            self.s.recv(1024)

    def zh(self):

        if self.combo.isChecked():

            self.memory += "ж+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("ж"), "utf-8"))

            self.s.recv(1024)

    def commas(self):

        if self.combo.isChecked():

            self.memory += "э+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("э"), "utf-8"))

            self.s.recv(1024)

    def slash(self):

        if self.combo.isChecked():

            self.memory += "|+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("|"), "utf-8"))

            self.s.recv(1024)

    def shift(self):

        if self.combo.isChecked():

            self.memory += "left shift+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("left shift"), "utf-8"))

            self.s.recv(1024)

    def z(self):

        if self.combo.isChecked():

            self.memory += "z+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("z"), "utf-8"))

            self.s.recv(1024)

    def x(self):

        if self.combo.isChecked():

            self.memory += "x+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("x"), "utf-8"))

            self.s.recv(1024)

    def c(self):

        if self.combo.isChecked():

            self.memory += "c+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("c"), "utf-8"))

            self.s.recv(1024)

    def v(self):

        if self.combo.isChecked():

            self.memory += "v+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("v"), "utf-8"))

            self.s.recv(1024)

    def b(self):

        if self.combo.isChecked():

            self.memory += "b+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("b"), "utf-8"))

            self.s.recv(1024)

    def n(self):

        if self.combo.isChecked():

            self.memory += "n+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("n"), "utf-8"))

            self.s.recv(1024)

    def m(self):

        if self.combo.isChecked():

            self.memory += "m+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("m"), "utf-8"))

            self.s.recv(1024)

    def leftarrow(self):

        if self.combo.isChecked():

            self.memory += "б+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("б"), "utf-8"))

            self.s.recv(1024)

    def rightarrow(self):

        if self.combo.isChecked():

            self.memory += "ю+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("ю"), "utf-8"))

            self.s.recv(1024)

    def dot(self):

        if self.combo.isChecked():

            self.memory += "?+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("?"), "utf-8"))

            self.s.recv(1024)

    def rightshift(self):

        if self.combo.isChecked():

            self.memory += "right shift+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("right shift"), "utf-8"))

            self.s.recv(1024)

    def leftctrl(self):

        if self.combo.isChecked():

            self.memory += "left control+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("|"), "utf-8"))

            self.s.recv(1024)

    def leftalt(self):

        if self.combo.isChecked():

            self.memory += "left alt+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("left alt"), "utf-8"))

            self.s.recv(1024)

    def space(self):

        if self.combo.isChecked():

            self.memory += "space+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("space"), "utf-8"))

            self.s.recv(1024)

    def rightalt(self):

        if self.combo.isChecked():

            self.memory += "right alt+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("right alt"), "utf-8"))

            self.s.recv(1024)

    def rightctrl(self):

        if self.combo.isChecked():

            self.memory += "right control+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("right control"), "utf-8"))

            self.s.recv(1024)

    def left(self):

        if self.combo.isChecked():

            self.memory += "left+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("left"), "utf-8"))

            self.s.recv(1024)

    def right(self):

        if self.combo.isChecked():

            self.memory += "right+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("right"), "utf-8"))

            self.s.recv(1024)

    def up(self):

        if self.combo.isChecked():

            self.memory += "up+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("up"), "utf-8"))

            self.s.recv(1024)

    def down(self):

        if self.combo.isChecked():

            self.memory += "sown+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("down"), "utf-8"))

            self.s.recv(1024)

    def win(self):

        if self.combo.isChecked():

            self.memory += "win+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("win"), "utf-8"))

            self.s.recv(1024)

    def prtsc(self):

        if self.combo.isChecked():

            self.memory += "printscreen+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("printscreen"), "utf-8"))

            self.s.recv(1024)

    def pause(self):

        if self.combo.isChecked():

            self.memory += "pause+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("pause"), "utf-8"))

            self.s.recv(1024)

    def scrlock(self):

        if self.combo.isChecked():

