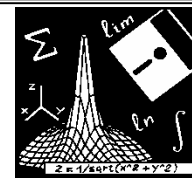


ЭЛЕКТРОНИКА



УДК 004.946:004.383.8.032.26

DOI: [https://doi.org/10.34680/2076-8052.2021.4\(125\).10-14](https://doi.org/10.34680/2076-8052.2021.4(125).10-14)

РАЗРАБОТКА VR. ОСОБЕННОСТИ СОВМЕСТИМОСТИ С НЕЙРОТЕХНОЛОГИЯМИ

Д.Я.Бурцева, Е.А.Менделеев, И.Л.Хонин, К.О.Докучаев, Р.В.Петров

VR DEVELOPMENT. FEATURES OF COMPATIBILITY WITH NANOTECHNOLOGIES

D.Ya.Burtseva, E.A.Mendelev, I.L.Khonin, K.O.Dokuchaev, R.V.Petrov

Новгородский государственный университет имени Ярослава Мудрого, Darya.Burtseva@novsu.ru

Статья посвящена обсуждению некоторых проблем, возникающих при взаимодействии нейротехнологий и виртуальной реальности при использовании наиболее применяемых в настоящее время межплатформенных сред разработки компьютерных игр (Unreal Engine 4, Unity). На сегодняшний день нейротехнологии достигли значительного развития, получив признание и применение во многих областях сферы общества. Они оказывают значительное влияние на окружающий нас мир, в связи с чем возможность их комбинированного с виртуальной реальностью применения является весьма ожидаемым явлением. Приводятся основные сведения о принципах разработки виртуальной реальности и совместимости с нейротехнологиями. Необходимо учитывать, что программирование виртуальной реальности в разных средах, имея общие основы, обладает индивидуальными особенностями реализации — от различной последовательности действий до полной замены некоторых шагов.

Ключевые слова: виртуальная реальность, нейротехнологии, очки виртуальной реальности, трехмерная графика, взаимодействие с виртуальными объектами

Для цитирования: Бурцева Д.Я., Менделеев Е.А., Хонин И.Л., Докучаев К.О., Петров Р.В. Разработка VR. Особенности совместимости с нейротехнологиями // Вестник НовГУ. Сер.: Технические науки. 2021. №4(125). С.10-14. DOI: [https://doi.org/10.34680/2076-8052.2021.4\(125\).10-14](https://doi.org/10.34680/2076-8052.2021.4(125).10-14)

The paper is devoted to the discussion of some problems arising from the interaction of neurotechnologies and virtual reality when using the most widely used a cross-platform game engine (Unreal Engine 4, Unity). At present, neurotechnologies have achieved significant development, having received recognition and application in many areas of society. They have a significant impact on the world around us, and therefore, the possibility of their combined use with virtual reality is a very expected phenomenon. The article provides basic information about the principles of virtual reality development and compatibility with neurotechnologies. Each of the approaches virtual reality programming in different game engine has common foundations, but has individual implementation features - from different sequences to complete replacement of some steps, which must be taken into account.

Keywords: virtual reality, neurotechnology, virtual reality glasses, three-dimensional graphics, interaction with virtual objects

For citation: Burtseva D.Ya., Mendelev E.A., Khonin I.L., Dokuchaev K.O., Petrov R.V. VR development. Features of compatibility with nanotechnologies // Vestnik NovSU. Issue: Engineering Sciences. 2021. №4(125). P.10-14. DOI: [https://doi.org/10.34680/2076-8052.2021.4\(125\).10-14](https://doi.org/10.34680/2076-8052.2021.4(125).10-14)

Введение

За последнее время во все сферы человеческой жизни активно внедряются новые технологии — от телефонов с сенсорным экраном до продвинутых нейрогарнитур, с высокой точностью считывающих показатели головного мозга пользователя. К подобным новшествам можно отнести и набирающую в последние годы популярность виртуальную реальность. Виртуальная реальность — созданный при помощи компьютерных средств виртуальный мир, передаваемый человеку посредством органов зрения и слуха, с незначительным привлечением тактильных ощущений. Создаваемые для такой реальности объекты могут действовать согласно как реальным, так и альтернативным законам физики, что делает взаимо-

действие с ними довольно увлекательным опытом [1]. Пользователь контактирует с искусственно созданным пространством при помощи устройств виртуальной реальности, состоящей из шлема, контроллеров и отслеживающих датчиков. Шлем — это прибор, содержащий один или несколько дисплеев, на которые передаётся визуальная информация. Для того чтобы пользователь имел возможность реально перемещаться в нереальном мире, данные устройства содержат системы слежения, состоящие из гироскопа, акселерометра и компаса, и имеют широкий угол обзора [1].

