

Разработка программы для системы реабилитации пациентов с нарушениями опорно-двигательного аппарата

Г. А. Машевский

*Санкт-Петербургский государственный
электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)*

Aniket@list.ru

Н. А. Гайдей

*Санкт-Петербургский государственный
электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)*

nikolai@gaidei.ru

Аннотация. Данная работа посвящена разработке программы для системы виртуальной реальности, которая должна работать совместно с тренажером для реабилитации пациентов с нарушениями функций опорно-двигательного аппарата.

Разработанная программа для системы реабилитации может использоваться в медицинских учреждениях для повышения эффективности лечебного процесса, а также способствовать более быстрому выздоровлению пациентов. Также полученная система может быть дополнительно оснащена датчиками контроля биологических показателей реабилитируемого, для более тонкого управления процессом лечения.

Ключевые слова: реабилитация; тренажер; программа; виртуальная реальность

I. ВВЕДЕНИЕ

Одной из ключевых тенденций современной медицины становится все более значительная роль реабилитации как завершающего этапа лечебного процесса. На сегодняшний день среди специалистов в области здравоохранения сложилось общее мнение о том, что недостаточно просто сохранить жизнь и здоровье пациента – важно также обеспечить его возвращение к полноценному образу жизни. В результате реабилитация сформировалась в самостоятельную отрасль со своими методами и подходами к решению поставленных задач. Разработано множество специализированных технических средств, и эта сфера продолжает стремительно развиваться. Однако остается актуальной проблема повышения мотивации пациентов и их вовлеченности в процесс восстановления. [1] Занятия на реабилитационных тренажерах могут показаться однообразными, а пациенты часто испытывают трудности в оценке правильности своих движений.

Параллельно с этим в последние годы наблюдается активное внедрение технологий дополненной и виртуальной реальности в повседневную жизнь. Хотя изначально они применялись преимущественно в развлекательной сфере, со временем их начали использовать и в других областях — образовании, промышленности, архитектуре. Практика показывает, что эти технологии повышают вовлеченность пользователей в различные процессы, что привело к популяризации концепции т. н. геймификации.

Виртуальная реальность постепенно находит применение и в медицине, в том числе в сфере реабилитации. Однако уровень распространения таких систем пока остается недостаточным. Чаще всего они создаются как комплексные решения, что увеличивает их стоимость и снижает доступность. [8] В то же время на практике используется большое количество простых и недорогих реабилитационных тренажеров, которые останутся востребованными в ближайшие годы. Это создает потребность в повышении их эффективности путем вовлечения пациентов в процесс восстановления.

Одним из перспективных решений этой проблемы является интеграция реабилитационных тренажеров с доступными устройствами виртуальной реальности, такими как VR-очки или VR-шлемы. Для этого потребуется оснащение тренажеров датчиками, позволяющими фиксировать движения пациента, а также разработка специализированного программного обеспечения, которое будет представлять полученные данные в игровой форме. [2]

Наиболее эффективное использование VR-технологий в реабилитации связано с восстановлением подвижности конечностей. Благодаря виртуальной реальности пациенты могут выполнять такие упражнения, как ходьба, подъем по лестнице или приседания, в безопасной цифровой среде. Это дает возможность избежать риска травм, повторять движения необходимое количество раз, а также регулировать сложность упражнений в соответствии с потребностями пациента.

Кроме того, VR способствует развитию координации и равновесия, что особенно важно при восстановлении после травм и заболеваний. Виртуальные тренировки способствуют формированию новых нейронных связей между мозгом и телом, ускоряя процесс реабилитации. Использование подобных технологий позволяет сократить сроки восстановления, которые при стандартных методиках могут занимать месяцы и даже годы. В результате пациент получает наглядную обратную связь, а тренировки становятся более увлекательными и эффективными. [3]

Актуальность данной темы обусловлена необходимостью повышения эффективности реабилитационного процесса.

Целью данной работы является разработка программы для VR-шлема, позволяющей визуализировать движения, выполняемые пациентом на реабилитационном тренажере.

Основные задачи исследования включают:

- Анализ существующих методик и тренажеров, используемых для реабилитации;
- Исследование движений, выполняемых на тренажерах, и создание сценариев, адаптированных для применения в реальной жизни;
- Разработка механизма подключения VR-программы к тренажеру для сбора данных о движениях пациента и синхронизации визуализации с реальным процессом;
- Организация сбора данных о пациенте для последующего анализа врачами;
- Обеспечение обработки информации о ходе тренировок с последующим ее представлением пациенту в наглядной форме.
- Объектом исследования является VR-программа, предназначенная для визуализации движений, выполняемых на реабилитационном тренажере.

