УДАЛЕННЫЙ МОНИТОРИНГ РЕАКЦИИ ПОЛЬЗОВАТЕЛЯ НА  
ВИРТУАЛЬНЫЙ КОНТЕНТ

Туев Д.В. 1, Видман В.В. 2

*1 НИ ТПУ, г. Томск (студент гр. 8ИМ31 ОИТ, ИШИТР), e-mail: dvt18@tpu.ru*

*2 НИ ТПУ, г. Томск (старший преподаватель ОИТ, ИШИТР), e-mail: vidman@tpu.ru*

**Аннотация**

В докладе рассматриваются способы подключения датчиков к портативным VR-устройствам и записи данных с них. Был реализован прототип программного комплекса для мониторинга VR-пользователя. Он состоит из приложений для ПК и VR-шлема. Приложение для VR воспроизводит видео и передает данные на ПК. Приложение на ПК синхронно выводит видео и отображает принимаемые данные.

**Ключевые слова:** 360-видео, виртуальная реальность, психиатрия, Android, Unity.

**Введение**

В современном мире, пронизанном быстрыми технологическими изменениями, виртуальная реальность (VR) вступает в область психиатрии, предоставляя уникальные возможности для исследования, диагностики и терапии различных психических расстройств. Виртуальная реальность, представляющая собой симуляцию окружающей среды с использованием компьютерной графики, открывает новые перспективы в области психического здоровья. Применение VR в диагностике психиатрических расстройств создает контролируемую среду для проведения наблюдений и тестирования, что может существенно повысить объективность и репрезентативность результатов.

Самыми доступными на момент выполнения исследования являются портативные VR-системы (состоящие из шлема и пары контроллеров) от компаний Meta[[1]](#footnote-1) и PICO. Они работают под управлением модифицированной версии операционной системы для мобильных устройств Android. Например, ОС на шлемах имеет более строгие ограничения доступа приложений к системе и отсутствие некоторых API, из-за чего не все Android-приложения поддерживаются. Для реализации VR-приложений на низком уровне используются проприетарные API от производителей устройств или открытый API OpenXR [2][7][8]. На высоком уровне часто используются коммерческие игровые движки, предоставляющие инструментарий для разработки трехмерных приложений, игр и симуляций.

Целью работы является проектирование и разработка приложения для удаленного мониторинга реакции пользователя на виртуальный контент.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. провести анализ способов подключения медицинских датчиков к портативным VR-устройствам;
2. спроектировать прототип с учетом медицинских и технологических требований;
3. реализовать прототип.

**Подключение датчиков к устройству**

Для проектирования приложения требуется рассмотреть способы подключения устройств к VR-системам.

*Подключение напрямую к устройству*

Android предоставляет API для подключения внешних устройств по Bluetooth. Это позволяет подключаться к датчикам, поддерживающим Bluetooth или Bluetooth Low Energy. Данная технология поддерживается в микроконтроллере ESP32 и многих смарт-часах. Вследствие ограниченности версий Android, использующихся на VR-шлемах, связь по Bluetooth может работать некорректно.

Альтернативным способом является подключение по USB и использование OTG-кабеля.   
Для Android существует библиотека, реализующая драйверы популярных USB-UART преобразователей [1]. Данный способ позволяет обмениваться данными с любыми современными микроконтроллерами или с ПК. Недостатком данного способа является наличие кабеля, способного повредить USB-порт на VR-шлеме при резком вращении головы пользователя.

*Подключение по локальной сети*

Любое приложение для Android имеет доступ к локальной сети устройства, даже на ограниченных версиях Android для VR-шлемов. Приложение может создавать TCP/IP-сокеты для подключения к другим хостам (TCP-клиент) или запускать TCP-сервер, к которому могут подключаться другие хосты.

*Подключение по локальной сети через посредника*

Данный способ является расширением предыдущего. На посреднике (ПК или смартфоне) запускается TCP-сервер или клиент. Посредник же подключается к устройству посредством Bluetooth, USB или иного способа, принимает данные с датчика и передает их по локальной сети на VR-шлем.