            self.memory += "scroll lock+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("scroll lock"), "utf-8"))

            self.s.recv(1024)

    def ins(self):

        if self.combo.isChecked():

            self.memory += "insert+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("insert"), "utf-8"))

            self.s.recv(1024)

    def home(self):

        if self.combo.isChecked():

            self.memory += "home+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("home"), "utf-8"))

            self.s.recv(1024)

    def pgup(self):

        if self.combo.isChecked():

            self.memory += "page up+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("page up"), "utf-8"))

            self.s.recv(1024)

    def end(self):

        if self.combo.isChecked():

            self.memory += "end+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("end"), "utf-8"))

            self.s.recv(1024)

    def pgdn(self):

        if self.combo.isChecked():

            self.memory += "page down+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("page down"), "utf-8"))

            self.s.recv(1024)

    def numlock(self):

        if self.combo.isChecked():

            self.memory += "numlock+"

        elif not self.combo.isChecked() and self.memory:

            try:

                self.memory = self.memory[:-1]

                self.s.send(bytes(self.memory, "utf-8"))

                self.s.recv(1024)

                self.memory = ""

            except:

                pass

        else:

            self.s.send(bytes(str("numlock"), "utf-8"))

            self.s.recv(1024)

if \_\_name\_\_ == "\_\_main\_\_":

    app = QApplication(sys.argv)

    ex = App()

    ex.show()

    sys.exit(app.exec())

**ПРИЛОЖЕНИЕ Б**

import socket

import keyboard

import sys

hostMACAddress = str(input("Введите мак адрес"))

port = str(input("Введите порт"))

hostMACAddress = 'e8:48:b8:c8:20:00' # The MAC address of a Bluetooth adapter on the server. The server might have multiple Bluetooth adapters.

port = 7 # 3 is an arbitrary choice. However, it must match the port used by the client.

backlog = 1

size = 1024

s = socket.socket(socket.AF\_BLUETOOTH, socket.SOCK\_STREAM, socket.BTPROTO\_RFCOMM)

s.bind((hostMACAddress,port))

s.listen(backlog)

while 1:

    client, address = s.accept()

    while 1:

        try:

            data = client.recv(size)

        except ConnectionError:

            break

        if not data: break

        if data:

            keys = str(data.decode("utf-8"))

            keyboard.press(keys)

            client.send(data)

            keyboard.release(keys)

            print(bytes.decode(data))

s.close()

sys.exit(0)