Об интересе к системам виртуальной реальности можно судить по количеству проданных экземпляров видеоигры Half-Life Alyx, выпущенной американской компанией Valve в 2019 г. специально для таких систем. По состоянию на 2021 г. общее число

проданных экземпляров достигло 860 тыс., а суммарная выручка составила 40,7 миллионов долларов. Во многом такую популярность игре принесло использование в игре эффекта погружения в сценарий. К тому же помимо увлекательного сюжета, что очень важно для технической составляющей виртуальной системы, у игры наличествует захватывающий игровой процесс, наполненный всевозможным взаимодействием с окружающей средой: любой объект в кадре можно взять в виртуальную руку и ударить об стену или выкинуть за пределы виртуального местоположения, а механики игровой стрельбы и перезарядки оружия были максимально приближены авторами к имеющейся реальности. Данный коммерческий успех способен вызвать интерес к разработке подобного типа проектов. Таким образом, увлечение способов взаимодействия с виртуальным миром может как привлечь новых пользователей, так и заинтересовать более опытных.

Нейротехнологии существенно влияют на жизнь человека и общества, хотя их присутствие остаётся невидимым. Они включают в себя разные технологии, в основном предназначенные для улучшения и исправления функционирования мозга, а также позволяют исследователям и врачам визуализировать мозговую деятельность. В общем, нейротехнологии позволяют нам лучше понимать, как функционирует мозг, и изучать различные аспекты сознания, мыслительной деятельности и высших психических функций человека.

В последнее время на мировом рынке электроники появилось достаточно большое количество устройств, которые способны получать разного рода биоэлектрические сигналы от человеческого организма, в том числе и электроэнцефалографические (ЭЭГ), и передавать их на специализированные устройства. ЭЭГ использует для чтения мозговых сигналов электроды, приложенные к голове, и регистрирует слабые электрические нервные импульсы, созданные психологическим состоянием человека или мыслительным процессом. Одним из нейроинтерфейсов потребительского класса на мировом рынке являются нейро-гарнитуры американской компании NeuroSky. Нейроинтерфейсы NeuroSky представляют собой одноканальные ЭЭГ-интерфейсы. Область применения нейроинтерфейсов и соответствующих компьютерных приложений — это тренировочные комплексы, системы развития ментальных способностей, системы изменения поведения, системы профилактики стрессов, работа с синдромом дефицита внимания и гиперактивности, системы по работе с психоэмоциональными состояниями и т. д.

Рассматривая глубину проблем с совместимостью виртуальной реальности и нейротехнологий, была поставлена цель: выбрать новый способ взаимодействия нейроинтерфейса с виртуальной реальностью и определить основные нюансы, возникающие при реализации предложенного метода.

Для достижения результата был решён ряд задач:

1. Определены основные этапы разработки виртуальной среды.

2. Разработан модуль для нового взаимодействия с виртуальной средой на двух популярных платформах Unreal Engine 4 и Unity.

1. Комбинированное применение с нейрогарнитурами

Очевидно, что системы виртуальной реальности (VR) сами по себе являются крайне перспективным направлением развития, но не стоит забывать, что сегодня существует множество других технологий, способных при их грамотном сочетании с VR-устройствами улучшить и расширить предлагаемый системами функционал. В частности, такими технологиями могут стать нейротехнологии.

