МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЁТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Асс. | |  |  | | | |  | | Е. Е. Майн |
| должность, уч. степень, звание | |  | подпись, дата | | | |  | | инициалы, фамилия |
| ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №3 | | | | | | | | | | |
| Маркерная дополненная реальность на базе ARCORE | | | | | | | | | | |
| по дисциплине: ВИРТУАЛЬНАЯ И ДОПОЛНЕННАЯ РЕАЛЬНОСТЬ | | | | | | | | | | |
|  | | | | | | | | | | |
| РАБОТУ ВЫПОЛНИЛА | | | | | | | | | | |
| СТУДЕНТКА ГР. | 4931 | | |  | 02.11.2022 |  | | Е.Ю. Ильченко | | |
|  |  | | |  | подпись, дата |  | | инициалы, фамилия | | |
|  |  | | |  |  |  | |  | | |

Санкт-Петербург 2022

**Цель работы:** Знакомство с пакетом разработки Google ARCore для дополненной  
реальности. Подготовка проекта Unity для работы с ARCore, создание простого приложения с использованием технологии маркерной дополненной реальности и реализация простейшего взаимодействия с объектами сцены через экран смартфона.

Порядок выполнения лабораторной работы  
1) Создать новый проект Unity с пресетом 3D.  
2) Установить необходимый набор плагинов ARCore Extensions для работы  
с AR Foundation.  
3) Настроить плеер Unity под особенности мобильного AR-приложения.  
4) Создать префаб примитива, который будет проецироваться на маркер  
(согласно варианту задания в таблице 1).  
5) Подобрать подходящее изображение для маркера и добавить его в сцену.  
6) Реализовать функцию проецирования объекта на экране телефона при  
наведении камеры на маркер.  
7) Добавить функцию изменения свойств спроецированного объекта при  
касании экрана смартфона (согласно варианту задания в таблице 1).  
8) Скомпилировать проект в файл .apk и установить его на телефон.  
9) Продемонстрировать результат преподавателю.  
10) Сформулировать выводы и оформить отчёт.

**Вариант задания**

Таблица 1 – Вариант задания ЛР

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Используемый примитив** | **Действие при касании** |
| 11 | Цилиндр | Изменение размера |

**Ход работы**

**Создание и настройка проекта**

Установили необходимый набор плагинов ARCore Extensions для работы с AR Foundation.(см. рис.1-2)

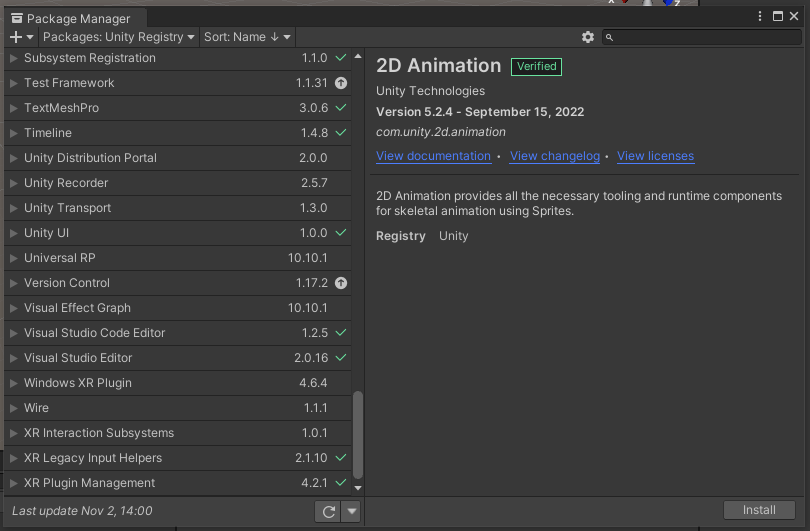
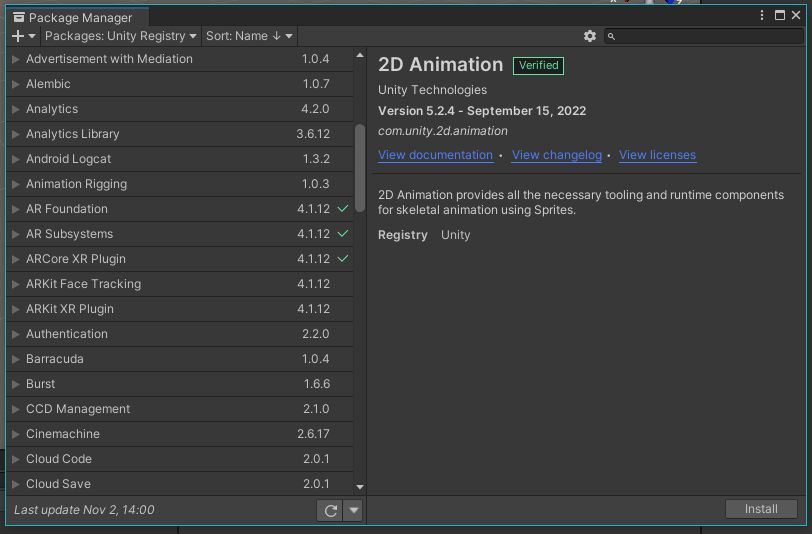