*Подключение по сети Интернет*

В данном случае датчик передает информацию на сервер, расположенный в облаке. Программа на VR-шлеме подключается к этому серверу по сети Интернет и принимает данные с датчика. Данный способ имеет место быть только если датчик не позволяет передавать данные по локальной сети напрямую, а только поддерживает связь со сторонним сервером, управляемым производителем датчика. Медицинские данные должны быть надежно защищены от доступа сторонних лиц, что в данном случае невозможно обеспечить.

**Функции прототипа**

Прототип приложения должен обладать следующим набором функций.

1. Прототип должен проигрывать контент в формате видео.
2. Прототип должен в VR-шлеме воспроизводить видео.
3. Прототип должен на ПК показывать изображение, которое видит VR-пользователь.
4. Прототип должен сохранять данные в файл.
5. Прототип должен работать на Meta Quest 2.
6. Прототип должен работать на PICO 4.

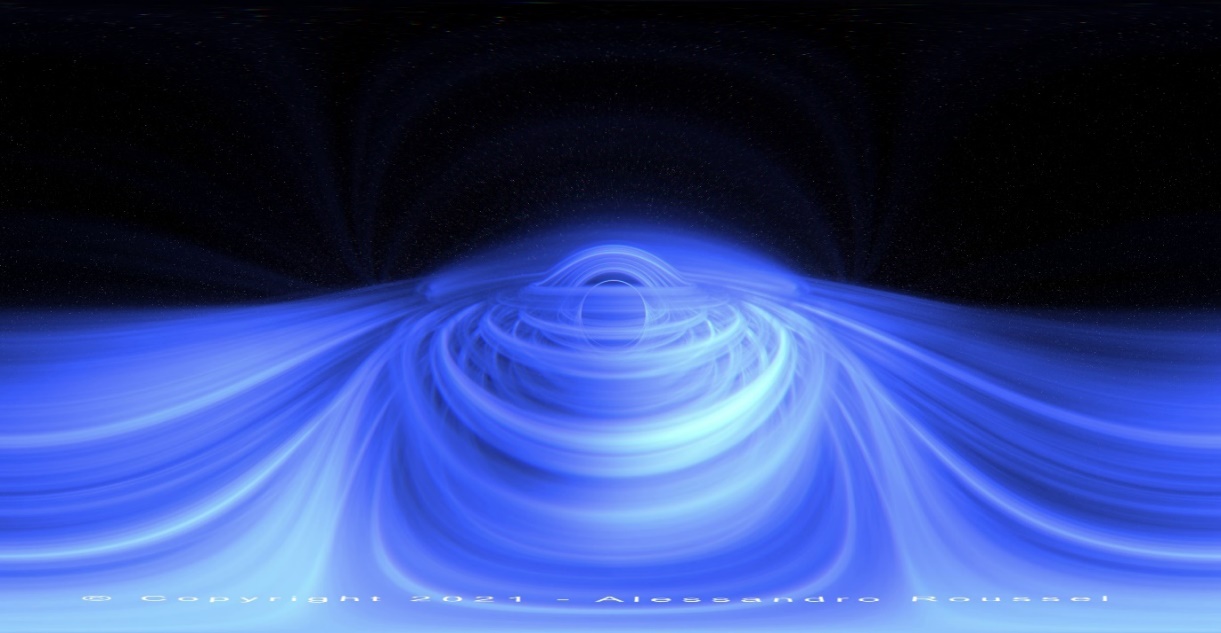
**Форматы видео**

Для реализации прототипа требуется исследовать, какие форматы видео используются в виртуальной реальности.

360-видео – это видео в сферическом формате. Оно может быть просмотрено в любом направлении. Для видео в формате 360 существует несколько способов проекции сферы на плоскую картинку.

1. Кубическая текстура (cubemap). Каждый кадр делится на сетку из 6 прямоугольников.   
   Каждая ячейка этой сетки является одной из граней куба.
2. Равнопромежуточная проекция [3] (equirectangular projection). Сфера проецируется на прямоугольник по определенной формуле. Пример представлен на рисунке 1.

YouTube также поддерживает формат «mesh», при использовании которого в видеофайл записывается 3D-модель с произвольными позициями вершин и UV-координатами.