При нынешнем уровне развития нейротехнологии уже сейчас применяют в комбинациях с виртуальной реальностью в целях обеспечения более качественной реабилитации, путём симуляций и погружений в определённые ситуации. Данная ситуация обусловлена тем, что в игровой форме человек усваивает, запоминает и просто воспринимает всё намного лучше — поэтому разработаны игры, построенные на принципе, где человеку необходимо самостоятельно контролировать какие-либо свои показатели. Многие пациенты клиник, использующих нейротехнологии, в силу того, что не могут наблюдать какие-либо изменения у себя, склонны к прекращению выполнения упражнений, будь то нахождение в состоянии сосредоточенности или расслабления. Контролируемые самим пользователем сигналы от головного мозга, необходимые при продвижении в игре, воспринимаются гораздо позитивней и показывают лучший результат в сравнении с использованием нейротехнологий без привлечения такого метода. Стоит отметить, что в данных разработках нейротехнологии, как правило, не предоставляют дополнительных возможностей пользователю, в связи с чем не используется весь потенциал подобной комбинации технологий.

Использовать нейротехнологии в качестве альтернативного способа взаимодействия с виртуальным миром, показывающим пациенту или игроку сигналы обратной связи, оказывается очень эффективно. Это обеспечивает устойчивую обратную связь, необходимую для самооценки либо для дальнейших психологических выводов.

2. Разработка виртуальной реальности

В целом разработку виртуальной реальности от концепции до сборки на межплатформенной среде разработки компьютерных игр можно разделить на следующие модули:

1. Концепция проекта

На данной стадии происходит поиск и генерация идеи проекта, определяется цель, место действия, количество и качество графики, уровень и проработка механик, а также целевая аудитория. Данные меры играют фундаментальную роль в успехе будущего проекта: от хорошо проработанной идеи зависит уровень эффективности вложенных человеко-часов и денежных ресурсов [2].

2. Командная работа

Когда идея утверждена, происходит работа с командой [3]. Определяется состав, в котором четко

определены роли между участниками. Разумеется, существуют варианты, когда один участник решает несколько задач, но по возможности подобных ситуаций стараются избегать: эффективность решения задач выше, когда есть специализация, и каждый, как на конвейере, собирает последовательно детали одного и того же элемента [2].

3. Двумерное проектирование

На данной стадии художники рисуют образы мест, персонажей, предметов, которые являются либо ключевыми в повествовании, либо же просто украшением. Во многом от того, насколько точно они интерпретируют идею проекта, зависит качество работы компьютерных модельеров, художников по текстурам и художников-аниматоров.

4. Трёхмерное проектирование

Когда наброски художников согласованы, основная работа переходит к специалистам трёхмерной графики, которые воплощают в реальность нарисованные идеи [3]. Здесь крайне важно, чтобы подобранные люди владели 3D-программами и понимали специфику трёхмерной графики. Даже если будет возможность взять готовую модель из интернета, то в случае необходимости ее надо будет подправить под определённые нужды или задачи.

В наше время существует огромное количество программ, позволяющих решать данные задачи, среди них наиболее популярными являются Blender и Autodesk Maya. Каждая из них обладает своими достоинствами и недостатками. Blender — бесплатная программа с открытым исходным кодом, имеющая огромный набор инструментов, однако полный функционал можно получить только благодаря дополнениям. Новичок может запутаться в интерфейсе или в том, какая команда за что отвечает, а также упустить потенциально полезную функцию из-за недостаточного описания функционала. Maya обладает полным набором инструментов, однако стоит довольно дорого — 1000\$ за год обслуживания. Необходимо учитывать финансовые возможности, а также задачи, которые стоят в проекте.

На данной стадии важно не только соблюдать правильную сетку модели (топологию), во многом влияющую на будущие текстуры и анимации, но и количество полигонов (многоугольников) и при необходимости оптимизировать модель. Если модель не имеет никаких погрешностей и ошибок, то ее постепенно подготавливают к процессу наложения растрового изображения на поверхность 3D-объекта в целях создания реалистичности (текстурирование). Поскольку текстурные карты и сама модель находятся в разных измерениях — 2D и 3D, необходимо некоторое связующее звено, называемое текстурными координатами (UV-развертка) — отображение полигонов трёхмерного объекта на 2D-плоскость.

UV делают при помощи специальных инструментов, интегрированных в трёхмерный редактор. Есть множество различных техник, но чаще всего модель разрезают на «швы», а пространство между ними вытягивают на текстурный квадрат и выпрямляют инструментами. На протяжении всего процесса создания UV-развертки разрешение и корректность

текстуры проверяются специальной шахматной картой «Checker map». От того, насколько корректно отобразятся «шахматы», будет ясно, корректно отобразится текстура или нет. Если UV-развертка устраивает специалиста, то модель переносится в «Adobe Substance Painter», где в дальнейшем уже визуализируется.