**ПРИЛОЖЕНИЕ В**

<?xml version="1.0" encoding="UTF-8"?>

<ui version="4.0">

 <class>Form</class>

 <widget class="QWidget" name="Form">

  <property name="geometry">

   <rect>

    <x>0</x>

    <y>0</y>

    <width>1370</width>

    <height>500</height>

   </rect>

  </property>

  <property name="windowTitle">

   <string>Form</string>

  </property>

  <widget class="QPushButton" name="esc\_vk">

   <property name="geometry">

    <rect>

     <x>0</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>esc</string>

   </property>

  </widget>

  <widget class="QPushButton" name="f1\_vk">

   <property name="geometry">

    <rect>

     <x>110</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>f1</string>

   </property>

  </widget>

  <widget class="QPushButton" name="f2\_vk">

   <property name="geometry">

    <rect>

     <x>165</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>f2</string>

   </property>

  </widget>

  <widget class="QPushButton" name="f3\_vk">

   <property name="geometry">

    <rect>

     <x>220</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>f3</string>

   </property>

  </widget>

  <widget class="QPushButton" name="f4\_vk">

   <property name="geometry">

    <rect>

     <x>275</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>f4</string>

   </property>

  </widget>

  <widget class="QPushButton" name="f5\_vk">

   <property name="geometry">

    <rect>

     <x>385</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>f5</string>

   </property>

  </widget>

  <widget class="QPushButton" name="f6\_vk">

   <property name="geometry">

    <rect>

     <x>440</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>f6</string>

   </property>

  </widget>

  <widget class="QPushButton" name="f7\_vk">

   <property name="geometry">

    <rect>

     <x>495</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>f7</string>

   </property>

  </widget>

  <widget class="QPushButton" name="f8\_vk">

   <property name="geometry">

    <rect>

     <x>550</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>f8</string>

   </property>

  </widget>

  <widget class="QPushButton" name="f9\_vk">

   <property name="geometry">

    <rect>

     <x>660</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>f9</string>

   </property>

  </widget>

  <widget class="QPushButton" name="f11\_vk">

   <property name="geometry">

    <rect>

     <x>770</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>f11</string>

   </property>

  </widget>

  <widget class="QPushButton" name="f12\_vk">

   <property name="geometry">

    <rect>

     <x>825</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>f12</string>

   </property>

  </widget>

  <widget class="QPushButton" name="f10\_vk">

   <property name="geometry">

    <rect>

     <x>715</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>f10</string>

   </property>

  </widget>

  <widget class="QPushButton" name="prtsc\_vk">

   <property name="geometry">

    <rect>

     <x>935</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>prtsc</string>

   </property>

  </widget>

  <widget class="QPushButton" name="scrlk\_vk">

   <property name="geometry">

    <rect>

     <x>990</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>scrlk</string>

   </property>

  </widget>

  <widget class="QPushButton" name="pause\_vk">

   <property name="geometry">

    <rect>

     <x>1045</x>

     <y>0</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>pause</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlock\_vk">

   <property name="geometry">

    <rect>

     <x>1155</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>num</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlockslash\_vk">

   <property name="geometry">

    <rect>

     <x>1210</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>/</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlockstar\_vk">

   <property name="geometry">

    <rect>

     <x>1265</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>\*</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlockminus\_vk">

