

Для начала нужно рассчитать вектор между NPC и нужным объектом путём вычитания положения цели из положения в данный момент NPC, рассчитать угол между этим вектором и направлением взгляда нашего NPC. Если абсолютное значение угла превышает заданный угол поля зрения NPC, то ваш NPC не может видеть данный объект. Нужно учитывать, что наш NPC может находиться за каким-либо объектом. Для данной проверки хорошо подойдёт построение бегущих лучей — это математический способ проверить, пересекается ли луч с какими-либо объектами, начиная со своего месторасположения и двигаясь в заданном направлении.

Для NPC важную роль играет также и слух. Если вы видите объект, значит, можете слышать его. Отслеживание шума является важным аспектом для стелс-игр. Для этого мы должны иметь полную информацию о находящихся рядом с нами объектах. Как и в случае со зрением, должно быть определённое расстояние, на котором NPC может услышать игрока. С каждым действием, которое может выполнить объект, должен быть определённый уровень звука. Для оптимизации можно записать звуки заранее либо же рассчитывать на основе разных влияющих факторов [2].

Поскольку есть слух и зрение, можно сделать и обоняние. Чтобы добавить запах, достаточно каждому объекту назначить свой номер запаха и его резкость. С течением какого-то времени запах может терять свои свойства, а дальше и вовсе пропасть. Данный аспект не столь важен для обычных NPC. Но если вы захотите добавить собаку, то её стоит научить распознавать запахи [2].

Часто в реальном мире мы сталкиваемся с временными объектами, будь то лужа или чей-то след от сапога. Чтобы наш NPC мог определить временный объект в виртуальном мире, лучше всего добавить невидимый объект, который будет виден только ИИ. Когда он будет встречать данный объект, он будет знать, что здесь произошло.

Если нам нужно, чтобы наш NPC куда-то двигался, можно воспользоваться принципом «Столкнуться и повернуть». Он является самым простым для передвижения NPC, работает следующим образом: 1) двигаться в направлении цели; 2) если столкнулся с объектом, повернуться в направлении, ближайшем к нашей цели, если нет определённого выбора, то он делается произвольно.

Алгоритм «Столкнуться и повернуть» можно запросто улучшить. Достаточно создать точку возврата, при разветвлении запомнить точку, и если один из путей окажется тупиком, вернуться к нашей точке и выбрать другой путь [2].

Если говорить об ИИ для шутеров, многие понимают, что стоять просто под огнём Bot не будет. Для расчёта, куда ему спрятаться, он будет исходить из геометрических расположений объектов и рельефа местности. Чтобы передвинуться в укрытие, он будет анализировать его размер и то, куда в данный момент стреляет игрок, дабы избежать его.

Заключение. Игровой ИИ является имитацией человеческого мышления. Исходя из вышесказанного можно понять общий принцип взаимодействия игрового ИИ с игроком, его поведение в виртуальном мире и взаимодействие с различными объектами.

Список цитируемых источников

1. *Баррат, Дж.* Последнее изобретение человечества: искусственный интеллект и конец эры Homo sapiens / Дж. Баррат ; пер. с англ. — М. : Альпина нон-фикшн, 2015. — 304 с.
2. *Шампандер, А. Дж.* Искусственный интеллект в компьютерных играх: как обучить виртуальные персонажи реагировать на внешние воздействия / А. Дж. Шампандер. — М. : ИД. Вильямс, 2007. — 764 с.

УДК 004.457

В. В. Кравченя, Е. Г. Шапович

Учреждение образования «Барановичский государственный университет», Барановичи

РАЗРАБОТКА ВИРТУАЛЬНОЙ ЭКСКУРСИИ «VR BARSU»

Введение. Развитие VR-технологии повлияло на абсолютно все сферы жизнедеятельности человека. В данный момент технологии виртуальной реальности широко применяются в различных областях человеческой деятельности: проектировании и дизайне, добыче полезных ископаемых, военных технологиях, строительстве, тренажёрах и симуляторах, маркетинге и рекламе, индустрии развлечений и т. д. [1].

В данной исследовательской работе будет производиться разработка виртуальной экскурсии по учреждению образования «Барановичский государственный университет».

Данный проект будет использоваться в большей степени в качестве рекламы. Любой человек, имеющий доступ в Интернет, сможет ввести соответствующий адрес и виртуально пройтись по ключевым местам БарГУ. Это будет способствовать интересу абитуриентов и их родителей при выборе университета.

Основная часть. Целью научно-исследовательской работы является создание виртуальной экскурсии по БарГУ с применением технологий VR.

Основные задачи при выполнении проекта: создание проекта с интуитивно понятным интерфейсом; применение в проекте VR-технологий; применение проекта веб-технологий; использование языка гипертекстовой разметки HTML5 и языка программирования JavaScript.