Текстуры создаются в «PBR Metallic Roughness», в котором используются алгоритмы физического рендеринга в реальном времени. Качество и степень проработки зависит прежде всего от задания и идеи проекта. Весь процесс следует принципу «от общего к частному». Сначала наносятся основные цвета и материалы, а затем — детали. В конце данного этапа текстуры экспортируются в нужных разрешениях и формате для последующего перемещения в межплатформенную среду разработки компьютерных игр (игровые движки).

В некоторых случаях требуется создать анимацию для модели. В таких ситуациях используется либо покадровая анимация, либо создается «скелет», если объект более сложный или же представляет организм (животные, человек). Записанная анимация экспортируется в игровой движок.

5. Движок игровой

После того как все необходимые элементы созданы, их перемещают в игровой движок, где собираются места визуализации (локации или места), прописываются взаимодействия (механики) будущего проекта, а также настраиваются схемы управления действующим персонажем. В роли управляющих устройств могут быть указаны как привычные всем клавиатура и мышь, так и контроллеры системы виртуальной реальности. На этой же стадии присоединяются и инновационные методы управления виртуальной реальностью (в нашем случае — нейрогарнитур).

Поскольку VR-системы представляют собой системы управления и вывода графической информации, то большая часть разработки проектов под них соответствует привычной последовательности этапов по созданию игр. Главной особенностью разработки под системы виртуальной реальности является работа со всевозможными взаимодействиями с созданными виртуальными объектами. Используются возможности физики для создания эффектов реалистичности и погружения в сценарный процесс.

3. Проблемы, возникающие при разработке нейромодуля на игровых движках Unreal Engine и Unity

Возможность применения виртуальной реальности и нейротехнологий имеет потенциал, в связи с чем встают задачи комбинации данных типов устройств. Варианты решения могут быть разнообразны, начиная с полной разработки программ с нуля и заканчивая попытками применения уже готовых решений. Ниже будут рассмотрены основные проблемы, возникающие при разработке подобного модуля для уже готовых игровых движков.

Проблемы разработки на Unity

При разработке нейромодуля на игровом движке Unity проблем, как правило, не возникает. Благодаря языку программирования C#, используе-

мому для написания файлов кода, имея документацию с алгоритмом считывания байтов данных с нейроинтерфейса, вполне возможно реализовать сам класс (модуль), работающий с нейрогарнитурой, и пару несложных механик на взаимодействии нейроинтерфейса и создать VR за несколько часов (при этом это будет готовый демонстрационный проект, показывающий новый способ взаимодействия с виртуальным миром посредством использования нейрогарнитур).

Проблемы разработки на Unreal Engine

Разработка нейромодуля на игровом движке Unreal Engine, в сравнении с Unity, является более сложной задачей.

При разработке модуля была использована IDE Microsoft Visual Studio 2019.

Должны быть учтены следующие проблемы:

1. В проекте не должно быть обработки исключений «try-catch».

Для обработки ошибок разного рода используются в основном два варианта: обработка исключений и возвращение кодов ошибок из методов или функций как результат их работы. Во втором случае, как только была вызвана функция или метод, необходимо сразу обработать её код ошибки с помощью конструкции if. Такие конструкции могут создать структуру кода со слишком глубокой вложенностью, что в дальнейшем не позволит легко поддерживать код в ситуации его дополнения или изменения в случае логической ошибки. Поэтому лучше использовать конструкцию «try-catch», что облегчает читаемость кода. Именно это описывал в своей книге Роберт Мартин [4]. Однако такой механизм обработки ошибок не поддерживается проектом Unreal Engine. Если с любым другим проектом существует специальная настройка, позволяющая включить конструкцию «try-catch» в проект, то в случае с проектом Unreal Engine такой настройки просто не существует, поэтому было необходимо выполнить обработку ошибок с помощью кодов ошибки, что, в свою очередь, увеличивает необходимое время на их обработку и читаемость кода.

2. Указание на ошибку в месте, где её нет.

Данная проблема появляется в ситуациях, когда необходимо добавить новый функционал или изменить существующий. При добавлении нового функционала возникают ошибки в местах, где их потенциально не должно быть, что в итоге не позволяет с первого раза скомпилировать проект для тестирования. Для этого необходимо пересобрать проект полностью, что расходует время.

