

Лабораторная работа № 2. Проект – разработка простого AR-Приложения для Android-устройства (смартфон, планшет и пр.). Создание в графическом редакторе Unity 3D сцены дополненной реальности: визуализация **видеоклипа**.

Объекты ДР – это объекты проекта, создаваемого с помощью средств платформы **Vuforia**.

Введение.

Работа по созданию приложений ДР заключается в заведении проекта и объектов проекта (Контент) в **Vuforia**, а разработка 3D-сцен для объектов этого проекта осуществляется в **Unity 3D**. При этом **Vuforia** отвечает за идентификацию проекта через **License key** (см. ниже), а привязка к будущей сцене виртуального 2D- или 3D-объекта (например, 3D-модели, плоских изображений, видеоклипов и пр.) будет осуществляться через определяемую в **Vuforia** метку (**Target**). Допустимые в используемой в Лабораторном практикуме версии **Vuforia Engine** типы таргетов будут подробно рассмотрены ниже.

ВАЖНО!! → вся работа с **Vuforia** (с проектом, объектами) осуществляется через **web-интерфейс**, иными словами, **Vuforia** является облачным приложением. А работа с **Unity-3D** осуществляется непосредственно на компьютере разработчика, т.е. локально.

Связь между облачным ведением проекта (в **Vuforia**) и локальной проработкой сцен Приложения ДР должна быть выполнена за счет импорта подготовленных объектов проекта из облака **Vuforia** в среду редактора **Unity-3D**.

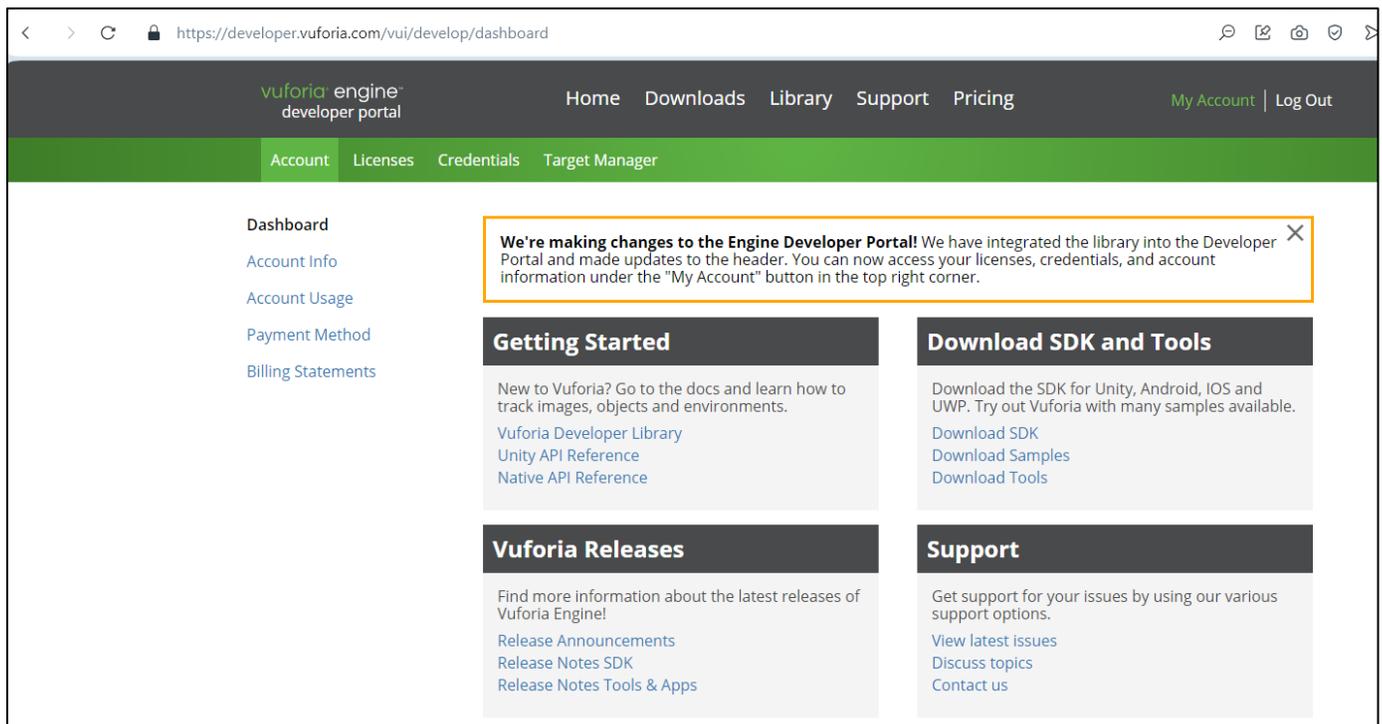
Рассмотрим типовую процедуру создания простого (игрового) **AR Приложения**.

Предлагается разработать приложение ДР для Android-устройств, в котором при наведении камеры устройства на реальную метку (таргет – изображение, например, на бумаге, или на дисплее) пользователь в области воспроизведения на экране мобильного устройства (МУ) увидит другое, заранее подготовленное 2D – изображение (лист инструкции по эксплуатации, пояснение, другую картинку, **видеоклип и т.п.**)

Предварительные условия для начала работы:

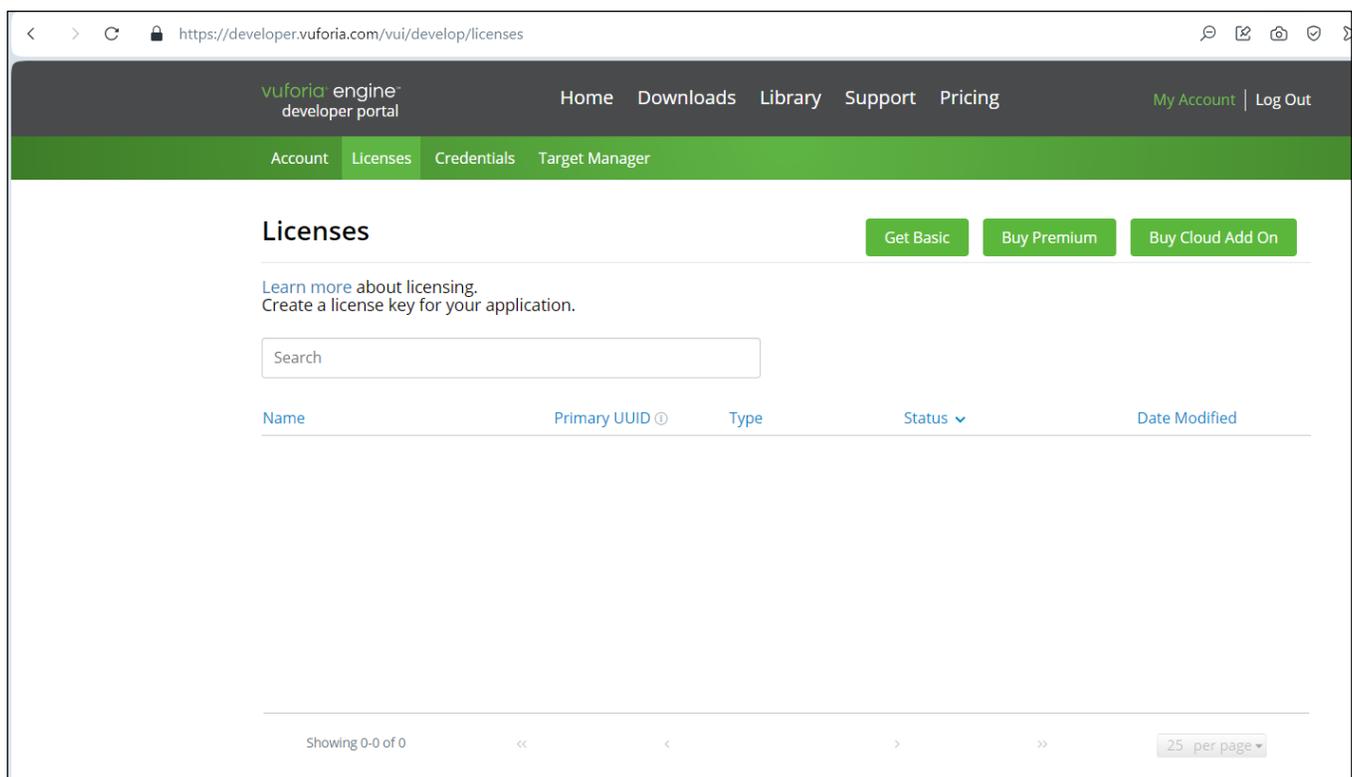
- Интернет-соединение локального компьютера;
- Наличие аккаунта пользователя **Vuforia** (результат успешного выполнения ЛР №1);
- Установленная на компьютере разработчика система **Unity 3D** (результат успешного выполнения ЛР №1);
- Заранее подготовленные изображения для метки (таргета) и контент (видеоклип);

На сайте <https://developer.vuforia.com/> осуществляем вход (не забывайте про **VPN**) с заведенными ранее логином/паролем. В результате получаем доступ к среде разработки **Vuforia** (облачное решение). Для ведения проекта средствами **Vuforia** необходимо выполнить две процедуры: получить **лицензионный код** на проект и сгенерировать метки (**Target**).



В открывшемся окне доступны две вкладки для последовательного выполнения этих процедур – **Licenses** и **Target Manager**.

1. Получение лицензии Vuforia. Переходим на вкладку **Licenses** для получения лицензии на создание Приложения ДР в среде **Unity**. Если ни одного проекта еще не создано, то в открывшемся окне по ссылке **Licenses** пусто:



Выбираем **Get Basic**, в результате переходим в режим создания лицензионного ключа:

The screenshot shows a web browser window with the URL <https://developer.vuforia.com/vui/develop/licenses/free/new>. The page header includes the Vuforia logo and navigation links: Home, Downloads, Library, Support, Pricing, My Account, and Log Out. A secondary navigation bar contains Account, Licenses, Credentials, and Target Manager. The main content area features a 'Back To Licenses' link, the heading 'Add a license key to your Basic plan', and a text input field for 'License Name *'. Below the field is the text 'You can change this later'. A checkbox is present with the text: 'By checking this box, I acknowledge that this license key is subject to the terms and conditions of the [Vuforia Developer Agreement](#)'. At the bottom are 'Cancel' and 'Confirm' buttons.

ВАЖНО!! В зависимости от выбранной версии **Vuforia Engine** на данном экране может отобразиться информация с характерными для студенческого режима использования **Vuforia** ограничениями. В этом случае для нас единственно важным является: сколько актов распознавания метки (**Reco**) можно выполнить в течение месяца. 1000 для некоммерческого использования – более чем достаточно.

В открывшемся поле необходимо ввести уникальное имя. В рамках данной ЛП вам достаточно одной лицензии (**License Name**), одной Базы данных меток (**Target Database**), одного проекта в **Unity 3D**. Исходя из этого для всех трех объектов **предлагается** выбрать одно и то же имя. Например – **AR-001**. Не забудьте отметить галочкой «**By checking this box**» и нажмите появившуюся клавишу **Confirm**.

This screenshot is identical to the one above, but the 'License Name *' input field now contains the text 'AR-001'. The checkbox below the field is now checked, indicating that the user has acknowledged the terms and conditions of the Vuforia Developer Agreement. The 'Confirm' button is now active.

В результате в списке лицензионного менеджера появится новая строка:

https://developer.vuforia.com/vui/develop/licenses

vuforia engine™ developer portal

Home Downloads Library Support Pricing My Account | Log Out

Account Licenses Credentials Target Manager

Licenses

Get Basic Buy Premium Buy Cloud Add On

Learn more about licensing.
Create a license key for your application.

Search

Name	Primary UUID ⓘ	Type	Status ▾	Date Modified
AR-001	N/A	Basic	Active	Feb 24, 2024

Для получения лицензионного ключа необходимо кликнуть по имени/ссылке **AR-001** в поле **Name**. В результате вы получите **license key**:

https://developer.vuforia.com/vui/develop/licenses/691bb30202304fdea4754719a8812189/info

vuforia engine™ developer portal

Home Downloads Library Support Pricing My Account | Log Out

Account Licenses Credentials Target Manager

Licenses > AR-001

AR-001

Edit Name Delete License Key

License Key Usage

Please copy the license key below into your app

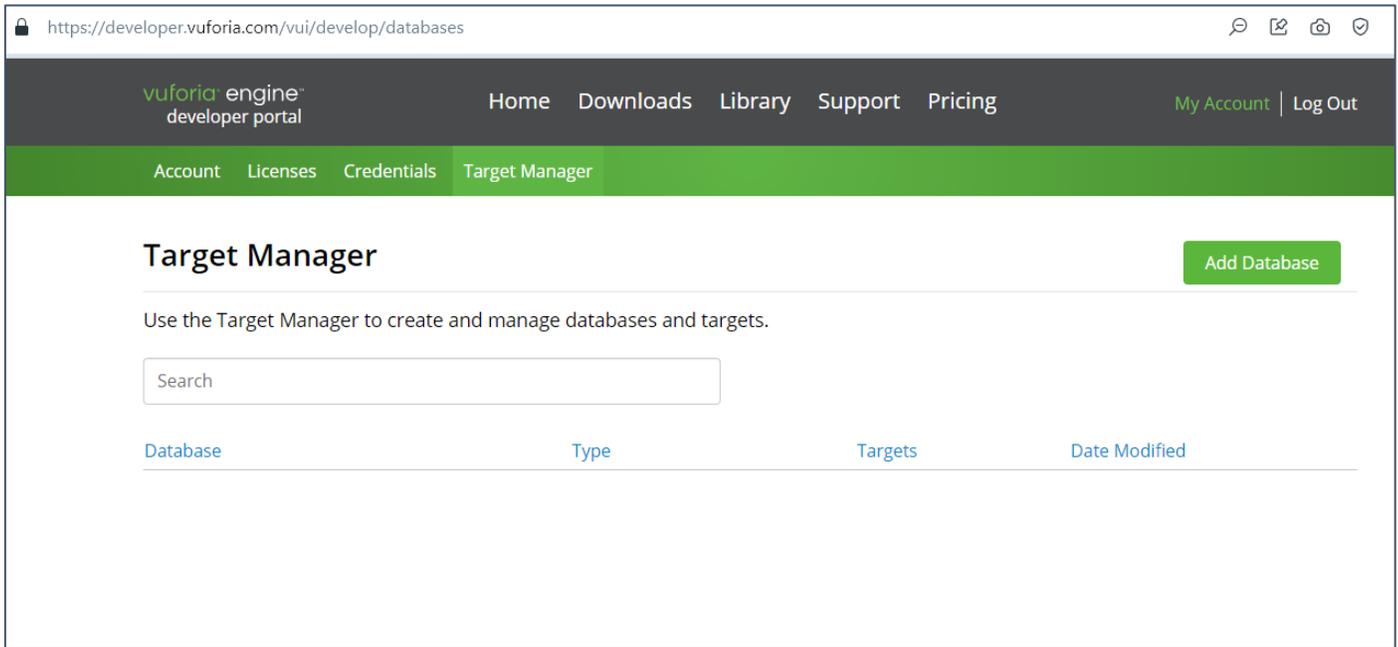
```
AXxFGv3/////AAABmcPkqu6OW0Mdv937pN11AKwdhkrvSLOxaFbUZ3yWeSYtQyEvQNFSZG0vf2uFoX1Weodua+WGZbq20Y2HSSKDXFZ
NEXS85Zj fWuZQB+ZctAG/2LFm8Ug3g01KGv+1PFGEJ7f3we/HtZPGkKD9DGae9E1ZgVuy115dKrmQU18VYNI0gnsd+g7J1/nw3F/DR/
qo31gBdUWP19enFMyhW+D4sm71zHWj kCjSL4EFJX0FRgRbQ+7EdSTUnYK8Er2jXxQ1CeNF3+JFCgPDKMaL/IPz1/LCzIvpWDnd2A2gC
0kspagiZy9tdUzFK2x1107GwecP5AR1Ua0wwKH5mhDS/1KhK/gVotWyXby78Bk+FnQXbaM9
```

Plan Type: Basic
Status: Active
Created: Feb 24, 2024 13:21
License UUID: 691bb30202304fdea4754719a8812189

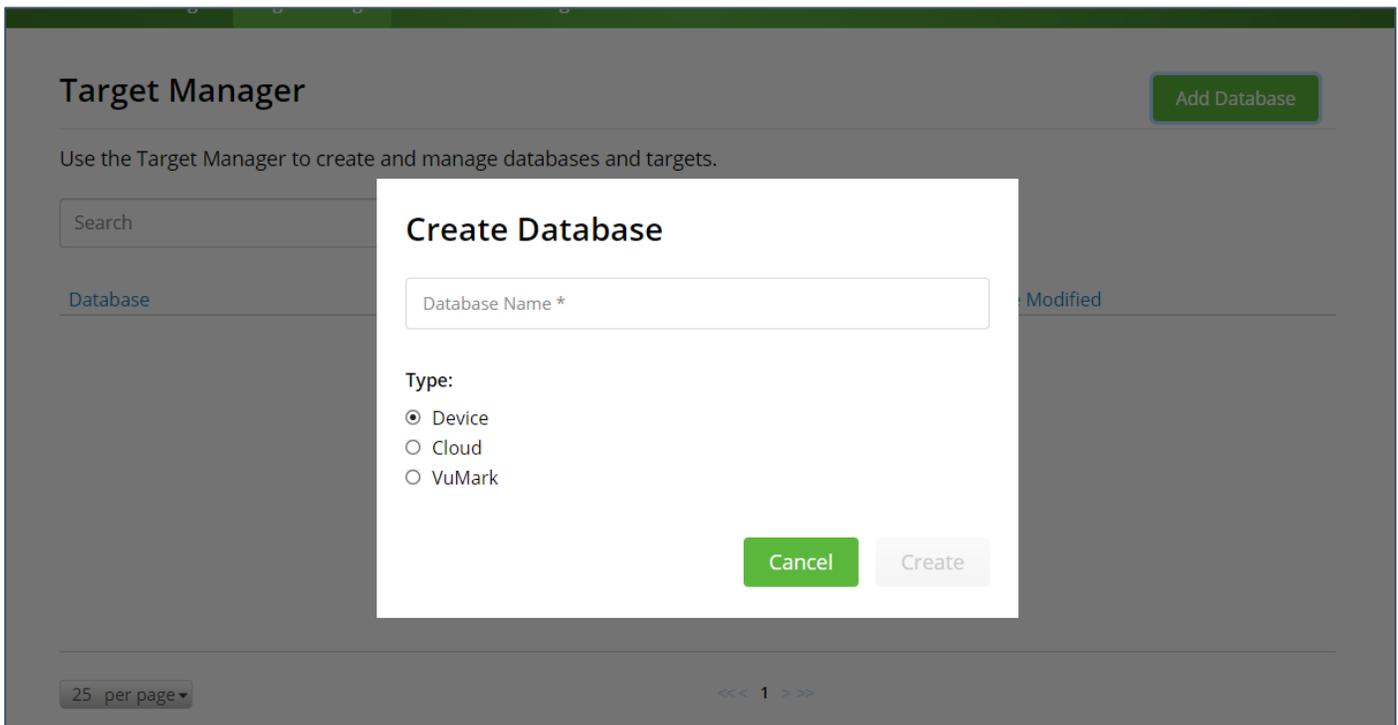
History:
License Created - Today 13:21

ВАЖНО!! Полученный лицензионный ключ необходимо скопировать и сохранить в любом текстовом редакторе (хороший стиль - не **Word!!**) для дальнейшего его использования в среде разработки (редакторе) **Unity 3D** на вашем локальном компьютере. Хотя с вашими логин/пароль на портале разработчика **developer.vuforia.com** для выбранного проекта всегда можно найти эту лицензию.

2. Создание базы данных таргетов. Переходим в менеджер меток – Target Manager:



Фактически на данном этапе начинается создание базы данных меток → **Add Database:**



Как предлагалось выше, будем использовать для БД меток то же имя, что и для лицензионного ключа – **AR-001**. Тип БД – **Device**. Это означает, что **БД меток будет располагаться на устройстве (смартфон, планшет, очки ДР и т.д.)**.

te and manage databases and targets.

Create Database

Database Name *
AR-001

Type:

- Device
- Cloud
- VuMark

[Cancel](#) [Create](#)

При нажатии кнопки **Create** переходим к созданию БД **AR-001**:

https://developer.vuforia.com/vui/develop/databases

vuforia engine[™]
developer portal

Home Downloads Library Support Pricing

My Account | Log Out

Account Licenses Credentials Target Manager

Target Manager

[Add Database](#)

Use the Target Manager to create and manage databases and targets.

Search

Database	Type	Targets	Date Modified
AR-001	Device	0	Feb 24, 2024

Далее переходим по ссылке **AR-001** в поле **Database** и начинаем формировать БД таргетов:

://developer.vuforia.com/targetmanager/project/targets?projectId=94773568ad114dfd8bec0b4cdc4750b4&av=false

vuforia engine[™]
developer portal

Home Downloads Library Support Pricing

My Account | Log Out

Account Licenses Credentials Target Manager

Target Manager > AR-001

AR-001

[Edit Name](#)

Type: Device

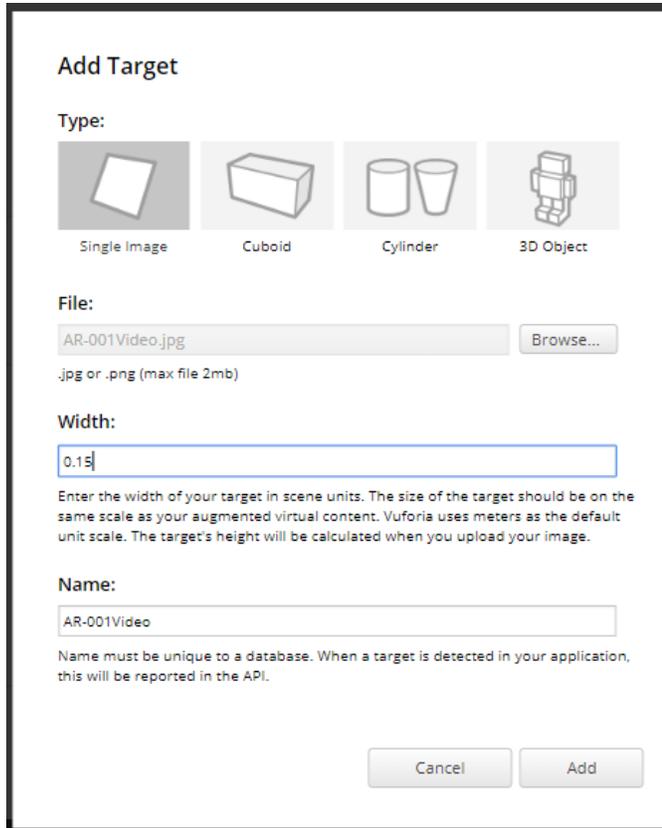
Targets (0)

[Add Target](#) [Download Database \(All\)](#)

Target Name	Type	Rating	Status	Date Modified
-------------	------	--------	--------	---------------

Для задач данной ЛР в качестве меток - таргетов будут использоваться плоские, хорошо распознаваемые изображения, например: **фотографии, картинки, логотипы, баркоды и т.д.** Такие объекты необходимо подготовить заранее: и в электронном формате для БД таргетов (**.jpg, .png**), и в виде твердой копии для использования при работе создаваемого Приложения ДР.

В появившемся окне выбираем → **Add Target** для заведения метки.



Add Target

Type:

Single Image Cuboid Cylinder 3D Object

File:

AR-001Video.jpg Browse...

.jpg or .png (max file 2mb)

Width:

0.15

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

AR-001Video

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Cancel Add

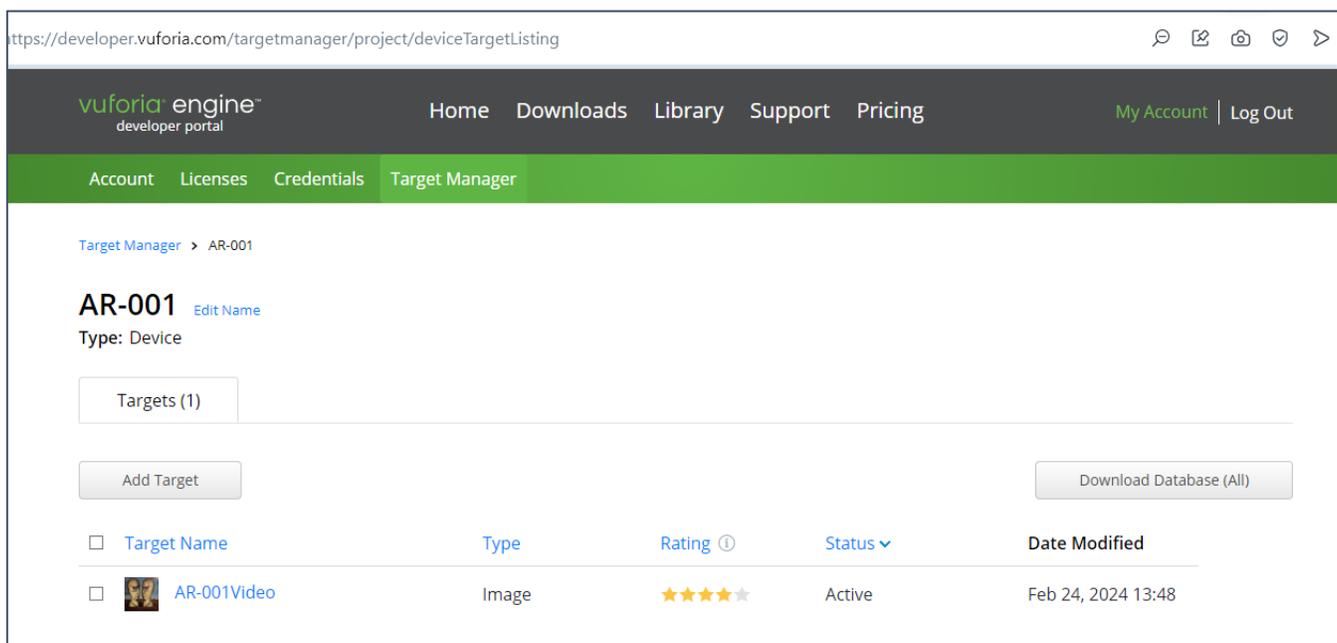
В открывшемся окне:

- выбираем тип метки – **Single Image** (обратите внимание, какие еще типы таргетов поддерживаются в **Vuforia Engine**),
- загружаем подготовленный заранее файл, содержащий метку - таргет (**File**),
- указываем реальный физический размер (ширина) твердой копии метки в метрах (**Width**) – в нашем случае 15 см.

Поле **Name** заполняется автоматически.

Для предварительного анализа возможности использования данного файла в качестве метки, выбираем кнопку **Add**.

Результат анализа качества метки приведен в окне:



https://developer.vuforia.com/targetmanager/project/deviceTargetListing

vuforia engine™
developer portal

Home Downloads Library Support Pricing My Account | Log Out

Account Licenses Credentials Target Manager

Target Manager > AR-001

AR-001 [Edit Name](#)

Type: Device

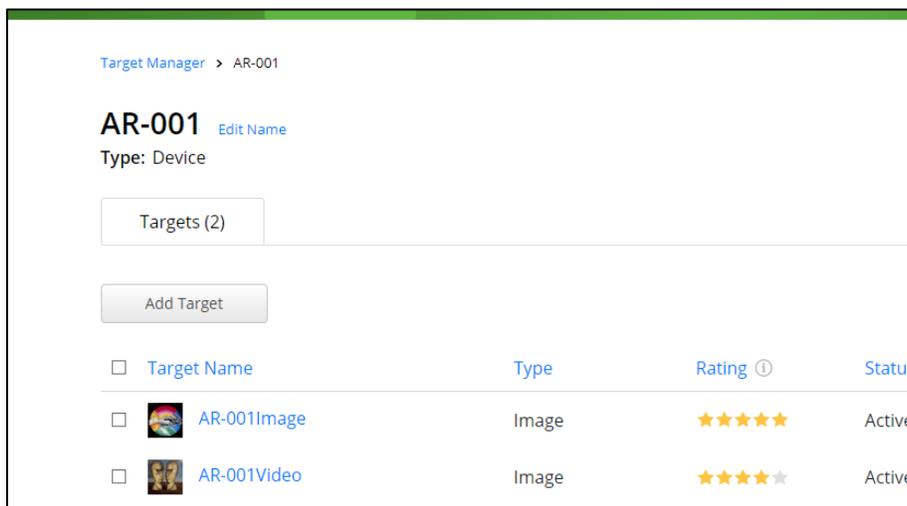
Targets (1)

Add Target Download Database (All)

<input type="checkbox"/>	Target Name	Type	Rating ⓘ	Status ▾	Date Modified
<input type="checkbox"/>	 AR-001Video	Image	★★★★☆	Active	Feb 24, 2024 13:48

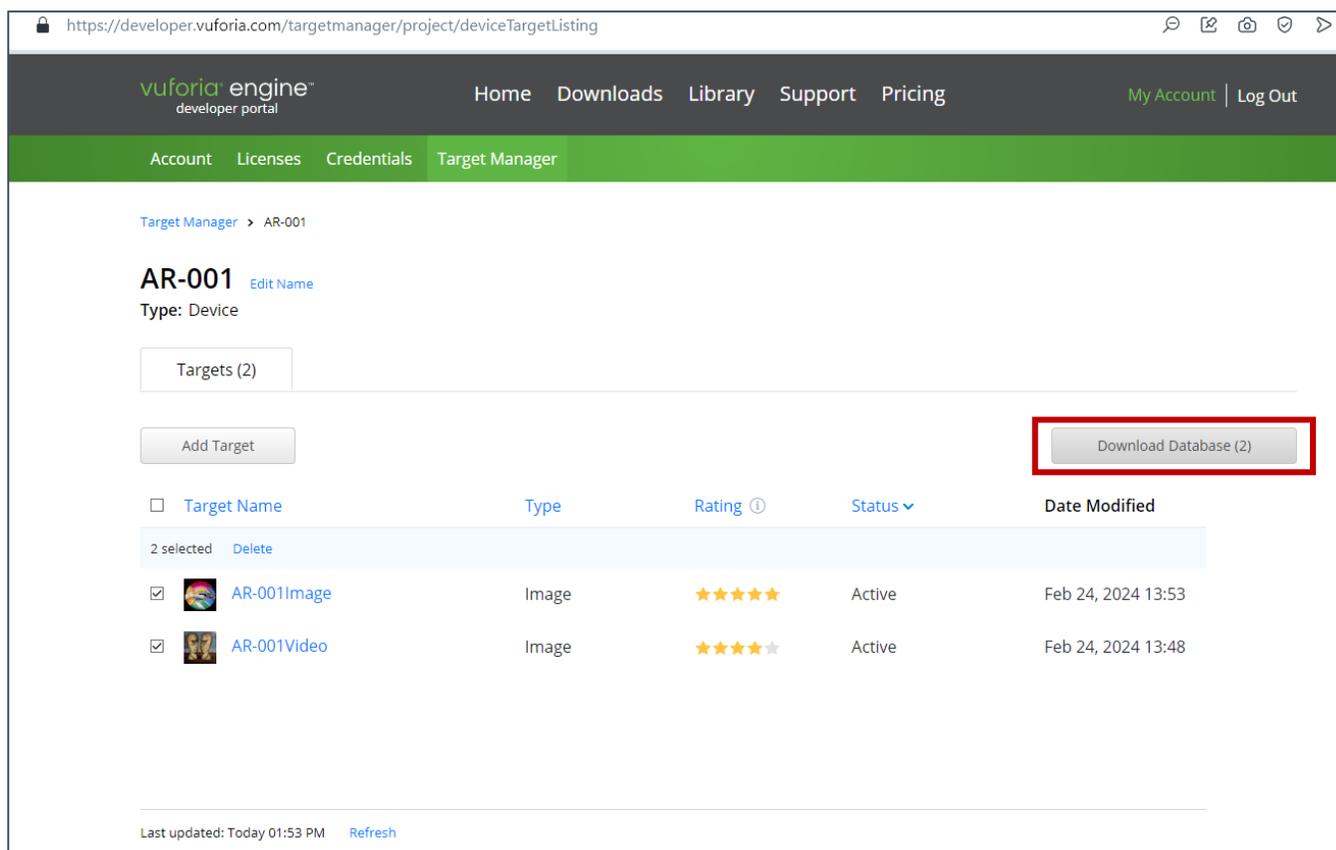
Выбранный нами файл изображения помещен в базу. Качество файла (количество желтых звездочек в поле **Rating**) – достаточное для дальнейшей работы и распознавания.

