

Массовая виртуальная реальность и прототипирование

М.З. Исмагилов maratismagil@hotmail.com

Аннотация: цель данной статьи - рассказать читателю о массовых технологиях виртуальной реальности, то есть тех, которые доступны не только в стенах НИИ и лабораториях промышленных предприятий, и о том как эти технологии можно применять для прототипирования. Сначала мы обсудим, когда именно средства виртуальной реальности стали массовыми и как они отличаются от более ранних средств. Далее, разберем какие сейчас есть аппаратные средства VR и как их можно классифицировать. Посмотрим на основные формы интерактивности доступные в массовой VR. Далее, рассмотрим программное обеспечение с помощью которого создаются виртуальные миры. Затем обсудим как именно выбирается программное и аппаратное обеспечение для прототипа. И наконец, рассмотрим два виртуальных прототипа, созданных в то время когда технологии VR еще не были массовыми, и обсудим как их можно было бы усовершенствовать применяя современные технологии.

Введение

Еще 10 лет назад технологии виртуальной реальности (далее VR) были уделом узких научных кругов и крайне ограниченно применялись в сфере медицины и производства. Однако, ситуация резко изменилась с появлением на рынке устройств VR для массового потребителя.

Потребительские устройства VR хотя и не обладают сопоставимой с профессиональными устройствами мультимодальностью, производительностью и надежностью, но превосходят их в массовости и экономичности. То же можно сказать и о программных средствах

обеспечения среды виртуальной реальности, где более массовые инструменты быстрее развиваются за счет поддержки сообщества.

Цель статьи в том, чтобы доказать возможность применения потребительских устройств виртуальной реальности для задач прототипирования и показать некоторые возможные способы построения прототипов и человеко-машинных интерфейсов в этом случае.

Устройства Виртуальной Реальности. Массовые и специализированные

Для этой статьи используем определение виртуальной реальности данное отечественным ученым А.И. Вороновым [4]. Виртуальная реальность - это «кибернетическое пространство», созданное с помощью компьютера, для полного погружения в которое техническими средствами достигается практически полная изоляция от внешнего мира. То есть, мы будем рассматривать VR как компьютерную технологию, а не как более абстрактное философское понятие, которое может и вовсе не иметь отношения к технике [3]. Также заметим, что не все устройства VR используют компьютер для погружения в виртуальную среду. История VR началась раньше компьютерной эры, с изобретения, в 1832, стереоскопа Чарльзом Уитстоном. Очевидно, что аналоговые устройства VR сейчас массово не применяются, поэтому касаться их в статье не будем.

Вообще, история VR включает в себя множество самых разнообразных устройств, с которыми можно ознакомиться в специальной литературе[1]. Мы же сейчас рассмотрим только современные и разделим их на устройства для массового потребителя и специализированные.

Все устройства VR можно разделить по генерируемым ими стимулам или раздражителями:

1. Устройства генерирующие визуальные стимулы

2. Звуковые стимулы
3. Тактильные стимулы
4. Кинестетические ощущения в виртуальном мире
5. Ощущения баланса и движения
6. Ощущения запаха и вкуса.

А можно разделить по техническому устройству на:

1. Стационарные устройства
2. Устройства, работающие по принципу относительности к какой-либо части тела пользователя.

И наконец, устройства VR можно классифицировать по доступности для пользователя.

1. Устройства для массового потребителя
2. Устройства для научных, медицинских, промышленных, коммерческих организаций.

Последняя классификация кажется автору наиболее важной, ведь в то время как специализированные VR комплексы используются очень ограниченным кругом организаций, то массовые - миллионами пользователей по всему миру. Для примера, по данным Statista Research Department от 3 мая 2019 года, совокупные продажи VR устройств Sony составили 2,2 миллиона, а устройств Oculus - 1,3 миллиона.

Как правило, массовые устройства VR во многом схожи друг с другом. Это обусловлено стремлением производителей снизить их себестоимость. Специализированные устройства VR, наоборот, очень разнообразны, чтобы максимально эффективно выполнять конкретные задачи.

Массовые устройства VR обычно ограничиваются функциями воспроизведения визуальных и аудио стимулов, а также опираются на кинестетические ощущения пользователя. Они все представляют из себя шлемы, которые крепятся к голове пользователя. Ярким примером

массового устройства ВР является шлем Oculus Rift (см. рис. 1). Шлем с функцией отслеживания положения закрепляется на голове пользователя с помощью ремешков и соединяется кабелем с компьютером, а по беспроводной связи он соединен с контроллерами правой и левой руки с функциями отслеживания позиции.