Предмет исследования – создание системы обратной связи для пациентов в процессе тренировок с применением технологий виртуальной реальности.

II. РАЗРАБОТКА ПРОГРАММЫ ДЛЯ СИСТЕМЫ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

A. Разработка концепции и структуры системы

Виртуальная реальность (VR) активно используется в тренировках на тренажерах, повышая когнитивные способности, мотивацию и продуктивность. Она создает интерактивные среды, имитирующие реальные условия, что улучшает освоение новых упражнений. Особенно это актуально для беговых дорожек, велотренажеров и других устройств. [4]

VR также снижает риск травм: пользователи могут безопасно отрабатывать технику движений, прежде чем приступить к реальным нагрузкам. [7] Интерактивные тренировки делают процесс более увлекательным, позволяя, например, бегать по виртуальным пейзажам, что повышает вовлеченность и комфорт. [5]

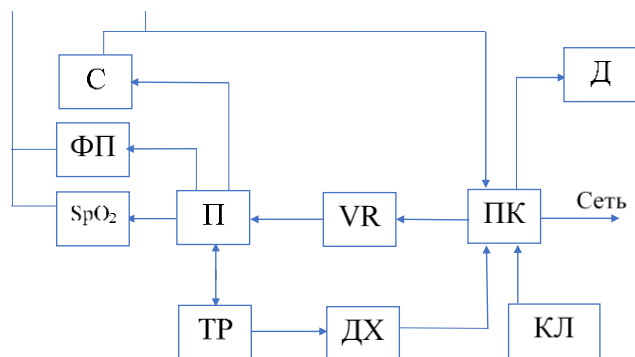


Рис. 1. Структурная схема биотехнической системы

ДХ – датчик Холла; С – спирометр; ФП – фотоплетизмограф; SpO2 – пульсоксиметр; КЛ – клавиатура; П – пациент; VR – устройство виртуальной

реальность; ТР – тренажер; Д – дисплей; ПК – персональный компьютер.

B. Выбор тренажера и контроллера

По рекомендации врачей ФГБУ НМИЦ ТО им. Р. Р. Вредена Минздрава России был выбран тренажер Total Abdominal для тренировки мышц пресса. Он оснащен подголовниками и ручками, поддерживает вращательные движения, укрепляет мышцы брюшной стенки, улучшает координацию и гибкость позвоночника.

Это оборудование используется для реабилитации после травм опорно-двигательной системы, так как снижает нагрузку на позвоночник и предотвращает повторные травмы. Оно также укрепляет мышцы корпуса, улучшая поддержку внутренних органов.

Для адаптации программы к пользователю необходимо точно фиксировать данные о перемещении частей тренажера. [6] Для этого был выбран датчик Холла, фиксирующий движения без задержек и искажений.

C. Применение датчика Холла

Датчик Холла установлен на неподвижной части тренажера, а магнит – на подвижной. Это устройство воспринимается Unreal Engine как аналог джойстика, позволяя передавать данные без сложных программных преобразований.

Основные преимущества датчика:

- Высокая точность измерений.
- Устойчивость к пыли, грязи, воде.
- Надежность и долговечность.
- Работа в широком диапазоне температур.

Эти характеристики делают его предпочтительным решением по сравнению с оптическими и электромеханическими датчиками. В геймпадах датчики Холла измеряют угол поворота и силу нажатия, а в тренажерах – фиксируют движения для точного контроля программы.

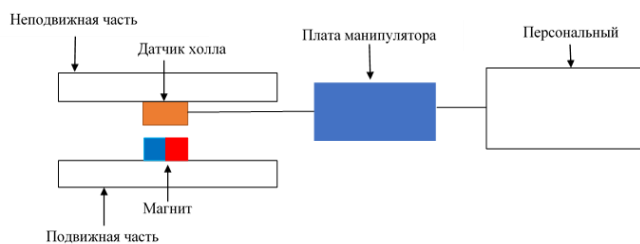


Рис. 2. Схема съема данных с тренажера

D. Дополнение системы средствами мониторинга биологических показателей пациента

В данном разделе рассматривается система медицинских датчиков для контроля ключевых биологических параметров пациента. Мониторинг основан на измерении:

- уровня кислорода в капиллярной крови,
- наполненности мелких сосудов кровью,
- функционального состояния дыхательной системы через жизненную емкость легких.