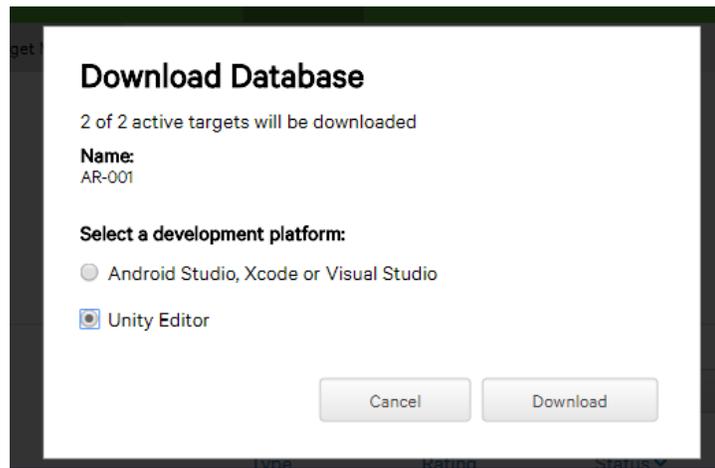
Добавим в формируемую **БД меток** еще одну метку для второй части данной ЛР → **Add Target**:



Обратите внимание на различные значения рейтинга - показателя качества изображения. В будущем можно исследовать влияние качества изображения, выбранного для таргета, на качество распознавания и отслеживания работающего Приложения ДР.

Для размещения этих меток в БД **AR-001** выбираем их в образовавшемся списке и загружаем → **Download Database (2)**.

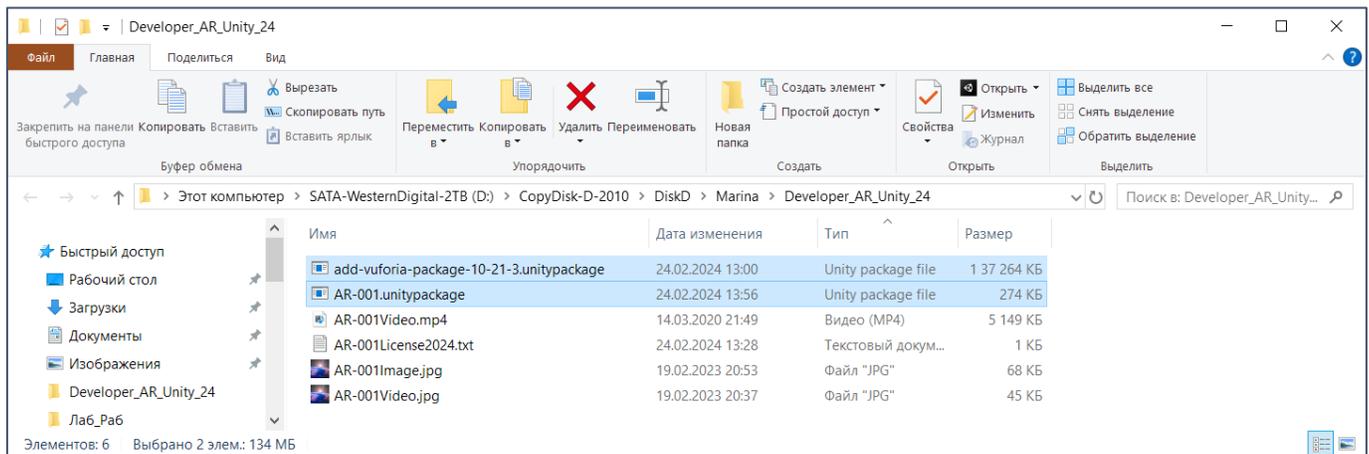




ВАЖНО!! Не забудьте выбрать платформу работы с объектами ДР – в нашем случае это **Unity Editor**. Для подтверждения правильности выбора и выполнения загрузки – нажмите **Download**.

В результате формируется образ Базы Данных **AR-001** для работы в среде **Unity 3D** в специальном формате - **.unitypackage**, который выгружается из облака **Vuforia** для сохранения в локальной файловой системе. Не забывайте требование к полному имени Проекта и его составляющих, сформулированному в ЛП№1:

все имена (включая путь и имя директории расположения проекта) должны состоять только из латинских букв и цифр!!!

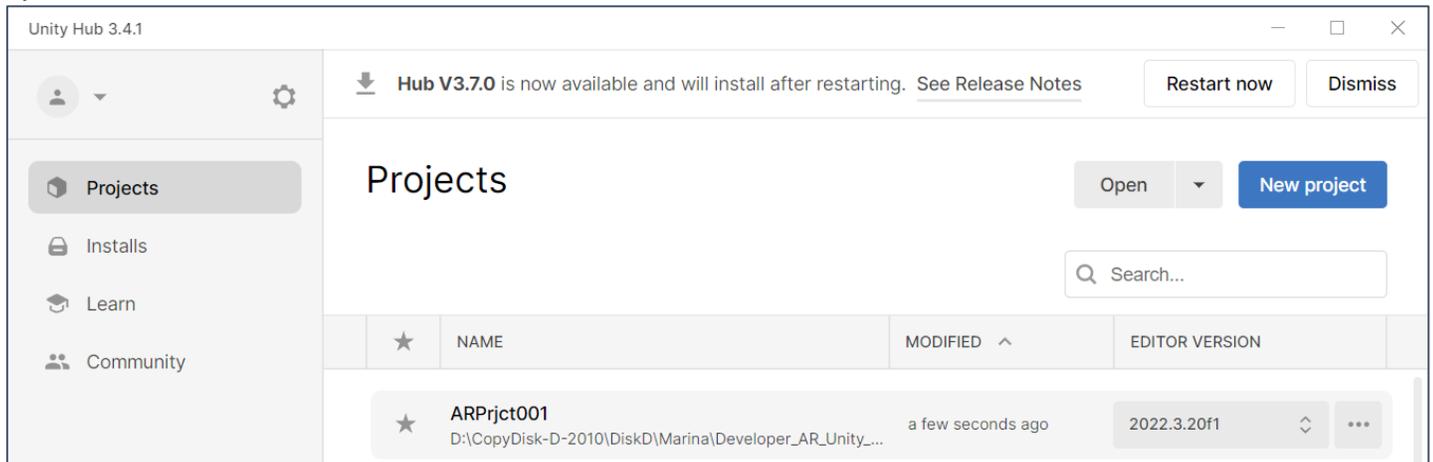


В результате выполнения двух описанных процедур в пп. **1.** и **2.** мы получили из облака **Vuforia** для локальной работы в **Unity 3D** следующие ресурсы:

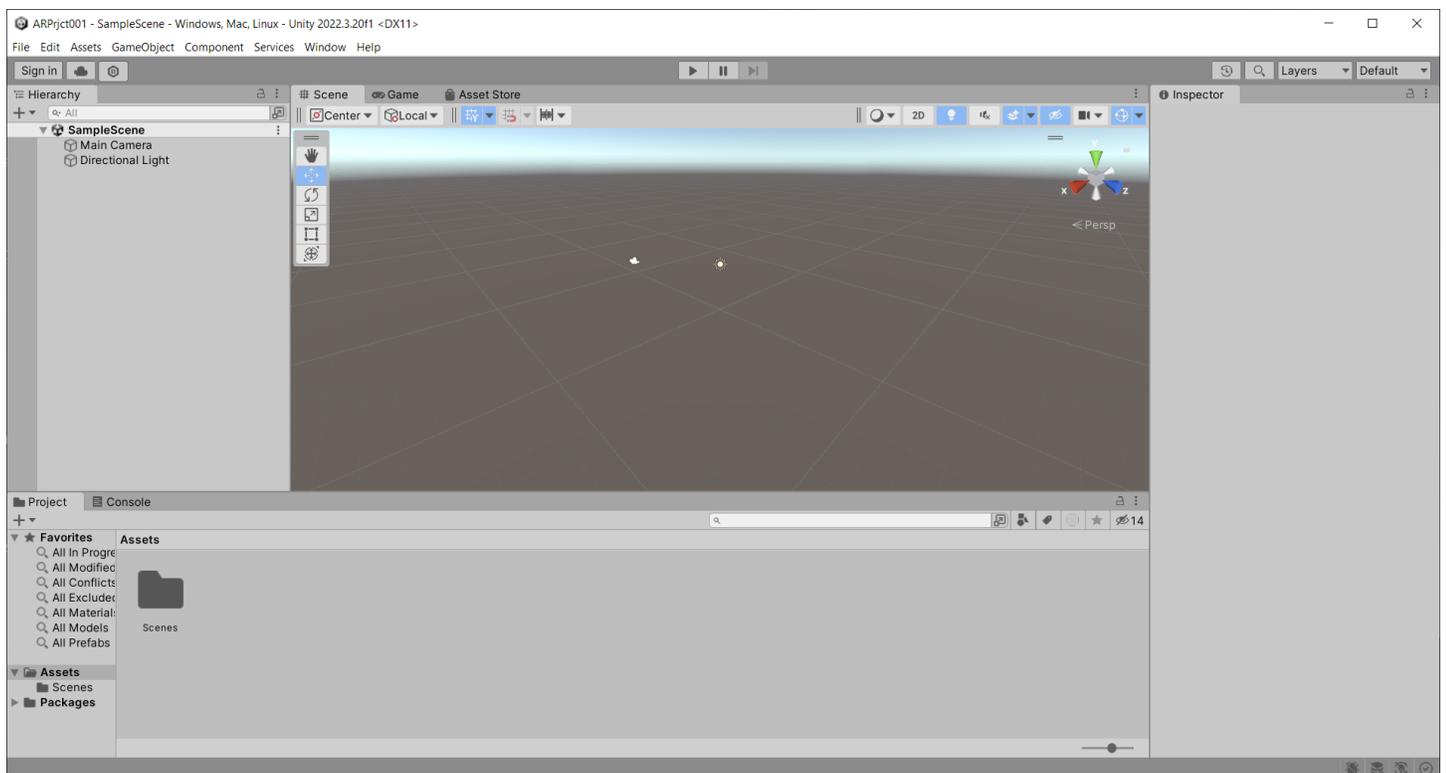
- Лицензионный ключ;
- Базу данных меток - два таргета: **AR-001Video.jpg**, **AR-001Image.jpg**;
- Заранее подготовленные объекты контента для воспроизведения их на экране **Android**-устройства в Приложении ДР: в рассматриваемом примере это видеоклип в формате **.mp4** – файл **AR-001Video.mp4**.

3. Настройка среды разработки Приложений ДР – Vuforia Engine & Unity 3D. Стартуем приложение **Unity 3D** (для загруженной в ЛРН№1 версии **Unity** это рекомендуется делать через **Unity Hub**) и выполняем настройки, необходимые для начала работы с **Vuforia**.

Если Первая ЛР выполнена корректно, вы увидите в открывшемся окне заведенный ранее проект.



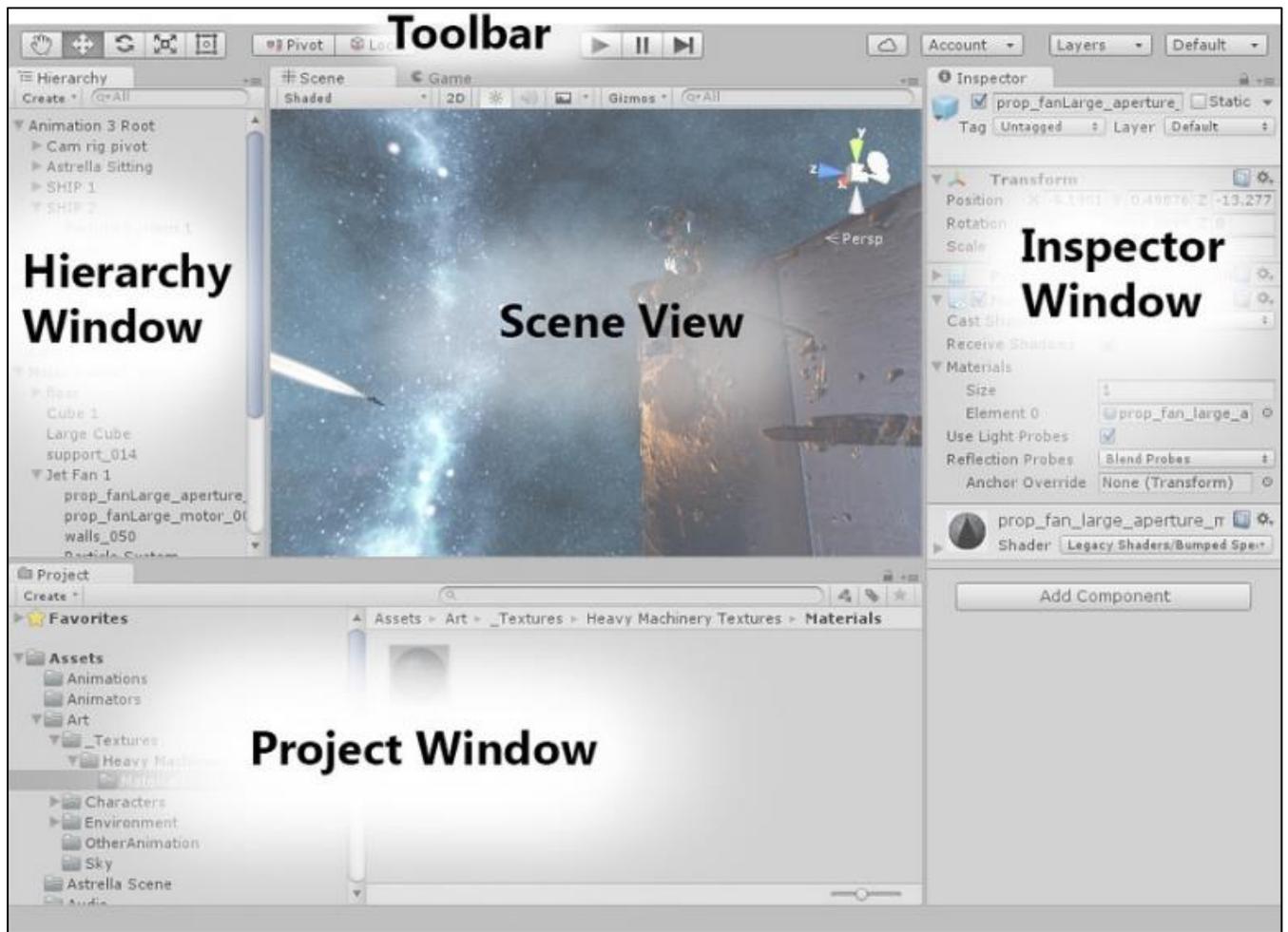
Стартуем проект (клик левой клавишей по имени в списке). Открывается редактор **Unity 3D**:



Документацию по работе в редакторе – см. по ссылке:

<https://docs.unity3d.com/Manual/UsingTheEditor.html>

В зависимости от версии **Unity**, структура его редактора, представленная ниже, может изменяться. Однако стандартный состав полей редактора остается неизменным.

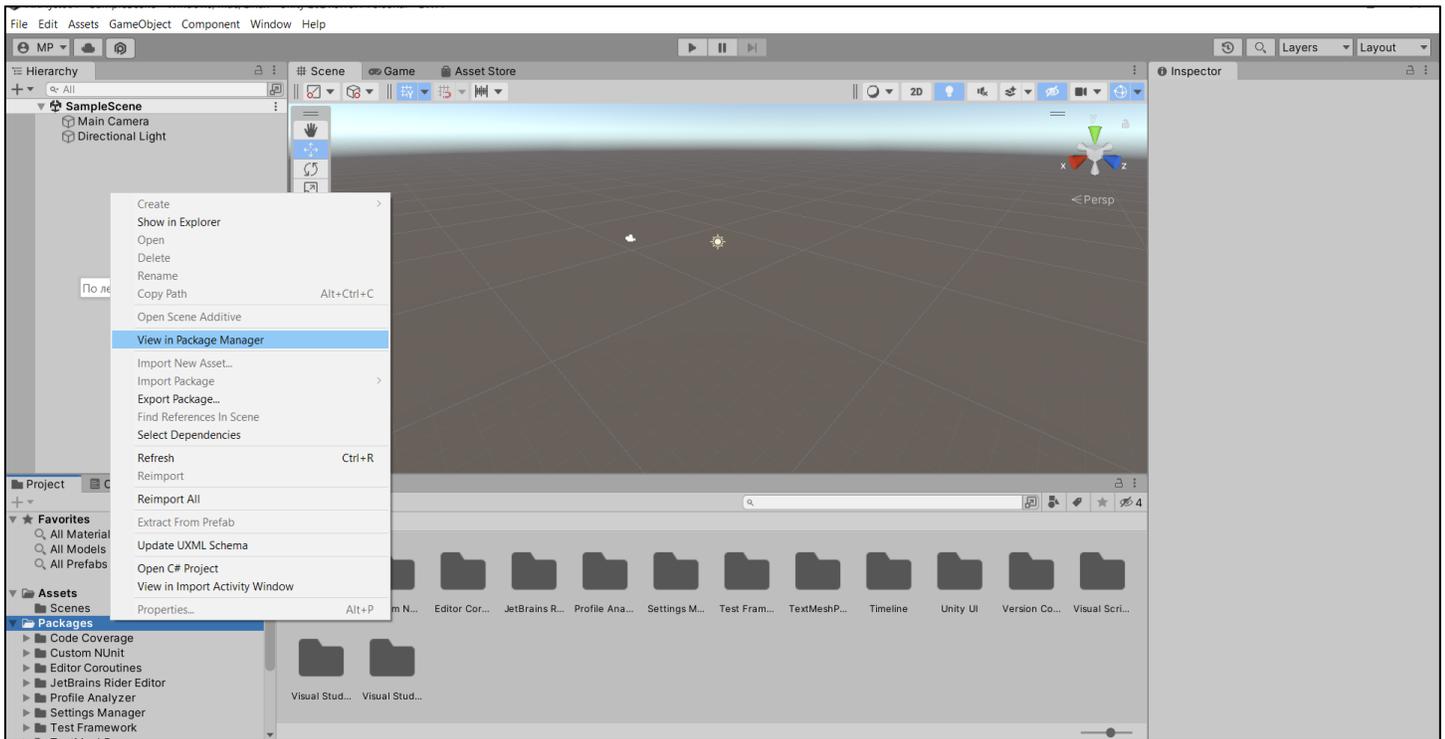


Обратите внимание - начальные действия по организации среды игрового движка **Unity 3D** для нашего проекта (Приложение ДР) происходят с использованием закладки **Scene** (область **Scene View**), группы элементов линейки **Toolbar**, и элементов закладки **GameObject**, входящей в состав ленты основного меню (**File Edit Assets GameObject ...**).

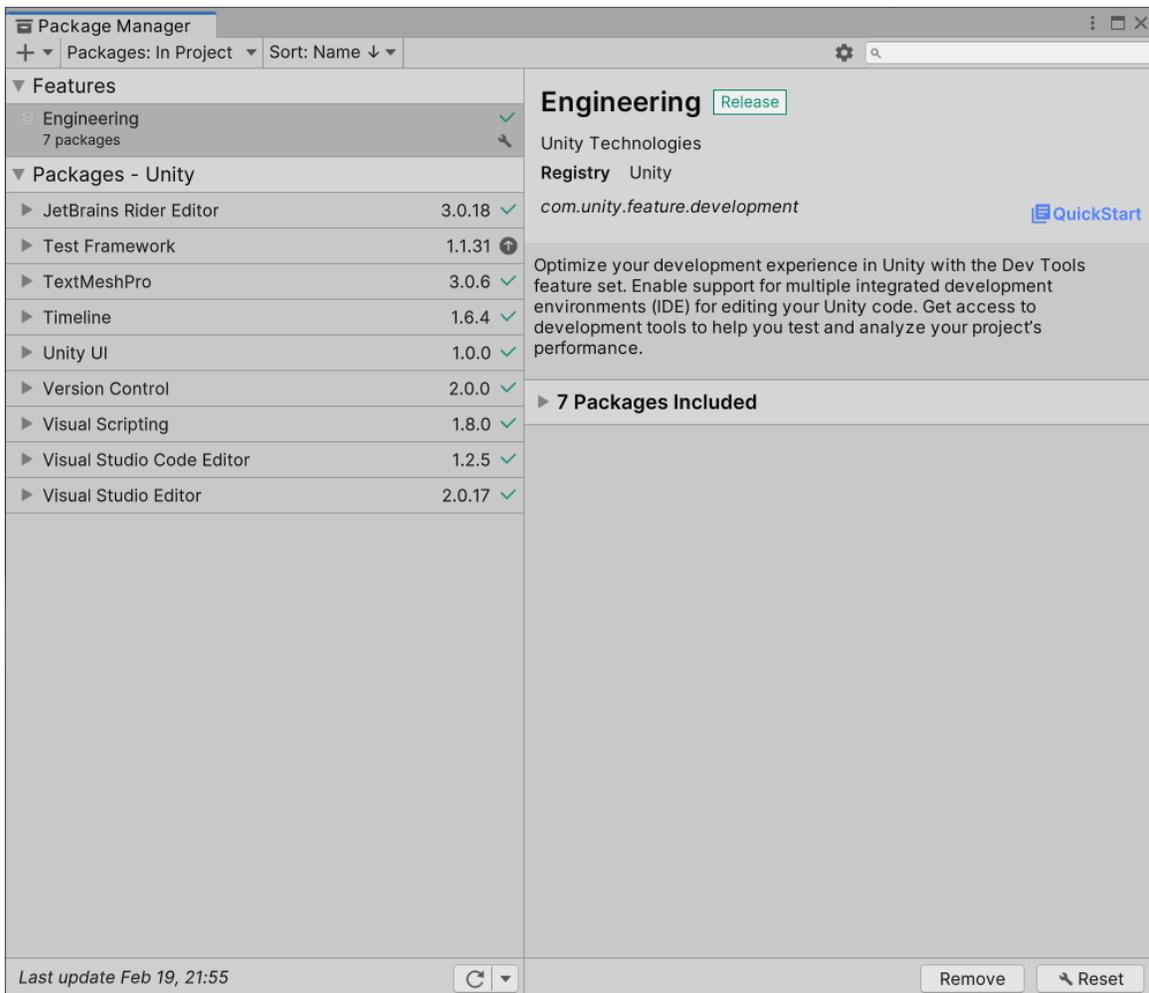
Без дополнительных настроек редактор **Unity 3D** работает в режиме **Виртуальной реальности**. По условиям Задания ЛР наше **Android**-устройство в Проекте должно работать в режиме **Дополненной реальности**. Как отмечалось выше, за режим ДР отвечает **Vuforia**. Это означает, что в список объектов **Unity 3D (GameObject)** необходимо добавить специфические объекты **Vuforia**. А работу с ними поддержать действующими лицензиями **Vuforia**.

Для этого необходимо файлы (пакеты → **packages**) типа **.unitypackage** загрузить в **Unity 3D**. Эти файлы (пакеты) вы заранее подготовили (в ЛР2 – база данных таргетов) и загрузили из облака **Vuforia** на локальную машину (см. ЛР1).

Для начала проверим, какие пакеты уже загружены в среду **Unity**. Для этих целей в области **Project Window** находим позицию **Packages**, по правой клавише мыши вызываем меню, в котором ищем позицию **View in Package Manager**.



В результате на экране появляется список уже загруженных в **Unity** пакетов, например:

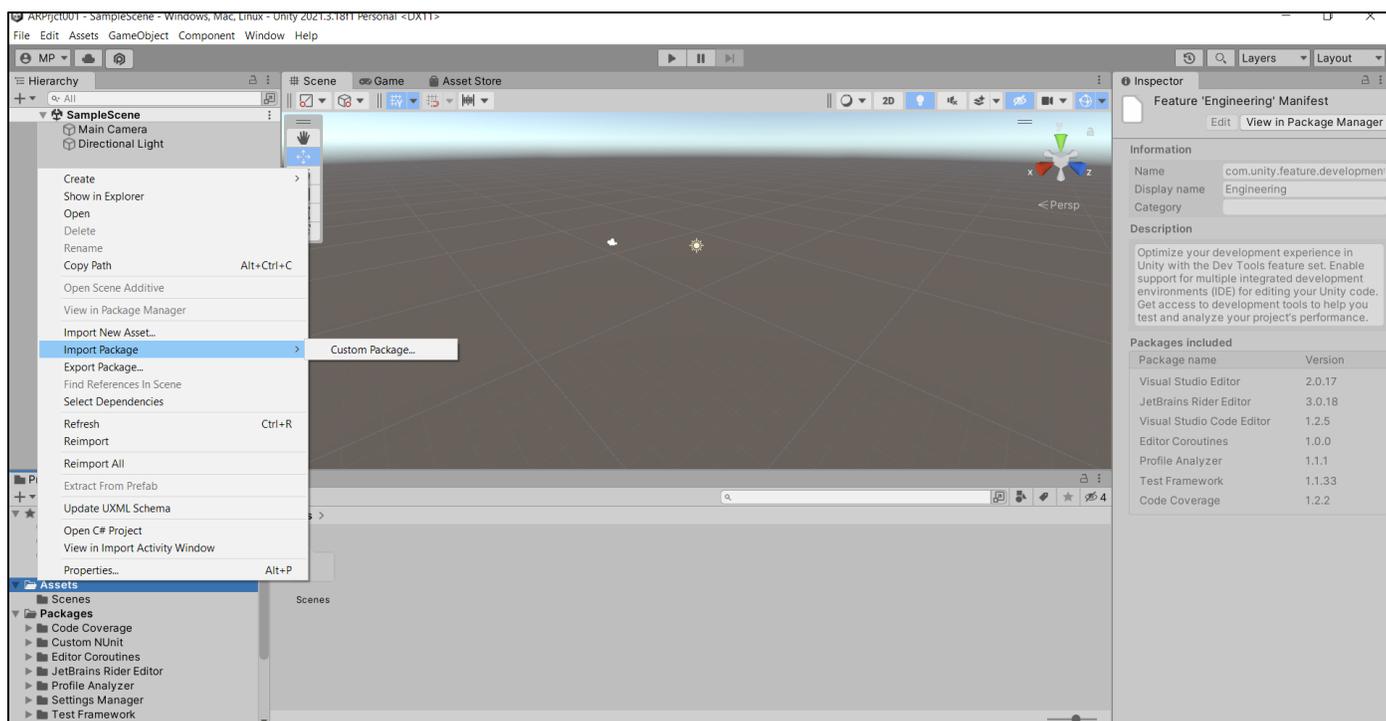


Если в этом списке не содержится пакет **Vuforia Engine**, его необходимо подгрузить. Этот файл (пакет) в специфическом формате **.unitypackage** был скачен и сохранен в рамках ЛР №1 в соответствующей директории на локальном компьютере пользователя.

