**Липецкий государственный технический университет**  
Факультет автоматизации и информатики  
Кафедра автоматизированных систем управления

Лабораторная работа №4

по предмету «Организация графических систем и систем мультимедиа»

Обработка потокового виде/аудио.

Максимов А.В.

Студент

Группа М-АС-21

Кургасов В.В.

Руководитель

Доцент

Липецк 2022 г.

Задание кафедры

Обработка потокового виде/аудио

Содержание

[Теоретическая часть 4](#_Toc103162793)

[Ход работы 4](#_Toc103162794)

[Вывод 9](#_Toc103162795)

[Приложение А 10](#_Toc103162796)

[Приложение Б 12](#_Toc103162797)

Теоретическая часть

Unity - межплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие

AR Foundation — это абстракция над несколькими технологиями дополненной реальности, в числе которых ARKit, ARCore, Magic Leap, HoloLens и другие. AR Foundation позволяет писать один код сразу для многих платформ.

ARCore, также известный как Google Play Services for AR, представляет собой набор для разработки программного обеспечения, разработанный Google, который позволяет создавать приложения дополненной реальности.

ARCore использует три ключевые технологии для интеграции виртуального контента с реальным миром, видимым через камеру смартфона или планшета:

* Шесть степеней свободы позволяют телефону понимать и отслеживать свое положение относительно мира.
* Понимание окружающей среды позволяет телефону определять размер и расположение плоских горизонтальных поверхностей, таких как земля или кофейный столик.
* Оценка освещенности позволяет телефону оценивать текущие условия освещения окружающей среды.

ARCore был интегрирован во множество устройств.

Ход работы

Для разработки приложения был использован игровой Unity вместе с библиотекой ARFoundation, предназначенной для работы с технологиями дополненной реальности.

Для начала создадим объект, который будет помещён в дополненную реальность. Объект представлен на рисунке 1.

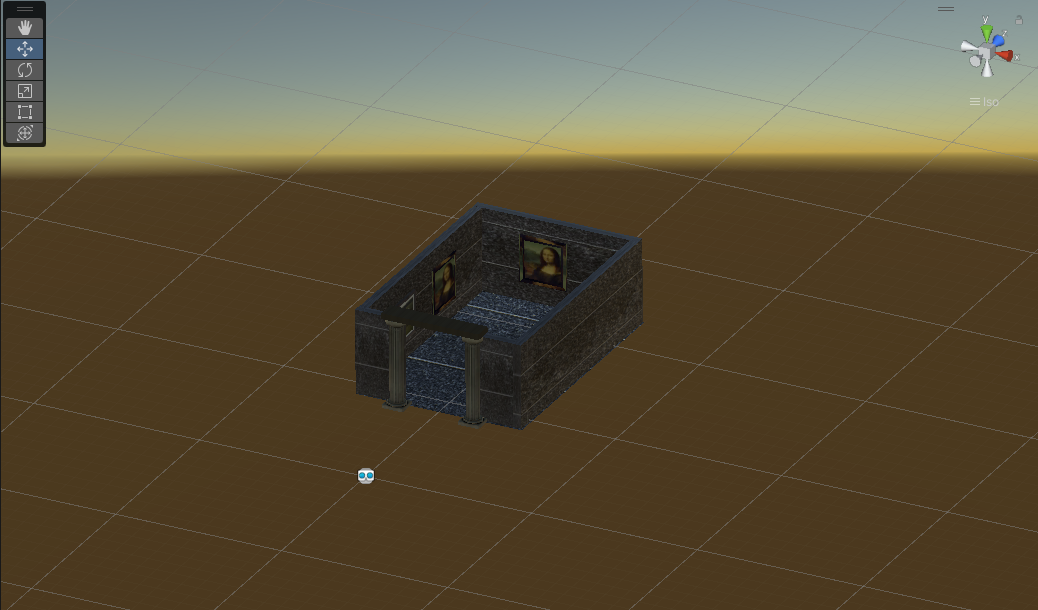


Рисунок 1 - Создание 3D объекта



Рисунок 2 – Структура объекта

После создания 3D объекта добавим на цену куб, который будет использован для сокрытия всей сцены, кроме входа. На куб добавим маскирующий шейдер. Код шейдера представлен в приложении Б.