   <property name="geometry">

    <rect>

     <x>1320</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>-</string>

   </property>

  </widget>

  <widget class="QPushButton" name="tilda\_vk">

   <property name="geometry">

    <rect>

     <x>0</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>`Ё~</string>

   </property>

  </widget>

  <widget class="QPushButton" name="one\_vk">

   <property name="geometry">

    <rect>

     <x>55</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>1!</string>

   </property>

  </widget>

  <widget class="QPushButton" name="two\_vk">

   <property name="geometry">

    <rect>

     <x>110</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>2@&quot;</string>

   </property>

  </widget>

  <widget class="QPushButton" name="three\_vk">

   <property name="geometry">

    <rect>

     <x>165</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>3#№</string>

   </property>

  </widget>

  <widget class="QPushButton" name="four\_vk">

   <property name="geometry">

    <rect>

     <x>220</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>4$;</string>

   </property>

  </widget>

  <widget class="QPushButton" name="five\_vk">

   <property name="geometry">

    <rect>

     <x>275</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>5%</string>

   </property>

  </widget>

  <widget class="QPushButton" name="six\_vk">

   <property name="geometry">

    <rect>

     <x>330</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>6^:</string>

   </property>

  </widget>

  <widget class="QPushButton" name="seven\_vk">

   <property name="geometry">

    <rect>

     <x>385</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>7&amp;?</string>

   </property>

  </widget>

  <widget class="QPushButton" name="eight\_vk">

   <property name="geometry">

    <rect>

     <x>440</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>8\*</string>

   </property>

  </widget>

  <widget class="QPushButton" name="win\_vk">

   <property name="geometry">

    <rect>

     <x>80</x>

     <y>275</y>

     <width>63</width>

     <height>50</height>

    </rect>

   </property>

   <property name="font">

    <font>

     <pointsize>24</pointsize>

    </font>

   </property>

   <property name="text">

    <string>⊞</string>

   </property>

  </widget>

  <widget class="QPushButton" name="nine\_vk">

   <property name="geometry">

    <rect>

     <x>495</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>9(</string>

   </property>

  </widget>

  <widget class="QPushButton" name="zero\_vk">

   <property name="geometry">

    <rect>

     <x>550</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>0)</string>

   </property>

  </widget>

  <widget class="QPushButton" name="minus\_vk">

   <property name="geometry">

    <rect>

     <x>605</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>-\_</string>

   </property>

  </widget>

  <widget class="QPushButton" name="plus\_vk">

   <property name="geometry">

    <rect>

     <x>660</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>+=</string>

   </property>

  </widget>

  <widget class="QPushButton" name="backspace\_vk">

   <property name="geometry">

    <rect>

     <x>714</x>

     <y>55</y>

     <width>161</width>

     <height>50</height>

    </rect>

   </property>

   <property name="font">

    <font>

     <pointsize>16</pointsize>

    </font>

   </property>

   <property name="text">

    <string>←</string>

   </property>

  </widget>

  <widget class="QPushButton" name="ins\_vk">

   <property name="geometry">

    <rect>

     <x>935</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>ins</string>

   </property>

  </widget>

  <widget class="QPushButton" name="home\_vk">

   <property name="geometry">

    <rect>

     <x>990</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>home</string>

   </property>

  </widget>

  <widget class="QPushButton" name="pgup\_vk">

   <property name="geometry">

    <rect>

     <x>1045</x>

     <y>55</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>pg up</string>

   </property>

  </widget>

  <widget class="QPushButton" name="del\_vk">

   <property name="geometry">

    <rect>

     <x>935</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>del</string>

   </property>

  </widget>

  <widget class="QPushButton" name="end\_vk">

   <property name="geometry">

    <rect>

     <x>990</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>end</string>

   </property>

  </widget>

  <widget class="QPushButton" name="pgdn\_vk">

   <property name="geometry">

    <rect>

     <x>1045</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>pg dn</string>

   </property>

  </widget>

  <widget class="QPushButton" name="down\_vk">

   <property name="geometry">

    <rect>

     <x>990</x>

     <y>275</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>↓</string>

   </property>

  </widget>

  <widget class="QPushButton" name="right\_vk">

   <property name="geometry">

    <rect>

     <x>1045</x>

     <y>275</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>→</string>

   </property>

  </widget>

  <widget class="QPushButton" name="left\_vk">

   <property name="geometry">

    <rect>

     <x>935</x>

     <y>275</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>←</string>

   </property>

  </widget>

  <widget class="QPushButton" name="up\_vk">

   <property name="geometry">

    <rect>

     <x>990</x>

     <y>220</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>↑</string>

   </property>

  </widget>

  <widget class="QPushButton" name="rightctrl\_vk">