  
*Рис 1. Кадр 360-видео в формате равнопромежуточной проекции  
Источник: Falling into a realistic Black Hole (VR 360°)  
Автор: ScienceClic English,* [*https://youtu.be/17tEg\_uTF\_A*](https://youtu.be/17tEg_uTF_A)

Формат VR180 был разработан YouTube. В отличие от 360-видео, его угол обзора – полусфера, но видео хранит изображение для двух глаз, тем самым предоставляя ощущение глубины. Спецификация VR180 [11] поддерживает только формат «mesh», описанный выше. Пример кадра в этом формате представлен на рисунке 2.

  
*Рис. 2. Кадр VR180-видео  
Источник: Meanwhile 4D  
Автор: TomSka,* [*https://youtu.be/3s2qAFfpJf4*](https://youtu.be/3s2qAFfpJf4)

**Фоновая запись данных**

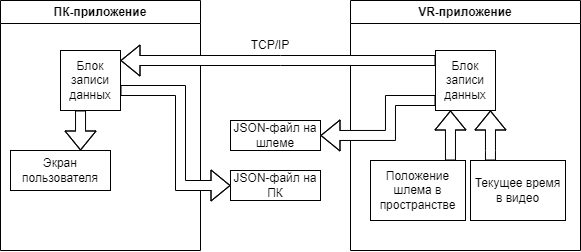
В ходе исследования возникла гипотеза: возможно разработать приложение, которое будет в фоновом режиме записывать данные, пока на переднем плане работает игра или VR-плеер. Данное приложение, при срабатывании предварительно настроенного триггера, делает снимок экрана, и тем самым получает направление взгляда. Были найдены следующие варианты реализации: WebXR-плеер [4], запуск второго VR-приложения на фоне, перехват через OpenXR API Layers [6], служба Android для создания скриншотов.

В результате анализа каждого варианта реализации самым перспективным оказался последний вариант. WebXR-плееров с открытым исходным кодом и активной поддержкой найдено не было. Ни Meta Quest, ни PICO не позволяют запускать несколько VR-приложений одновременно и не поддерживают OpenXR API Layers.

Для проверки гипотезы был создан прототип варианта реализации с использованием службы Android. Данный прототип после запуска создает снимки экрана каждые 5 секунд и сохраняет во внутреннее хранилище. В результате разработки прототипа было выявлено, что Meta Quest 2 не поддерживает Media Projection API [5], из-за чего приложение-прототип закрывается с ошибкой при попытке запустить службу. Но при тестировании на PICO 4 прототип запустился и функционировал корректно. Однако, такая реализация не имеет доступа к VR-датчикам (например, направлению взгляда глаз).

**Система записи данных**

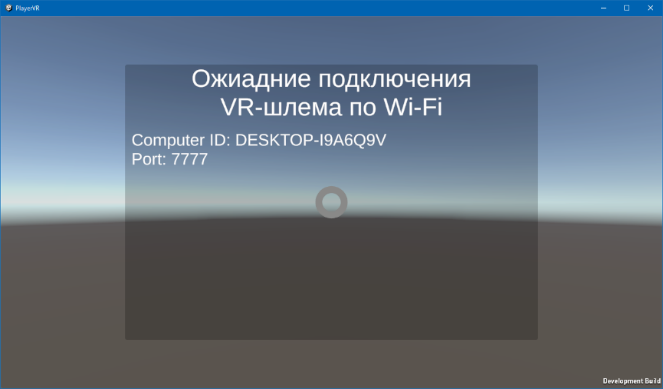
Одними из функций прототипа являются запись данных в файл и их отображение.Дляэтого прототип программы должен иметь систему записи данных, собирающую информацию со всех настроенных датчиков. Реализованная система предоставляет интерфейсы для приема данных с датчиков (например, из VR-системы или по Bluetooth) и их последующей передачи (например, в файл или по сети). Наличие синхронизации изображения подразумевает клиент-серверную архитектуру и передачу данных от VR-шлема к ПК. Для этого и на ПК, и на VR-шлеме создается по одному экземпляру блока записи данных. По TCP/IP данные передаются с VR-шлема на ПК и выводятся на экране. Диаграмма обмена данными между приложениями представлена на рисунке 3.