Рис. 1 Шлем ВР Oculus Rift DK1

Специализированные устройства ВР часто работают не только по каналам визуального и аудио восприятия, но и по другим. Наиболее яркий пример - это авиа-тренажеры, которые перемещают тело пользователя в пространстве и тем самым воздействуют на его вестибулярный аппарат. Многие из таких устройств стационарные. Другой пример - комплекс CAVE [1], который вместо проекции картинки на экран перед глазами проецирует ее на специальные стены с помощью мощных проекторов, как показано на рис. 2. Очевидно, что такие специализированные ВР устройства сложны в производстве и обслуживании и дороги, в ближайшей перспективе они не подходят для массового рынка.

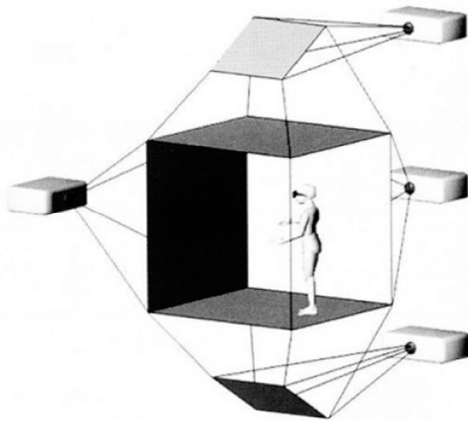


Рис. 2 Комплекс CAVE

Массовые устройства VR и их классификация

Объектом исследования в статье являются именно устройства массового VR. То есть те, при конструировании и производстве которых производитель взял курс на массового потребителя.

Первым действительно массовым устройством VR можно считать шлем Oculus Rift DK1, который был создан в 2012 году и поступил в продажу летом 2013 года. Стоит отметить, что это был “комплект разработчика”, то есть устройство не предназначенное для конечного потребителя. Однако, таких комплектов компания Oculus VR продала более 100000 штук, поэтому уже тогда можно было говорить о массовой технологии.

Причины, по которым компания продавала комплект разработчика, а не устройство для рядовых пользователей сейчас вполне очевидны. На тот момент система просто не могла обеспечить пользовательский опыт допустимого качества. Достаточно лишь упомянуть, что экран DK1 имел разрешения 640*800 на каждый глаз.

Позже компания Oculus была приобретена корпорацией Facebook и сейчас они производят уже пользовательскую (CV) версию шлема с существенно улучшенной эргономикой и характеристиками.

Сейчас на рынке представлены устройства от разных производителей. Самые популярные устройства это: GearVR, Oculus Go/Quest, PSVR, Oculus Rift, HTC Vive, Samsung HMD Odyssey. Несмотря на разных производителей, многие из этих устройств функционируют по схожим принципам.

Все эти шлемы можно классифицировать по характеру и положению вычислительного модуля, который и обеспечивает VR:

1. Стационарные устройства VR. Они не выполняют всех функций по обработке виртуального мира и подключаются к компьютеру или другому вычислительному устройству. (PSVR, Oculus Rift, HTC Vive)
2. Автономные устройства VR. Могут работать самостоятельно (Oculus Go/Quest).

Практически все стационарные устройства VR используют операционную систему Windows. Почти все автономные устройства используют Android. Такое разделение понятно, поскольку так исторически сложилось, что стационарные компьютеры используют процессоры на базе x86 совместимой архитектуры, а мобильные устройства используют процессоры на ARM.

Кроме автономности есть и другая принципиально важная характеристика - количество степеней свободы шлема и контроллеров (если они есть). Современные VR устройства могут иметь 3 или 6 степеней свободы (см. рис. 3). Это не зависит от автономности/стационарности, хотя обычно стационарные шлемы имеют больше степеней свободы. Связано такое разделение с принципиальными отличиями в реализации механизма отслеживания наклона/поворота и позиции шлемов. Для того, чтобы рассчитать углы поворота шлема в него достаточно установить акселерометр и гироскоп. Для определения позиции же необходима сложная оптическая или иная система, которая будет сравнивать положение

шлема с элементами окружающей среды. Обычно, такая система использует стационарные камеры в комнате пользователя, и только самые современные шлемы, такие как Oculus Quest и Rift S используют камеры внутри самого шлема.