Для обеспечения точности и стабильности работы система включает несколько специализированных датчиков, процессор обработки данных и интерфейс для вывода информации на дисплей. Такой подход позволяет медицинскому персоналу отслеживать важные показатели в режиме реального времени, что особенно важно в процессе реабилитации.

Дополнительно возможно подключение модуля для записи данных на внешние носители, что позволит анализировать динамику изменений состояния пациента.

Для измерения кислорода в крови в систему интегрирован пульсоксиметр, оптимальным вариантом является прибор в формате зажима («прищепка»). Измерение проводится путем фиксации пальца пациента внутри устройства и активации сенсора.

Контроль наполнения сосудов кровью осуществляется с помощью фотоплетизмографа, который анализирует изменения объема крови в периферических сосудах.

Оценка работы дыхательной системы во время упражнений производится с применением компактного спирометра, фиксирующего параметры легочной функции. [9]

Е. Алгоритм работы программы

Программное обеспечение реализовано на базе Unreal Engine 5.1.1 – платформы для создания фотореалистичных графических сред. Встроенная система визуального скриптинга Blueprint позволяет ускорить разработку, упростить внесение изменений и адаптировать программу под современные технологии.

Blueprint – графический язык программирования в Unreal Engine, позволяющий создавать интерактивные события без написания кода на C++. Работа основана на системе узлов, соединенных линиями, где каждый узел отвечает за определенное действие (например, анимацию или перемещение объекта).

Использование Blueprints позволяет:

- оперативно вносить правки в логику работы программы;
- тестировать изменения без перезапуска приложения;
- адаптировать систему под новые требования.

Алгоритм также включает механику загрузки визуальных объектов в Unreal Engine на уровне кода. При запуске приложение подгружает компонент класса Level, отвечающий за начальное окружение, включая интерфейс и стартовую сцену для пользователя.

Автоматический перевод Blueprint в C++ позволяет повысить производительность работы приложения и сделать его более оптимизированным. Unreal Engine использует компилятор Blueprints для генерации исходного кода на C++. Этот код можно использовать в дальнейшем при создании более сложных проектов и при их оптимизации.

Кроме того, перевод Blueprint в C++ может значительно сократить время на разработку в случае, когда функционал уже реализован в других частях кода. В этом случае достаточно просто перенести узлы Blueprints в нужный компонент класса C++.



Рис. 3. Алгоритм работы программы

Система Blueprint представляет собой набор узлов, каждый из которых является визуализированным элементом кода. Узлы содержат переменные и функции, а их соединение формирует целостную структуру программы. Хотя физическое расположение узлов не влияет на функциональность, порядок их соединения определяет логику выполнения.

Использование Blueprint позволяет разделять программу на независимые компоненты, которые обращаются друг к другу по мере необходимости. Такой подход снижает нагрузку на центральный процессор и улучшает производительность, обеспечивая поддержку на устройствах с разной вычислительной мощностью.

Программа состоит из четырех ключевых компонентов:

- BP_Spawner – отвечает за создание объектов в сцене.
- BP_Tunnel – управляет перемещением объектов в виртуальном пространстве.
- BP_UI – обрабатывает отображение пользовательского интерфейса.
- tester – используется для тестирования функций.

Настройка игрового режима в Unreal Engine

Для корректной работы проекта разработан GameMode, определяющий параметры управления пользователем и отображения окружения. В Unreal Engine GameMode задает правила работы виртуального пространства, включая порядок загрузки уровней, условия завершения сессии и обработку действий пользователя.

Разработанный VRGameMode включает:

- обработку данных с контроллеров VR-устройств,
- управление камерой, обеспечивающей отображение сцены,
- настройку физической модели взаимодействия с объектами.

Этот компонент определяет ключевые параметры сессии, такие как игровая логика, интерфейс и динамические параметры. Например, GameMode может регулировать длительность сессии, изменять погодные условия или управлять количеством активных элементов в сцене.