  
*Рис. 3. Диаграмма обмена данными между устройствами*

**Результаты**

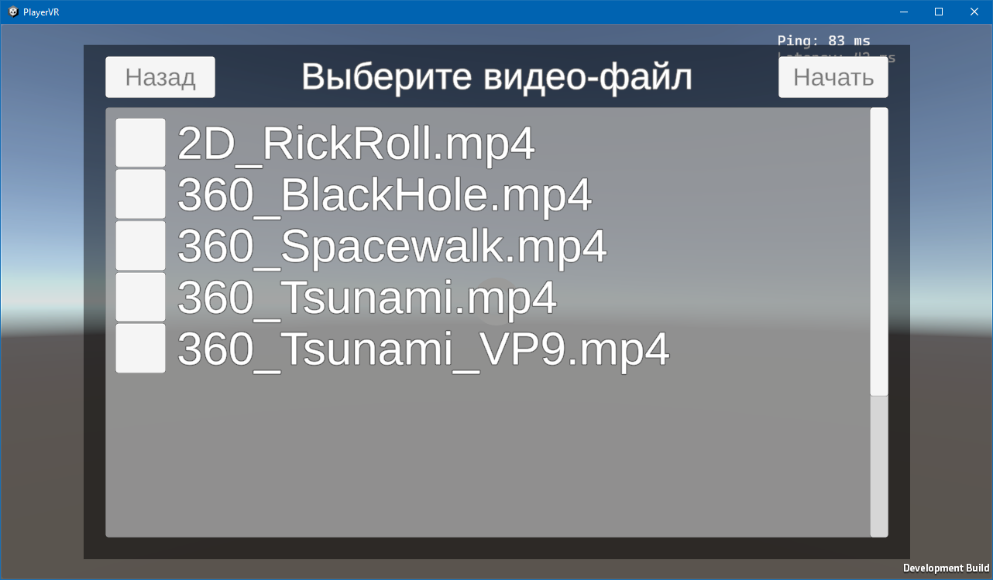
Для реализации прототипа был выбран движок Unity. Он бесплатен для личного и академического использования, поддерживает VR на ПК и портативных шлемах, а также и имеет компонент для воспроизведения видео в качестве текстуры. Для декодирования видео из файла в текстуру используется компонент Video Player из Unity. Поддерживается видео в контейнере MP4 и кодеке H.264.

При запуске ПК-приложения, оно запускает сервер и ожидает подключения VR-приложения (рис. 4). Для реализации сетевого подключения была использована библиотека Netcode for GameObjects [1]. VR-приложение выводит список всех найденных серверов (рис. 5).

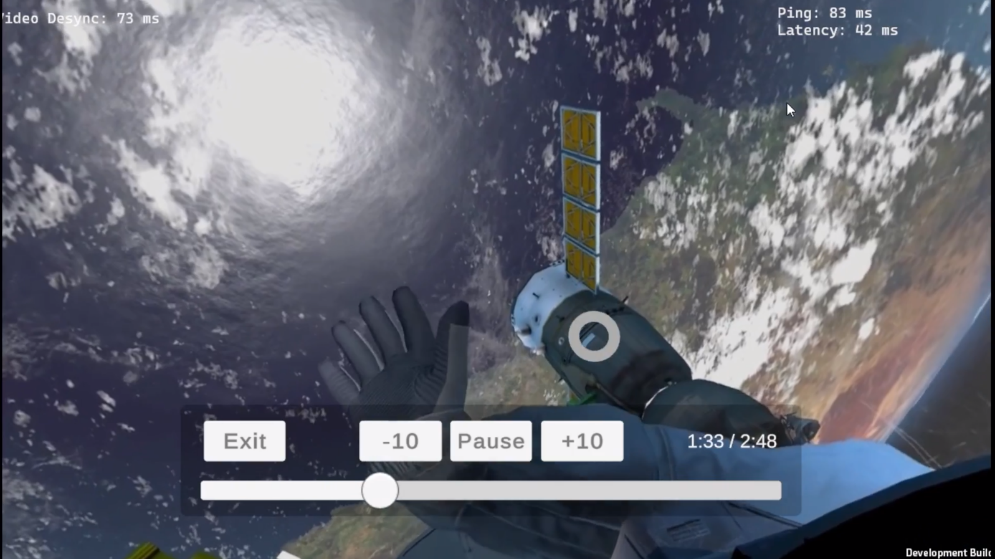
  
*Рис. 4. Снимок стартового экрана ПК-приложения*

  
*Рис. 5. Снимок стартового экрана VR-приложения*

После подключения VR-шлема, пользователю предоставляется список видео, доступных для проигрывания (рис. 6). После выбора видео, оно проигрывается и на VR-шлеме, и на ПК.