Рис.1-2 Установка необходимых пакетов

Произвели необходимую настройку проекта Unity для сборки под Android (см. рис. 3 -4)

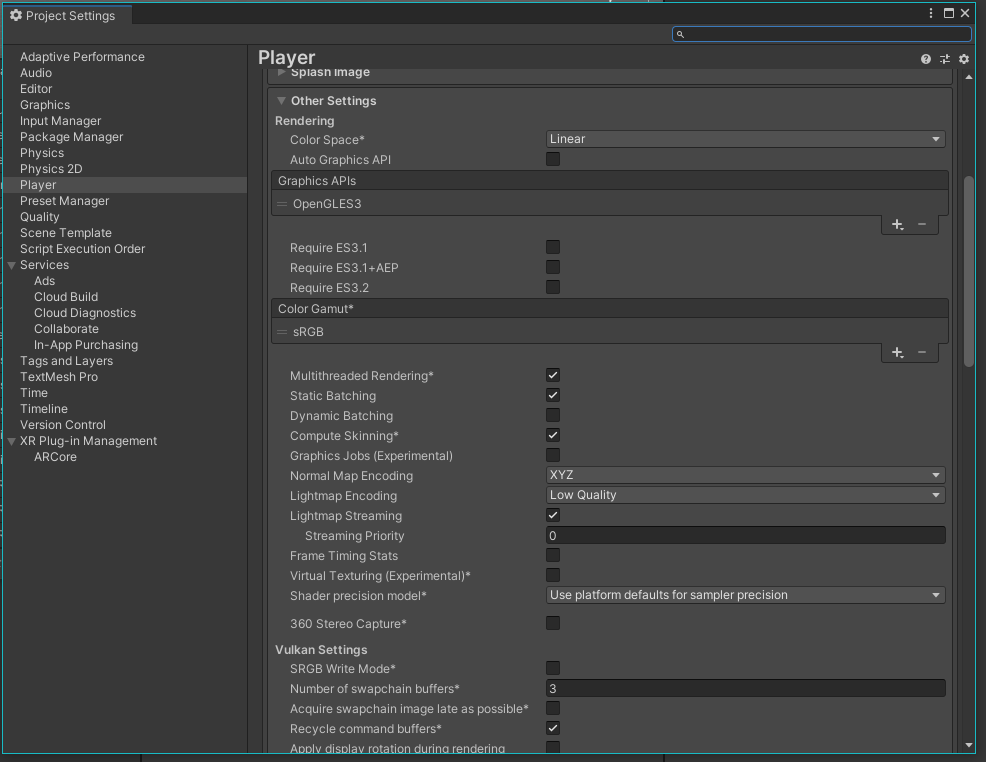
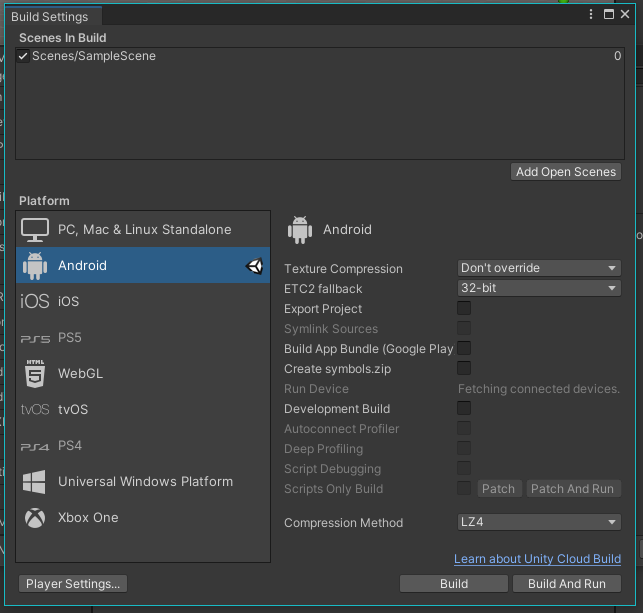