Рис. 3 Степени свободы шлема VR

Интерактивность в виртуальных мирах

Для полноценной работы в VR необходим не только вывод информации, но и ее ввод. При этом нужно помнить, что устройства ввода должны тоже быть достаточно дешевыми, компактными и простыми, как и сами шлемы. В качестве контр-примера возьмем профессиональные устройства ввода Phantom Omni с активной тактильной обратной связью [1]. Устройство позволяет свободно перемещать и вращать “перо” в виртуальном пространстве и ощущать его взаимодействие с предметами. Однако, высокая сложность устройства, его массивность и требования к точности не позволяют его назвать массовым.

Потребительские устройства ввода тоже весьма разнообразны. Многие из них несут в себе принципиально разные идеи о том, каким должно быть взаимодействие с виртуальным миром. Устройства ввода можно классифицировать по модальности прямой и обратной связи:

- 1) Стандартные устройства без функции отслеживания положения.
- 2) Устройства с функцией отслеживания положения.
- 3) Микрофоны.
- 4) Устройства отслеживающие направление взгляда.

Стандартные устройства ввода работают так же, как и с обычным компьютером. Это: клавиатура, мышь, геймпад или джойстик. Несмотря на кажущуюся простоту, они могут быть весьма эффективны для задач где требуется высокая надежность распознавания команды, высокая скорость и обратная связь. Большинство традиционных устройств используют кнопки, а кнопка - лучший инструмент для передачи бинарной информации. Минусом такого подхода является снижение эффекта присутствия в VR и невозможность для оператора увидеть само устройство. Впрочем, можно использовать 3d-модель или изображение устройства ввода в самом виртуальном мире, там где пользователь ожидает его увидеть [1]. На этой репрезентации устройства также можно давать подсказки по функциям кнопок.

Устройства с функцией отслеживания положения могут быть сами отслеживаемыми, или могут отслеживать положение пользователя в реальном мире. Первый вариант более популярный, так как обеспечивает большую надежность и точность отслеживания. Ярким примером может служить контроллер Oculus Touch. Каждый контроллер Oculus Touch работает в режиме шести степеней свободы и дает оператору весьма правдоподобную информацию о положении его рук в VR. С другой стороны, система работает на основе оптических датчиков, говорить о высокой точности которых не приходится, а движения рук аватара просчитываются по законам инверсной кинематики и корректно определить положение, например, локтя пользователя, невозможно. Но даже несмотря на перечисленные недостатки эта схема ввода очень

популярна в ВР. Во многом еще и благодаря тому, что на контроллерах расположены традиционные джойстики и кнопки, которые приносят плюсы стандартных устройств ввода в опыт ВР.

Кроме полноценных контроллеров с 6 степенями свободы используются также и виртуальные “указки” с 3 степенями. Это контроллеры Oculus Go, Gear VR и некоторых других устройств на основе мобильной технологий. Они не могут передавать информацию о своем положении и поэтому указки привязаны к некой фиксированной точке относительно виртуальной головы пользователя.

Есть также класс устройств которые не нужно держать в руках, при этом они следят непосредственно за частями тела пользователя. Например, Microsoft Kinect который снимает пользователя с помощью камеры с функцией определения глубины. Такой подход выглядит удобным, но он работает только в ограниченной зоне видимости камеры и может давать различные результаты при разной освещенности или даже разной форме тела пользователя.

Все устройства с отслеживанием положения частей тела в пространстве существенно усиливают эффект присутствия.

Микрофоны предназначены для записи звука голоса пользователя. Обычно они встроены в корпус шлема и комплектуются средствами для шумоподавления. Пользователь подает голосовые команды через микрофон, а приложение транслирует их в понятный системе формат.

Устройства отслеживающие направление взгляда - это гироскоп и акселерометр в шлеме ВР. С одной стороны они используются для обеспечения правильного отображения визуальных стимулов в ВР, с другой - возможно использовать их в качестве устройств ввода. Как правило, ввод направлением взгляда применяют в том случае если нет доступа к другим методам ввода. Например, шлемы Google Cardboard и Samsung GearVR без доступа к внешним контроллерам. В этом случае в приложении на экране

рисуют прицел, который пользователь наводит на нужный ему элемент интерфейса. Когда прицел наведен, пользователь должен какое-то время держать его в зоне элемента и, после срабатывания таймера, элемент интерфейса срабатывает. Конечно, такой метод ввода очень неудобен, так как пользователю приходится постоянно заботиться о направлении взгляда и напрягать шею, а также ждать срабатывания таймера каждый раз.