Для создания виртуальной экскурсии «VR BarSU» используется программное обеспечение линейки Adobe. Для обработки иллюстраций и фотографий используется программа AdobePhotoshop CC3, для прорисовки векторных изображений — программа AdobeIllustrator CC3, для создания некоторых анимированных элементов дизайна — программа AdobeFlash CC. Работа по форматированию сопровождающих текстов произведена посредством программы AdobeIndesignCC.

Использование программ одной линейки (в данном случае от компании Adobe) значительно упрощает задачу. Форматы файлов всей линейки схожи или совместимы, что делает работу приятной и удобной. Единственным исключением является программа Image Composite Editor от корпорации Windows, в которой собраны панорамы из вертикальных снимков помещений.

Для дальнейшего экспорта и сборки проекта, для его просмотра на любом современном браузере как компьютеров, так и мобильных устройств использовался язык гипертекстовой разметки HTML5 и язык программирования JavaScript. В HTML5 реализовано множество новых синтаксических особенностей. Например, элементы <video>, <audio> и <canvas>, а также возможность использования SVG и математических формул. Эти новшества разработаны для упрощения создания и управления графическими и мультимедийными объектами в сети без необходимости использования сторонних API и плагинов, например Adobe Flash Player, поддержку которого в скором будущем перестанут осуществлять все браузеры, построенные на базе Chromium [2].

Представим панель навигации работы с панорамным туром (рисунок 1). При наведении на каждую из кнопок выводится соответствующая текстовая информация о ее назначении, при нажатии — осуществляется операция.

Это стрелочки, позволяющие перемещаться левее, правее, вверх, вниз. Можно воспользоваться кнопкой, при клике на которую панорама автоматически перемещается сама. С помощью навигации можно приблизиться к определенному месту в панораме и отдалиться. Также при просмотре панорамы необязательно использовать имеющиеся кнопки. Просмотр можно осуществлять с помощью мышки. Нажимая на левую ее клавишу и ведя ее вправо, изображение тоже перемещается. Тот же принцип управления действует при желании повернуть налево, посмотреть вверх или вниз. Изображения кнопок наглядны и понятны, они не требуют дополнительных разъяснений. Также в навигации предусмотрена кнопка, кликнув на которую открывается информация о зале, в котором мы находимся в настоящий момент.

Имеется возможность перейти в полноэкранный режим просмотра панорамы, для этого тоже есть соответствующая кнопка. По факту панорама не разворачивается в имеющемся окне. С нажатием соответствующей кнопки пользователь переходит на другую страницу с панорамой большего формата. На странице с панорамой большего формата также есть навигация с тем же набором кнопок. Чтобы вернуться в обычный режим, необходимо нажать кнопку, это обозначающую.

Представим панораму по актовому залу (рисунок 2). При наведении на двери, имеющиеся в актовом зале, пользователь может осуществить переход в другое место университета. На этом же рисунке представлен результат наведения мышки на дверь, после наведения отображается название перехода.



Рисунок 1 — Панель навигации

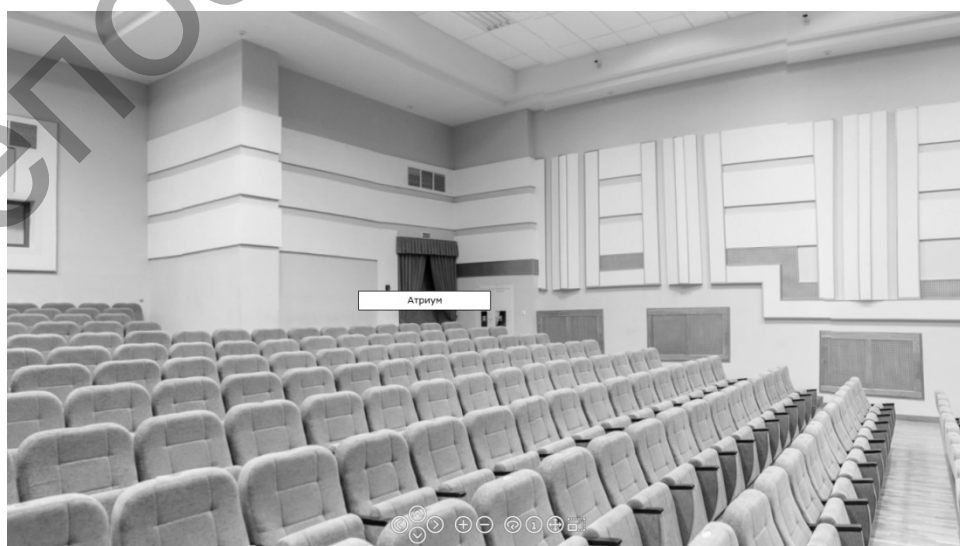


Рисунок 2 — Панорама «Актовый зал»

Виртуальная экскурсия включает в себя тур более чем по 50 помещениям университета. Все связи тура проработаны так, чтобы не было большой «вложенности» помещений. Таким образом, пользователь всегда легко сможет найти главную страницу тура и не потеряться.