   
*Рис. 6. Список видео в ПК-приложении*

Во время проигрывания видео, направление взгляда выводится пользователю ПК в виде кольца (рис. 7).

   
*Рис. 7. Отображение видео в ПК-приложении*

**Заключение**

На начальном этапе работы был проведен анализ способов подключения различных датчиков к портативным VR-системам. Было выявлено, что наиболее универсальными способами подключения является подключение по локальной сети устройства напрямую или с помощью посредника – ПК, который принимает данные от датчика и передает их по локальной сети.

В ходе проектирования и разработки прототипа приложения возникла гипотеза, что существует возможность разработать фоновую службу, которая будет записывать необходимые данные. Был разработан прототип, подтверждающий гипотезу. Однако, этот прототип имеет два существенных ограничения: он работает только на VR-шлемах PICO и не имеет доступа к системе пространственного отслеживания, а значит не имеет возможность фиксировать ориентацию шлема в пространстве и направление взгляда.

В результате выполнения работы был спроектирован и реализован прототип приложения для воспроизведения VR-контента в видеоформате 360 и фоновой фиксации медицинских показателей. Прототип состоит из приложения для ПК и приложения для VR-шлема. И ПК, и VR-шлем, синхронно воспроизводят одно и то же видео. VR-приложение отправляет данные на ПК, который отображает их на экране.

**Список использованных источников**

1. About Netcode for GameObjects – Unity3D [Электронный ресурс]. – Режим доступа: <https://docs-multiplayer.unity3d.com/netcode/current/about/> (дата обращения: 29.03.2024)
2. About OpenXR & Unity – PICO developer [Электронный ресурс]. – Режим доступа: <https://developer-global.pico-interactive.com/document/unity-openxr/> (дата обращения: 29.03.2024)
3. Equirectangular Projection [Электронный ресурс]. – Режим доступа: <https://mathworld.wolfram.com/EquirectangularProjection.html> (дата обращения: 29.03.2024)
4. Getting Started with PWAs\*\* [Электронный ресурс]. – Режим доступа: <https://developer.oculus.com/documentation/web/pwa-gs/> (дата обращения: 29.03.2024)
5. Media Projection API – Android [Электронный ресурс]. – Режим доступа: <https://developer.android.com/reference/android/media/projection/MediaProjection.html> (дата обращения: 29.03.2024)
6. OpenXR Loader – Design and Operation [Электронный ресурс]. – Режим доступа: <https://registry.khronos.org/OpenXR/specs/1.0/loader.html> (дата обращения: 29.03.2024)
7. OpenXR Specification [Электронный ресурс]. – Режим доступа: <https://registry.khronos.org/OpenXR/specs/1.0/html/xrspec.html> (дата обращения: 29.03.2024)
8. OpenXR Support for Meta Quest and Quest 2[[2]](#footnote-2)\*\* [Электронный ресурс]. – Режим доступа: <https://developer.oculus.com/documentation/native/android/mobile-openxr/> (дата обращения: 29.03.2024)
9. Unity User Manual [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/Manual/index.html> (дата обращения: 29.03.2024)
10. usb-serial-for-android [Электронный ресурс]. – Режим доступа: <https://github.com/mik3y/usb-serial-for-android> (дата обращения: 29.03.2024)
11. VR180 Video Format Specification [Электронный ресурс]. – Режим доступа: <https://github.com/google/spatial-media/blob/master/docs/vr180.md> (дата обращения: 29.03.2024)

1. *компания Meta Platforms Inc. внесена в реестр экстремистских организаций, ее деятельность в России признана экстремистской деятельностью* [↑](#footnote-ref-1)
2. *веб-сайт Oculus принадлежит компании Meta Platforms Inc, которая внесена в реестр экстремистских организаций, ее деятельность в России признана экстремистской деятельностью* [↑](#footnote-ref-2)