Массовые программные средства виртуальной реальности

Контент виртуальных миров представляет собой визуальные и аудио символы производимые устройством VR.

Аудио информация во всех популярных шлемах передается одинаково, через встроенные в систему динамики, или через подключенные к шлему наушники.

Визуальная информация представляется пользователю в форме изображения на экране/экранах шлема. Однако, способы обработки и хранения этой информации можно разделить на две группы:

- 1) Двухмерная панорамная графика
- 2) Трехмерная графика

В первом случае 2D изображение растягивается на весь сектор обзора пользователя, или только на его часть. Погружение в VR обеспечивается за счет возможности осматривать предметы вокруг. Визуальная информация может выводиться в виде статичных изображений или видео. Картинка может быть одинаковой для левого и правого глаза, или разной, для поддержания бинокулярного зрения. Бинокулярное зрение играет очень важное значение для определения глубины визуальных стимулов [1], поэтому для полноценного эффекта присутствия необходимо именно стерео-изображение. Обычно используют полный 360-градусный панорамный обзор либо 180-градусный. Неполный сектор обзора

используется потому, что пользователю зачастую неудобно и незачем смотреть на предметы за его спиной, а кроме того, такое ограничение экономит память и вычислительные ресурсы устройства ВР.

3D графика реального времени рассчитывается каждый кадр, с частотой обычно не ниже 60 Гц [2], и выводится на экран шлема. Как и в традиционной 3D графике, расчет производится на основе математической модели мира в памяти вычислительного устройства. В ВР ситуация отлична только тем, что для визуализации необходима своевременная информация о положении шлема и устройств ввода (если они есть). 3D графику, как правило, выводят только в стерео-режиме.

Для прототипов критически важна скорость. Поэтому написание низкоуровневых программ для управления шлемом ВР почти никогда не бывает допустимым. Даже более того, прямое использование программных интерфейсов (например DirectX или OpenGL) для доступа к отдельным компонентам ВР-системы сейчас избыточно. Для прототипирования требуются программные платформы максимально высокого уровня. В качестве таких платформ могут служить:

- 1) САD-системы.
- 2) Среды разработки компьютерных игр.
- 3) Браузерные API для доступа к ВР устройствам.

САD. Обычно, плагины для поддержки ВР не могут существовать сами по себе и представляют собой расширения для САD-системы. В большинстве случаев САD-системы предназначены для проектировщиков и инженеров в строго определенной сфере производства (архитектурное, промышленное проектирование), поэтому их сложно назвать массовыми и доступными для широкого круга пользователей. Хотя отметим, что наиболее универсальные и популярные системы, например такие пакеты как 3Ds Max и Blender могут применяться для создания панорамного

контента, то есть изображений и видео формата 180 и 360 градусов. Для этого достаточно только установить специальным образом настроенную камеру и произвести визуализацию требуемых кадров стандартным образом.

Среды разработки компьютерных игр. Так называемые “игровые движки”. Несмотря на название могут использоваться далеко не только для компьютерных игр. Следует отличать их от игровых конструкторов - специальных средств для создания игр. Среды разработки игр же представляют собой универсальные системы способные выводить компьютерную графику и получать информацию с устройств ввода. Поскольку это, в грубом приближении, и требуется для прототипов в VR, то игровые движки представляют собой очень перспективное направление.

Главное их преимущество именно в массовости, ведь их уже используют тысячи коммерческих организаций и индивидуальных пользователей по всему миру. Пользователи сред разработки компьютерных игр не только используют их для своих целей, но и участвуют в тестировании новых версий движка, дополняют его собственными инструментами. Степень универсальности сред разработки игр выше чем в САД. Связано это с тем, что в САД работа с логикой виртуальных объектов имеет конечной целью обычно итоговую 3D модели, чертежей, спецификаций и т.п. В компьютерных играх же целью является сам процесс игры, а следовательно и виртуальное пространство и объекты в нем могут быть какие угодно и взаимодействовать по каким угодно законам. Ниже в статье будет рассмотрена среда разработки игр Unity применительно к прототипированию.

Браузерные API. Это программные интерфейсы в глобальной сети, которые способны создать иммерсионную среду VR при помощи браузера.

Сейчас сложно представить себе компьютер, на котором не установлен браузер и у которого нет выхода в интернет. Многие шлемы VR

подключаются к компьютеру и поэтому могут использовать браузер. Автономные же устройства имеют предустановленный браузер, мобильную версию для ARM-процессоров. Сейчас почти каждый современный браузер поддерживает спецификацию WebGL, с помощью которой способен принимать команды на рендер 3D-моделей и транслировать их видеоадаптеру используя шейдерный язык GLSL. Далее, полученный кадр браузер размещает в шаблоне обычной веб-страницы.