Заключение. С развитием новых технологий и новых возможностей у человека появляются и новые потребности. Потребности в более интересно представленной информации, в доступности этой информации. Это касается и проявления интереса людей к архитектуре и материально-технической базе университета. Созданное приложение достаточно интуитивно понятное в управлении, представленная в нем информация максимально наглядная. Связи между страницами и разделами логичны, меню легкодоступное. Применение данного приложения на выставках, днях открытых дверей сделает эти мероприятия еще более интересными. Можно отметить, что результаты научно-исследовательской работы будут способствовать повышению результатов профориентационной работы, а также повышать престиж учреждения образования.

Список цитируемых источников

1. Виртуальная реальность [Электронный ресурс]. — Режим доступа: <http://www.nvidia.ru/object/virtual-reality-technology-ru.html>. — Дата доступа: 12.02.2018.
2. Лоусон, Б. Изучаем HTML5. Библиотека специалиста / Б. Лоусон, Р. Шарп. ; пер. Т. Качковская, Е. Шикарева. — СПб. : Питер, 2011. — 272 с.

УДК 004.42

С. Ю. Мальчиков, А. И. Калько

Учреждение образования «Барановичский государственный университет», Барановичи

РАЗРАБОТКА ОБОЛОЧКИ ДЛЯ ВЫПОЛНЕНИЯ СТУДЕНЧЕСКИХ ТЕСТОВ

Введение. Целью работы является разработка оболочки для тестирования, которая бы содержала в себе вопросы по курсу дисциплины, разработанные преподавателем.

Задачами данного проекта являются: создание базы данных для хранения данных теста; разработка клиент-серверного приложения с доступом к базе данных, которое будет удобно в использовании.

Актуальность выбранной темы заключается в том, что прогресс не стоит на месте и входит в каждую сферу жизнедеятельности человека, в том числе и в сферу образования. Одним из таких нововведений является тестирование, которое необходимо уметь не только разработать, но и создать.

Основная часть. Язык программирования, на котором реализована система, заслуживает большого внимания, так как разработчик будет погружен в него с начала конструирования и до самого конца.

Синтаксис объектно ориентированного языка Java, разработанного Sun Microsystems, Inc., напоминает C и C++. Java — платформенно-независимый язык: исходный код Java сначала преобразуется в байт-код, который может выполняться на любой платформе в среде, известной как «виртуальная машина». Java широко используется для создания веб-приложений [1].

Каждый программист имел дело исключительно с последовательным программированием. Все действия, выполняемые программой, выполнялись друг за другом, т. е. последовательно.

Последовательное программирование способно решить многие задачи. Однако для некоторых задач бывает удобно организовать параллельное выполнение нескольких частей программы, чтобы создать у пользователя впечатление одновременного выполнения этих частей, или, если на компьютере установлено несколько процессоров, чтобы они действительно выполнялись одновременно [2].

Многопоточное программирование может рассматриваться как стратегия устранения привязок. Оно помогает отделить выполняемую операцию от момента её выполнения. В однопоточных приложениях «что» и «когда» связаны так сильно, что просмотр содержимого стека часто позволяет определить состояние всего приложения. Программист, отлаживающий такую систему, устанавливает точку прерывания и узнаёт состояние системы на момент остановки [3].

Архитектура клиент-сервер предъявляет специфические требования как к клиенту, так и к серверу. Программа, удовлетворяющая этим требованиям, может считаться клиент-серверным приложением, выполняющим распределенную обработку данных [4].

Разработка оболочки тестов стала актуальной темой для многих университетов и одновременно доступной для простых пользователей. Данное приложение позволит пользователям создавать и проходить тестирование на различные темы.

Разработано сетевое приложение «Оболочка студенческих тестов». В серверной части приложения реализовано создание, редактирование, управление и запуск тестов. Задача тестирующего — создать тест, по возможности его отредактировать, далее необходимо по сети раздать тест тестируемым, после чего тест будет выполнен, тестирующий имеет право его удалить или оставить для дальнейшего использования. В клиентской части доступно только прохождение теста, где тестируемый получит вопросы и ответит на них, при желании он может их пропустить. Время тестирования ограничено. Представим интерфейс клиентской части (рисунок 1).