УДК 004.946

РАЗРАБОТКА АДАПТИВНОЙ ИНТЕРАКТИВНОЙ МНОГОПОЛЬЗОВАТЕЛЬСКОЙ VR-ЭКСКУРСИИ НА ПРОИЗВОДСТВЕННЫЙ ОБЪЕКТ С ИСПОЛЬЗОВАНИЕМ СРЕДЫ UNITY

Дощинский М.С. (филиал МАГУ в г. Апатиты.)
Научный руководитель – кандидат технических наук, **Тоичкин Н.А.**
(филиал МАГУ в г. Апатиты.)

Введение. В производственных помещениях/локациях горнодобывающих предприятий и других крупных производственных объектов законодательно установлены возрастные ограничения на проведения образовательных и познавательных экскурсий, с целью ознакомления с производственным циклом и условиями труда. Однако данное посещение может быть полезно потенциальным работникам, в частности, обучающимся студентам и школьникам в плане выбора дальнейшей профессиональной деятельности. В связи с этим актуальной является задача разработки VR-экскурсий в такие потенциально опасные места, как, например, шахты закрытого типа, в которых осуществляется разработка различных полезных ископаемых, в частности, добыча минерала апатит на предприятиях компании ФосАгро. Таким образом, для решения вышеупомянутой проблемы принято решение совместно с корпоративным учебным центром компании ФосАгро разработать интерактивную многопользовательскую VR-экскурсию, основанную на снимках, сделанных в формате 360˚, для VR-платформы от производителя HTC [1].

Основная часть. Для разработки VR-экскурсии было принято решение использовать существующий игровой движок, что помогает сэкономить ресурсы на создание физического и графического движков с нуля, и позволяет создать надежное программное решение на основе проверенной технологии разработки. После анализа существующих решений, таких как Unreal Engine, Godot, Unity, Cry Engine, за основу разработки был взят Unity из-за низкого порога вхождения, большого количества обучающего контента и встроенной возможности разработки под VR.
Процесс создания VR приложений с использованием Unity можно представить с помощью следующих шагов:
1). После установки среды Unity, требуется создать проект с VR-шаблоном. В этом случае Unity сразу импортирует все необходимые для работы библиотеки. Продукты компании HTC основаны на открытом стандарте OpenXR [2], который необходимо подключить в настройках проекта.
2). Создается класс *InputManager*, который в начале работы приложения осуществляет поиск и фиксирует устройства ввода (контроллеры), после чего при необходимости считывает и преобразовывает данные ввода с контроллеров, предоставляя другим классам доступ к ним, в виде свойств и событий. Это избавляет от необходимости поиска устройств ввода в других классах и, следовательно, от дублирования кода.
Для реализации передвижения была использована библиотека NavMeshComponents [3]. Основными ее классами, необходимыми в разработке являются *NavMeshAgent* – класс, хранящий все характеристики передвигаемого объекта и отвечающий за передвижение и поворот объекта в пространстве, и *NavMeshSurface* – класс, который запекает на сцене поверхность, по которой в дальнейшем будет передвигаться *NavMeshAgent*. Передвижение осуществляется на основе значений двухосевого джойстика на контроллере.
3). Далее реализуется логика смены локаций. Т.к. использование панорамных изображений в 360 градусов исключает смену локации посредством передвижения, было принято решение разработать логику их принудительного упорядоченного переключения на основе смены сцен

с использованием класса *SceneManager* из библиотеки SceneManagement. Таким образом, здесь реализовано два класса: *LevelsData* – хранилище сцен, имеющее свойства и методы для получения сцен другими классами и *LevelsLoader* – компонент, являющийся связующим между вводом и хранилищем, так же выполняющим непосредственное переключение сцен. Переключение осуществляется после нажатия определенной кнопки на контроллере.
4). Следующим шагом необходимо реализовать систему локального соединения. Для этого подходит библиотека Photon [4]. На все объекты, которые должны быть синхронизированы на сцене добавляется компонент *PhotonView*. Весь код, который должен быть синхронизирован так же перерабатывается под библиотеку Photon, для чего применяются специальные методы, при вызове которых исполнение методов происходит для объекта с определенным *ViewID* на всех клиентах, подключенных к одной комнате.
5). В завершении создается меню, в котором будет три управляющих кнопки: подключение к комнате, создание комнаты, выход из приложения. Для взаимодействия с ними реализуется «лазерный указатель», привязанный к контроллеру. Когда лазерный указатель попадает в триггер кнопки, она считается активной и подсвечивается соответствующим цветом, а при ее нажатии, на контроллере исполняется логика, закрепленная за активной кнопкой.

Выводы. Таким образом, было разработано многопользовательское приложение VR-экскурсии, готовое к интеграции контента и последующему использованию. В дальнейшем рассматривается возможность написания внешнего редактора контента, исключающего необходимость использования Unity для создания новых экскурсий.

Список использованных источников:
1. Vive Pro - VR шлем профессионального уровня. URL: https://www.vive.com/ru/product/vive-pro-full-kit/ (дата обращения: 15.02.2023).
2. Лендинговая страница описания стандарта OpenXR. URL: https://www.khronos.org/openxr/ (дата обращения: 15.02.2023).
3. Программный код библиотеки NavMeshComponents. URL: https://github.com/Unity-Technologies/NavMeshComponents/tree/master/Assets (дата обращения: 15.02.2023).
4. Библиотека Photon. URL: https://assetstore.unity.com/packages/tools/network/pun-2-free-119922 (дата обращения: 15.02.2023).