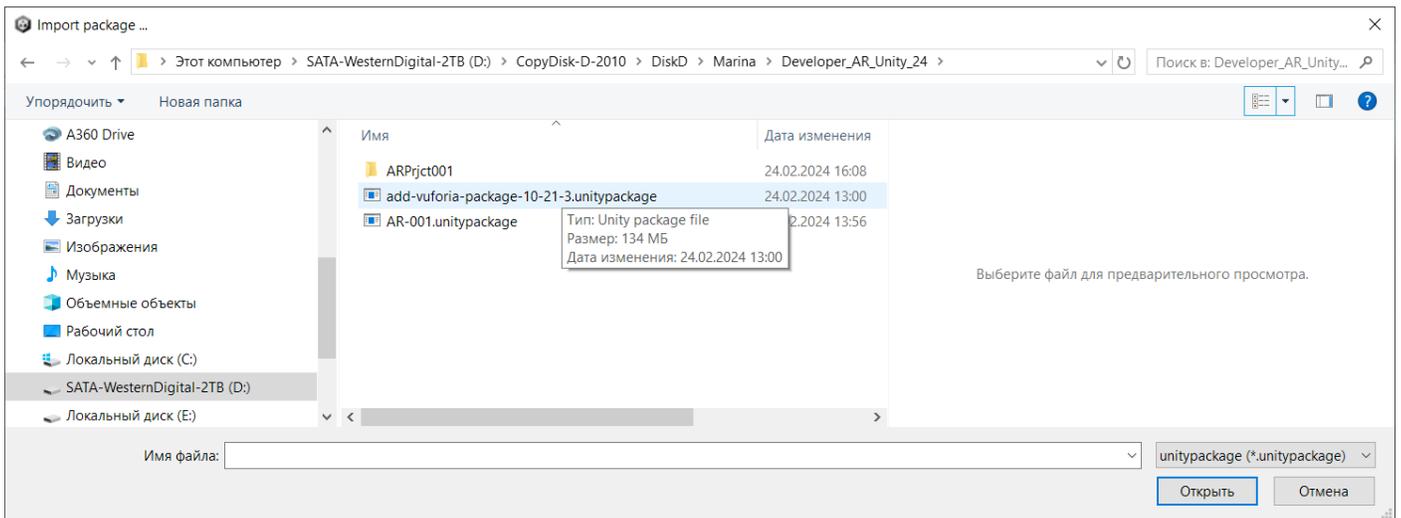
ВАЖНО!! Существует несколько способов связывания среды разработки сцен Приложения ДР – **Unity 3D** - и платформы ДР – **Vuforia Engine**. Вы можете найти их самостоятельно. Логичнее всего по этому поводу поискать рекомендации от компании разработчика **Vuforia Engine** на сайте <https://developer.vuforia.com/> → в разделе **Library** → **Getting Started** → **Development Environments** → **Unity**.

В данном описании мы воспользуемся одной из рекомендаций, в соответствии с которой для подключения Vuforia в проект Unity необходимо выполнение следующих шагов:

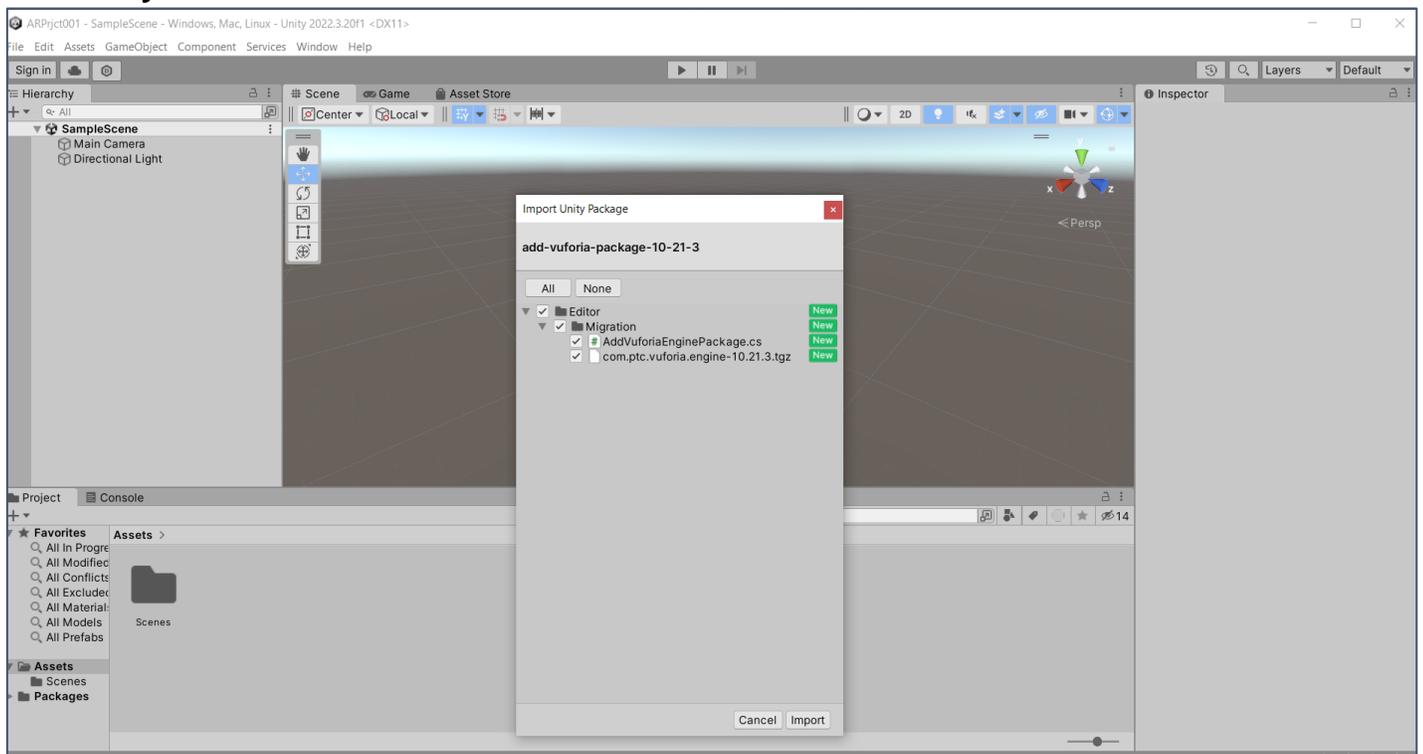
- В области **Project Window** в строке **Assets** → кликаем правой клавишей мыши (RMB) →



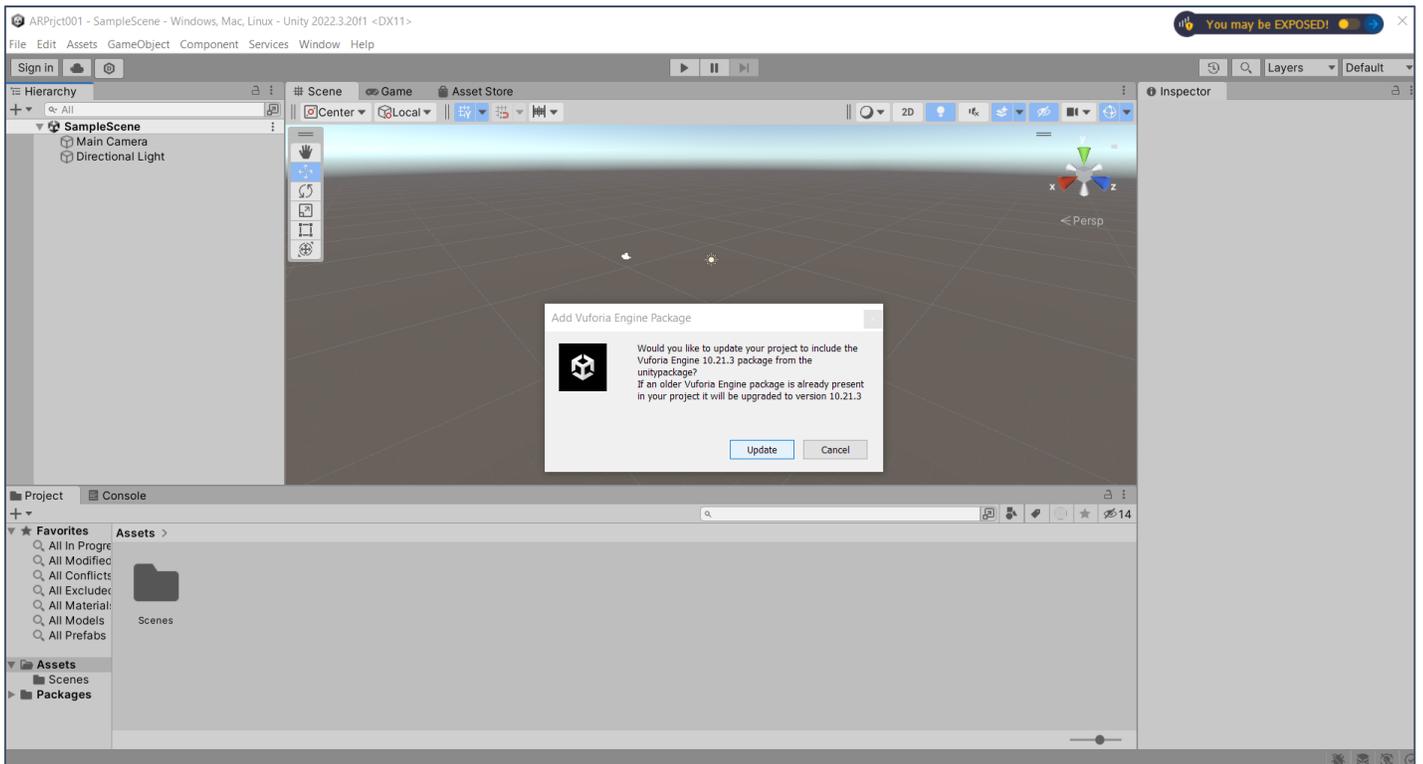
В элементе выпадающего меню **Import Package** выбираем действие **Custom Package**, после чего открывается навигатор файловой системы (ФС) вашего локального рабочего места, с помощью которого необходимо найти в локальной ФС сохраненный в ЛР№1 файл в формате **.unitypackage**:



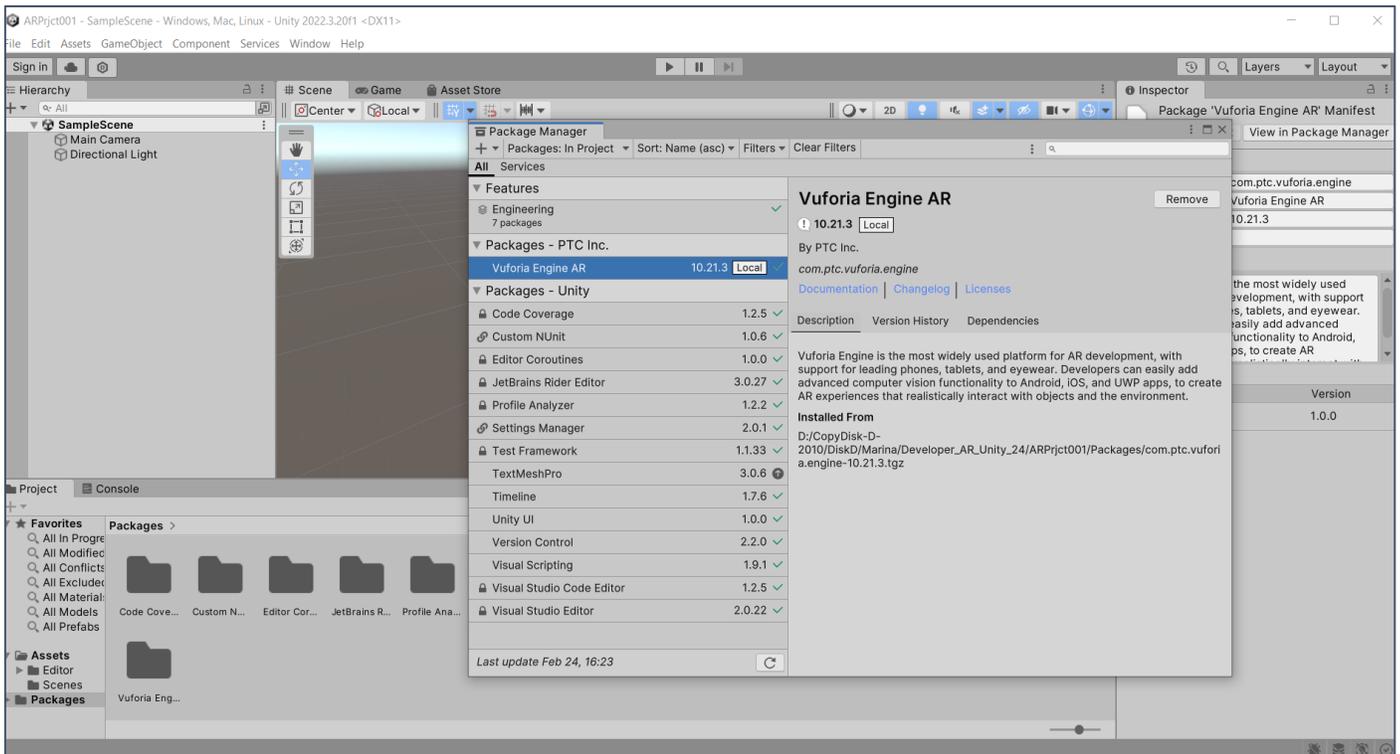
- Выбираем пакет одной из последних версий **Vuforia Engine** и импортируем его в среду **Unity**:



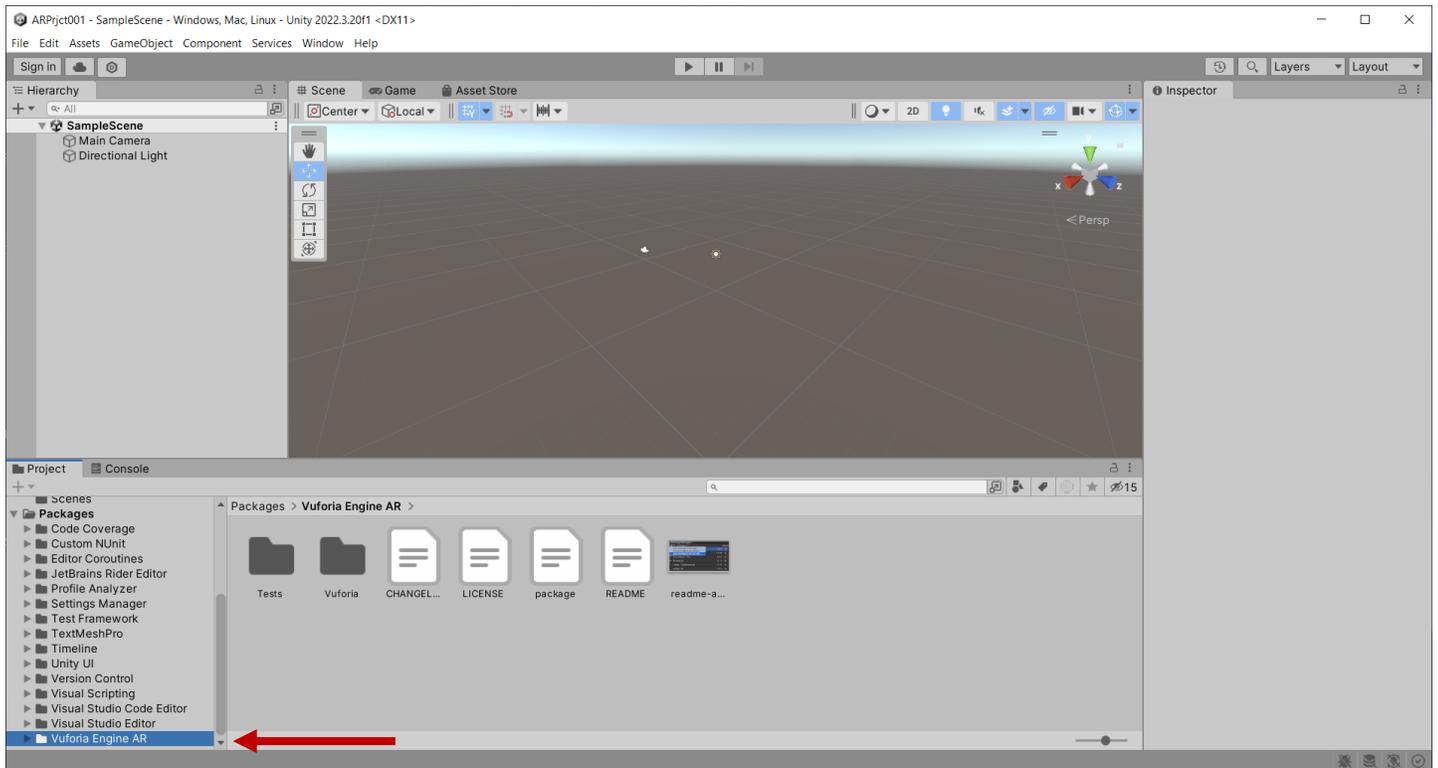
- После выбора опции **Import** появляется запрос на обновление Проекта – подключение пакета **Vuforia** указанной версии.



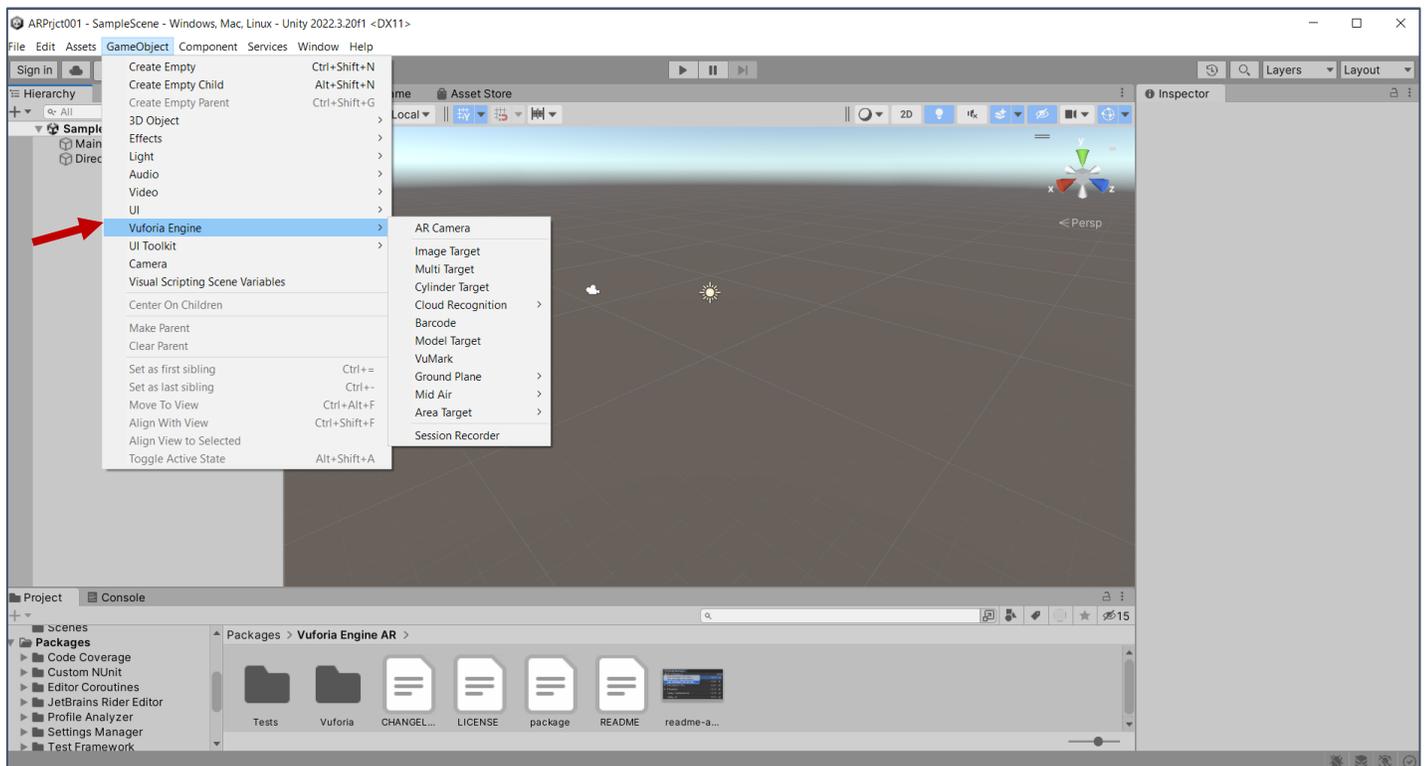
Выбираем опцию **Update**. После процесса загрузки всех элементов пакета, **Vuforia** появляется в обновленном списке загруженных в Проект пакетов:



- В результате мы получаем обновленный список пакетов в области **Project Window** редактора **Unity 3D**:



Убедимся, что у **Unity 3D** появились новые **Game Object** из пакета **Vuforia**:



На этом инсталляция пакета **.unitypackage** для работы с **Vuforia** в **Unity 3D** завершена. Базовые настройки для работы **Unity 3D** с **Vuforia** выполнены. Переходим к созданию Приложения ДР.

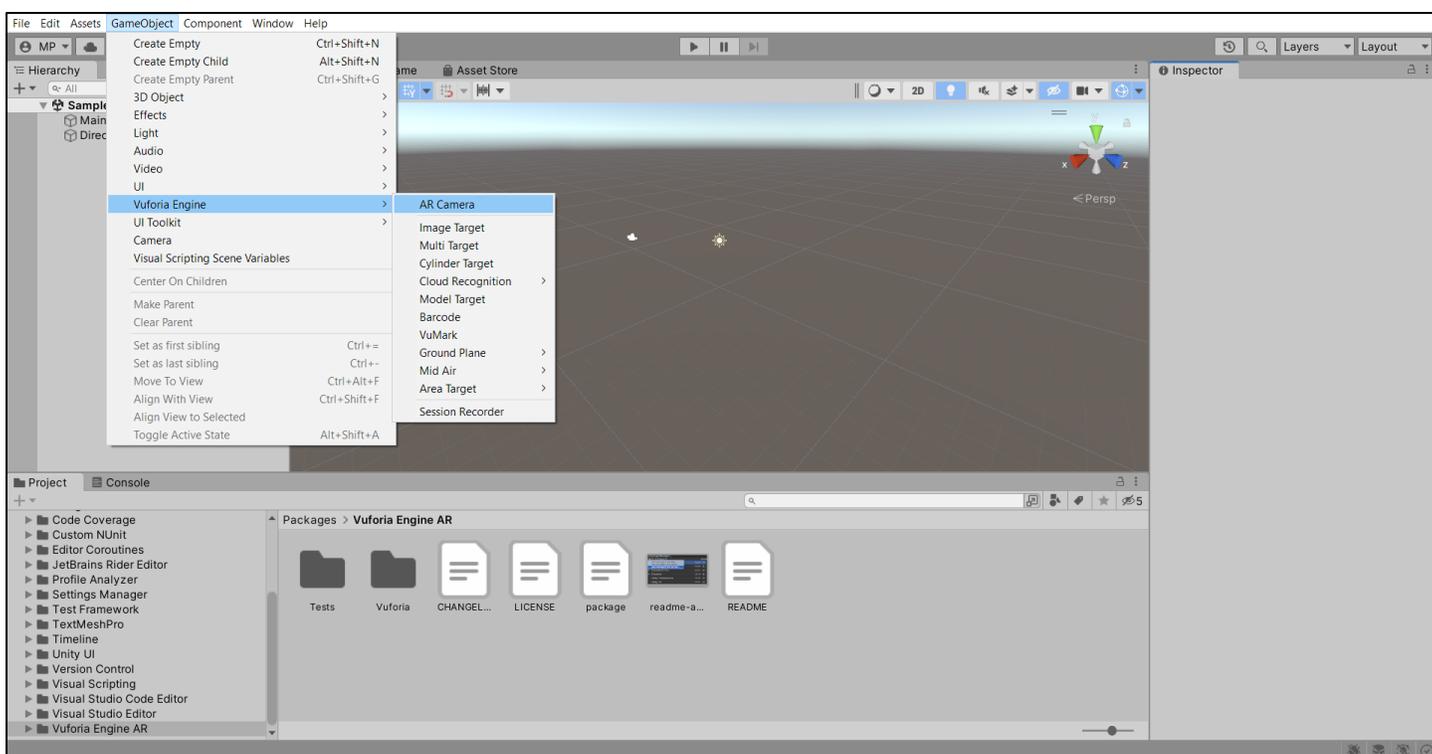
4. Создание Приложения ДР в среде Vuforia + Unity 3D для просмотра видеоклипа.

В стандартном режиме разработки Приложений в редакторе **Unity 3D Main Camera** не позволяет выполнять визуализацию реального окружения, транслируемого камерой МУ, а работает только с **виртуальными объектами сцены в виртуальном пространстве (режим ВР)**.

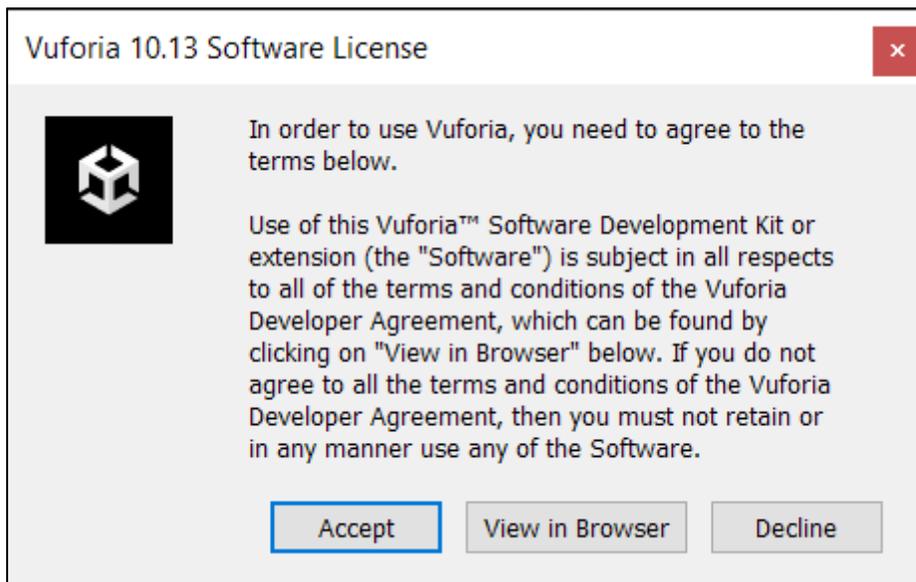
Для работы в другом режиме - **режиме ДР** - необходима «камера», способная выполнять визуализацию **виртуальных** объектов сцены (элементы контента) на фоне **реального окружения**. Для переключения нашего проекта из режима **виртуальной реальности** в режим **Дополненной реальности** необходимо заменить в наборе иерархических объектов (область экрана редактора **Hierarchy**) камеру **Main Camera** на камеру дополненной реальности из набора настроек **Vuforia**.

Для этого выполним следующие вызовы:

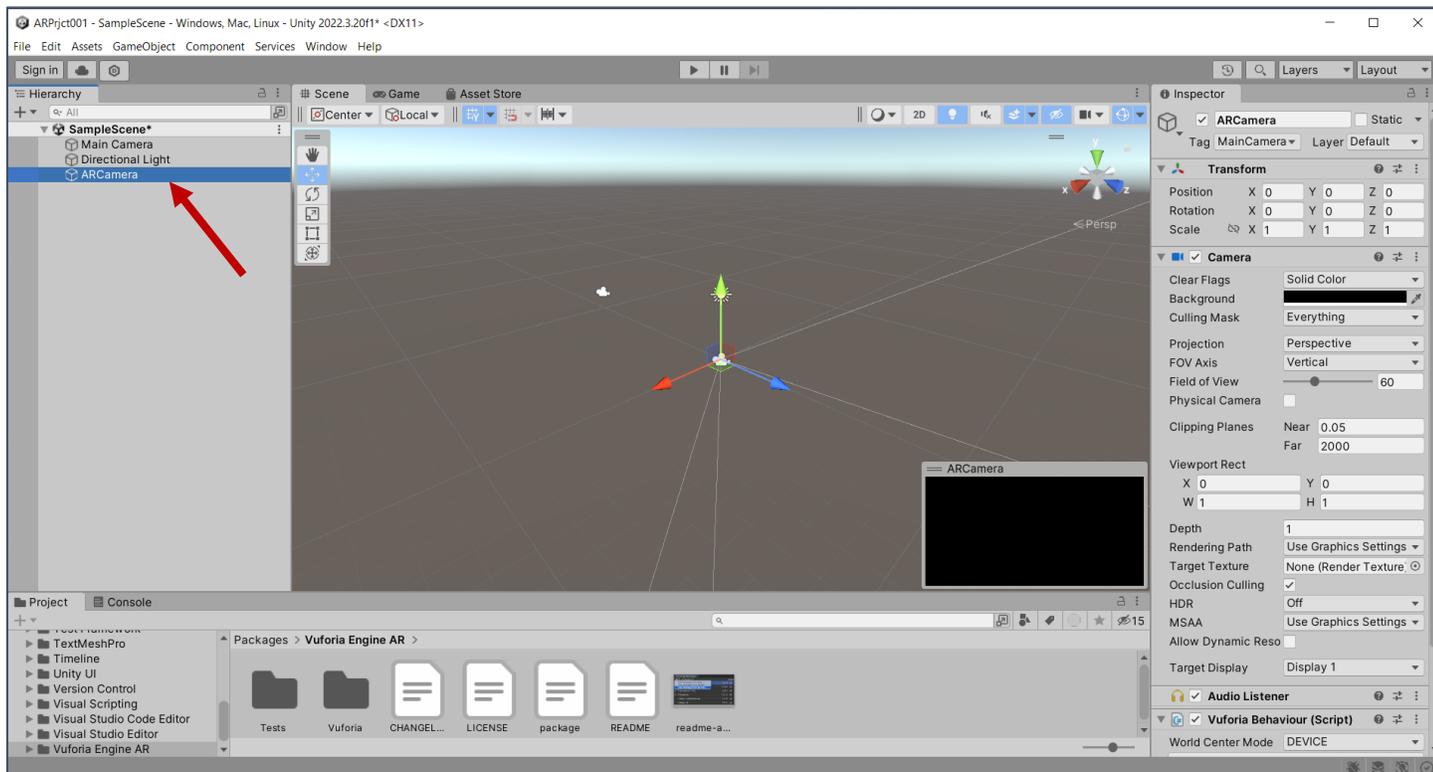
- **Game Object → Vuforia → AR Camera**



При этом у вас может появиться запрос на подтверждение использования **Vuforia Engine**:

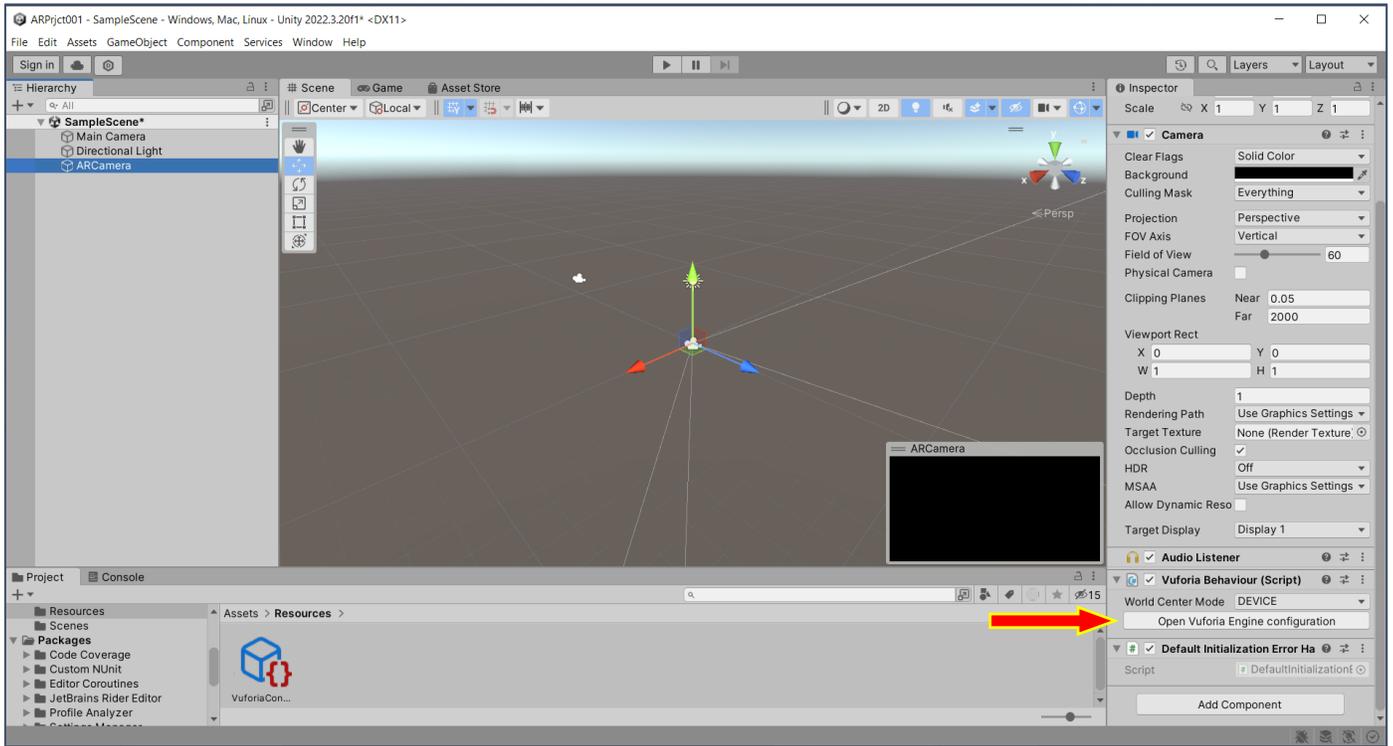


В области **Hierarchy** появляется **Vuforia**–объект **ARCamera**.