Запуск программы в Unreal Engine

Алгоритм работы проекта основан на пошаговой загрузке компонентов:

- Загрузка ресурсов – модели, текстуры, звуковые эффекты.
- Подготовка сцены – размещение объектов в виртуальном пространстве.
- Создание динамических элементов – настройка объектов, управляемых Blueprint.
- Обработка входных данных – управление элементами на основе пользовательского ввода.
- Запуск сцены – отображение интерфейса и начало взаимодействия.

Для инициализации используется Blueprint Player Start, который определяет стартовые координаты пользователя и окружения. Этот алгоритм встроен в движок и не требует ручного добавления объектов в код, что делает его удобным для проектов различной сложности – от простых визуализаций до полноценных VR-приложений.

Для оптимизации вычислительных ресурсов и унификации виртуального пространства, независимо от продолжительности раунда и количества подходов, был разработан алгоритм циклического формирования «тоннеля», по которому передвигается пользователь во время работы программы. Реализация основана на отслеживании положения объектов в пространстве и фиксации пересечения их коллизий (физических границ виртуальных элементов).

Так как приложение не предполагает сложных взаимодействий между объектами, коллизии были заданы с использованием примитивных форм. При этом визуальные характеристики моделей могут быть любыми, независимо от сложности их геометрии.

Механизм зацикливания построен на взаимодействии двух ключевых структурных элементов: BP_Spawner и BP_Tunnel. Каждый из них имеет физические параметры, включая коллизию. Для определения момента прохождения объекта Pawn через «тоннель» используется проверка пересечения границ объектов BP_Spawner и BP_Tunnel.

Так как объект BP_Tunnel ограничен по координате X, после завершения пересечения его границ с BP_Spawner, программа автоматически создает новый сегмент тоннеля перед предыдущим. Когда BP_Spawner полностью входит в новую область тоннеля, старый сегмент удаляется. Этот процесс позволяет создать бесконечную структуру «тоннеля» и адаптировать его в зависимости от заданных параметров упражнения.

Алгоритм зацикливания реализован с помощью следующих узлов Blueprint:

- On Component Begin Overlap (Box) – фиксирует пересечение объектов.
- On Component Begin Overlap (Box1) – аналогично, отслеживает другую часть тоннеля.
- Cast to BP_Spawner – приведение к нужному классу объекта.
- Target is Actor (Get Actor Transform) – получение координат объекта.
- Target is Actor (Destroy Actor) – удаление устаревших объектов.
- SpawnActor BP Tunnel – генерация нового сегмента тоннеля.

Таким образом, реализованная система обеспечивает плавное и непрерывное движение пользователя по виртуальному пространству, минимизируя нагрузку на аппаратные ресурсы.

Входные данные в программе представлены параметрами времени и количества упражнений, которые устанавливает врач. Эти параметры хранятся в переменных типа float, что позволяет изменять их динамически во время выполнения программы. Такой подход обеспечивает обновление значений после завершения одного действия и перехода к следующему.

Для обозначения начала и окончания раунда используется переменная типа boolean, которая регистрирует смену состояний и динамически изменяет свое значение, обеспечивая корректное управление процессом выполнения программы.

При активном использовании узла Blueprint Event Tick возможно снижение производительности приложения, особенно если он применяется в нескольких структурных элементах. Однако в данном проекте нет большого количества одновременно выполняемых процессов, поэтому использование Event Tick не оказывает значительного влияния на быстродействие программы. Это, в свою очередь, позволяет снизить минимальные системные требования к оборудованию, на котором будет запускаться данное приложение в медицинском учреждении.

Для хранения переменных, доступ к которым требуется в разное время выполнения программы и в различных средах исполнения, используется специально созданный компонент класса Game Instance.

Game Instance в Unreal Engine представляет собой объект, который создается при запуске программы и остается в памяти до ее завершения. Он выполняет функцию глобального контейнера для хранения данных и настроек, обеспечивая доступ к информации из любых частей программы.

Этот компонент может использоваться, например, для хранения параметров выполнения визуализации, таких как уровень сложности, громкость звука и другие настройки. Кроме того, он позволяет управлять функциями, необходимыми на протяжении всего выполнения программы, например, фиксировать текущее состояние пользователя в виртуальном пространстве.

Game Instance также может взаимодействовать с различными системами Unreal Engine, включая анимацию, физику, звуковые эффекты и другие модули. Он играет ключевую роль в передаче данных между уровнями виртуальной среды, при этом практически не влияя на производительность системы. Благодаря этому Game Instance является эффективным инструментом для работы с проектами любого масштаба в Unreal Engine.