Рис.3 – 4 – Настройка проекта

**Создание и настойка AR – окружения**

Поскольку приложение дополненной реальности будет работать с камерой смартфона, штатная камера Unity под названием «Main Camera» здесь не нужна. Удалили её со сцены. Вместо неё будет используем специальную AR-камеру (см. рис. 5).

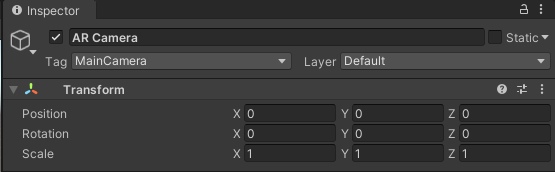


Рис. 5 – Подготовка AR-камеры

Добавили в сцену объекты «XR –>AR Session Origin» и «XR –> AR Session» (см.рис.6).



Рис.6 – Добавление в сцену AR Session Origin и AR Session

**Проецирование объекта**

Подготовили проецируемы объект (см.рис.7-8).Создали цилиндр.



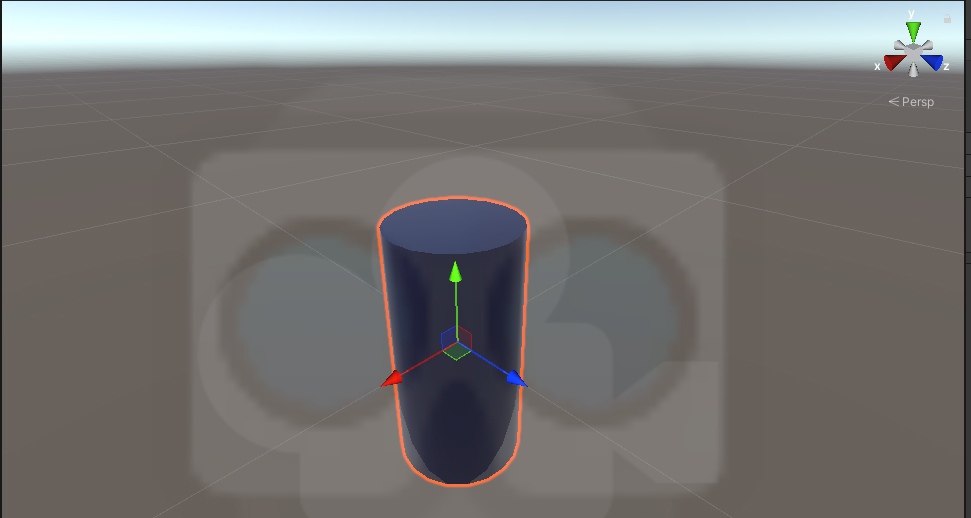


Рис. 7 – 8 – Создание и настройка проецируемого объекта

Добавьте в ассеты проекта папки «Prefabs» и «Materials» (см.рис.9).