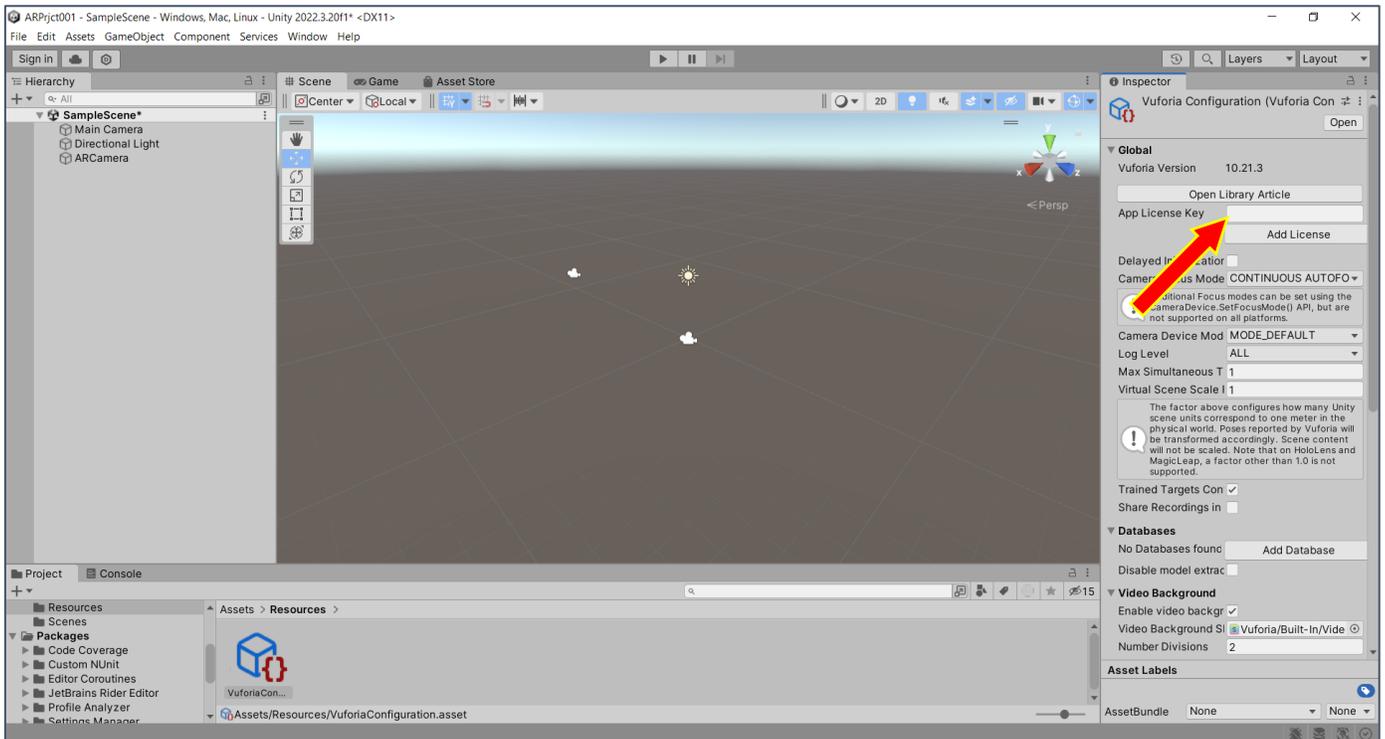


Для работы с объектами **Vuforia** необходимо перевести их под действие сгенерированной вами выше лицензии. Делать мы это будем через верхний уровень объектов **Vuforia**. Установленный объект **ARCamera** является верхним уровнем иерархии всех объектов **Vuforia** разрабатываемой сцены →

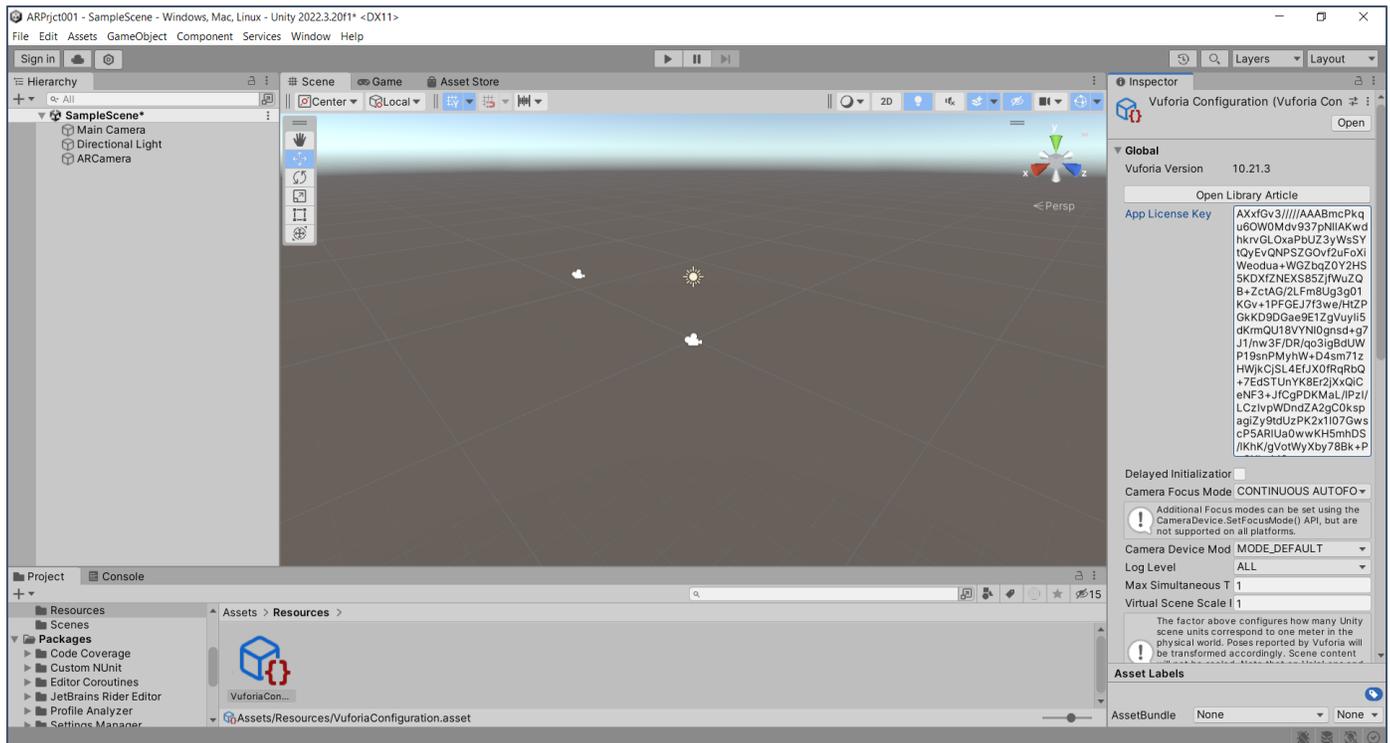
- В области **Hierarchy** выбираем **ARCamera** → в области **Inspector** находим и выбираем поле **Open Vuforia Engine configuration**.



- Теперь в Inspector'e появилось окно ввода лицензии **App License Key**:



- Ранее на локальной машине вы сохранили текст лицензии (**license key**). Находим этот текст и копируем в это поле.

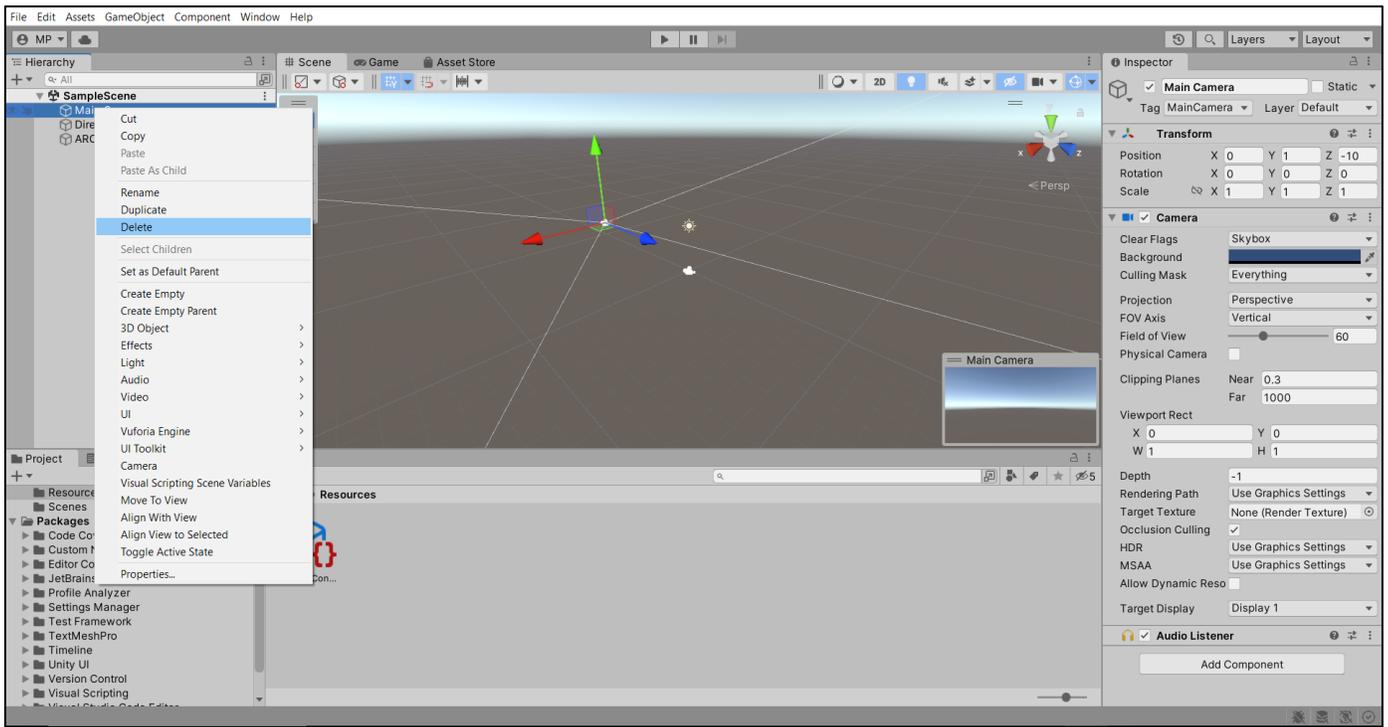


ВАЖНО!! Если Вы выберете на этом этапе поле **Add License**, то вновь попадете на этап ее генерации, который в данной ЛР был выполнен в первую очередь. Иными словами, лицензия автоматом не устанавливается при конфигурировании, ее необходимо скопировать в поле **App License Key**, предварительно сгенерировав.

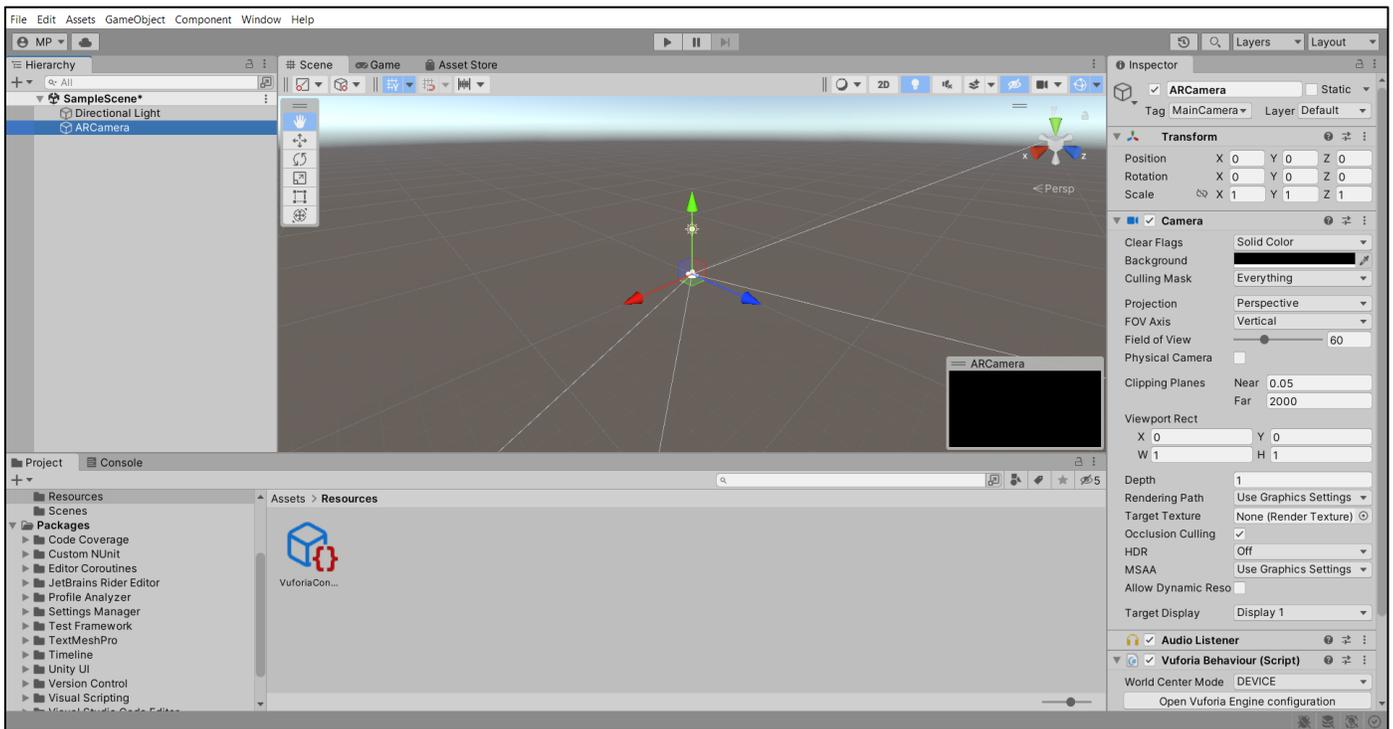
Но если вы забыли это сделать в свое время – здесь у вас есть возможность исправить эту ошибку.

Итак, лицензию вы установили.

Теперь выполняем замену камеры: оставляем **AR Camera** и удаляем **Main Camera**. **Main Camera** должна быть удалена обычным способом – маркируем **Main Camera** в меню иерархии и выбираем в падающем контекстном меню (**RMB**) функцию **Delete**.

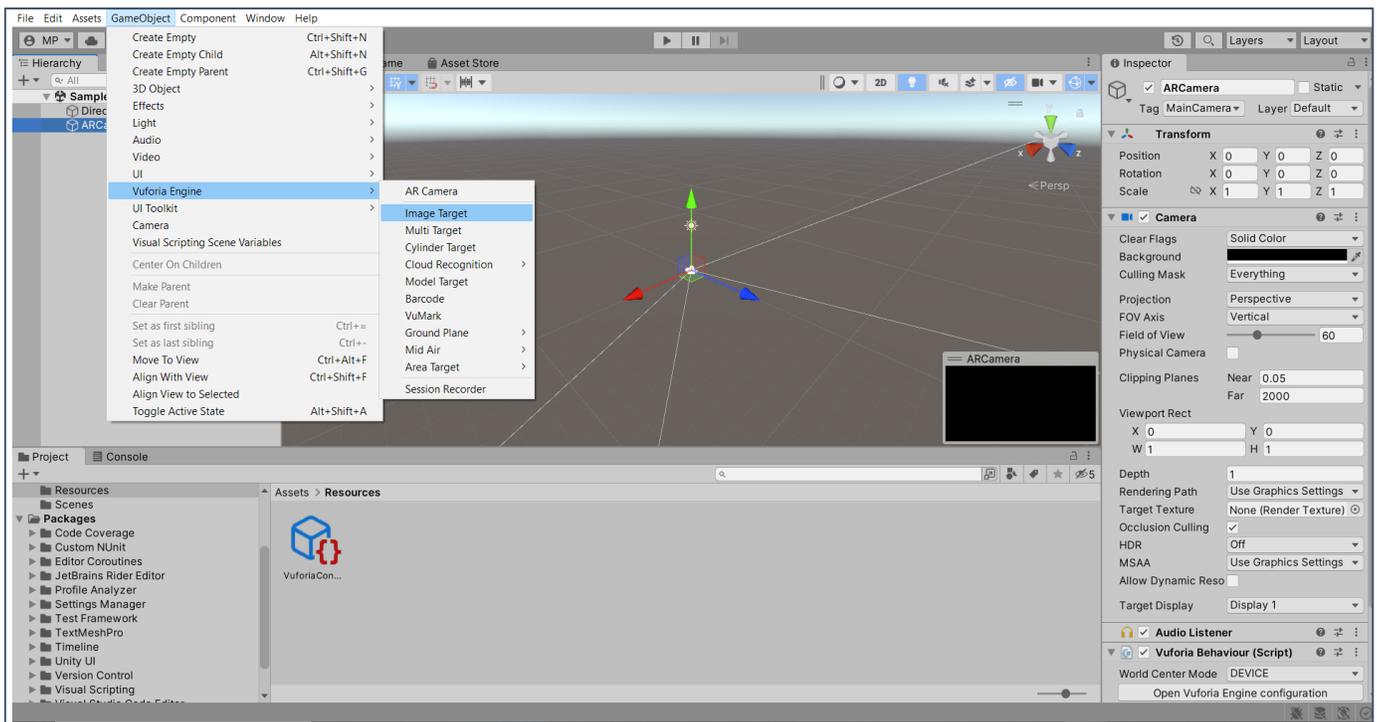


В результате в **Hierarchy** остается только одна камера, свойства которой отображены в области **Inspector**.

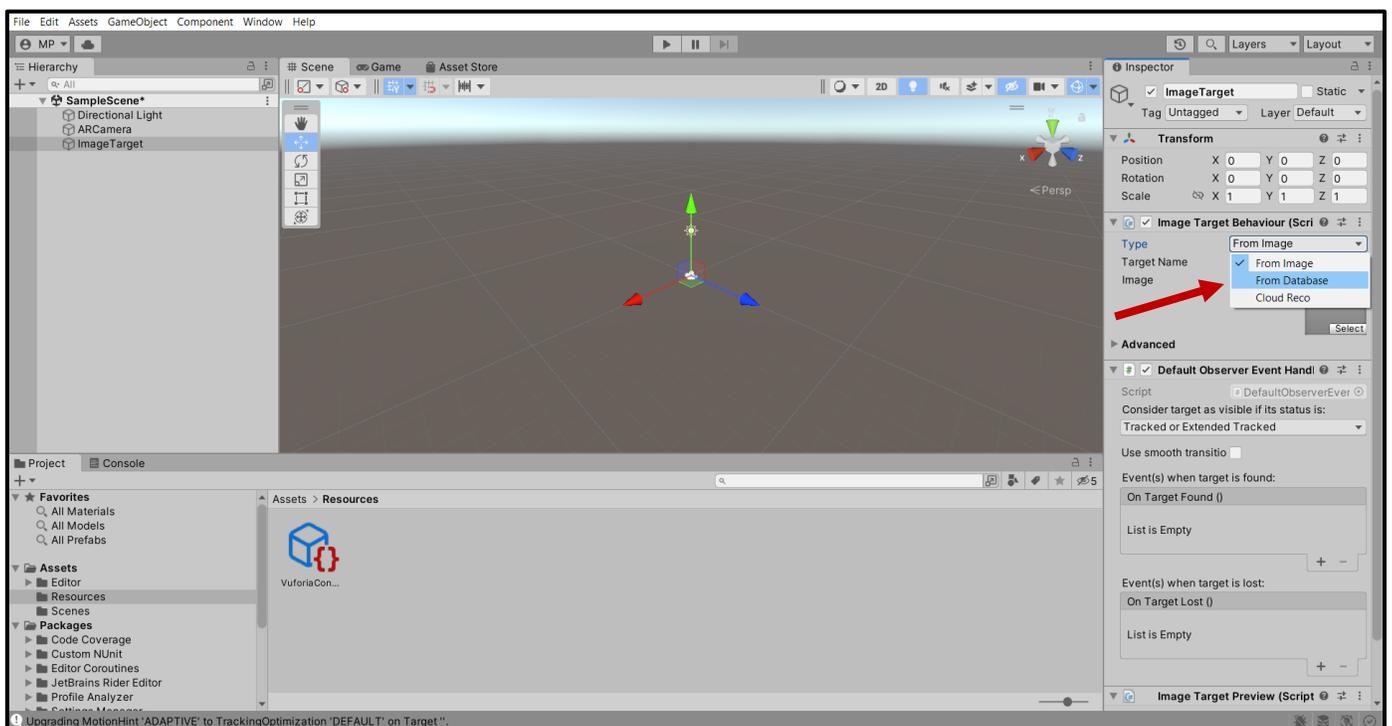


- Далее необходимо загрузить Базу данных таргетов.

Для этого осуществляем вызовы в закладке **Game Object** → **Vuforia Engine** → **Image Target**:

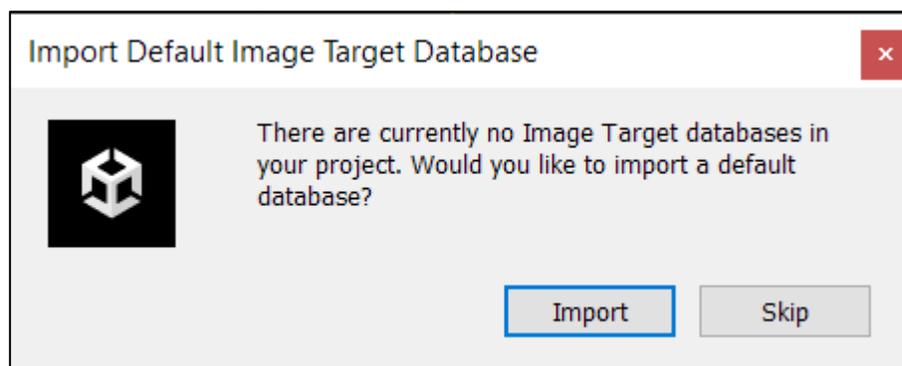
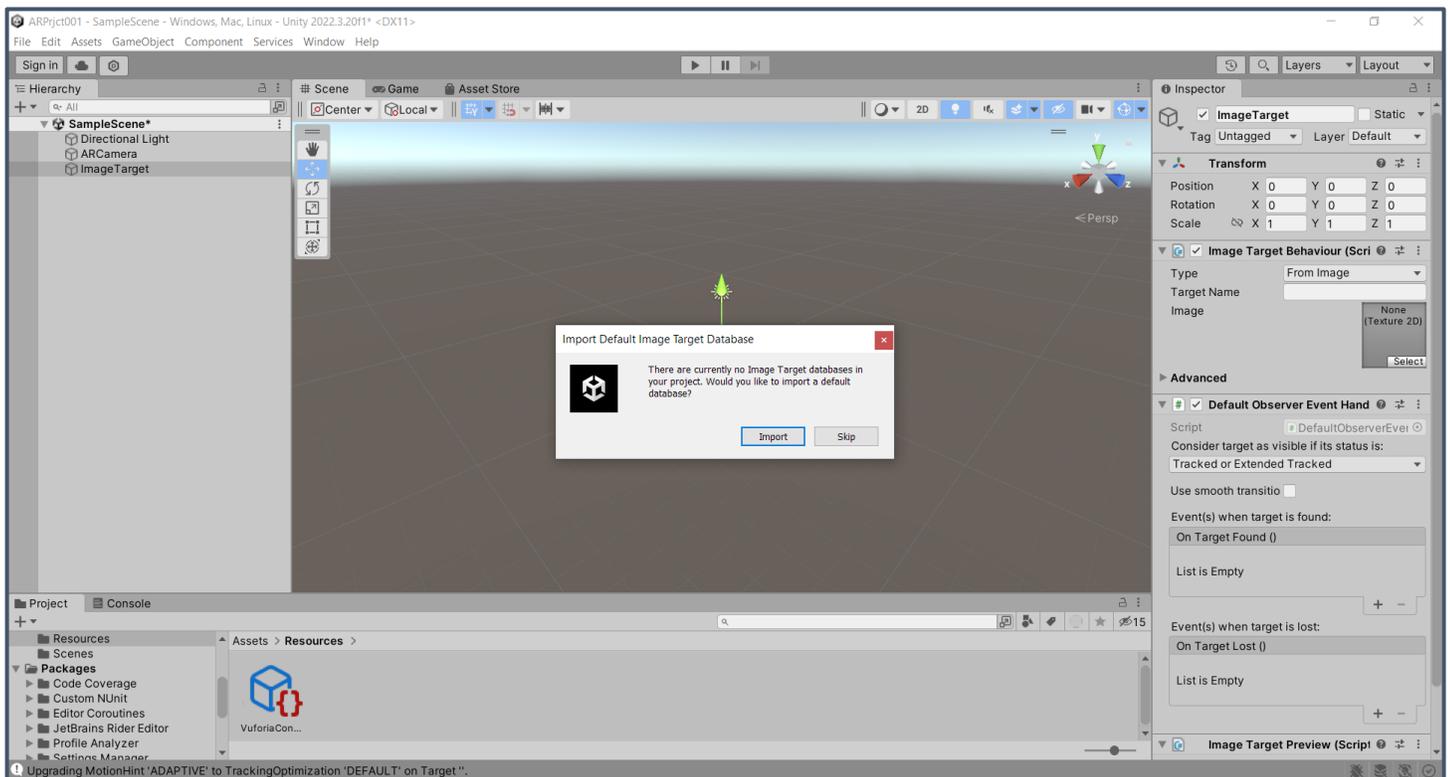


В результате в области **Hierarchy** появляется объект **Image Target**. Выберем его и в области **Inspector** определим источник изображения, которое будет ассоциировано с таргетом → в разделе **Image target Behaviour (Script)**. По умолчанию таким источником является простое изображение – **Image**. Нам же необходимо в качестве источника указать одну из записей сформированной ранее **базы данных таргетов**, сохраненной вами в локальной ФС в виде специфического объекта в формате **.unitypackage**.