3. Медленная обратная связь от IDE на синтаксические ошибки в программном коде.

Подобного рода трудность замедляет процесс отладки. Перед запуском компиляции IDE может выдать пустой список с ошибками. После того как компиляция проекта будет запущена, лишь через некоторое время могут возникнуть ошибки, связанные с синтаксисом языка, что требует лишних затрат на тестирование.

4. Сборка модуля в отдельный плагин для подключения к другим проектам.

После окончания написания класса необходимо собрать модуль в отдельный плагин для удобного подключения к другим проектам. Исходя из документации, представленной разработчиками движка, для его сборки необходимо нажать на кнопку «Package» рядом с созданным модулем, но данный модуль по умолчанию собирается под различные платформы. Поскольку настройки целевых платформ для плагина обнаружено не было, на настройку необходимых библиотек для сборки на эти платформы может потребоваться значительное время.

5. Проблема с поддержкой файлов Windows.

Для реализации механизма подключения к нейрошлему и считывания с него данных было использовано клиент-серверное взаимодействие, для чего необходимы файлы Windows, содержащие реализацию сокетов. После компиляции проекта могут возникать ошибки, указывающие на файлы Windows. Для их решения необходимо определить макрос, который убирает из этих файлов часть функционала для корректной работы.

6. Разработка модуля в виде статической Blueprint-библиотеки.

В нашем случае нейромодуль был реализован в виде статической Blueprint-библиотеки для использования его в игровом движке, в связи с чем все поля и методы класса, работающего с нейроинтерфейсом, статичны. В результате этого проект Unreal Engine запускался таким образом, что значения полей класса запоминались даже после остановки тестовой работы приложения, в связи с чем возникал ряд ошибок (как при подключении, так и при закрытии соединения с нейроинтерфейсом). По этой причине потребовалась реализация дополнительной обработки ошибок. Если, в сравнении с Unity, движок стирал из памяти значение в статической переменной после окончания тестового запуска приложения, то Unreal Engine сохраняет данные в полях класса до тех пор, пока рабочее окно движка не будет закрыто.

7. Реализация автоматического подключения модуля Unreal Engine к нейроинтерфейсу.

При разработке нейромодуля была использована официальная документация компании NeuroSky, чип которых применяется в используемой нейрогарнитуре. Само же устройство подключается к компьютеру по Bluetooth-соединению. Исходя из документации, для подключения к нейрошлему необходимо знать номер COM-порта, чтобы с него получать необходимые для работы данные. Номер порта можно легко узнать из дополнительных параметров Bluetooth и передать его в качестве параметра функции, устанавливающей соединение с нейроинтерфейсом. Однако не всё так просто, как это кажется на первый взгляд. Дело в том, что операционная система назначает COM-порт из списка доступных для устройства, чтобы установить с ним соединение. К примеру, на одном компьютере мы увидели, что порт COM4 соединён с нашим устройством и указали его в качестве параметра. После чего собранный проект отдали конечному пользователю, предварительно протестировав. Однако может возникнуть такая ситуация, что данный порт у пользователя уже занят

другим устройством (например, Bluetooth-наушниками), поэтому при подключении нейрогарнитуры по Bluetooth-соединению ей выделяется другой COM-порт, в результате чего данные не будут сниматься с нейроинтерфейса, а соответственно, не будут работать механики, основанные на нём. Поэтому было важно реализовать автоматическое подключение модуля, что потребовало альтернативных решений этой задачи.

8. Смена протокола обмена данными между компьютером и нейроинтерфейсом.

Использование программы «ThinkGear Connector», реализующей данный механизм путём передачи данных в удобном для разработчика виде, позволяет осуществить механизм автоматического подключения к нейрогарнитуре. Данная программа использует другой протокол обмена данными, для чего необходимо переписать алгоритм подключения и считывания данных, на что требуется дополнительное время.

9. Поиск сторонней библиотеки для анализа строк формата JSON.

Программа «ThinkGear Connector» передаёт данные разработчику в формате JSON строк. Если в предыдущей версии модуля считывание данных происходило в виде считывания байтов данных с нейроинтерфейса напрямую через COM-порт, то в данном случае через «ThinkGear Connector» мы получаем JSON строку с данными нейрошлема. Поэтому необходимо было найти стороннюю библиотеку с документацией для анализа таких строк.