Технология WebGL получила свою адаптацию для ВР в виде WebVR. Суть технологии та же, но только вместо холста в шаблоне используется весь спектр обзора 3D-гарнитуры. Естественно, в этом случае браузер также получает доступ к информации о положении шлема в пространстве, и о состоянии органов ввода, если они есть.

Главное преимущество браузерных технологий - в их мультиплатформенности. В идеальном случае одно и то же веб-приложение будет одинаково работать и на компьютере с x86 процессоре, и на мобильном устройстве с ARM. Недостаток вытекает из этого преимущества, ведь так как браузеры очень разнообразны, то сложно гарантировать безошибочную работу WebVR на всех устройствах и во всех браузерах. К тому же, браузер выступает еще одним слоем абстракции в приложении, что ведет к проблемам с производительностью, особенно на мобильных устройствах.

Общие подходы к созданию прототипов в виртуальной реальности

Задача создания прототипов с использованием массовых технологий ВР требует продуманной стратегии и правильного выбора оборудования и ПО. При работе над любым прототипом основные цели заключаются в том чтобы:

- 1) В том чтобы представить прототип, который будет достаточно полно отражать необходимые в данный момент свойства и функции проекта.
- 2) Представить прототип в максимально короткий срок.

Логично начать с выбора оборудования для прототипа. Мы уже рассмотрели основные типы массовых аппаратных средств ВР, все они достаточно похожи друг на друга и по сути, сейчас выбор стоит между мобильными беспроводными устройствами или проводными стационарными. Очевидно, что использование легкого беспроводного шлема в условиях сложного виртуального мира с фотореалистичной графикой и активным взаимодействием с предметами, почти невозможно. Но и использование более сложного в эксплуатации и дорогого шлема с беспроводными контроллерами для простой типовой презентации архитектурного проекта, будет избыточно.

Выбор программной платформы несколько сложнее и зависит от сути прототипа и предъявляемых к нему требований. В самых простых случаях можно обойтись просто набором из панорамных изображений и видео. В самых сложных - написать собственное приложение с интерфейсом для связи с SDK конкретного шлема.

После выбора аппаратной и программной платформы и подготовки краткого плана разработки приступают к созданию прототипа. Одна из особенностей прототипов в ВР в том, что они требуют итеративного подхода к разработке. Итеративный процесс разработки - это выполнение работ параллельно с непрерывным анализом полученных результатов и корректировкой предыдущих этапов работы.

Почему именно итеративный процесс уместен в большинстве случаев? Потому что на данный момент, ВР и феномен присутствия до конца не изучены, чрезвычайно сложно предсказать как те или иные решения повлияют на пользовательский опыт. И в то же время,

пользовательский опыт в VR намного важнее чем в традиционных виртуальных прототипах, поскольку негативный опыт будет вызывать дискомфорт и полученный прототип просто невозможно будет использовать. Благо, часто в ПО предусмотрены инструменты для быстрых итераций. Так, например, Unity позволяет подключить шлем к компьютеру и запустить проект на нем прямо из редактора.

Сравнение прототипов в Unity на базе массовых устройств VR и традиционных специализированных решений.

Для выявления преимуществ и недостатков массового VR полезно будет сравнить его возможности с тем что было до эпохи массового VR. Например, в автомобильной промышленности применение технологий VR началось в 1990-х годах [6], что гораздо раньше чем появление первого потребительского устройства VR.

Для начала, рассмотрим модель для исследований в сфере психологии [7] которая описана в работе Авербух Н.В. и была создана в Уральском Федеральном Университете. Целью модели было исследование феномена присутствия с помощью серии экспериментов. Сама модель с аппаратной стороны представлена шлемом VR Emagin Visor Z800, а с программной - компьютерной игрой GTA: San Andreas.

Про шлем Emagin Visor Z800 можно сказать то, что он во многом похож на нынешние потребительские устройства VR, но разработан и запущен в производство раньше, в 2005 году. Он достаточно часто встречается в научных исследованиях. Шлем этот стационарный, он подключается к компьютеру проводом через специальный коммутационный блок. По характеристикам вывода графики и отслеживании позиции он отстает от современных аналогов [8] [9], сравнение с популярным сейчас Oculus Rift CV1 приведено в табл. 1.