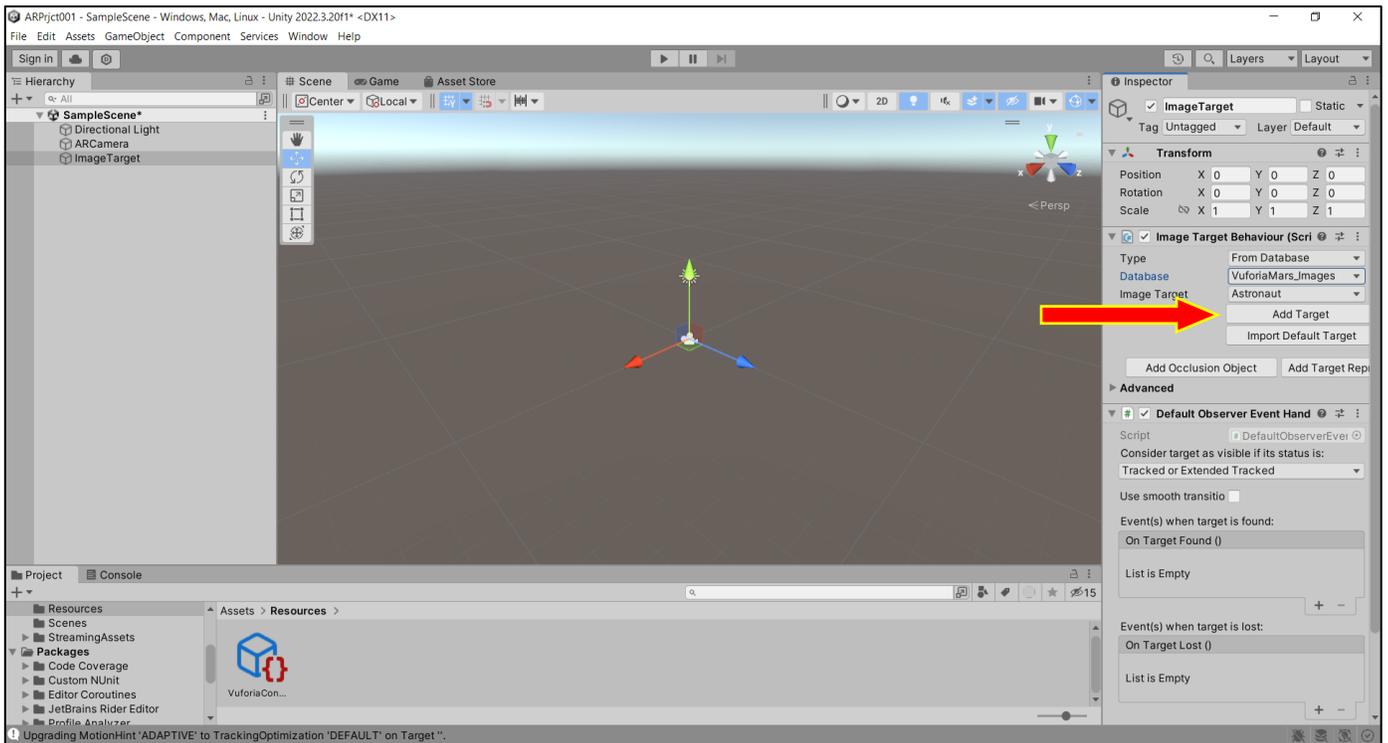
- Для этого, в падающем меню (см. рисунок выше) выбираем позицию **From Database**.

При попытке работы с базой данных таргетов (особенно при первом обращении) появляется запрос на подключение к какой-либо конкретной базе данных (через структуру, отображаемую обычно в формате **.unitypackage**)



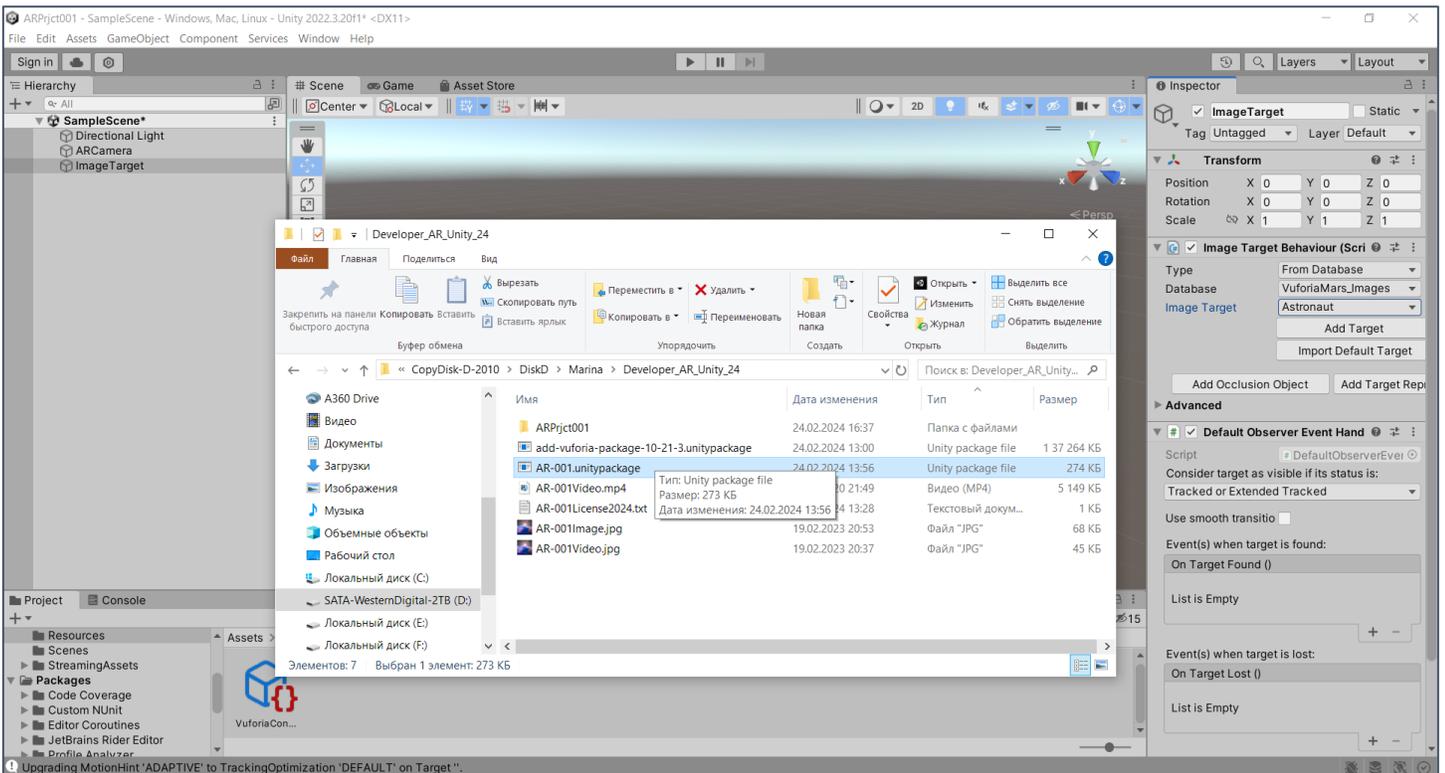
В этом случае выбираем **Import**

- В результате в области **Inspector**, в разделе **Image target Behaviour (Script)** появляется не только база данных таргетов образцов **Vuforia Engine** («Марсианская» серия карточек), но также и возможность добавить нашу собственную базу данных таргетов → **Add Target**:

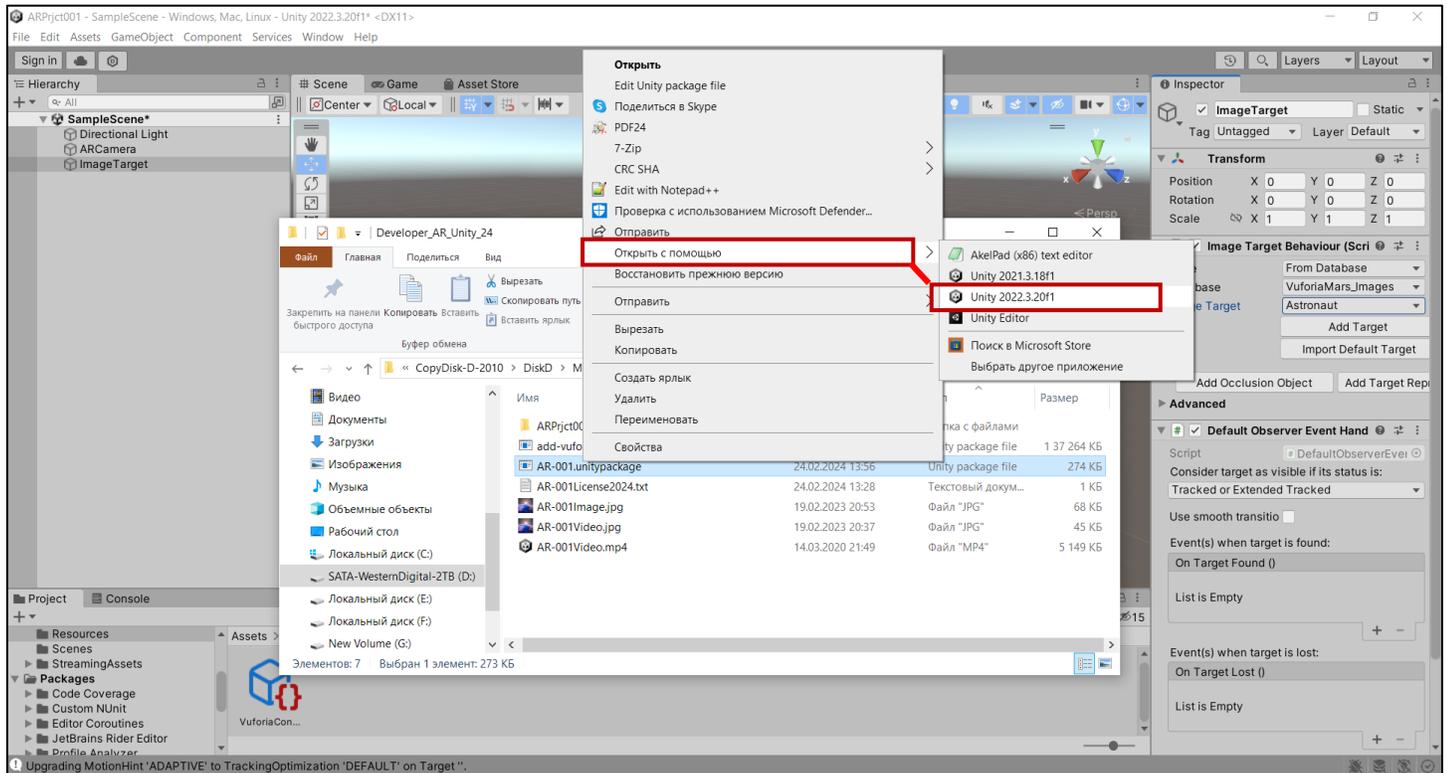


- Добавляем подготовленную заранее БД таргетов в сцену нашего проекта **AR-001**.

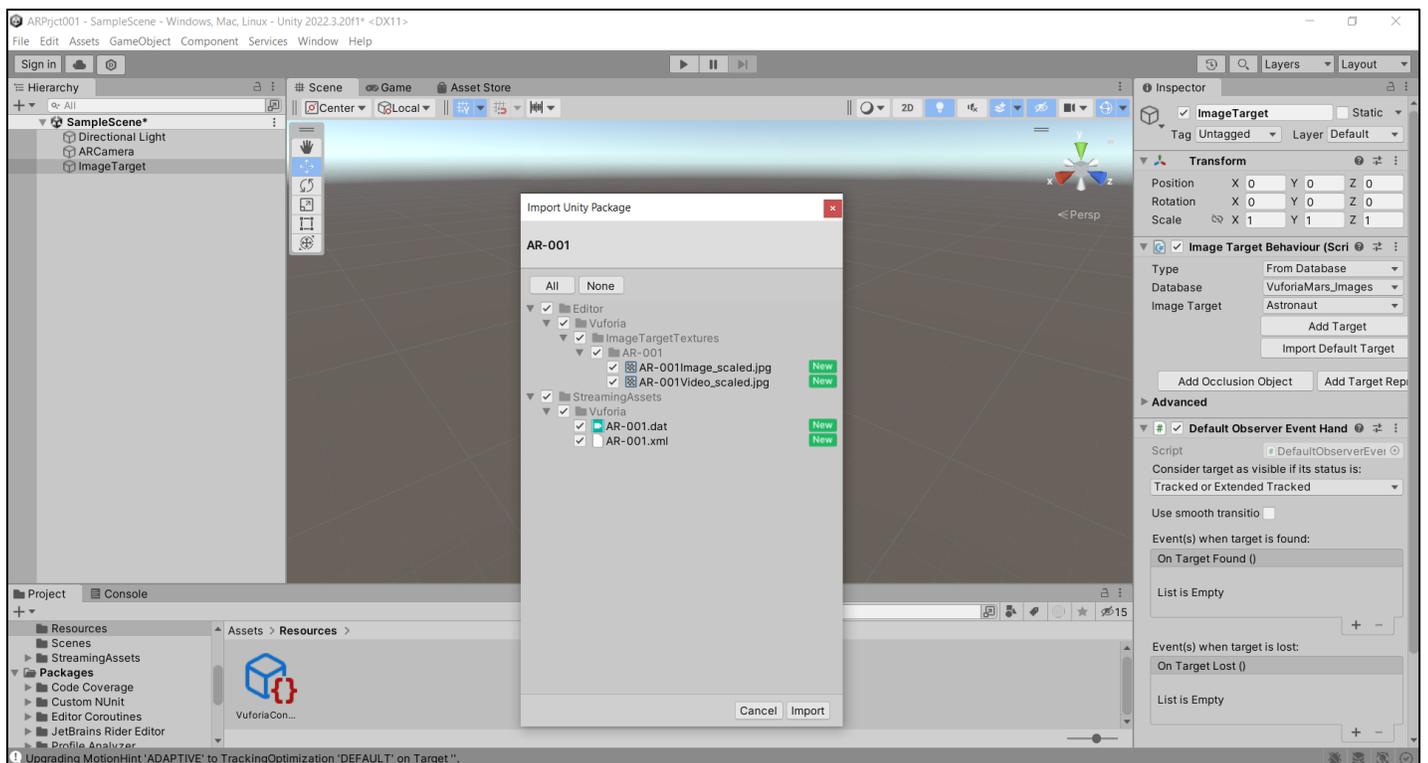
Для этого в локальной файловой структуре находим эту ранее подготовленную БД таргетов (в конце раздела 2. данного описания),



по правой клавише мыши вызываем контекстное меню работы с БД и выбираем действие **(ВОЛШЕБСТВО!!)** Open with → Unity 2022.3.20f1:

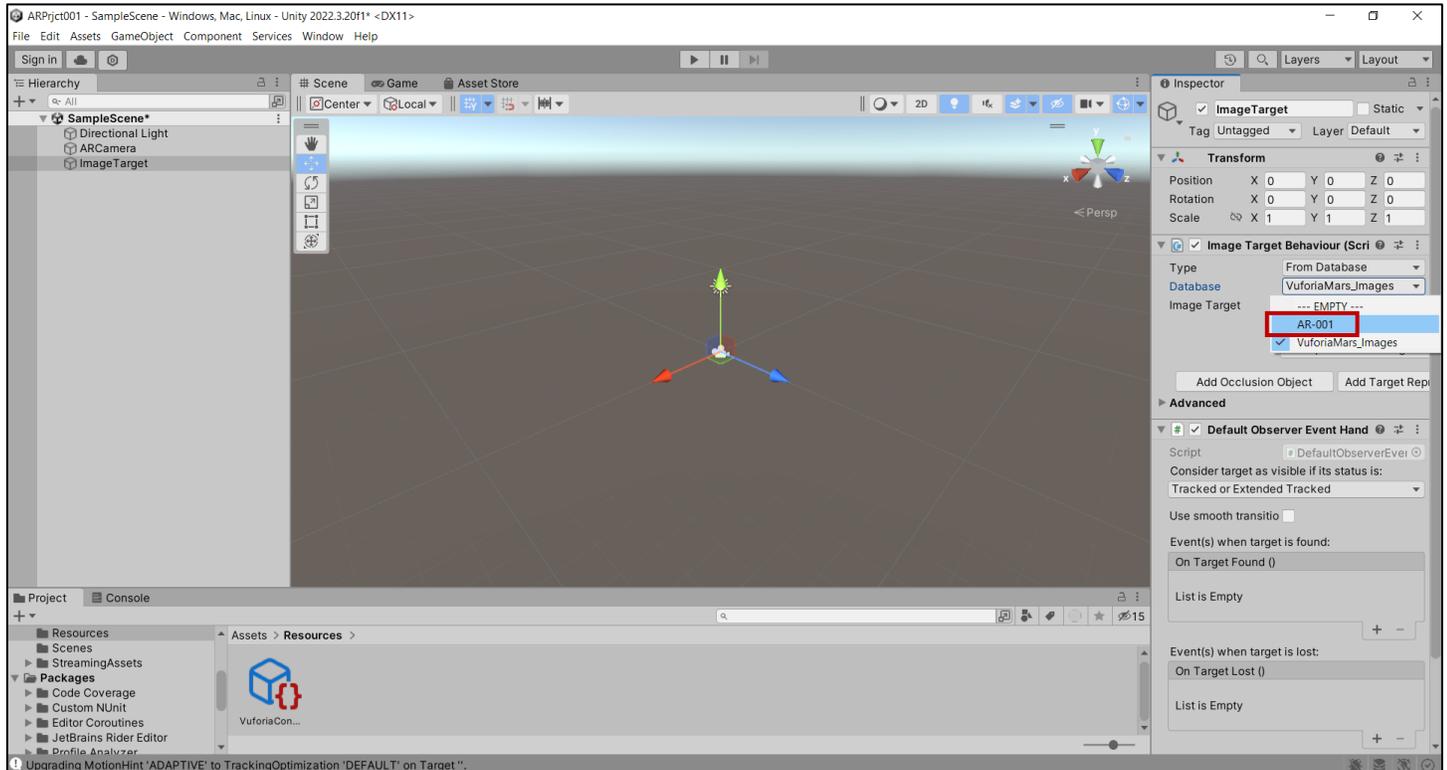


В результате в окне среды **Unity 3D** появляется диалоговое окно импорта БД таргетов:

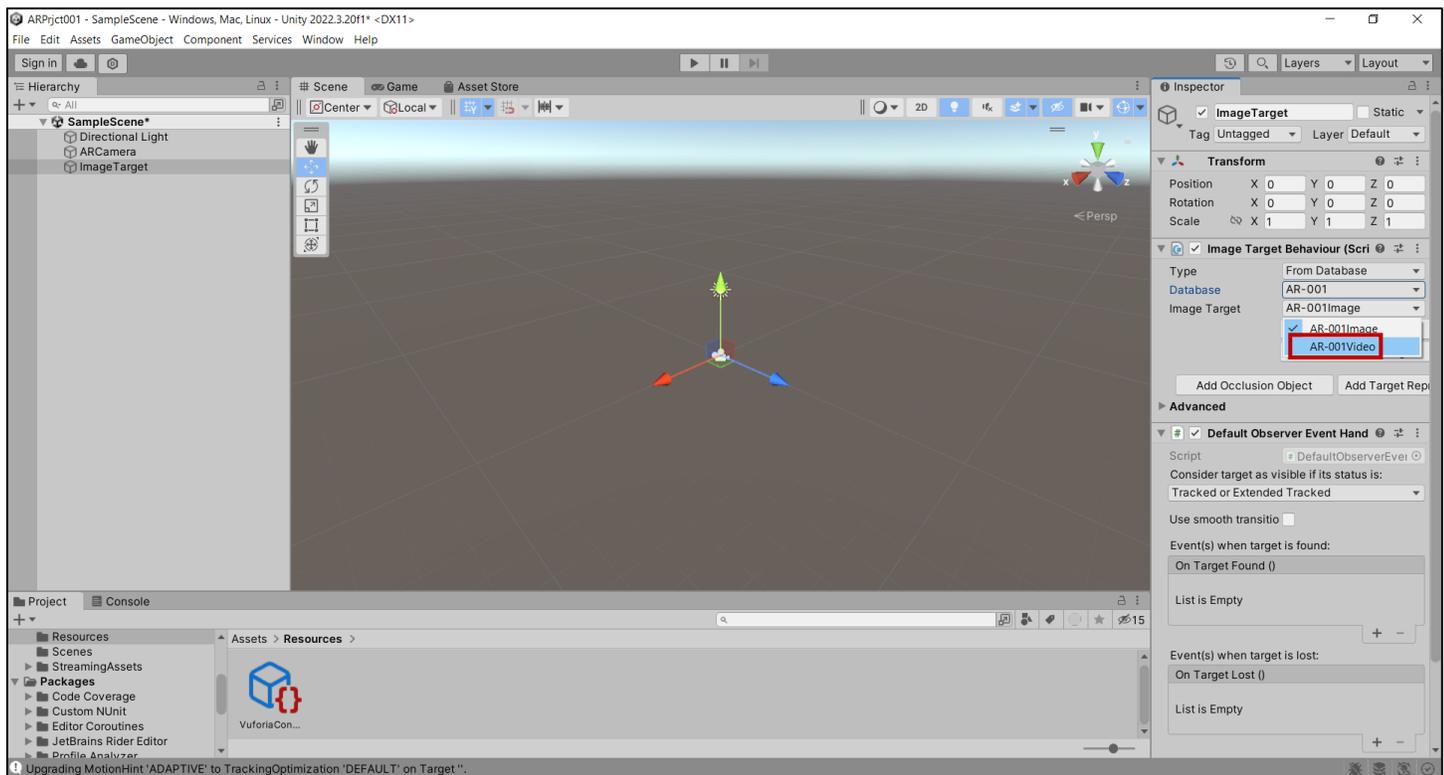


- Выполняем импорт нашей БД таргетов проекта **AR-001** путем нажатия клавиши **Import**.

Проверить результат данного действия можно, выбрав в окне **Hierarchy** позицию **ImageTarget**. Далее, в проявившемся окне **Inspector** можно убедиться в наличии БД таргетов проекта **AR-001**, выбрав выпадающий список в области **Image Target Behaviour** → **Database**:



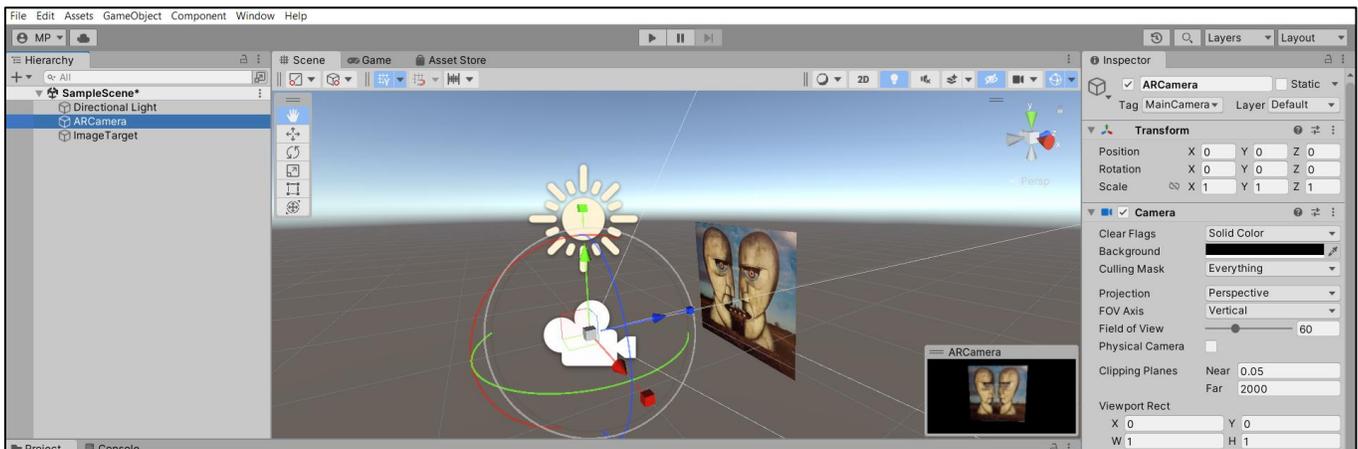
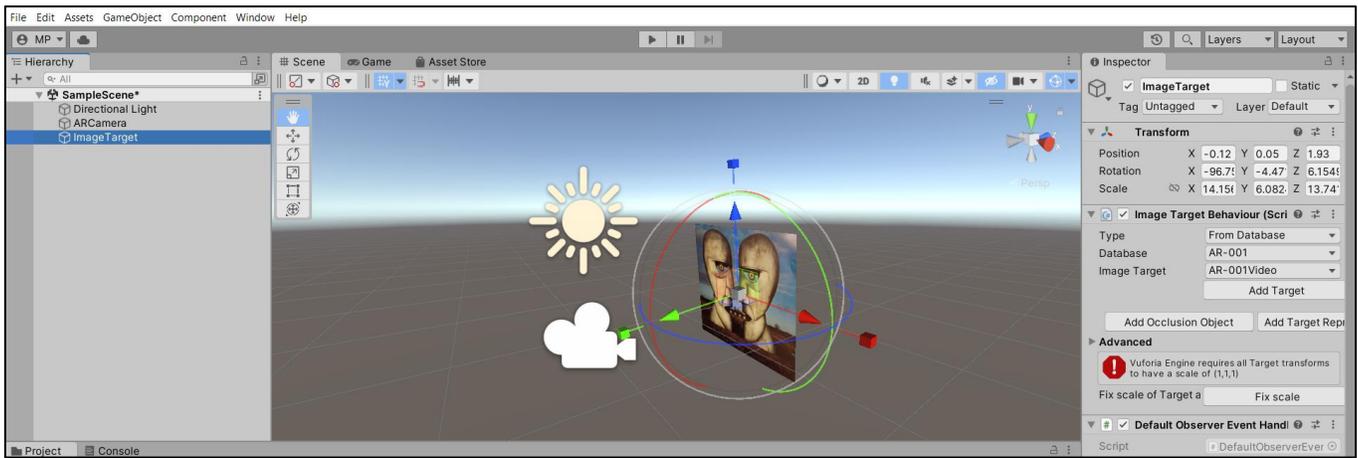
Из БД таргетов **AR-001** выбираем нужный нам **Image Target**, для данной ЛР это **AR001Video.jpg**.



Теперь в поле сцены имеются два интересующих нас объекта **Vuforia**:

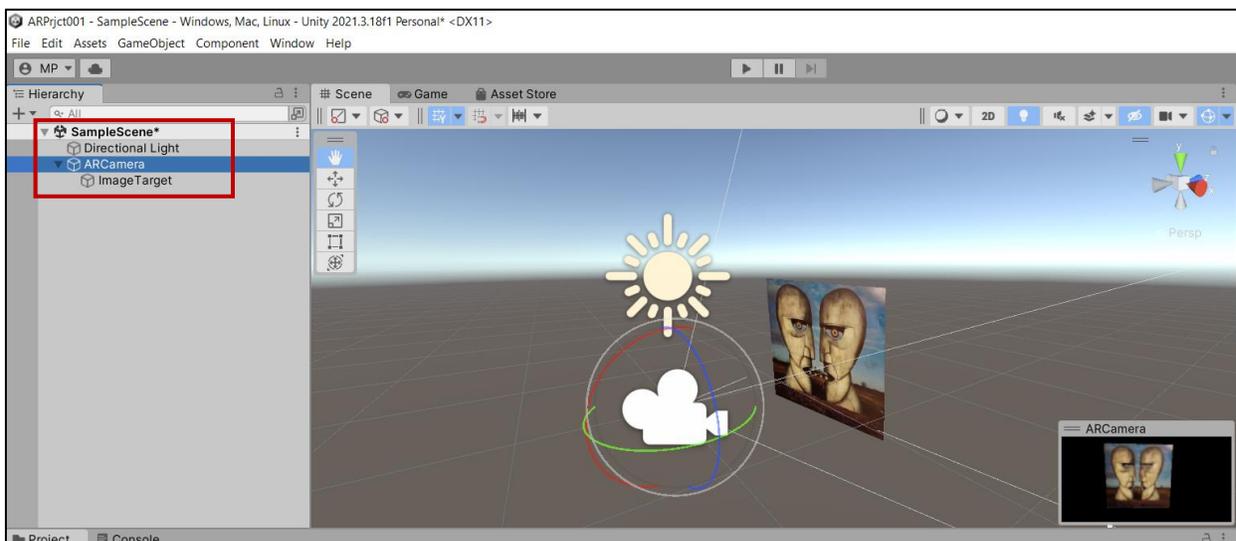
- **AR Camera** и
- **Image Target**.

Нам необходимо добиться правильного взаимного расположения камеры и таргета: например, в соответствии с вашим видением сцены, фронтально перед камерой, горизонтально на столе или на полу и пр. Контролировать правильность выполнения этого действия можно в окне предварительного просмотра (выбрать **AR Camera** в области **Hierarchy**) в нижней правой части области **Scene View** – окно **Camera Preview**. Используйте ручки **Transform (Position, Rotation, Scale)** в области **Inspector**. Удобнее всего воспользоваться инструментом **Gismo** , который размещен в области **Scene View** и позволяет визуально изменять взаимное расположение камеры и таргета, наблюдая результат наиболее удобным способом. Объекты для манипуляций - **AR Camera** и **Image Target** - выбираются в **Hierarchy**. Рекомендуется не менять положение камеры (оставьте ее в положении начала координат - 0,0,0 по трем осям), а манипулировать (вращать, поднимать, отодвигать от камеры, масштабировать) таргетом, наблюдая результат в окне предварительного просмотра, поочередно переключаясь в **Hierarchy** с **Image Target** на **AR Camera**:

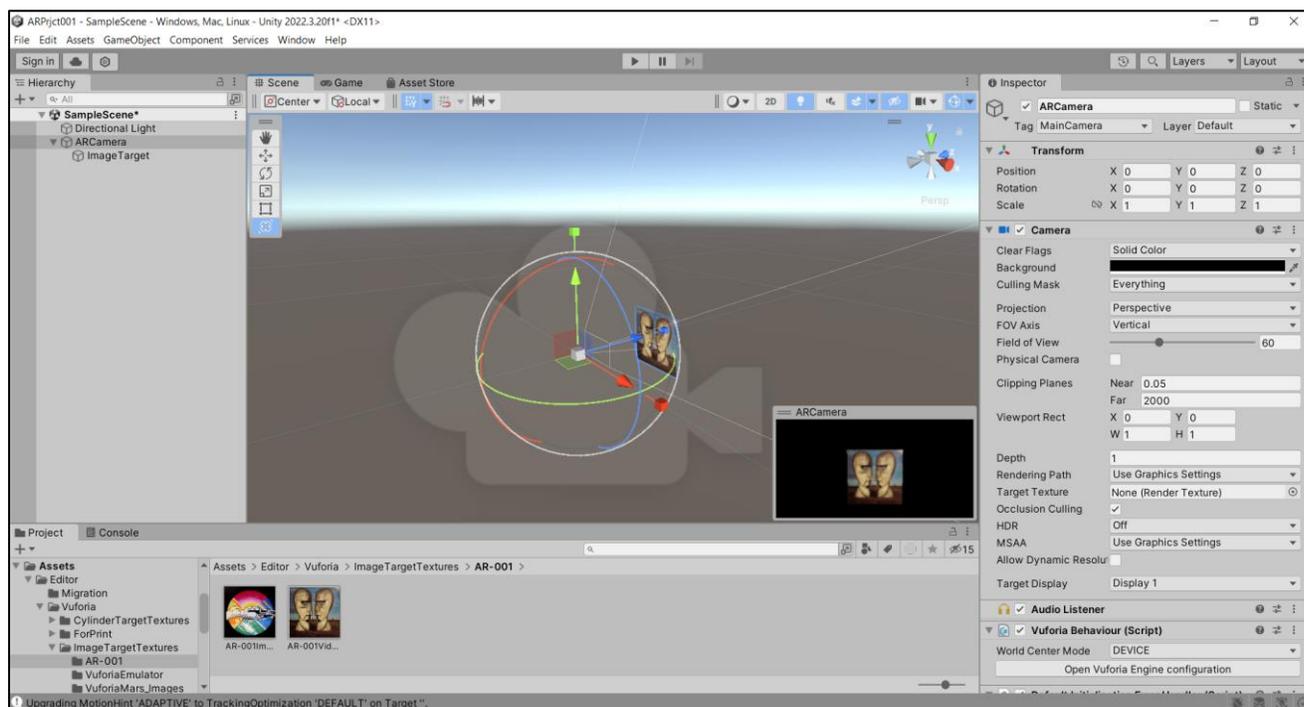


Хорошим результатом является такой, когда изображение **Image Target** занимает от 30% до 60% окна **Camera Preview**. Добейтесь этого.