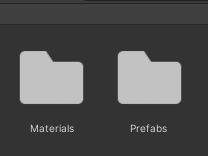


Рис. 9 – Добавление папок «Prefabs» и «Materials»

Создали новый материал и применили его к цилиндру (см. рис. 10).

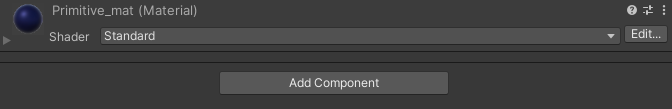


Рис.10 – Добавление материала цилиндра

Уменьшили цилиндр в размерах так, чтобы был сопоставим с реальными  
размерами маркера на экране (см.рис.11).

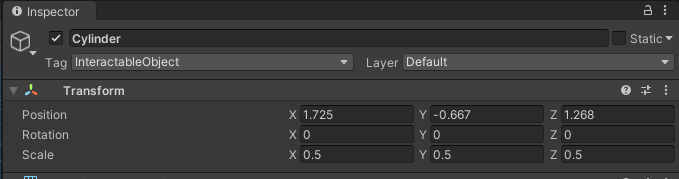


Рис. 11 – Изменение размеров цилиндра

Сохранили примитив в префаб и удалили его со сцены (см.рис.12).

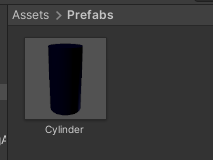


Рис.12 – Сохранение цилиндра в префаб

**Взаимодействие с объектами**

Отслеживать объекты сцены проще всего по их тэгу (Tag). Поэтому  
задайли нашему примитиву новый тэг «InteractableObject» (см.рис.13).

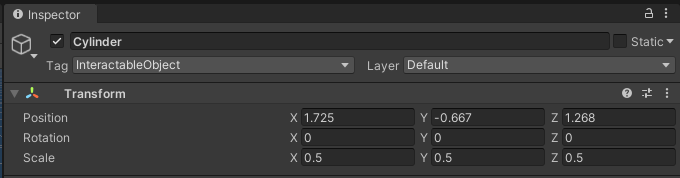


Рис. 13 – Добавление тега для цилиндра

В ассетах проекта создали папку «Scripts» и добавьте туда новый скрипт C# под названием «ObjectRaycast» (см.рис.14).

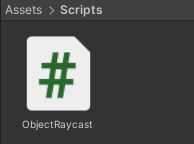


Рис. 14 – Добавление скрипта

Применили его к объекту «AR Session Origin». В Инспекторе назначили AR-  
камеру и активный объект (см.рис.15)

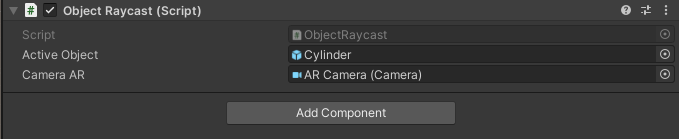


Рис.15 – Применение скрипта и определение переменных

Добавили примитиву компонент «Rigidbody» для придали ему физические  
свойства. Уберали гравитацию (параметр «Use Gravity») и сделали примитив  
кинематическим («Is Kinematic») (см.рис.16).

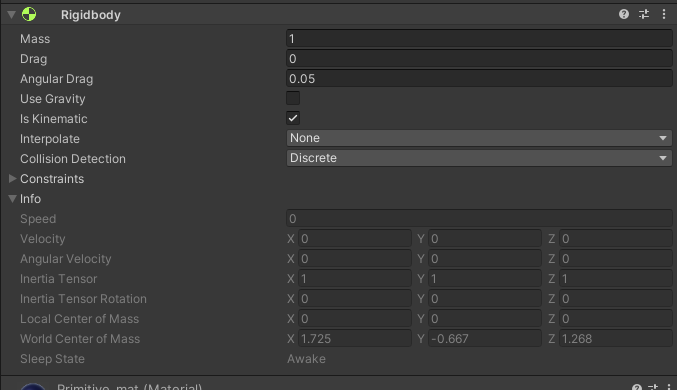


Рис.16 – Настройка физики примитива

Демонстрация объекта (см. рис. 17- 19)