Нода Get Widget предназначена для получения доступа к компоненту Widget, который был создан в среде Blueprint. Она позволяет управлять свойствами виджетов, включая их положение, размер, цвет и отображаемую информацию.

Для корректного использования Get Widget необходимо предварительно создать компонент Widget в Unreal Editor и добавить его в пользовательский интерфейс. Этот процесс выполняется в режиме конструктора интерфейса в окне View Port, где доступные элементы размещаются на панели виджетов. После сохранения компонент становится частью виджета и отображается в виртуальном пространстве в процессе работы программы.

Алгоритм работы данного механизма основан на переменных типа float и string. Их начальные значения устанавливаются врачом при составлении плана тренировки или задаются автоматически. Далее переменные string передаются в интерфейс компонента WB_UI, где их отображение осуществляется через преобразование в текст с помощью Blueprint To Text (Float).

Blueprints VRspectator – это компонент Unreal Engine, обеспечивающий возможность наблюдения за

виртуальной средой в режиме реального времени с помощью VR-устройств или стандартного экрана компьютера. Его работа основана на передаче визуальной информации из VR-пространства на дополнительный экран с использованием многопоточности и взаимодействия с пользовательским интерфейсом.

Настройка Blueprints VRspectator начинается с создания специального компонента, который интегрируется в проект. Этот процесс включает в себя определение параметров с использованием программ Blender или Maya. Параметры позволяют выбрать, какие именно визуальные элементы должны транслироваться из VR-среды и каким образом они будут представлены на дополнительном экране.

Дополнительно Blueprints VRspectator можно улучшить с помощью внешних инструментов, таких как Oculus VR или OpenVR, которые обеспечивают точное отслеживание перемещения пользователя в виртуальном пространстве.

Функциональные возможности VRspectator варьируются в зависимости от выбранной платформы. Для реализации проекта была выбрана Meta Quest 2, как наиболее совместимая среда, что подтверждено данными из табл. 1.

В рамках данного проекта Blueprint VRspectator активируется только при наличии подключенного и корректно работающего VR-устройства, а также заданных параметров поля зрения.

Параметры включения и отключения режима наблюдателя можно настраивать непосредственно в программе. Для этого в VR-приложение добавляются две кнопки: «Активировать режим наблюдателя» и «Отключить режим наблюдателя». После нажатия первой, приложение переключается в режим просмотра, предоставляя пользователю возможность наблюдать за процессом со стороны. В этом режиме доступна настройка положения камеры для обзора объектов внутри виртуальной среды. Однако в рамках данного проекта такая функция не используется, чтобы избежать путаницы для пациента. Вместо этого режим установлен по умолчанию и управляется переменной boolean.

F. Настройка поля зрения

Field Of View (угол обзора) определяет видимый пользователем сектор пространства внутри VR-приложения. Его значение варьируется в пределах 180–360 градусов.

Уменьшенный угол обзора повышает детализацию отображаемого объекта.

Расширенный угол обзора позволяет видеть больше объектов в сцене.

Настройка угла обзора выполняется через соответствующую ноду, после чего ее значение передается в ноду, фиксирующую текущие параметры.

Параметр Set Relative Rotation позволяет менять положение камеры наблюдателя относительно заданной точки. Например, если камера установлена над объектом, пользователь может изменять ее ориентацию для лучшего обзора.

ТАБЛИЦА I.

№	Режим	Meta Quest 2	Valve Index	PS VR 2	Описание
1	Disabled				Для достижения наилучшей производительности на HMD в этом режиме отключается вывод Spectator Screen.
2	SingleEyeLetterboxed				Этот режим предназначен в основном для отладочных целей, показывая на экране только SingleEye в формате letterbox.
3	Distorted				Этот режим поддерживается только компанией Oculus. В частности, это режим отладки, специфичный для Oculus, показывающий отклонения по палитре и т.д.
4	SingleEye				Как и режим "Undistorted", это режим отладки только для SingleEye. Поскольку этот режим растягивает сцену, он может быть полезен для выявления небольших артефактов в сцене.
5	SingleEyeCroppedToFill				В этом режиме отображение для одного глаза обрезается так, чтобы заполнить весь SingleEye.
6	Texture				В этом режиме отображается полноэкранный вид указанного UTexture
7	TexturePlusMirror				В этом режиме текстура отображается в одном окне для рендеринга, а изображение – в другом.