Обратите внимание на положение камеры и таргета в **Hierarchy!! Image Target** и **AR Camera** в процессе этих манипуляций находятся на одном уровне. После достижения требуемого результата объект **Image Target** в **Hierarchy** должен занять подчиненное по отношению к **AR Camera** положение – выполняется т.н. **Parenting** (перетаскиванием одного объекта на другой), установление связей «Родитель-потомок». Т.о. закрепляется связка камера-таргет.



По мере выполнения всех этих действий следите за изменением наполнения области **Project** редактора **Unity 3D** и содержанием появляющихся там папок, в том числе и тех, которые имеют отношение к нашему проекту **AR-001**:



Теперь надо разместить контент, связанный с меткой, в сцене ДР.

В нашем случае метка (таргет), объект **Image Target**, уже размещен в сцене.

Контент в данной ЛР – это подготовленный файл, видеоклип – **AR-001Video.mp4**. Находится в файловой структуре на локальной машине.

Фактически в проектируемой сцене мы хотим разместить виртуальный экран, на котором будет демонстрироваться видеоклип **AR-001Video.mp4**. Демонстрация начинается только после того, как на **Мобильном Устройстве (МУ)** с установленным на нем **Приложением ДР**, происходит распознавание физического объекта – в нашем случае плоского изображения - **Image Target**, попавшего в обзор видеокамеры МУ. На данном этапе мы будем заниматься размещением наилучшим образом в **3D-пространстве** сцены виртуального **2D-экрана** и выводом на этот экран элемента контента ДР – видеоклипа (**AR-001Video.mp4**).

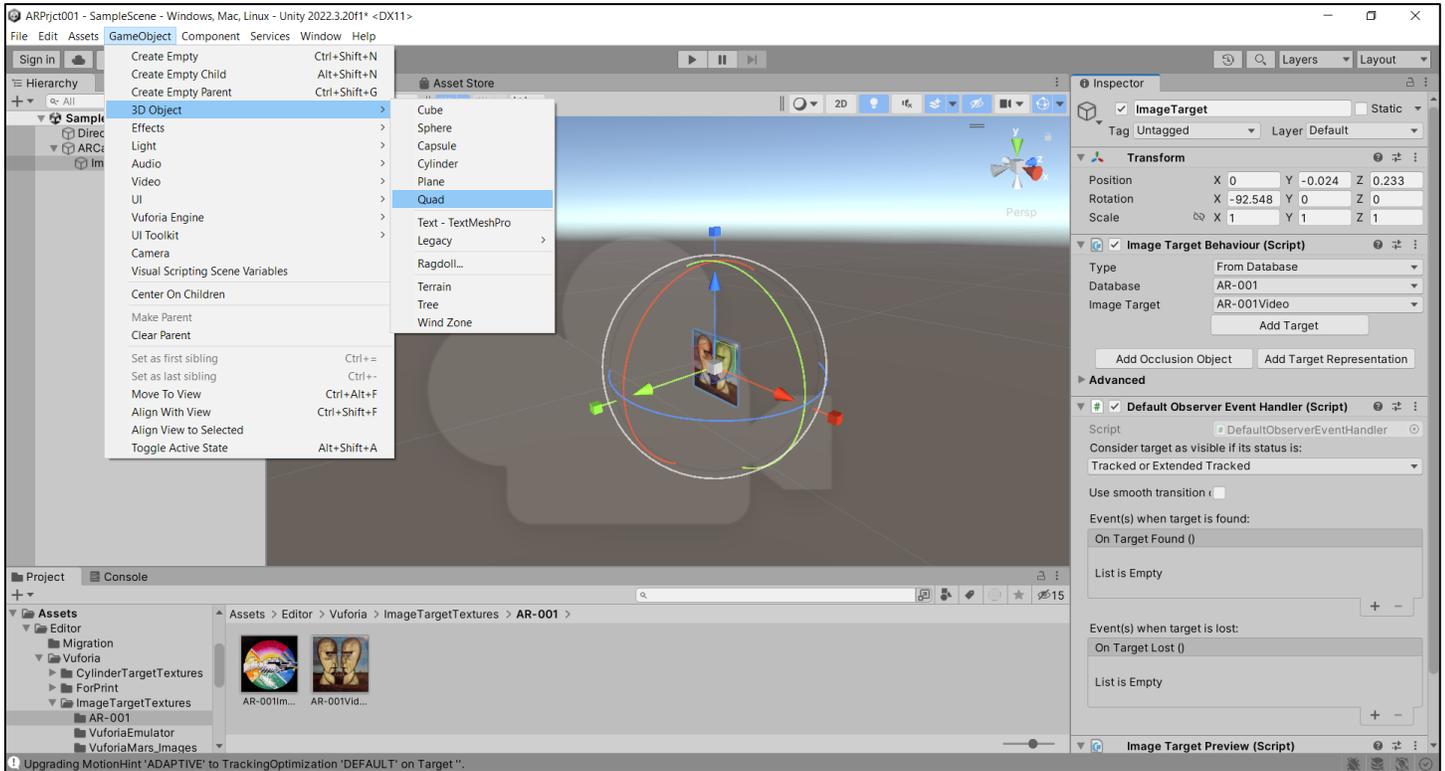
Для размещения контента в сцене **Unity 3D** воспользуемся функционалом редактора. Для этого в системе существует большое количество собственных объектов. Эти объекты можно увидеть в основной панели меню **Unity 3D** в группе **Game Object**. Один из них, чаще всего используемый для размещения **2D-контента** в трехмерном пространстве, – **Quad**.

Т.о. в **Unity 3D** связываем **Image Target** с размещаемым контентом через объект (шаблон, контейнер) **Quad**. Для такого связывания в иерархии (области **Hierarchy**) **Quad** должен располагаться под **Image Target**.

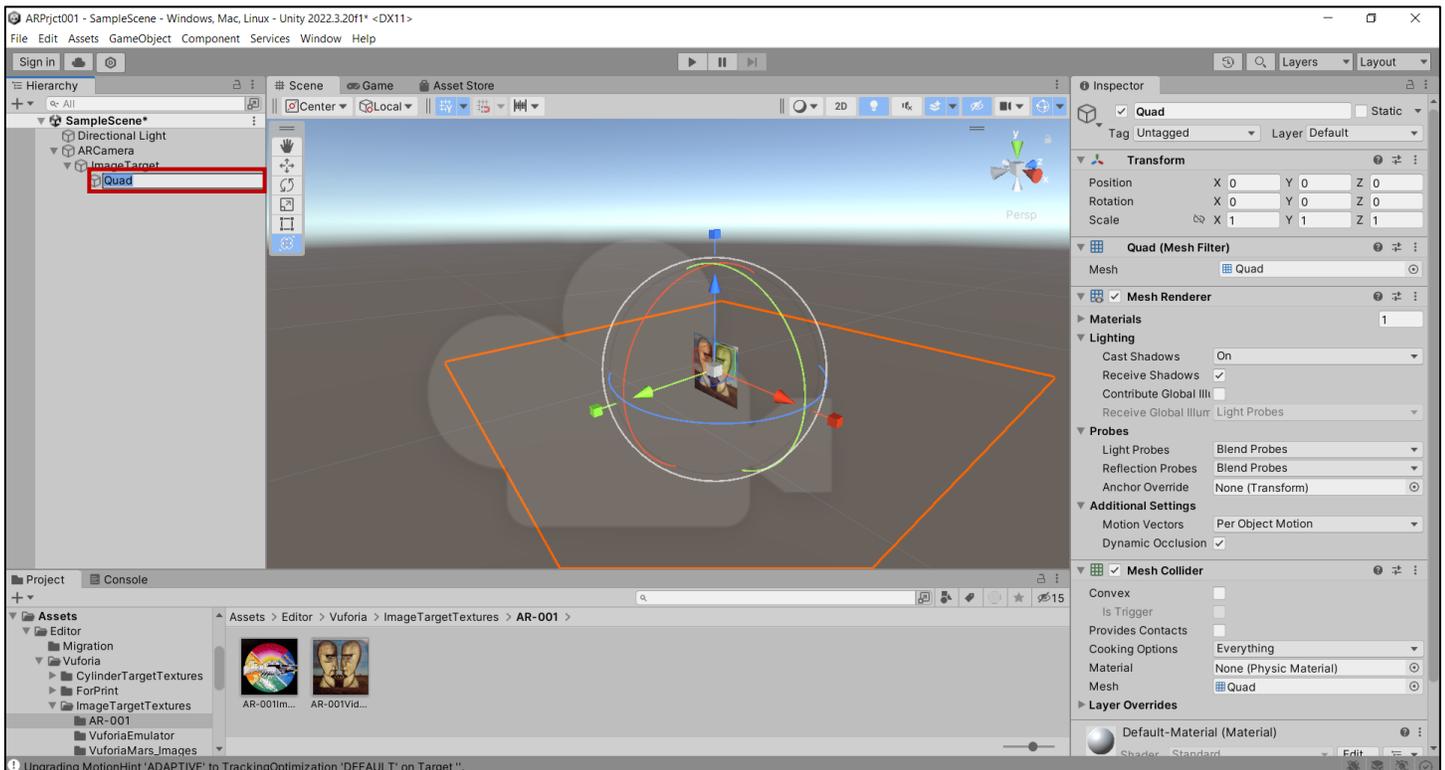
Последовательность вызовов для получения результата:

- В области **Hierarchy** выбираем **Image Target**;
- По правой клавише мыши в выпадающем меню находим строчку **3D Object**;
- В связанном с ней списке альтернатив выбираем **Quad**.

Unity 3D: 3D Object → Quad, контейнер 2D-объекта:

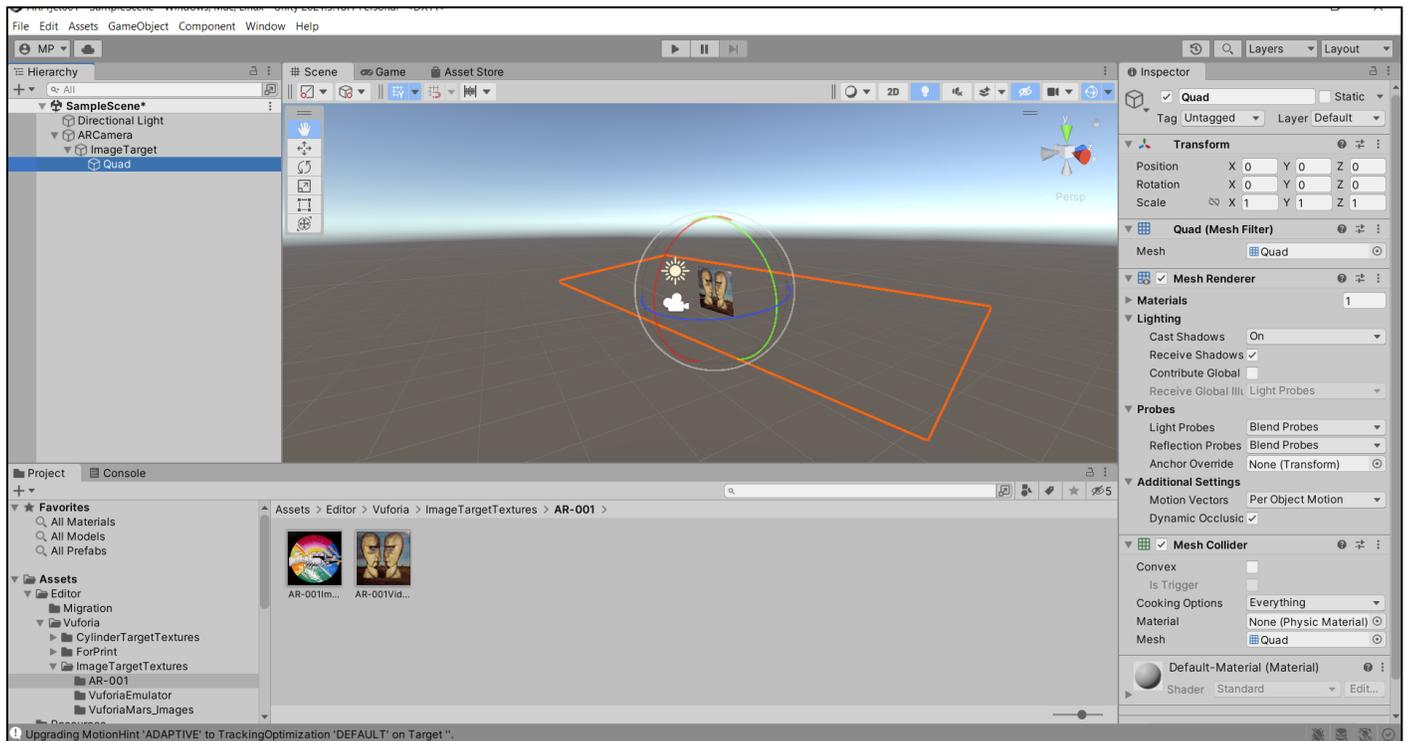


В результате в сцене появляется новый объект **Quad**, а в **Inspector'e** – информация о нем:

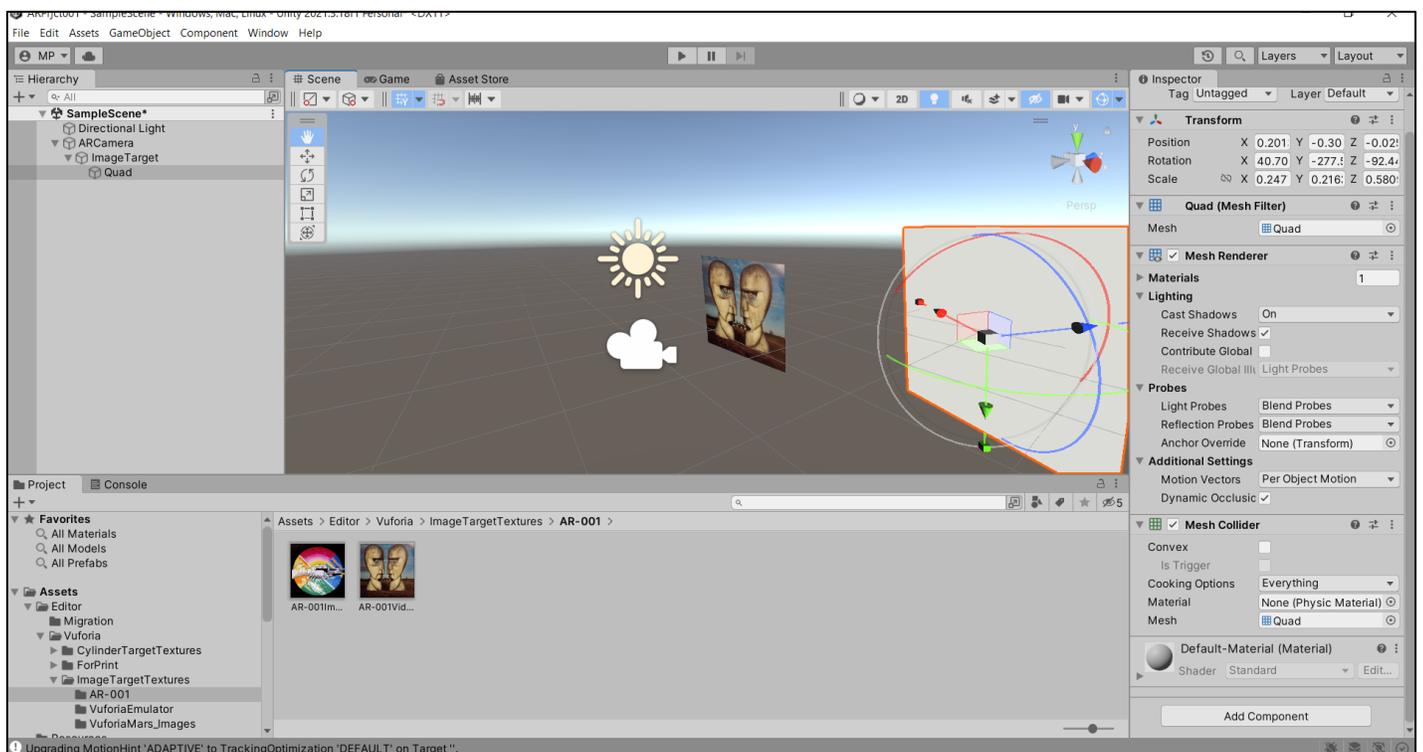


Для удобства работы со сценой можно воспользоваться привычными функциями средней клавиши мыши (уменьшить/увеличить).

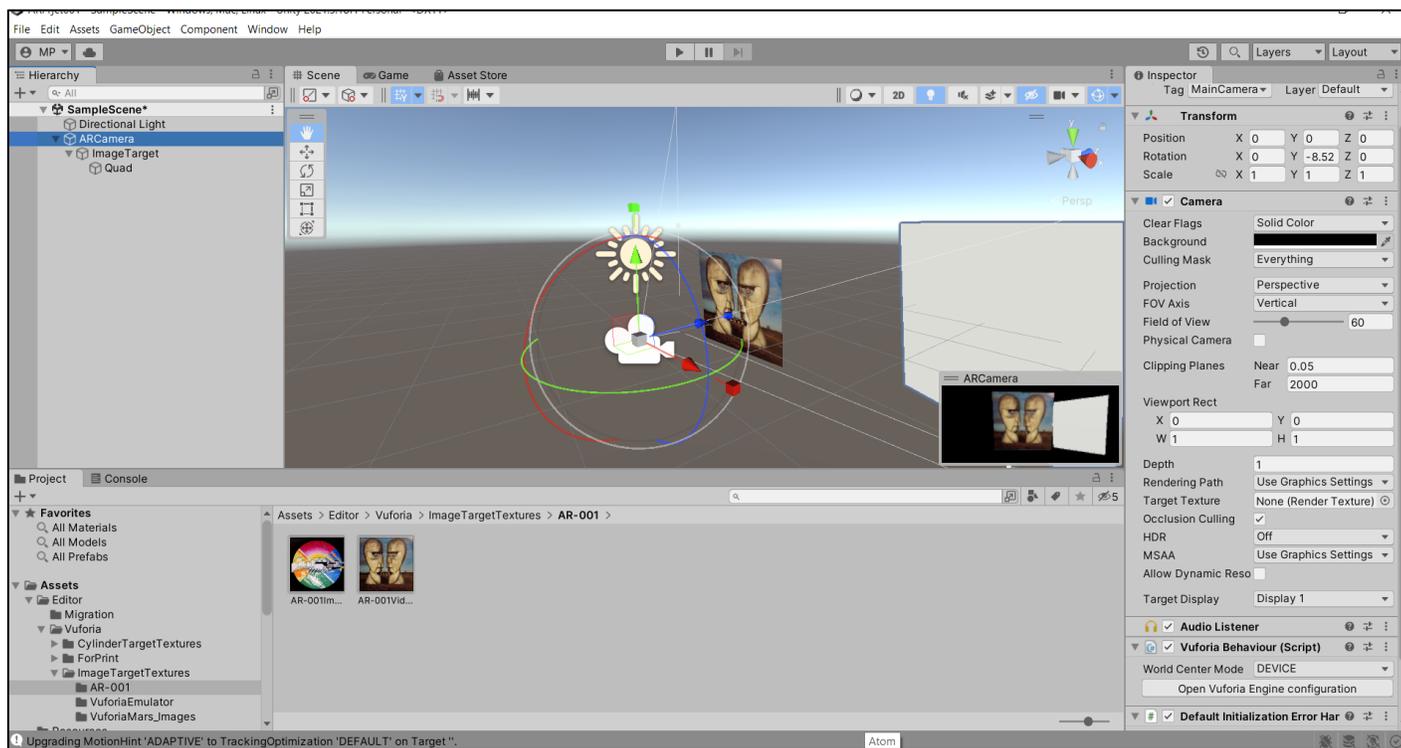
Теперь в сцене видны все объекты контента.



Не меняя расположения камеры и таргета, найдем наиболее удобное положение для контейнера видеоклипа – **Quad'a**. Например, экран с видео (**Quad** с видео) будет расположен на таком же расстоянии от камеры, что и таргет, но с небольшим разворотом по вертикали (**ось «Y»**). Одновременно уменьшим и размеры экрана. Вы можете выбрать расположение самостоятельно (с помощью инструмента **Gismo** или ручек **Transform** в **Inspector'e**).

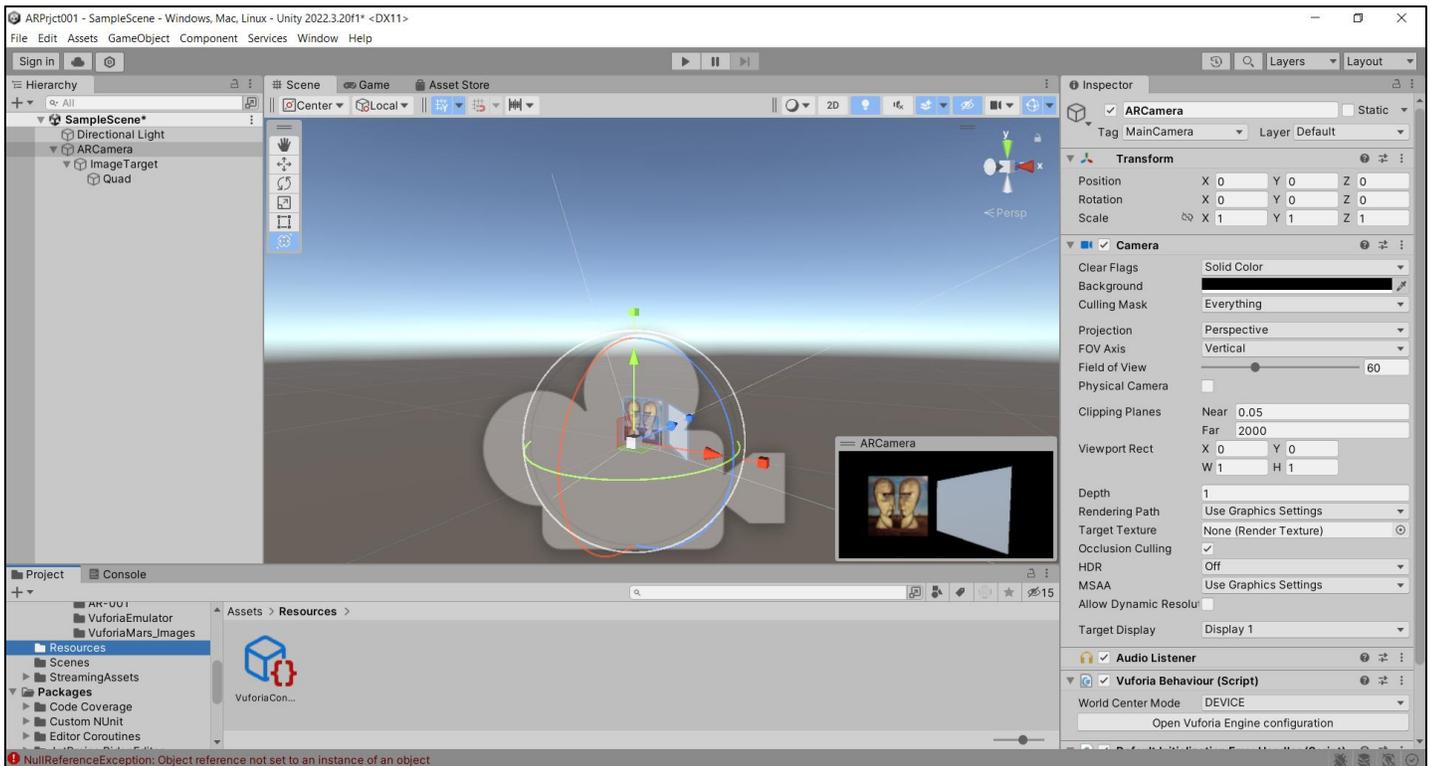


Рекомендуем манипулировать только **Quad**’ом, а **ARCamera** использовать для контроля результата в окне **PreView**. Если вы смотрите на экран фронтально, он не прозрачен!!

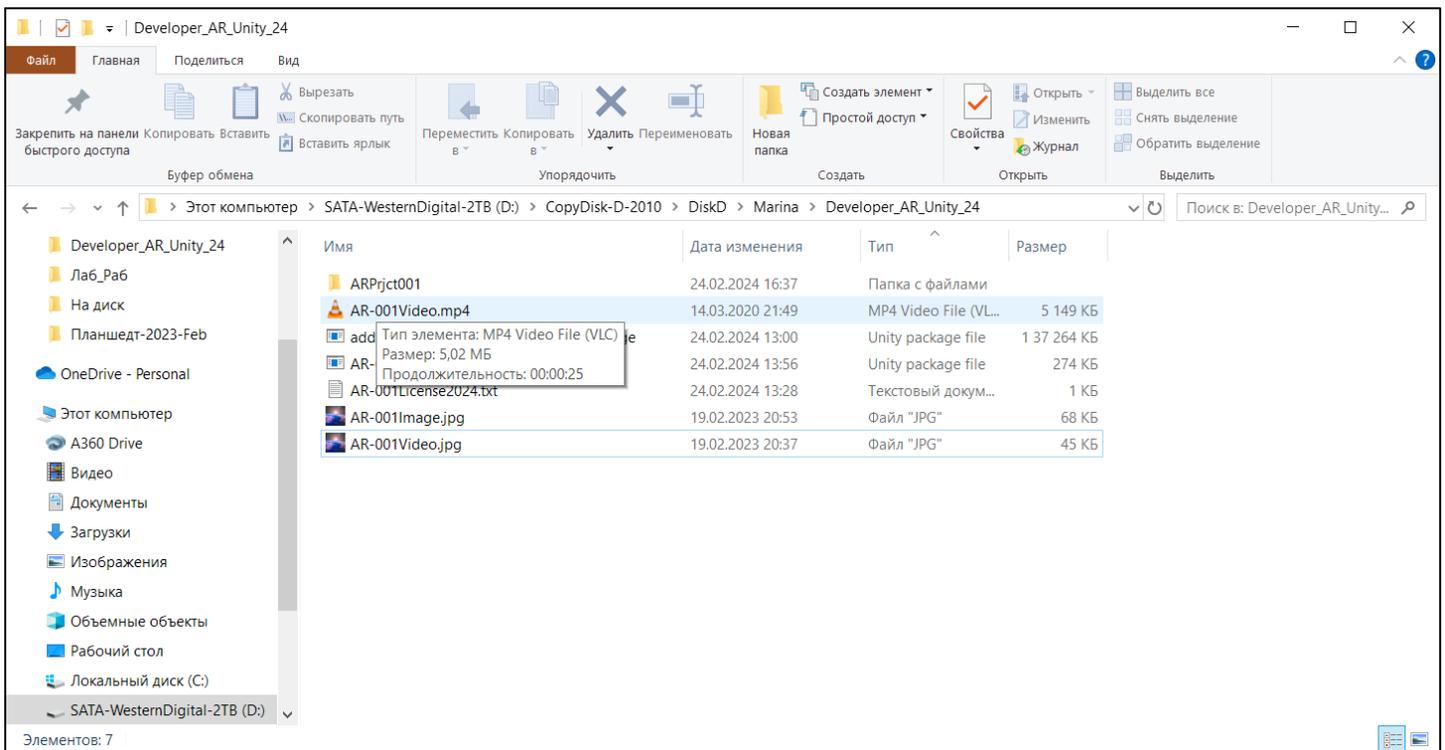


В результате сцену можно считать предварительно сформированной – в ней размещены **ARCamera**, таргет и контейнер для **2D**-контента – видеоклипа.