Табл. 1 Сравнение характеристик Rift CV1 и Z800

Характеристики	Oculus Rift CV1	Emagin Visor Z800
Угол обзора, градусов	110	80
Разрешение на глаз, пикселей	1080*1200	800*600
Количество степеней свободы шлема (DOF)	6	3
Частота работы трекера, Гц	1000	125

Стоит заметить, что современный шлем Oculus обычно продается с контроллерами Oculus Touch, которые позволяют использовать трехмерные “виртуальные руки” во время работы, а Z800 использует стандартные методы ввода и в отдельных случаях управление направлением взгляда.

Как ни странно, но разница в разрешении экрана современного шлема 2018 года выпуска и Z800 - 2005 года не так уж и велика. К тому же, оба шлема используют экраны на базе OLED. Примем во внимание, что угол обзора человека без шлема, когда его голова неподвижна, составляет 210 градусов, а разрешающая способность каждого глаза более чем 300000 * 300000 пикселей [1]. На этом фоне характеристики Oculus Rift не выглядят чем-то прорывным по сравнению с Z800. Наиболее серьезным оказалось отличие в скорости трекинга с степенями свободы шлема. Частота считывания данных с датчиков серьезно повышена у Oculus. Это наводит

на мысль о важности низкой задержки (Latency) для поддержания присутствия и отсутствия негативных эффектов “морской болезни”. Отсутствие же позиционного трекинга на Z800 объясняется тем, что в 2005 году просто не было достаточно эффективных и дешевых технологий для такой задачи.

Компьютерная игра - это яркий представитель специализированного программного обеспечения. В игре можно выполнять только действия заранее прописанные программистами в соответствии с ее правилами. А так как исходный код игр закрыт, то и правила изменить нельзя. В исследованиях УФУ использовалась игра GTA: San Andreas, а именно та ее часть, где игрок управляет вертолетом [7]. Можно утверждать, что программное обеспечение отвечало целям эксперимента.

Однако, игра в первую очередь служит для развлекательных целей. Поэтому, в статье [7] можно заметить некоторые эпизоды во время экспериментов, когда игрок, например, падал с вертолета и принимался бегать по городу пешком. Было ли такое поведение изначально допустимо в эксперименте? В этом возникают сомнения. И в то же время, никаких механизмов контроля среды эксперимента у психологов не было, в чем и заключается главный недостаток компьютерных игр.

В среде разработки компьютерных игр Unity изначально проект и сцена пусты и никаких игровых правил просто не существует. Исследователь сам должен создать среду эксперимента, расставляя объекты по сцене, и определить правила с помощью кода. Однако, это вовсе не значит, что разработчик должен обязательно работать с чистого листа. В открытом доступе, или за небольшую плату, существуют тысячи готовых библиотек для Unity [5]. Например, готовый летательный аппарат подходящий для данного эксперимента можно было бы взять из бесплатного пакета Standard Assets. Пакет включает не только графическую 3D-модель, но и параметрическую динамическую и физическую модель,

контроллер управления, компонент просчета столкновений с другими объектами и многое другое.

Подведем итоги, сравнив подходы к созданию моделей для психологических тестов на базе компьютерной игры и среды разработки игр в табл. 2.

Табл. 2 Сравнение компьютерной игры и среды разработки

Компьютерная игра (GTA: San Andreas)	Среда разработки игр (Unity)
<p>Среда эксперимента жестко определена игровыми правилами.</p> <p>Ее почти невозможно изменить на стадии проведения опытов.</p>	<p>Исследователь должен сам создать подходящую виртуальную среду для эксперимента.</p> <p>Среду можно изменить даже во время проведения экспериментов.</p>
<p>Требует умения играть в компьютерные игры.</p>	<p>Требует базовых навыков программирования и знания среды разработки игр.</p>
<p>Для сбора данных исследователь должен самостоятельно визуально контролировать состояние среды.</p>	<p>Возможно написать собственные классы для автоматизации сбора данных.</p>

Рассмотрим следующий случай. Разработка в MATLAB-Simulink модели визуализации в виртуальной реальности управляемого пространственного движения подводного аппарата [10]. В этом случае исследователь Скобелев М.М из МГТУ им. Н.Э. Баумана использовал приложения Virtual Reality Toolbox и VR Builder в среде MATLAB. К

сожалению, научный труд не описывает аппаратную сторону реализации ВР и то, была ли она проведена вообще, поэтому сравнивать мы будем только программные аспекты.