В данном проекте динамическая настройка этого параметра в процессе работы не требуется, поэтому он устанавливается заранее. При необходимости оператор может изменить его перед запуском программы.

В ходе проведенного исследования и разработки программного обеспечения для системы реабилитации пациентов с нарушениями опорно-двигательного аппарата было показано, что применение технологий виртуальной реальности способствует созданию биологической обратной связи. [10] Это позволяет пациенту получать информацию о своем прогрессе в игровой форме, что повышает его вовлеченность в процесс реабилитации и, потенциально, увеличивает ее эффективность.

III. ЗАКЛЮЧЕНИЕ

В результате проделанной работы были получены следующие ключевые результаты:

Разработана структурная схема системы реабилитации, использующей технологии виртуальной реальности для пациентов с нарушениями опорно-двигательного аппарата.

Выбран и интегрирован в систему тренажер и датчик Холла для определения его положения в пространстве. Также подобраны дополнительные сенсоры для регистрации физиологических параметров пациента.

Определено, что Unreal Engine является оптимальной платформой для реализации VR-решений в реабилитационных проектах. Принято решение использовать Blueprints для создания ключевых функциональных элементов симулятора. В процессе разработки применялись инструменты Unreal Editor, обеспечивающие конструирование виртуальной среды для выполнения реабилитационных упражнений. Использование Blueprints позволило эффективно обрабатывать движения пациента, улучшить производительность системы, а также сократить затраты времени и ресурсов на разработку.

Полученные результаты подтверждают перспективность использования VR-технологий в реабилитационных процессах, обеспечивая гибкость настройки программного обеспечения под индивидуальные потребности пациентов и повышая эффективность восстановления.

БЛАГОДАРНОСТЬ

Авторы хотят выразить особую благодарность врачам ФГБУ НМИЦ ТО им. Р. Р. Вредена, которые активно участвовали в процессе исследования, предоставляя свои знания и опыт. Их помощь в корректировке медицинских аспектов программы и готовность поделиться профессиональными рекомендациями были необходимы для создания эффективной и безопасной системы реабилитации для пациентов.

СПИСОК ЛИТЕРАТУРЫ

- [1] Виртуальная реальность в нейрореабилитации [электронный ресурс] – URL: https://beka.ru/downloads/issues/robots_7.pdf
- [2] Технологии VR на службе реабилитационной медицины [электронный ресурс] – URL: https://habr.com/ru/company/russian_rehab_industry/blog/447558/
- [3] Virtual Reality in Medicine [электронный ресурс] – URL: <https://www.webmd.com/a-to-z-guides/features/virtual-reality-medicine>
- [4] Прорывные технологии VR/AR в медицине [электронный ресурс] – URL: https://dtf.ru/s/vr_ar/1264712-proryvnye-tehnologii-vr-ar-v-medicine
- [5] Virtual reality in rehabilitation and therapy [электронный ресурс] – URL: https://www.researchgate.net/publication/261368673_Virtual_reality_in_rehabilitation_and_therapy
- [6] Virtual reality in the diagnostic and therapy for mental disorders: A systematic review [электронный ресурс] – URL: <https://www.sciencedirect.com/science/article/pii/S0272735822000988?via%3Dihub>
- [7] Does intervention using virtual reality improve upper limb function in children with neurological impairment: a systematic review of the evidence [электронный ресурс] – URL: <https://pubmed.ncbi.nlm.nih.gov/21401370/>
- [8] Computer-Mediated Therapies for Stroke Rehabilitation: A Systematic Review and Meta-Analysis [электронный ресурс] – URL: [https://www.strokejournal.org/article/S1052-3057\(22\)00150-1/fulltext](https://www.strokejournal.org/article/S1052-3057(22)00150-1/fulltext)
- [9] Effectiveness of conventional versus virtual reality-based balance exercises in vestibular rehabilitation for unilateral peripheral vestibular loss: results of a randomized controlled trial [электронный ресурс] – URL: <https://pubmed.ncbi.nlm.nih.gov/25842051/>
- [10] Technical aspects of virtual augmented reality-based rehabilitation systems for musculoskeletal disorders of the lower limbs: a systematic review [электронный ресурс] – URL: https://www.researchgate.net/publication/366836603_Technical_aspects_of_virtual_augmented_reality-based_rehabilitation_systems_for_musculoskeletal_disorders_of_the_lower_limbs_a_systematic_review