   <property name="geometry">

    <rect>

     <x>800</x>

     <y>275</y>

     <width>75</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>ctrl</string>

   </property>

  </widget>

  <widget class="QPushButton" name="list\_vk">

   <property name="geometry">

    <rect>

     <x>732</x>

     <y>275</y>

     <width>63</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>list</string>

   </property>

  </widget>

  <widget class="QPushButton" name="fn\_vk">

   <property name="geometry">

    <rect>

     <x>664</x>

     <y>275</y>

     <width>63</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>fn</string>

   </property>

  </widget>

  <widget class="QPushButton" name="rightalt\_vk">

   <property name="geometry">

    <rect>

     <x>596</x>

     <y>275</y>

     <width>63</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>alt</string>

   </property>

  </widget>

  <widget class="QPushButton" name="leftalt\_vk">

   <property name="geometry">

    <rect>

     <x>147</x>

     <y>275</y>

     <width>63</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>alt</string>

   </property>

  </widget>

  <widget class="QPushButton" name="space\_vk">

   <property name="geometry">

    <rect>

     <x>215</x>

     <y>275</y>

     <width>375</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>─</string>

   </property>

  </widget>

  <widget class="QPushButton" name="leftctrl\_vk">

   <property name="geometry">

    <rect>

     <x>0</x>

     <y>275</y>

     <width>75</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>ctrl</string>

   </property>

  </widget>

  <widget class="QPushButton" name="leftshift\_vk">

   <property name="geometry">

    <rect>

     <x>0</x>

     <y>220</y>

     <width>105</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>shift</string>

   </property>

  </widget>

  <widget class="QPushButton" name="capslock\_vk">

   <property name="geometry">

    <rect>

     <x>0</x>

     <y>165</y>

     <width>87</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>caps</string>

   </property>

  </widget>

  <widget class="QPushButton" name="tab\_vk">

   <property name="geometry">

    <rect>

     <x>0</x>

     <y>110</y>

     <width>75</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>tab</string>

   </property>

  </widget>

  <widget class="QPushButton" name="q\_vk">

   <property name="geometry">

    <rect>

     <x>80</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Qй</string>

   </property>

  </widget>

  <widget class="QPushButton" name="w\_vk">

   <property name="geometry">

    <rect>

     <x>135</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Wц</string>

   </property>

  </widget>

  <widget class="QPushButton" name="e\_vk">

   <property name="geometry">

    <rect>

     <x>190</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Eу</string>

   </property>

  </widget>

  <widget class="QPushButton" name="r\_vk">

   <property name="geometry">

    <rect>

     <x>245</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Rк</string>

   </property>

  </widget>

  <widget class="QPushButton" name="t\_vk">

   <property name="geometry">

    <rect>

     <x>300</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Tе</string>

   </property>

  </widget>

  <widget class="QPushButton" name="y\_vk">

   <property name="geometry">

    <rect>

     <x>355</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Yн</string>

   </property>

  </widget>

  <widget class="QPushButton" name="u\_vk">

   <property name="geometry">

    <rect>

     <x>410</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Uг</string>

   </property>

  </widget>

  <widget class="QPushButton" name="i\_vk">

   <property name="geometry">

    <rect>

     <x>465</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Iш</string>

   </property>

  </widget>

  <widget class="QPushButton" name="o\_vk">

   <property name="geometry">

    <rect>

     <x>520</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Oщ</string>

   </property>

  </widget>

  <widget class="QPushButton" name="p\_vk">

   <property name="geometry">

    <rect>

     <x>575</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Pз</string>

   </property>

  </widget>

  <widget class="QPushButton" name="leftbracket\_vk">

   <property name="geometry">

    <rect>

     <x>630</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>х[{</string>

   </property>

  </widget>

  <widget class="QPushButton" name="rightbracket\_vk">

   <property name="geometry">

    <rect>

     <x>685</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>ъ}]</string>

   </property>

  </widget>

  <widget class="QPushButton" name="a\_vk">

   <property name="geometry">

    <rect>

     <x>92</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Aф</string>

   </property>

  </widget>

  <widget class="QPushButton" name="s\_vk">

   <property name="geometry">

    <rect>

     <x>147</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Sы</string>

   </property>

  </widget>

  <widget class="QPushButton" name="d\_vk">

   <property name="geometry">

    <rect>

     <x>202</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Dв</string>

   </property>

  </widget>

  <widget class="QPushButton" name="f\_vk">

   <property name="geometry">

    <rect>

     <x>257</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Fа</string>

   </property>

  </widget>

  <widget class="QPushButton" name="g\_vk">

   <property name="geometry">

    <rect>

     <x>312</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Gп</string>