4. Требования к компьютеру для разработки нейромодуля и VR

Перед тем как приступить к разработке, стоит учитывать, что компьютерная техника должна соответствовать хотя бы минимальным требованиям, в противном случае работа будет невозможна.

Так, минимальные требования представляют собой следующее:

- видеокарта — NVIDIA GeForce GTX 970, AMD Radeon R9 290, аналогичная или более новая модель;

- процессор — Intel Core i5-4590/AMD FX 8350 аналогичная или выше;

- ОЗУ — 4 Гб или более;

- видеовыход — HDMI 1.4, DisplayPort 1.2;

- совместимость с Bluetooth EBR;

- USB — 1x USB 2.0;

- ОС — Windows 10.

Рекомендуемые требования:

- видеокарта — NVIDIA GeForce GTX 1060, AMD Radeon RX 480 аналогичная или более новая модель;

- процессор — Intel Core i5-4590/AMD FX 8350 аналогичная или выше;

- ОЗУ — 4 Гб или более;

- видеовыход — HDMI 1.4, DisplayPort 1.2;
- совместимость с Bluetooth EBR;
- USB — 1x USB 2.0;
- ОС — Windows 10.

Заключение

Разработка приложения виртуальной реальности является весьма трудоёмким процессом, а в совокупности со способом взаимодействия с нейрогарнитурой требуется значительное количество времени. Однако, как видно из приведённых нами основных модулей разработки виртуальных сред, подключение нейротехнологии будет происходить на отдельном этапе, что не повлияет на работу остальных элементов. Заметим, что нет необходимости менять устоявшиеся схемы разработки под новый способ взаимодействия с виртуальным миром, что важно для дальнейшего планирования задач разработки программ. Учитывая потенциал виртуальной реальности, перспективы и возможности дальнейшего роста, в силу того, что направление является довольно молодым, предложение об объединении VR и нейротехнологий представляется перспективным. Таким образом, с учётом определённых сложностей и нюансов, вполне возможно реализовать личностное взаимодействие с виртуальной реальностью путём применения нейротехнологий.

1. Рахматуллаев А.Н., Иманбек Р.К., Рахимова А.Р. Технология виртуальной реальности [Электронный ресурс] // Молодой ученый. 2021. №18 (360). С.50-58. URL: <https://moluch.ru/archive/360/80615/> (дата обращения: 01.10.2021).
2. Особенности VR-разработки. Часть 1. С чего начать? [Эл. ресурс]. URL: <https://dtf.ru/gamedev/628149-osobennosti-vr-razrabotki-chast-1-s-chego-nachat> (дата обращения: 20.09.2021).
3. Максим М. Миллионы миллиардов. Как стартовать в игровой индустрии, работая удаленно, заработать денег и создать игру мечты. СПб.: Питер, 2021. 320 с.
4. Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста / Пер. с англ. СПб.: Питер, 2013. 464 с.

References

1. Rakhmatullaev A.N., Imanbek R.K., Rakhymova A.R. Tekhnologiya virtual'noy real'nosti [Virtual reality technology]. Molodoy uchenyy, 2021, no. 18 (360), pp. 50-58. Available at: <https://moluch.ru/archive/360/80615/> (accessed 01.10.2021).
2. Osobennosti VR-razrabotki. Chast' 1 — S chego nachat'? [Features of VR development. Part 1 — Where to start?]. Available at: <https://dtf.ru/gamedev/628149-osobennosti-vr-razrabotki-chast-1-s-chego-nachat>. (accessed 09/20/2021).
3. Mikheenko M. Millions of billions. Kak startovat' v igrovoy industrii, rabotaya udalennno, zarabotat' deneg i sozdat' igru svoey mechty [How to start in the gaming industry, working remotely, make money and create a dream game]. St. Petersburg. "Peter" Publ., 2021. 320 p.
4. Martin R. Clean Code: Creation, Analysis and Refactoring. Programmer's library. (Russ. ed.: Martin R. Chisty kod: sozdaniye, analiz i refactoring. St. Petersburg. "Peter" Publ., 2013. 464 p.).