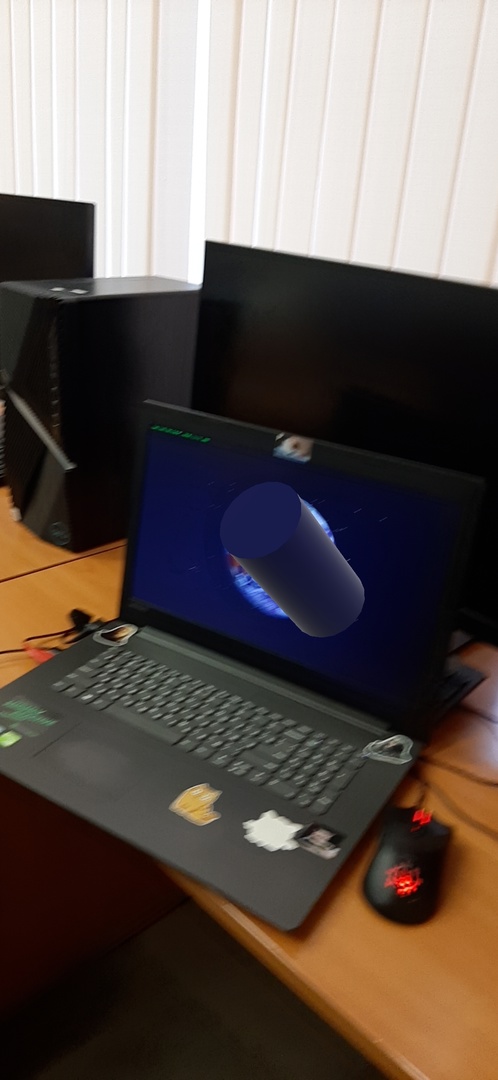


Рис.17 – Результат работы до нажатия на цилиндр



Рис.18 – Результат работы после нажатия на цилиндр



Рис.19 – Результат работы после нажатия на цилиндр

**Скрипт**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class ObjectRaycast : MonoBehaviour

{

//--------------ПЕРЕМЕННЫЕ----------------------------

// Объект, который будет реагировать на рейкаст

bool size = false;

public GameObject activeObject;

private AudioSource \_audioSource;

// Определяем камеру, из которой будут исходить лучи

public Camera cameraAR;

void Update()

{

// Поиск объекта в сцене по тэгу

activeObject =

GameObject.FindGameObjectWithTag("InteractableObject");

// Получаем компонент Рендерер аrтивного объекта (куба)

var cubeRenderer = activeObject.GetComponent<Renderer>();

// Если пользователь хотя бы раз нажал на экран

if (Input.touchCount > 0)

{

// Записываем в переменную информацию о касании

Touch touch = Input.GetTouch(0);

// Смотрим фазу касания – она должна быть начальной

if (touch.phase == TouchPhase.Began)

{

// Если касание есть, из камеры посылается луч в

Ray ray = cameraAR.ScreenPointToRay(touch.position);

// Сюда записываются объекты, в которые попал луч

RaycastHit hitObject;

if (Physics.Raycast(ray, out hitObject))

{

// Если луч попал в цилиндра

if (hitObject.transform.gameObject == activeObject)

{// Меняем размер цилиндра

if

(activeObject.gameObject.transform.localScale.x <= 0.1f)

size = true;

if

(activeObject.gameObject.transform.localScale.x >= 0.5f)

size = false;

if (!size)

activeObject.gameObject.transform.localScale += new Vector3(-0.2f, -0.2f, -0.2f);

else

activeObject.gameObject.transform.localScale += new Vector3(0.2f, 0.2f,0.2f);

}

}

}

}

}

}

**Выводы**

В ходе выполнения лабораторной работы были получены навыки работы с   
пакетом разработки Google. Был подготовлен проект Unity для работы с ARCore, создано простое приложение, реализовано взаимодействие с объектами сцены с помощью экрана смартфона.