Virtual Reality Toolbox является расширением для MATLAB и Simulink [11]. Это специализированное программное обеспечение, так как за пределами среды MATLAB его использовать невозможно.

Изначально, автор работы создал динамическую модель движения подводного аппарата с помощью визуального программирования в MATLAB. Далее он создал 3D модель аппарата, используя программное обеспечение AutoCad, 3Ds Max, и сохранил ее в формате VRML. Затем создал виртуальную среду в Virtual Toolbox и соединил MATLAB модель с этой средой, при этом используя 3D модель как репрезентацию подводного аппарата. Модуль Simulink в данном случае использовался для симуляции управления оператором, в нем была записана последовательность управляющих сигналов.

Заявленные цели проекта были достигнуты. Но можно ли было ускорить разработку и добиться лучших результатов используя массовые средства разработки для ВР, такие как Unity? Автор считает что да, возможно. Среда разработки Unity дала бы следующие преимущества:

C# скрипты вместо визуальных схем MATLAB. Они уступают в наглядности но делают модель компактной. Компактность заключается в возможности инкапсуляции отдельных компонентов системы и повторном использовании кода. Например, посмотрим на модуль управления подводным аппаратом с джойстика:

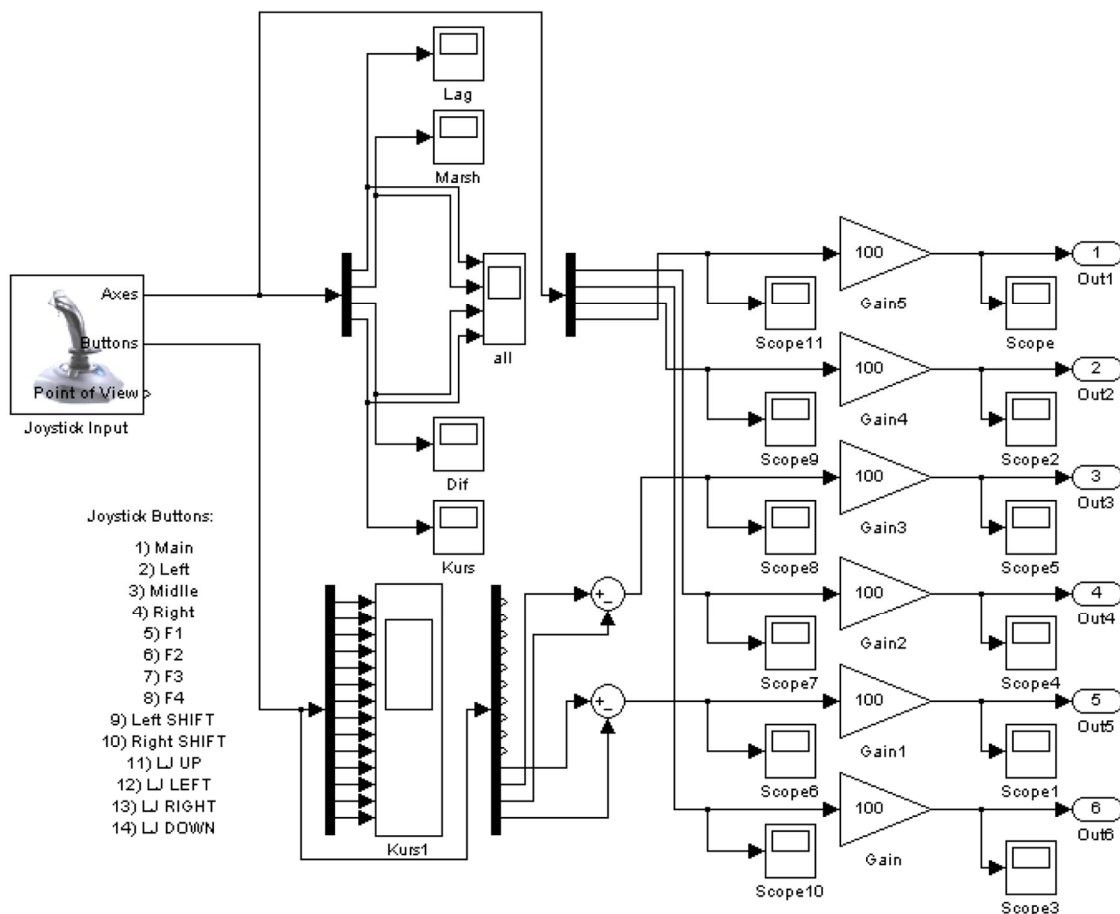


Рис. 4 Модуль джойстика в MATLAB

На этапе вывода управляющих сигналов модуль становится громоздким. Каждый выходной сигнал использует три собственные сущности типа Scope и Gain и при редактировании с каждым из них нужно работать отдельно. Для того чтобы скорректировать чувствительность джойстика нам нужно отредактировать 6 элементов Gain. В то время как в Unity можно создать класс который будет умножать чувствительность по каждому каналу на одну и ту же переменную, а все выходящие сигналы хранить в массиве или списке из 6 элементов.