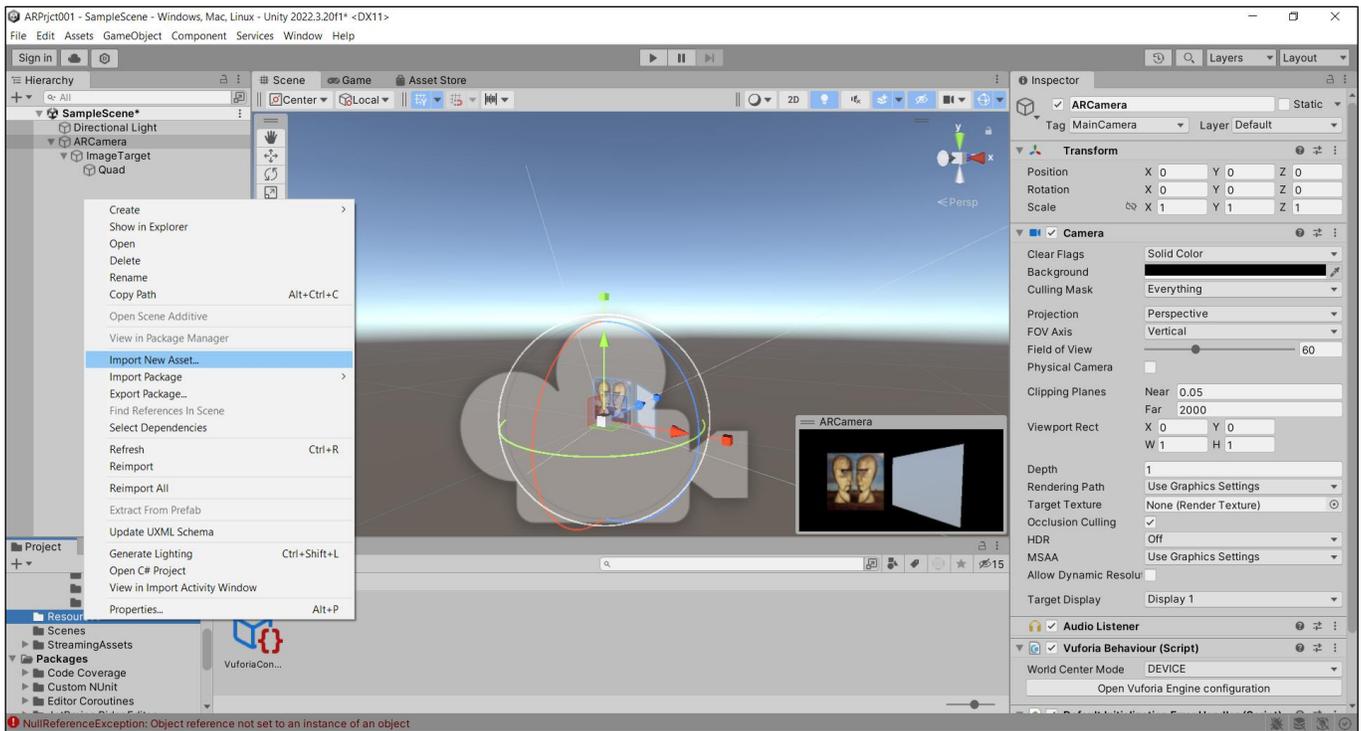
ВАЖНО!! Элементы контента такого типа (пользовательские видеоклипы, изображения и пр.) нашего Приложения ДР при работе с **Unity** должны быть помещены в раздел ресурсы (**Resources**) библиотеки активов (**Assets**). Пока этот раздел пустой:



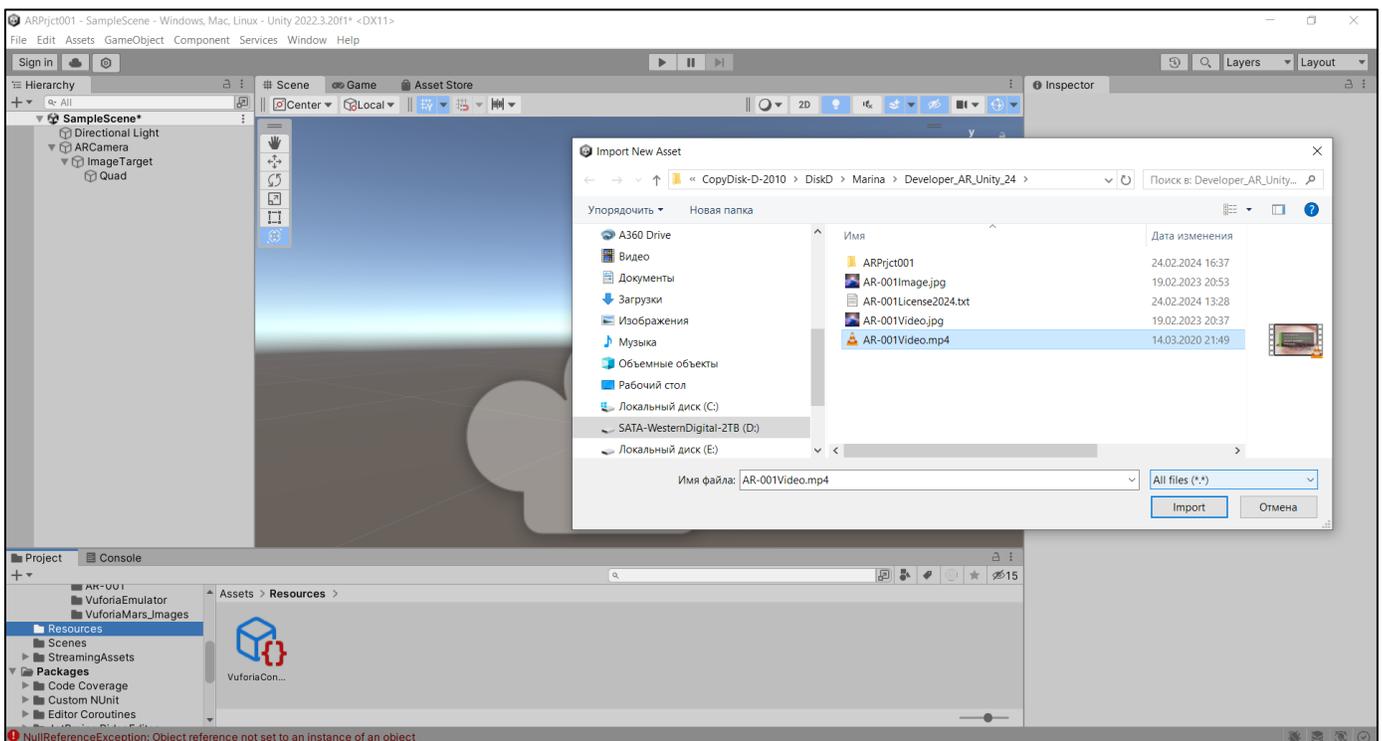
В локальной ФС находится заранее подготовленный файл в формате **.mp4** для использования его в Приложении ДР - **AR-001Video.mp4**:



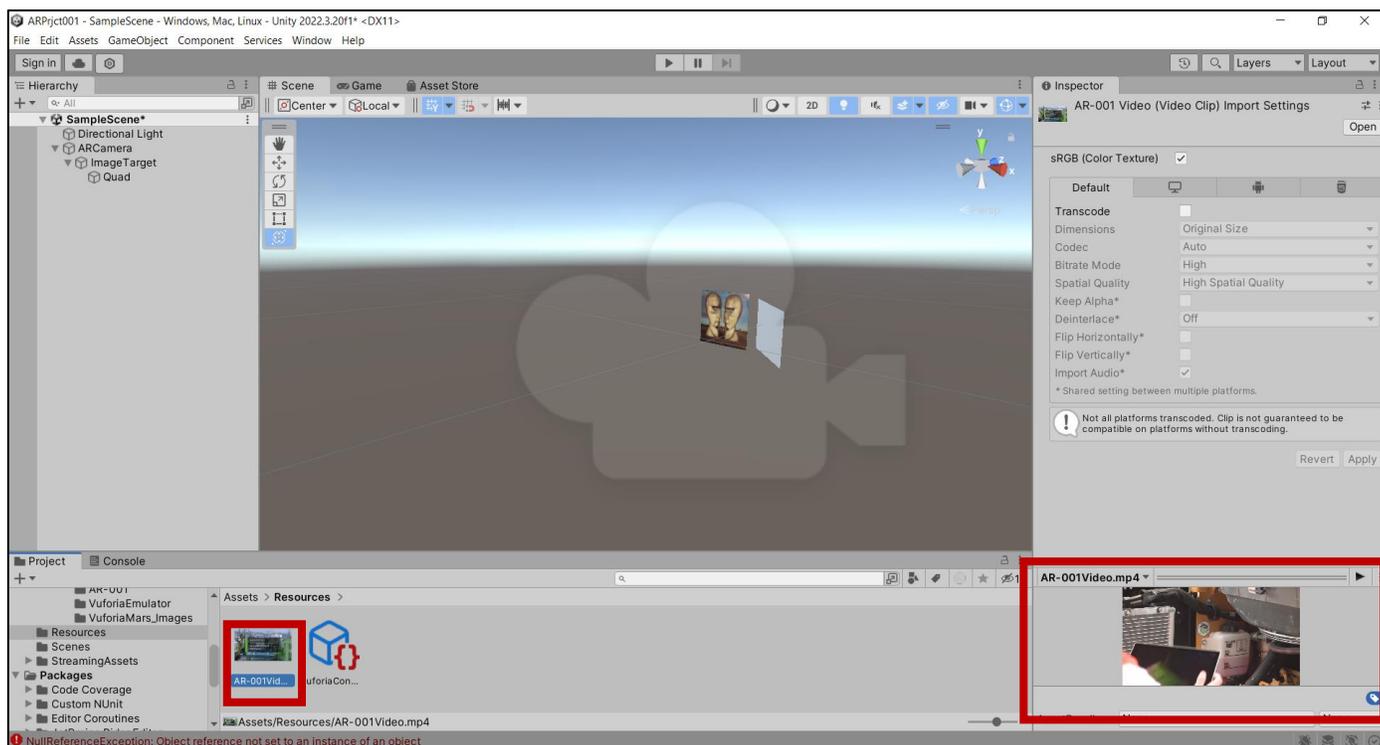
Для того, чтобы поместить элемент контента ДР - видеоклип в подраздел **Resource** раздела **Assets** перейдем на закладку **Resource Project Window** редактора **Unity**, в ее свободной области по правой клавише мыши (**RMB**) вызываем меню, в котором выбираем позицию **Import New Asset**:



В результате попадаем в локальную файловую систему, где находим выбранный файл с видеоклипком:

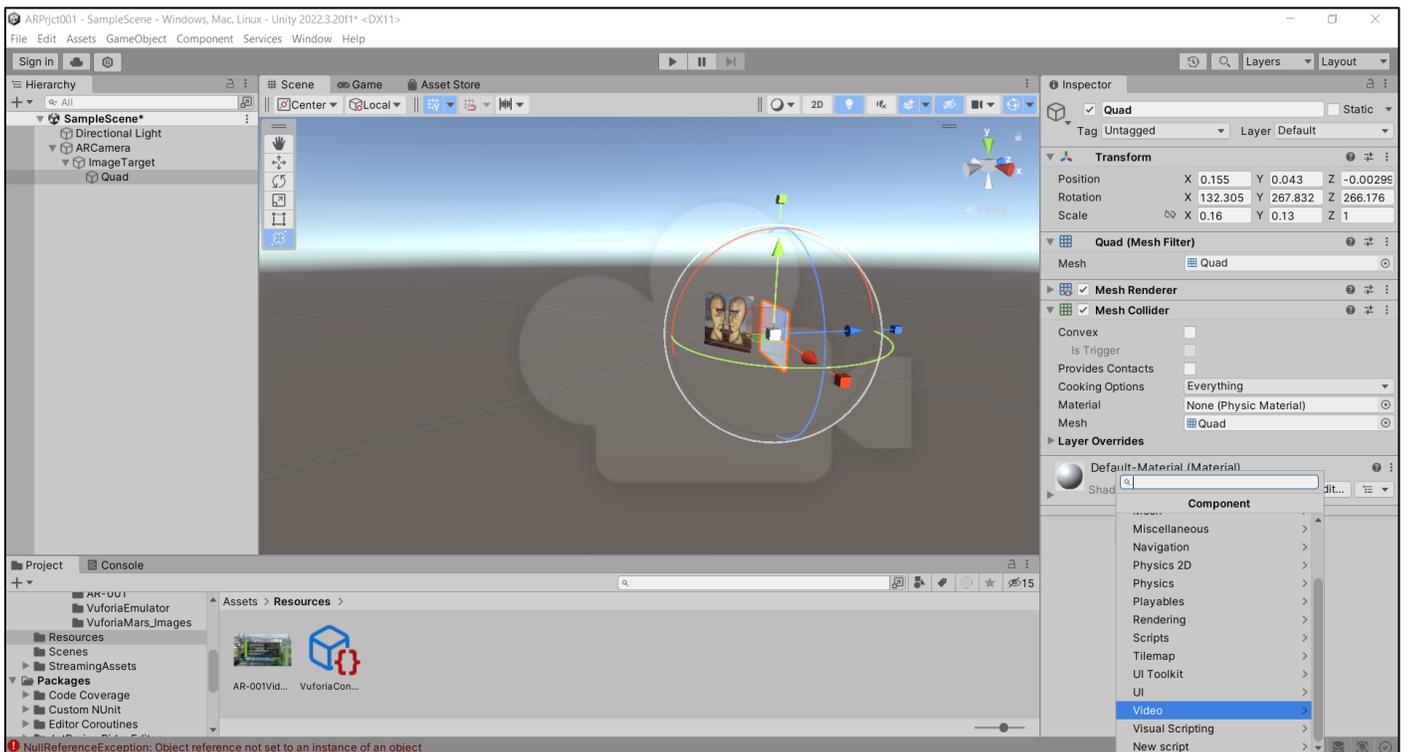
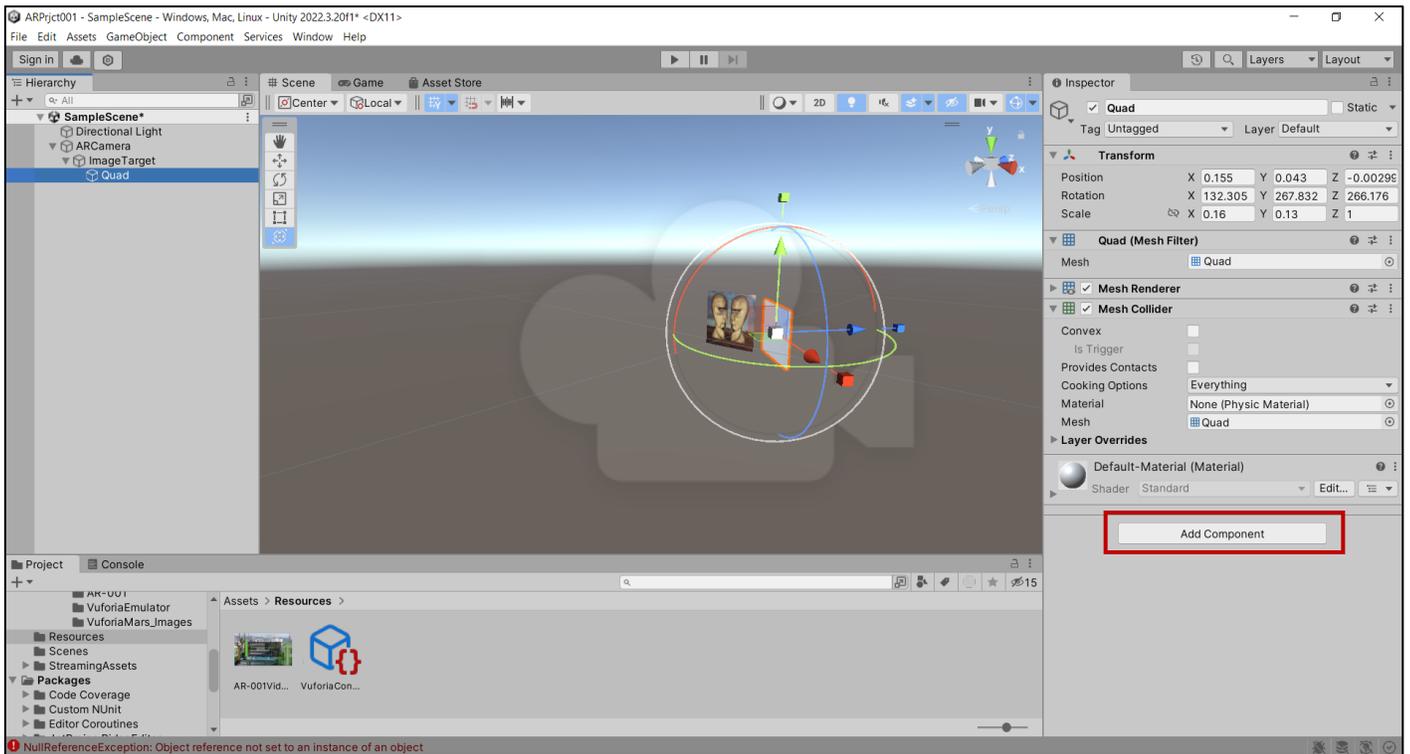


Импортируем выбранный файл, в результате чего он появляется в разделе ресурсы (**Resources**) библиотеки активов (**Assets**) в области **Project Window** редактора **Unity**:

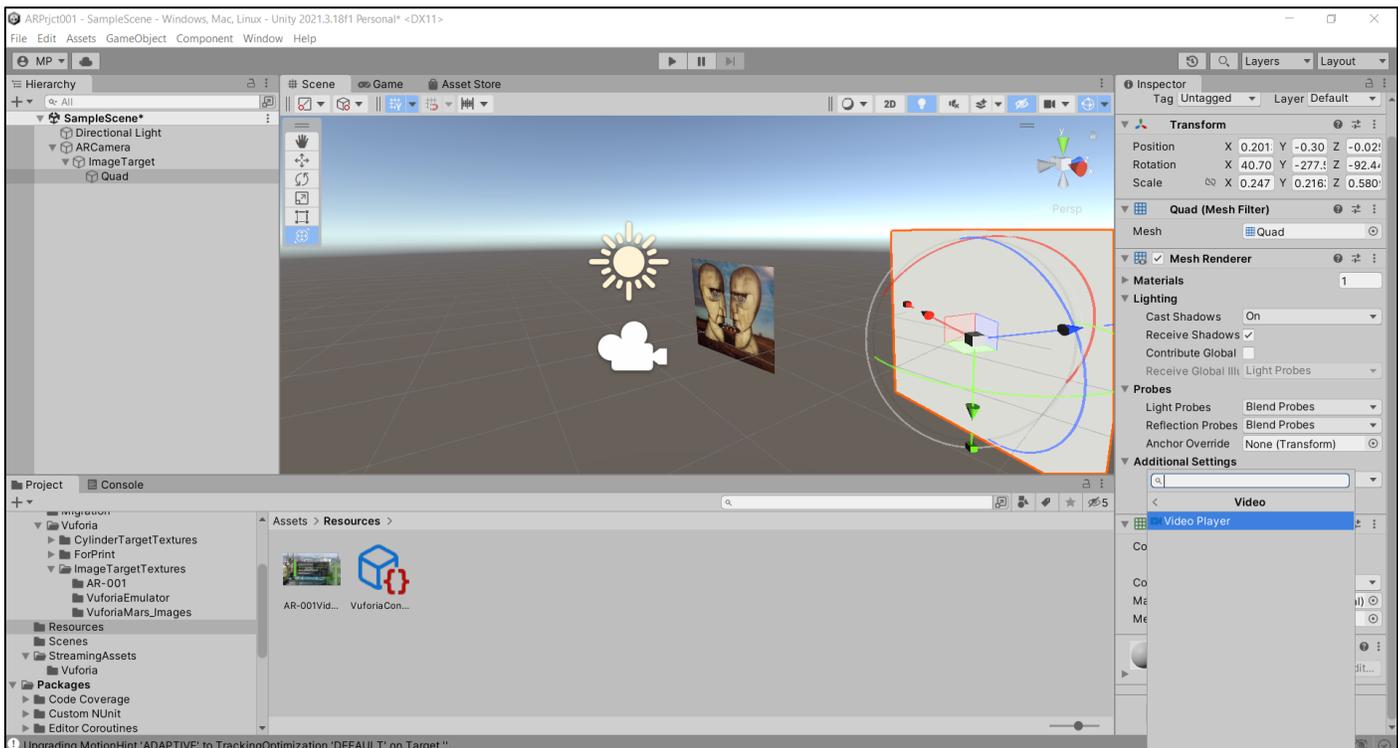
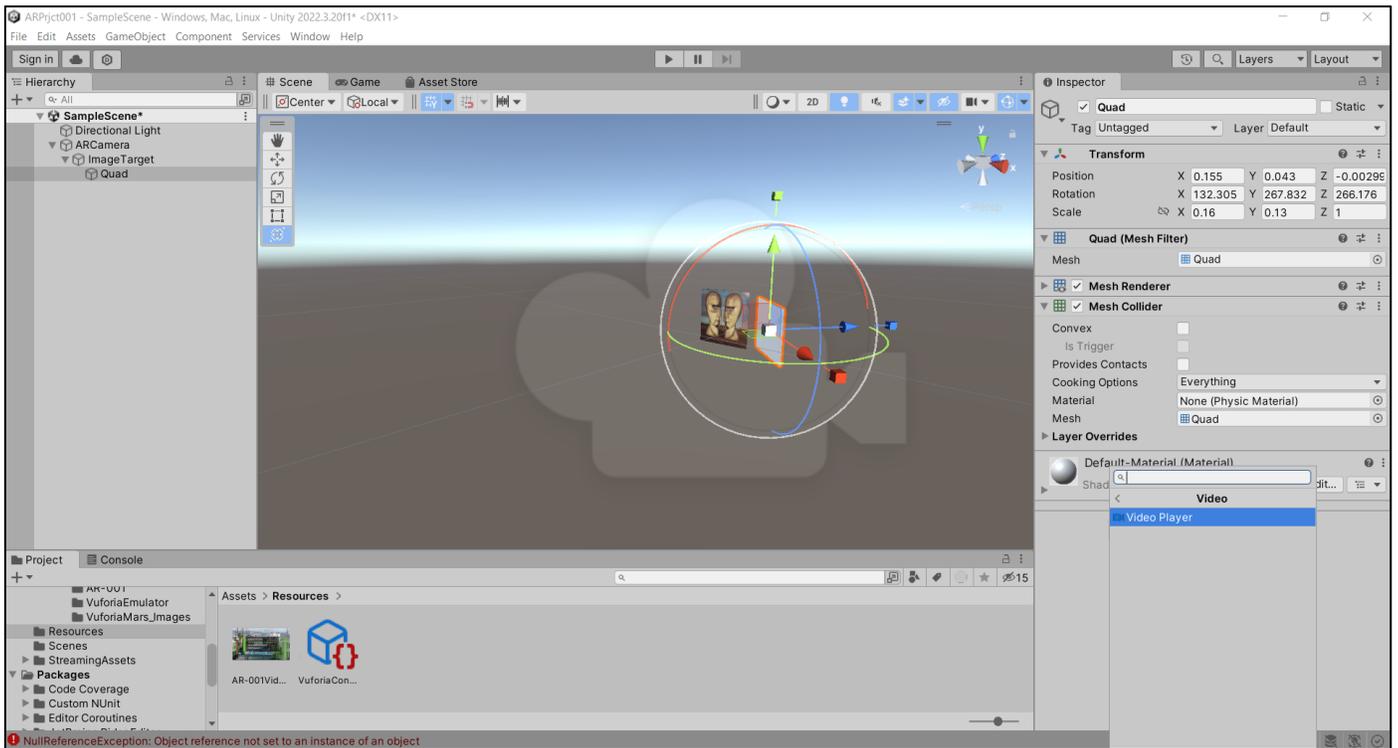


Следует иметь в виду, что по логике работы **Unity 3D контентом является как визуализируемый объект (в нашем случае видеоклип), так и «устройство», этот объект визуализирующее.** У нас это – заранее подготовленный файл с видеоклипом - **AR-001Video.mp4**, и компонент **Unity 3D**, отвечающий за визуализацию объектов типа «видео» - **Video Player**. **Оба эти объекта (AR-001Video.mp4 и Video Player) необходимо связать с контейнером Quad.**

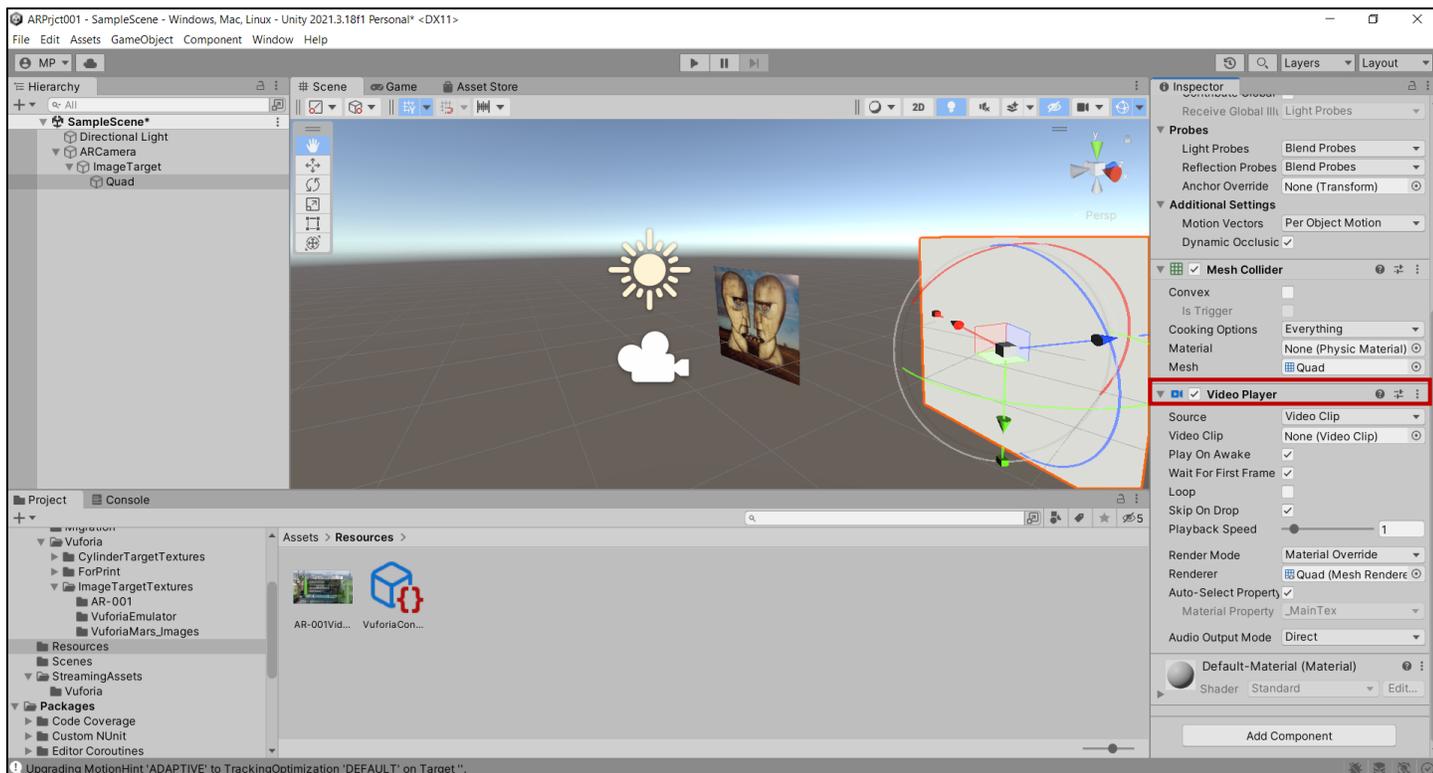
Начинаем наполнять контентом контейнер **Quad**. В первую очередь помещаем в контейнер компонент **Video Player**. Для этого в области редактора **Hierarchy** выбираем **Quad**, а в появившемся **Inspector'e** – кнопку **Add Component**:



В выпадающем списке находим компонент **Video Player** по двухступенчатому пути выбора:
Add Component → **Video** → **Video Player**:

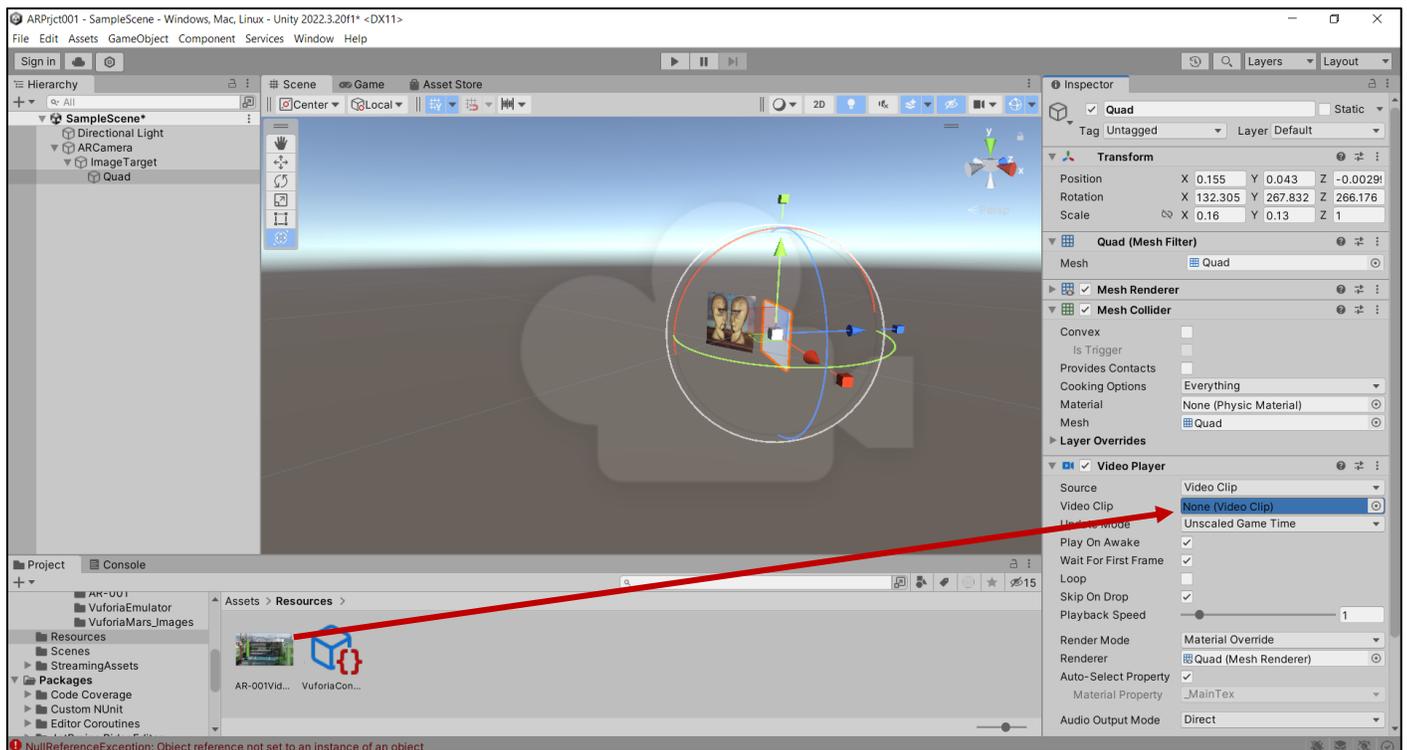


В результате в **Inspector**'е для **Quad** появится область настройки видеоплеера:



ВАЖНО!! Обратите внимание на выбранные и установленные параметры **Video Player**.