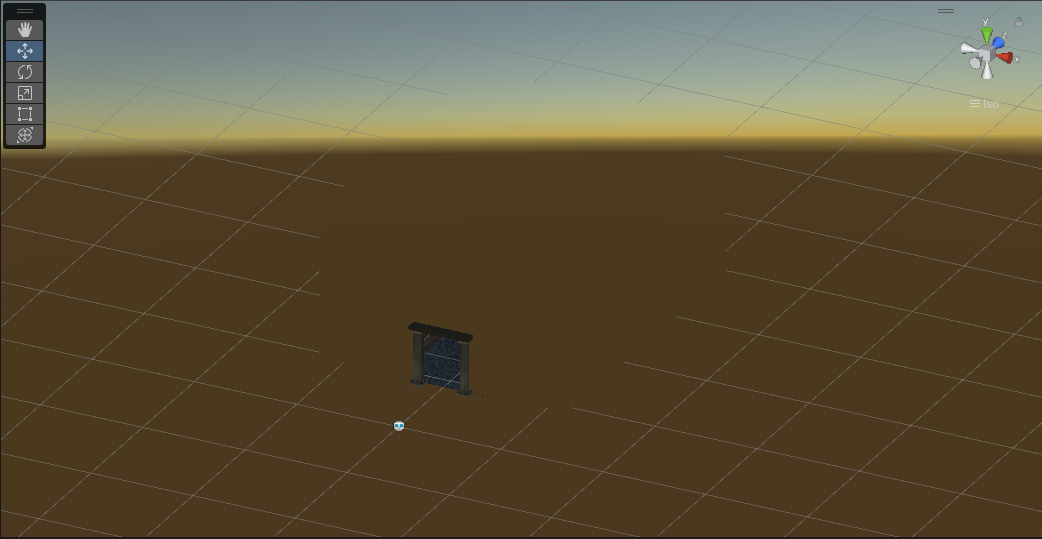


Рисунок 3 – 3D объект после добавления маскирующего куба

Далее создадим два объекта: AR Session и AR Session Origin (см. рисунок 5)



Рисунок 4 – Структура проекта после добавления AR Session и AR Session Origin

Целью AR Session Origin является преобразование отслеживаемых объектов, таких как плоские поверхности и характерные точки, в их конечное положение, ориентацию и масштаб в сцене Unity. Поскольку устройства AR предоставляют свои данные в «пространстве сеанса», которое является немасштабируемым пространством относительно начала сеанса AR, AR Session Origin выполняется соответствующее преобразование в пространство Unity.

AR Session отвечате за положение телефона пользователя и фиксирует перемещение человека в пространстве.

К AR Session Origin подключаем компоненты, показанные на Рисунке 5.

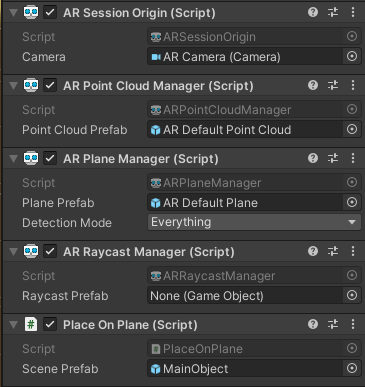


Рисунок 5 – Компоненты AR Session Origin

AR Point Cloud Manager создаёт облака точек, наборы характерных точек. Характерная точка - это определенная точка в облаке точек, которую устройство использует для определения своего местоположения в мире. Характерные точки обычно представляют собой примечательные элементы среды, которые устройство может отслеживать между кадрами.

Облако точек - это набор характерных точек, которые могут меняться от кадра к кадру. Некоторые платформы создают только одно облако точек, в то время как другие объединяют свои характерные точки в разные облака точек в разных областях пространства.

Облако точек считается отслеживаемым, а отдельные характерные точки - нет. Однако характерные точки могут быть однозначно идентифицированы между кадрами, поскольку они имеют уникальные идентификаторы.

AR Plane Manager создаёт объект класса GameObject для каждой обнаруженной плоскости в среде. Плоскость — это плоская поверхность, представленная позой, размерами и граничными точками. Граничные точки выпуклые.

Примерами элементов среды, которые могут быть обнаружены как плоскости, являются горизонтальные столы, полы, столешницы и вертикальные стены.

Скрипт Place on Plane помещает 3D объект в пространство дополненной реальности. Код скрипта представлен в приложении А.

Вывод

В ходе выполнения лабораторной работы было разработано приложение, позволяющее пользователю поместить 3D объект в пространство дополненной реальности. Для разработки использовался язык программирования C#, межплатформенная среда разработки игр Unity и библиотека для работы с дополненной реальностью AR Foundation. Программа была собрана под мобильную операционную систему Android.

Приложение А

Скрипт для отображения 3D объекта в дополненной реальности

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.XR.ARFoundation;

using UnityEngine.XR.ARSubsystems;

public class PlaceOnPlane : MonoBehaviour

{

ARRaycastManager ARRaycastManager;

private Vector2 touchPosition;

public GameObject ScenePrefab;

static List<ARRaycastHit> Hits = new List<ARRaycastHit>();

private void Awake()

{

ARRaycastManager = GetComponent<ARRaycastManager>();

ScenePrefab.SetActive(false);

}

// Start is called before the fist frame update

private void Start()

{

}

// Update is called once per frame

private void Update()

{

if (Input.touchCount>0)

{

touchPosition = Input.GetTouch(0).position;

if (ARRaycastManager.Raycast(touchPosition, Hits, TrackableType.PlaneWithinPolygon))

{

var hitPose = Hits[0].pose;

ScenePrefab.SetActive(true);

ScenePrefab.transform.position = hitPose.position;

LookAtPlayer(ScenePrefab.transform);

}

}

}

void LookAtPlayer(Transform scene)

{

var LookDirection = Camera.main.transform.position - scene.position;

LookDirection.y = 0;

scene.rotation = Quaternion.LookRotation(LookDirection);

}

}

Приложение Б

Шейдер для маскировки объекта

Shader "Custom/Mask"

{

SubShader{

Tags

{

"Queue" = "Geometry-1"

}

ZWrite On

ColorMask 0

Pass {}

}

}