У Virtual Reality Toolbox нет поддержки потребительских устройств VR на уровне ядра. У Unity есть, в случае проводных шлемов, таких как Oculus Rift, можно запустить приложение прямо из редактора Unity Editor среды и сразу выйти в виртуальную реальность. VR Toolbox

тоже возможно связать с экосистемой шлема, но для этого нужно написать собственный программный интерфейс для связи с Oculus SDK или иным интерфейсом оборудования.

VRBuilder поддерживает только формат 3D-моделей .vrmf. Формат текстовый и в данный момент устарел. Изначально формат был создан в 1994 году и предназначался для web-технологий. Идеология формата основана на подключаемых к браузеру плагинах. Сейчас же все популярные браузеры имеют встроенную поддержку WebGL для трехмерной графики и необходимость в этом формате отпала. Unity же поддерживает большинство популярных форматов 3D данных, таких как: .fbx, .obj, .blend и прочих.

Заключение

Мы рассмотрели современные массовые технологии виртуальной реальности и классифицировали аппаратные и программные средства. Мы выделили основные подходы к созданию прототипов в VR и провели сравнительный анализ современных методов с более ранними, до массового распространения VR.

Рынок VR постоянно растет, в нем почти каждый год появляются новые устройства и новое ПО. Несмотря на некоторые сложившиеся стандарты, например в сфере интерактивности, сложно предсказать каким он будет через несколько лет. Автору представляется, что будущие устройства будут конкурировать друг с другом по параметрам разрешения экрана и угла обзора, а также появятся более совершенные беспроводные технологии как для самих шлемов, так и для контроллеров.

В любом случае, чем более массовыми и дешевыми станут технологии VR, тем более интенсивно они будут применяться для

прототипирования в самых различных сферах производства, науки, культуры, образования и медицины.

Библиографический список

1. Jason Jerald, The VR Book: Human-Centered Design for Virtual Reality. Waterloo: ACM Books, 2016. 636 с.

2. Oculus Unity Documentation Archive [Электронный ресурс]. - Режим доступа: URL: <https://developer.oculus.com/documentation/unity/latest/concepts/unity-archive/> (дата обращения: 17.06.2019).

3. Лобанков И.Д. Современные концепции виртуальной реальности. Вестник ПАГС. 2015.

4. Воронов А.И. Философский анализ понятия «виртуальная реальность»: автореф. дис ... канд. филос. наук. СПб., 1999.

5. Tuliper Adam. Unity : Developing Your First Game with Unity and C# [Электронный ресурс]. - Режим доступа: URL: <https://msdn.microsoft.com/en-us/magazine/dn759441.aspx> (дата обращения: 17.06.2019).

6. Zimmermann P. Virtual Reality Aided Design: A Survey of the Use of VR in Automotive Industry. Dordrecht: Springer, 2008.

7. Авербух Н.В. Субъективный метод изучения особенностей переживания феномена присутствия. Национальный психологический журнал № 3(31), 2018.

8. Шлем виртуальной реальности eMagin Z800 3D Visor [Электронный ресурс]. - Режим доступа: URL: https://www.hwp.ru/articles/SHlem_virtualnoy_realnosti_eMagin_Z800_3D_Visor/ (дата обращения: 17.06.2019).

9. Oculus Rift [Электронный ресурс]. - Режим доступа: URL: https://en.wikipedia.org/wiki/Oculus_Rift (дата обращения: 17.06.2019).

10. Скобелев М.М. Разработка в MATLAB-Simulink модели визуализации в виртуальной реальности управляемого пространственного движения подводного аппарата. М: ФГБОУ ВПО "МГТУ им. Н.Э. Баумана", 2011.

11. Соловьев С.Г. Virtual Reality Toolbox - обзор [Электронный ресурс]. - Режим доступа: URL: <http://matlab.exponenta.ru/virtualreality/vrt.php#1> (дата обращения: 17.06.2019).