   </property>

  </widget>

  <widget class="QPushButton" name="h\_vk">

   <property name="geometry">

    <rect>

     <x>367</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Hр</string>

   </property>

  </widget>

  <widget class="QPushButton" name="j\_vk">

   <property name="geometry">

    <rect>

     <x>422</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Jо</string>

   </property>

  </widget>

  <widget class="QPushButton" name="k\_vk">

   <property name="geometry">

    <rect>

     <x>477</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Kл</string>

   </property>

  </widget>

  <widget class="QPushButton" name="l\_vk">

   <property name="geometry">

    <rect>

     <x>532</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Lд</string>

   </property>

  </widget>

  <widget class="QPushButton" name="zh\_vk">

   <property name="geometry">

    <rect>

     <x>587</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>ж;:</string>

   </property>

  </widget>

  <widget class="QPushButton" name="commas\_vk">

   <property name="geometry">

    <rect>

     <x>642</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>э'&quot;</string>

   </property>

  </widget>

  <widget class="QPushButton" name="slash\_vk">

   <property name="geometry">

    <rect>

     <x>697</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>|/\</string>

   </property>

  </widget>

  <widget class="QPushButton" name="z\_vk">

   <property name="geometry">

    <rect>

     <x>110</x>

     <y>220</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Zя</string>

   </property>

  </widget>

  <widget class="QPushButton" name="x\_vk">

   <property name="geometry">

    <rect>

     <x>165</x>

     <y>220</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Xч</string>

   </property>

  </widget>

  <widget class="QPushButton" name="c\_vk">

   <property name="geometry">

    <rect>

     <x>220</x>

     <y>220</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Cc</string>

   </property>

  </widget>

  <widget class="QPushButton" name="v\_vk">

   <property name="geometry">

    <rect>

     <x>275</x>

     <y>220</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Vм</string>

   </property>

  </widget>

  <widget class="QPushButton" name="b\_vk">

   <property name="geometry">

    <rect>

     <x>330</x>

     <y>220</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Bи</string>

   </property>

  </widget>

  <widget class="QPushButton" name="n\_vk">

   <property name="geometry">

    <rect>

     <x>385</x>

     <y>220</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Nт</string>

   </property>

  </widget>

  <widget class="QPushButton" name="m\_vk">

   <property name="geometry">

    <rect>

     <x>440</x>

     <y>220</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>Mь</string>

   </property>

  </widget>

  <widget class="QPushButton" name="leftarrow\_vk">

   <property name="geometry">

    <rect>

     <x>495</x>

     <y>220</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>б&lt;,</string>

   </property>

  </widget>

  <widget class="QPushButton" name="rightarrow\_vk">

   <property name="geometry">

    <rect>

     <x>550</x>

     <y>220</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>ю&gt;.</string>

   </property>

  </widget>

  <widget class="QPushButton" name="dot\_vk">

   <property name="geometry">

    <rect>

     <x>605</x>

     <y>220</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>/?,.</string>

   </property>

  </widget>

  <widget class="QPushButton" name="rightshift\_vk">

   <property name="geometry">

    <rect>

     <x>660</x>

     <y>220</y>

     <width>215</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>shift</string>

   </property>

  </widget>

  <widget class="QPushButton" name="enter\_vk">

   <property name="geometry">

    <rect>

     <x>750</x>

     <y>110</y>

     <width>121</width>

     <height>101</height>

    </rect>

   </property>

   <property name="font">

    <font>

     <pointsize>24</pointsize>

    </font>

   </property>

   <property name="text">

    <string>↵</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlockseven\_vk">

   <property name="geometry">

    <rect>

     <x>1155</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>7 home</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlockeight\_vk">

   <property name="geometry">

    <rect>

     <x>1210</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>8↑</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlocknine\_vk">

   <property name="geometry">

    <rect>

     <x>1265</x>

     <y>110</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>9 pg up</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlockplus\_vk">

   <property name="geometry">

    <rect>

     <x>1320</x>

     <y>110</y>

     <width>50</width>

     <height>105</height>

    </rect>

   </property>

   <property name="text">

    <string>+</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlocksix\_vk">

   <property name="geometry">

    <rect>

     <x>1265</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>6→</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlockfive\_vk">

   <property name="geometry">

    <rect>

     <x>1210</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>5</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlockfour\_vk">

   <property name="geometry">

    <rect>

     <x>1155</x>

     <y>165</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>4←</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlockenter\_vk">

   <property name="geometry">

    <rect>

     <x>1320</x>

     <y>220</y>

     <width>50</width>

     <height>105</height>

    </rect>

   </property>

   <property name="text">

    <string>enter</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlockthree\_vk">

   <property name="geometry">

    <rect>

     <x>1265</x>

     <y>220</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>3 pg dn</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlocktwo\_vk">

   <property name="geometry">

    <rect>

     <x>1210</x>

     <y>220</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>2↓</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlockone\_vk">

   <property name="geometry">

    <rect>

     <x>1155</x>

     <y>220</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>1 end</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlockzero\_vk">

   <property name="geometry">

    <rect>

     <x>1155</x>

     <y>275</y>

     <width>105</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>0 ins</string>

   </property>

  </widget>

  <widget class="QPushButton" name="numlockdel\_vk">

   <property name="geometry">

    <rect>

     <x>1265</x>

     <y>275</y>

     <width>50</width>

     <height>50</height>

    </rect>

   </property>

   <property name="text">

    <string>del ,</string>

   </property>

  </widget>

  <widget class="QLabel" name="port\_label">

   <property name="geometry">

    <rect>

     <x>10</x>

     <y>335</y>

     <width>150</width>

     <height>51</height>

    </rect>

   </property>

   <property name="sizePolicy">

    <sizepolicy hsizetype="Preferred" vsizetype="Preferred">

     <horstretch>32</horstretch>

     <verstretch>32</verstretch>

    </sizepolicy>

   </property>

   <property name="font">

    <font>

     <pointsize>32</pointsize>

    </font>

   </property>

   <property name="text">

    <string>PORT:</string>

   </property>

  </widget>

  <widget class="QLabel" name="mac\_label">

   <property name="geometry">

    <rect>

     <x>240</x>

     <y>335</y>

     <width>120</width>

     <height>51</height>

    </rect>

   </property>

   <property name="sizePolicy">

    <sizepolicy hsizetype="Preferred" vsizetype="Preferred">

     <horstretch>32</horstretch>

     <verstretch>32</verstretch>

    </sizepolicy>

   </property>

   <property name="font">

    <font>

     <pointsize>32</pointsize>

    </font>

   </property>

   <property name="text">

    <string>MAC:</string>

   </property>

  </widget>

  <widget class="QSpinBox" name="port\_spinbox">

   <property name="geometry">

    <rect>

     <x>170</x>

     <y>335</y>

     <width>60</width>

     <height>50</height>

    </rect>

   </property>

   <property name="font">

    <font>

     <pointsize>22</pointsize>

    </font>

   </property>

   <property name="minimum">

    <number>0</number>

   </property>

   <property name="maximum">

    <number>16</number>

   </property>

  </widget>

  <widget class="QPushButton" name="connect\_vk">

   <property name="geometry">

    <rect>

     <x>2</x>

     <y>387</y>

     <width>1365</width>

     <height>110</height>

    </rect>

   </property>

   <property name="font">

    <font>

     <pointsize>32</pointsize>

    </font>

   </property>

   <property name="styleSheet">

    <string notr="true">#connect\_vk

{

background: rgba(196,0,0,255);

}</string>

   </property>

   <property name="text">

    <string>CONNECT</string>

   </property>

  </widget>

  <widget class="QRadioButton" name="comboenable\_rb">

   <property name="geometry">

    <rect>

     <x>1164</x>

     <y>10</y>

     <width>201</width>

     <height>31</height>

    </rect>

   </property>

   <property name="font">

    <font>

     <pointsize>8</pointsize>

    </font>

   </property>

   <property name="text">

    <string>enable combination</string>

   </property>

  </widget>

  <widget class="QTextEdit" name="mac\_textarea">

   <property name="geometry">

    <rect>

     <x>340</x>

     <y>335</y>

     <width>1020</width>

     <height>50</height>

    </rect>

   </property>

   <property name="font">

    <font>

     <pointsize>24</pointsize>

    </font>

   </property>

  </widget>

 </widget>

 <resources/>

 <connections/>

</ui>