Далее, методом **drag-n-drop** перемещаем видеоклип (**AR-001Video.mp4**) из **Project Window** в поле **Video Clip** раздела **Video Player Inspector**'а Quad:



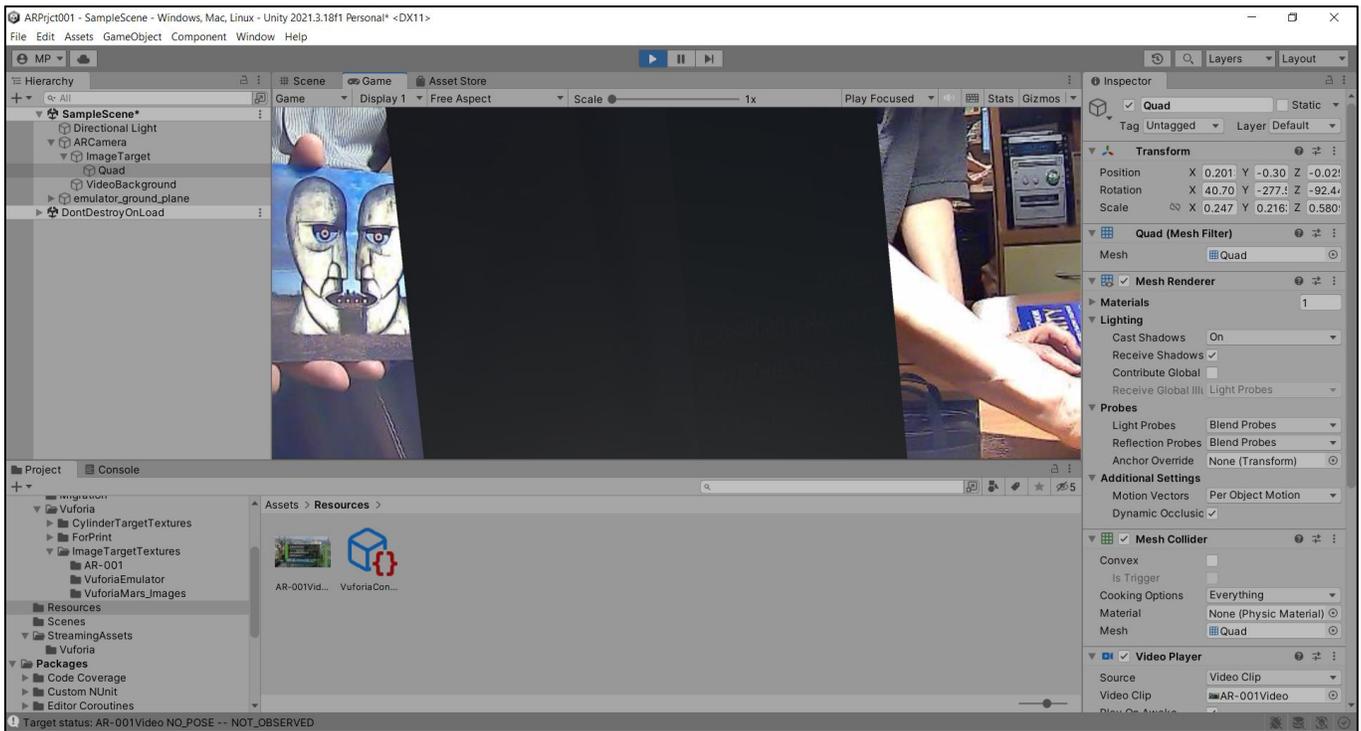
ВАЖНО!! Таким образом в **Unity 3D** сформирована сцена Приложения ДР:

- В сцену включена **ARCamera** вместо **Main Camera Unity 3D**;
- Импортирован таргет – **Image Target**;
- С таргетом связан контейнер **Quad** для размещения видеоклипа;
- В **Quad** в качестве контента загружен **визуализатор – плеер** – и подготовленный видеоклип.

Сохраняем достигнутый результат в **Unity 3D: File → Save Project**

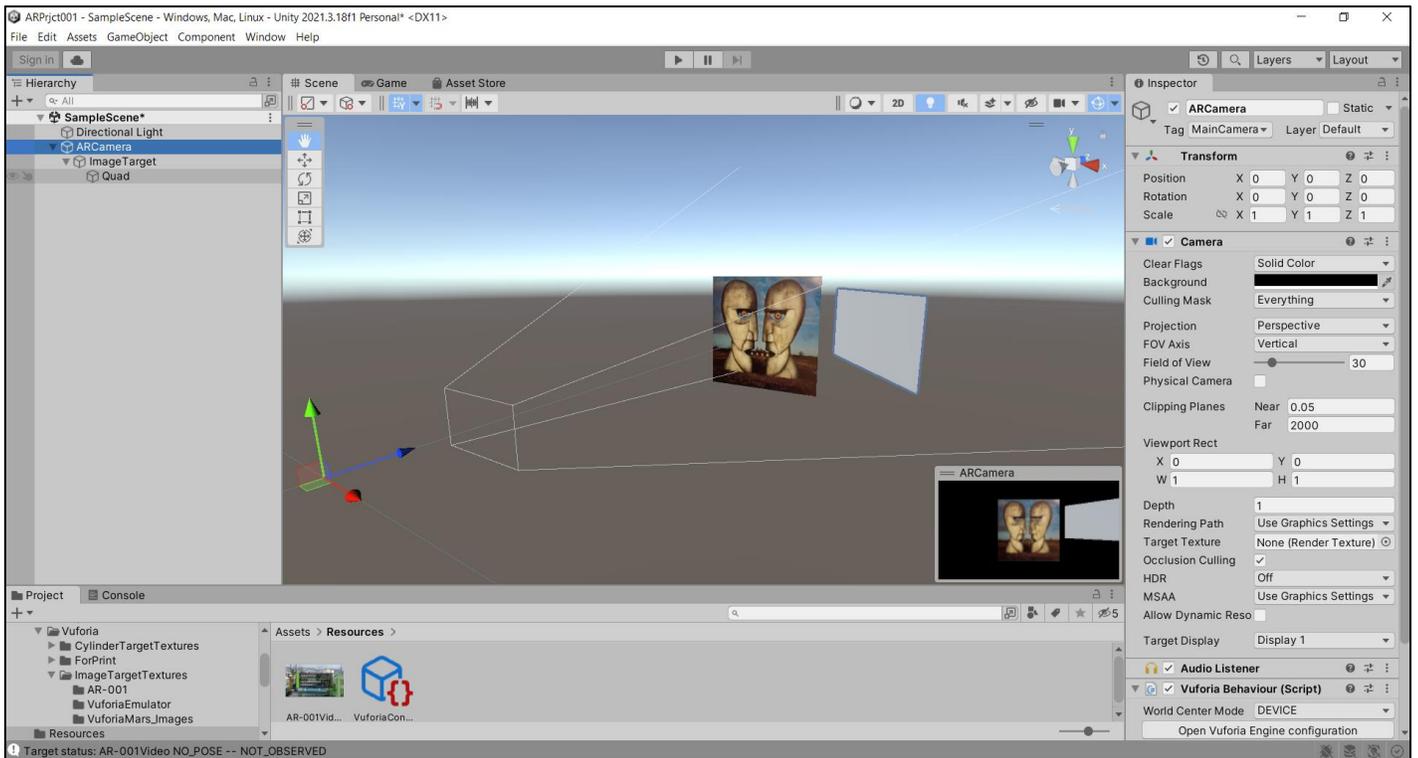
Проверить работоспособность разрабатываемого Приложения ДР непосредственно в **Unity 3D** можно на локальной машине, если она снабжена видеокамерой.

Предпросмотр достигнутого результата выполняется в режиме **Game** (закладка) в области **Scene View** редактора, нажатием на клавишу **Play**:

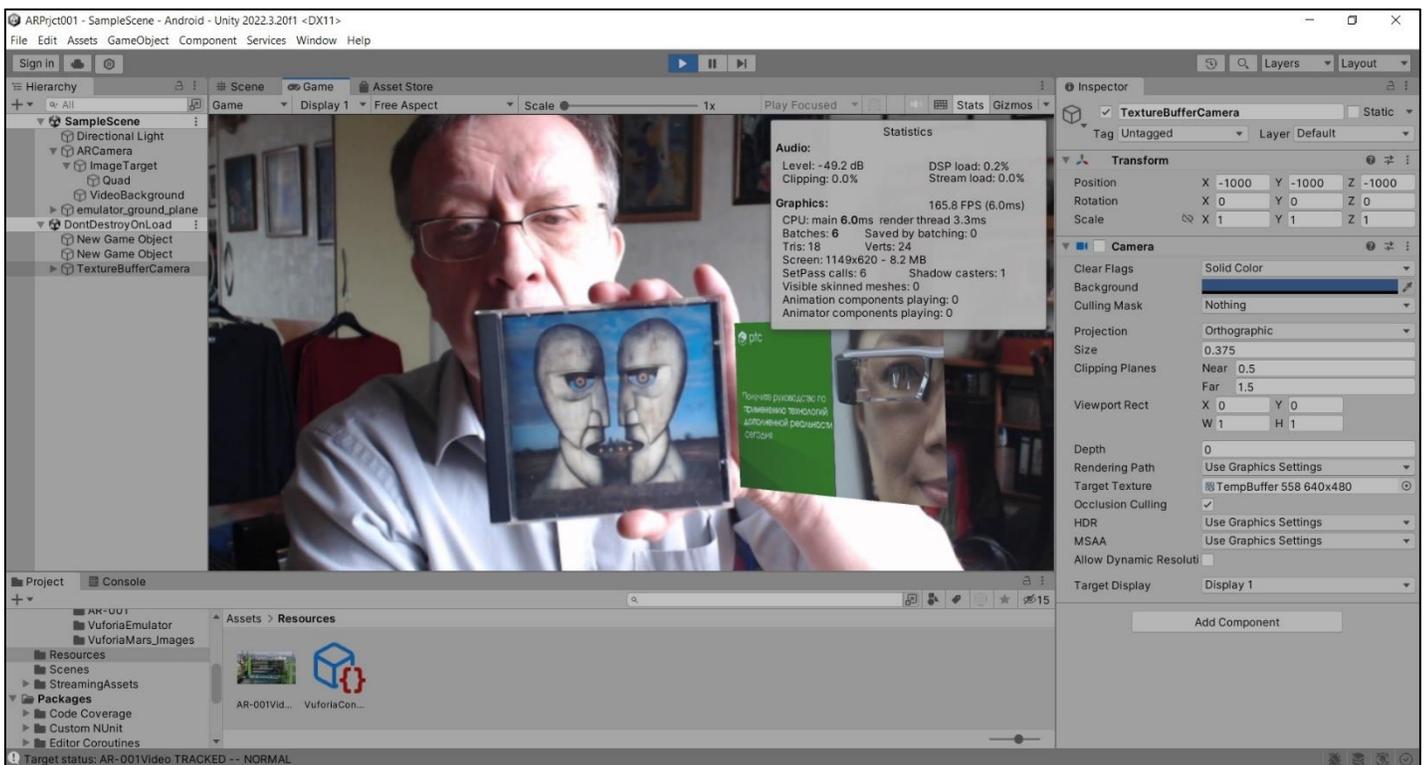


Полученный результат показывает, что таргет и клип работают, но бывает так, что сцена сформирована некорректно – экран, скорее всего, развернут к зрителю тыльной стороной и взаимное расположение таргета и экрана возможно следует изменить таким образом, чтобы в область трансляции МУ попадал весь экран (**Quad**).

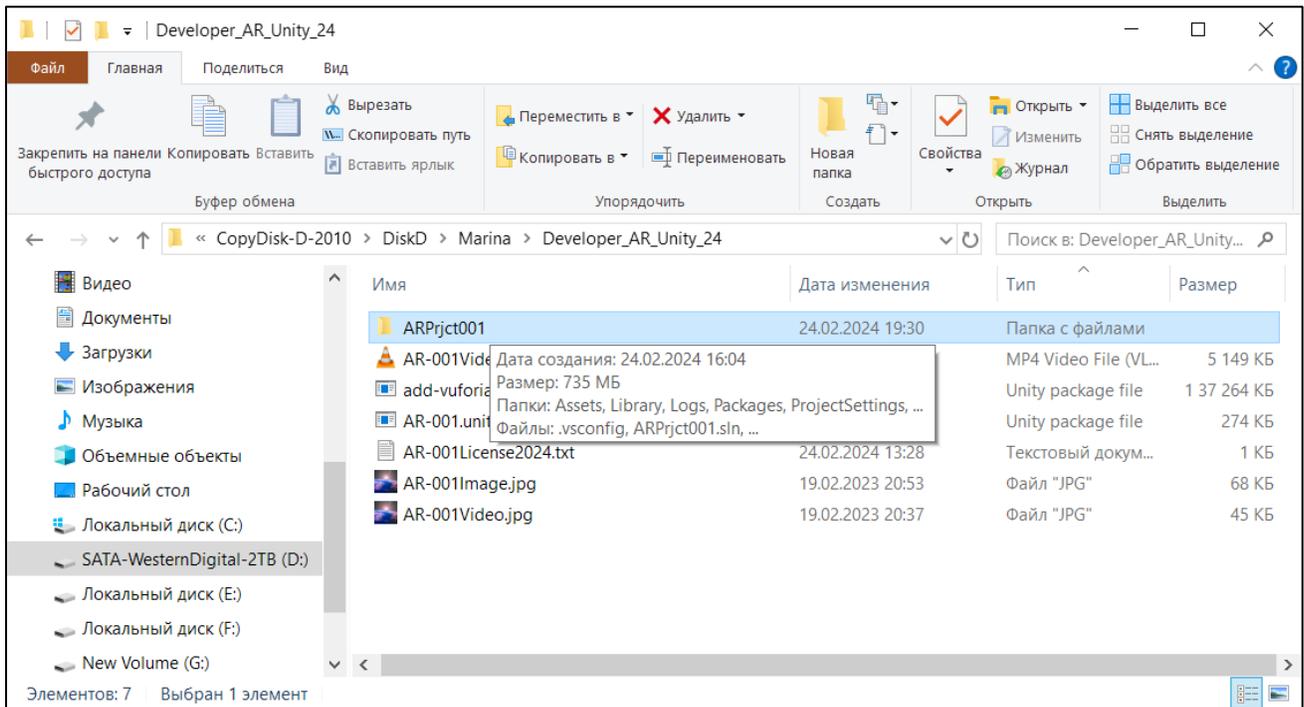
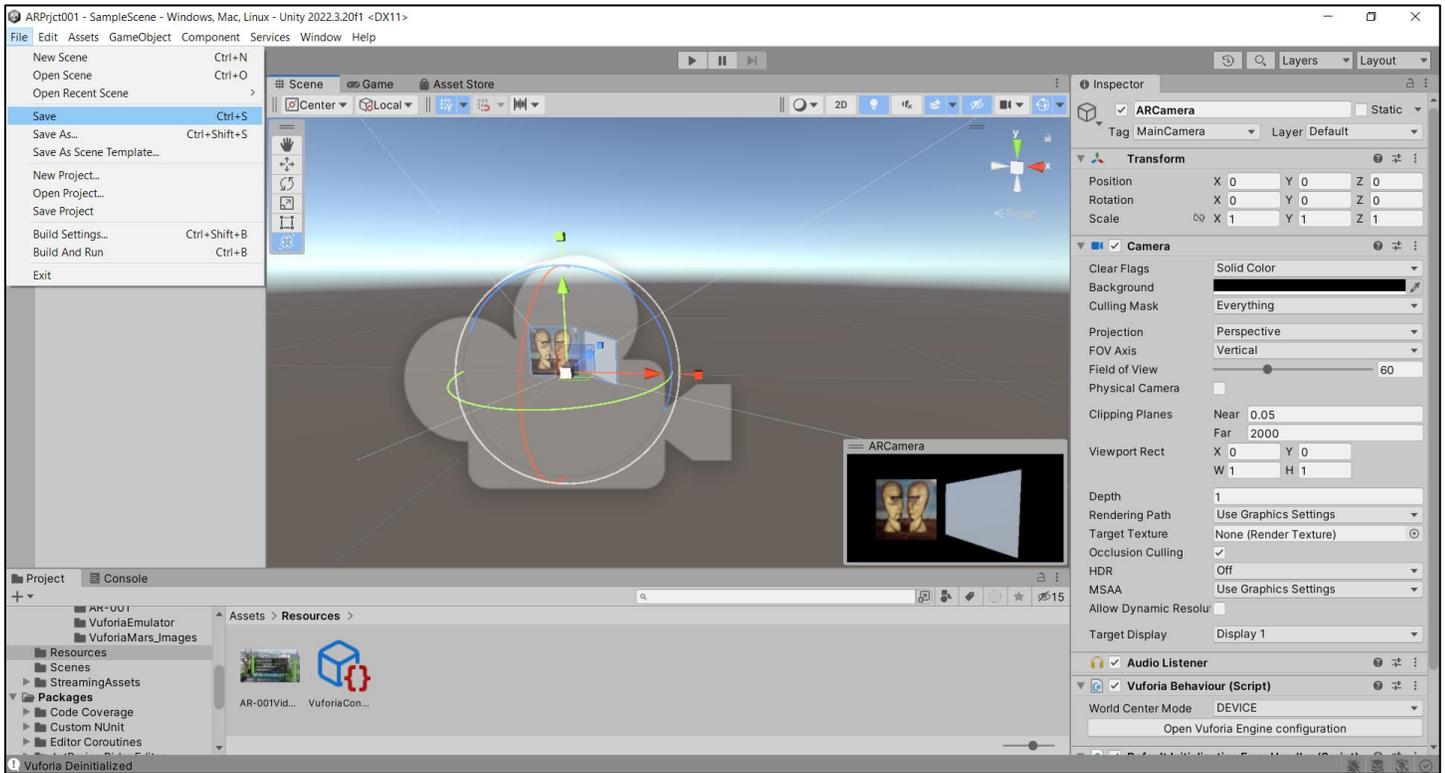
В случае обнаружения некорректно сформированной сцены ДР-приложения необходимо еще на стадии работы в **Unity Editor** добиться приемлемого результата → убедимся в том, что **Quad** развернут правильно, его размеры соответствуют размерам таргета, имеют верное соотношение сторон и т.д. Рекомендуется располагать камеру в точке начала координат (0,0,0), с углом поворота по трем осям – 0, и масштабным коэффициентом = 1 по всем трем осям →



В результате проверка в режиме **Game** → **Play** дает удовлетворительный результат.



Разработка сцены закончена. Здесь можно сохранить полученный результат – сцену - не выходя из **Unity 3D**. Это может пригодиться в дальнейшем при доработке сцены, при прерывании сеанса работы в **Unity 3D** и т.д. Для сохранения сцены выполнить: **File**→**Save** или **File**→**Save as**. Сохранение производится в локальной ФС на вашей локальной машине.



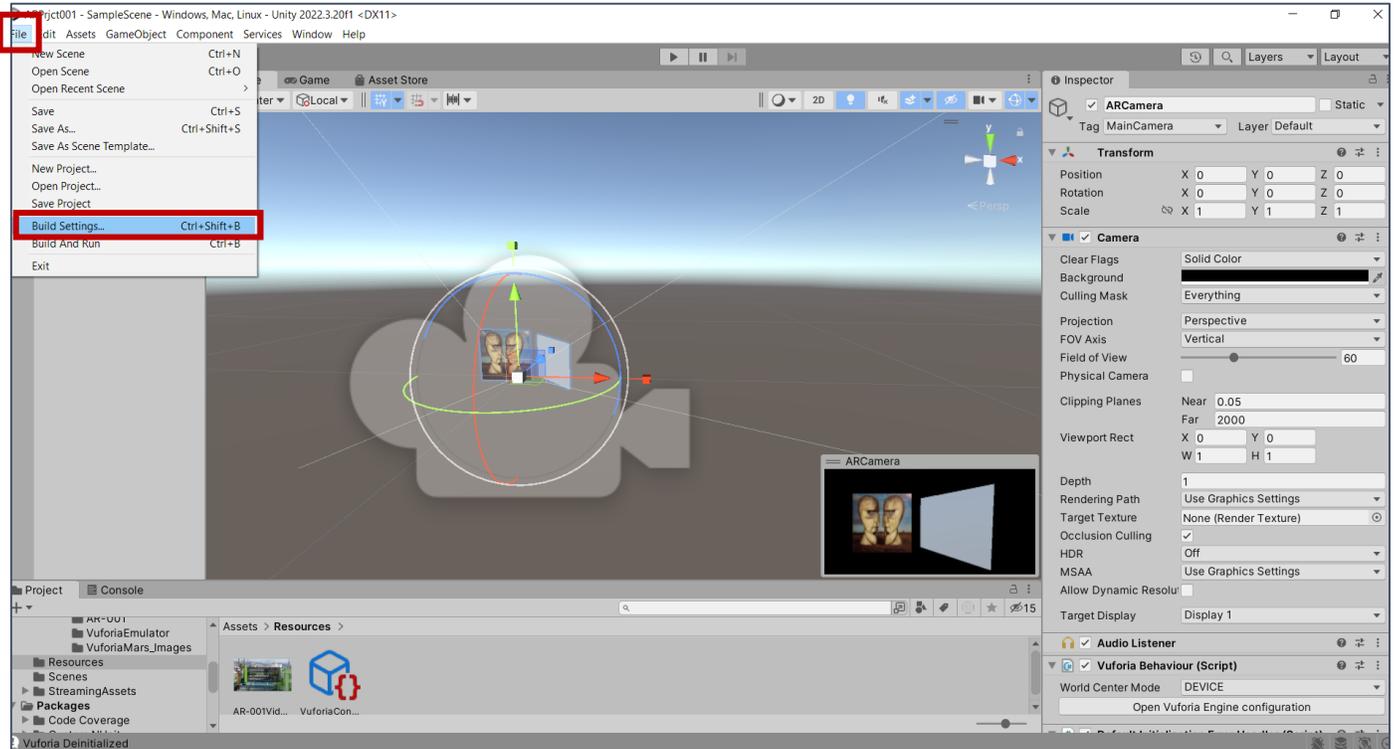
5. Создание файла .ark для загрузки Приложения ДР на Android-MY.

Разрабатываемое приложение должно будет использовать камеру конкретного **Android**-мобильного устройства. **Unity 3D** в нашем проекте только подготавливает универсальный драйвер **AR Camer**'ы для типового **Android**-устройства. Все особенности конкретных, передовых **Android**-устройств (стерео, 4K, и т.д.) требуют дополнительного

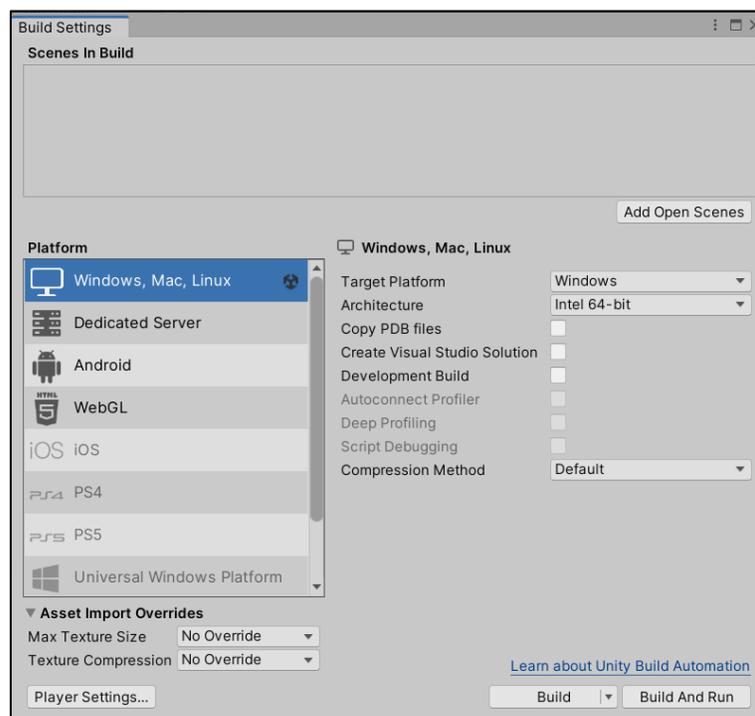
программирования, что находится за пределами данной ЛР. Формат файла загружаемого Приложения ДР для **Android**-устройств – это **.apk**.

5.1. Для создания файла .apk включаем в редакторе Unity 3D режим Build:

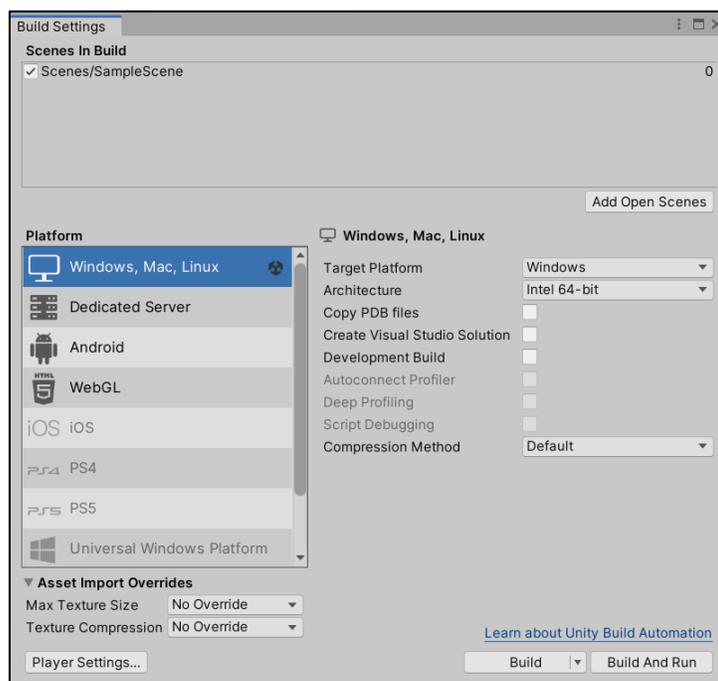
File→Build Settings:



Все предварительные настройки – выбор сцены, настройки ОС устройства – осуществляются в открывшемся окне **Build Settings**.



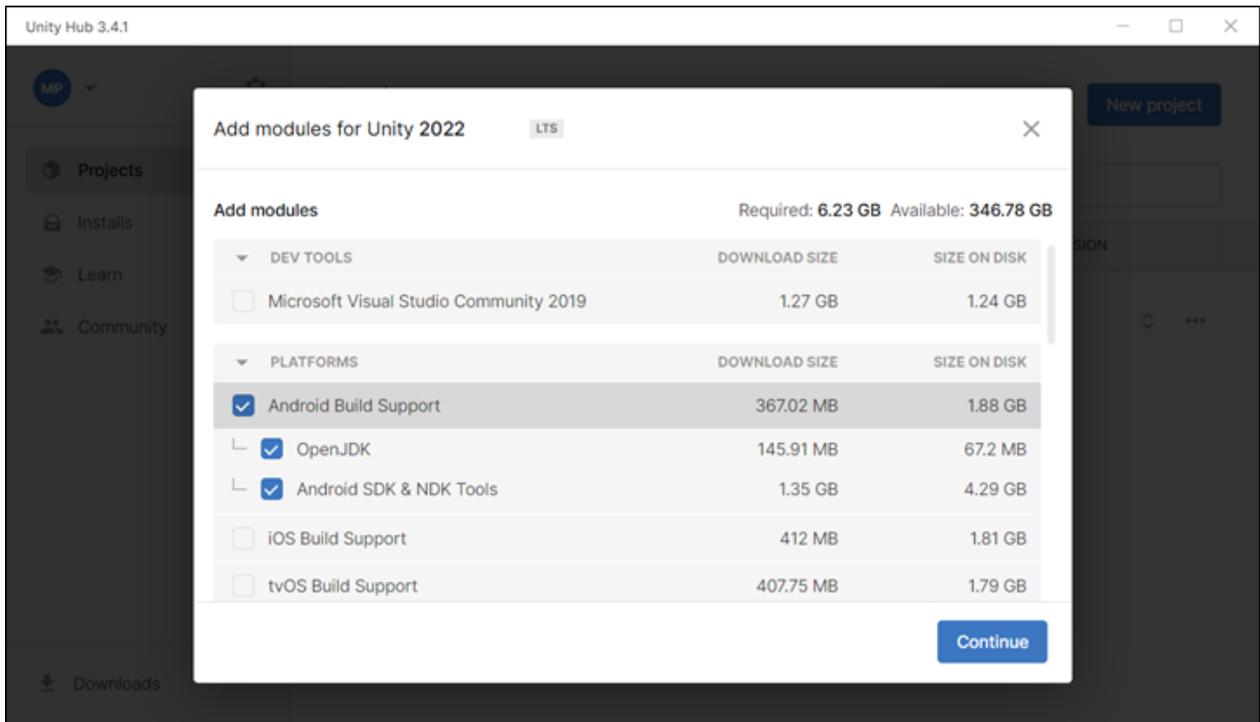
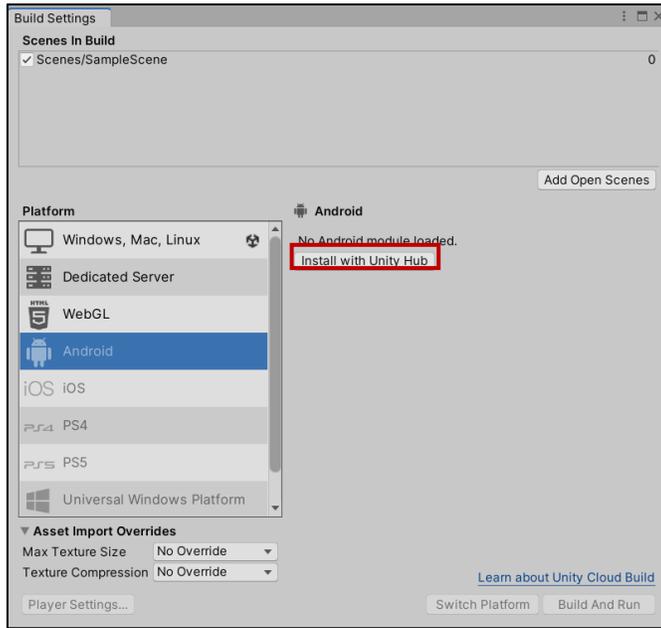
Для выбора сцены нужно использовать кнопку **Add Open Scene**, пометив в случае необходимости нужную из списка:

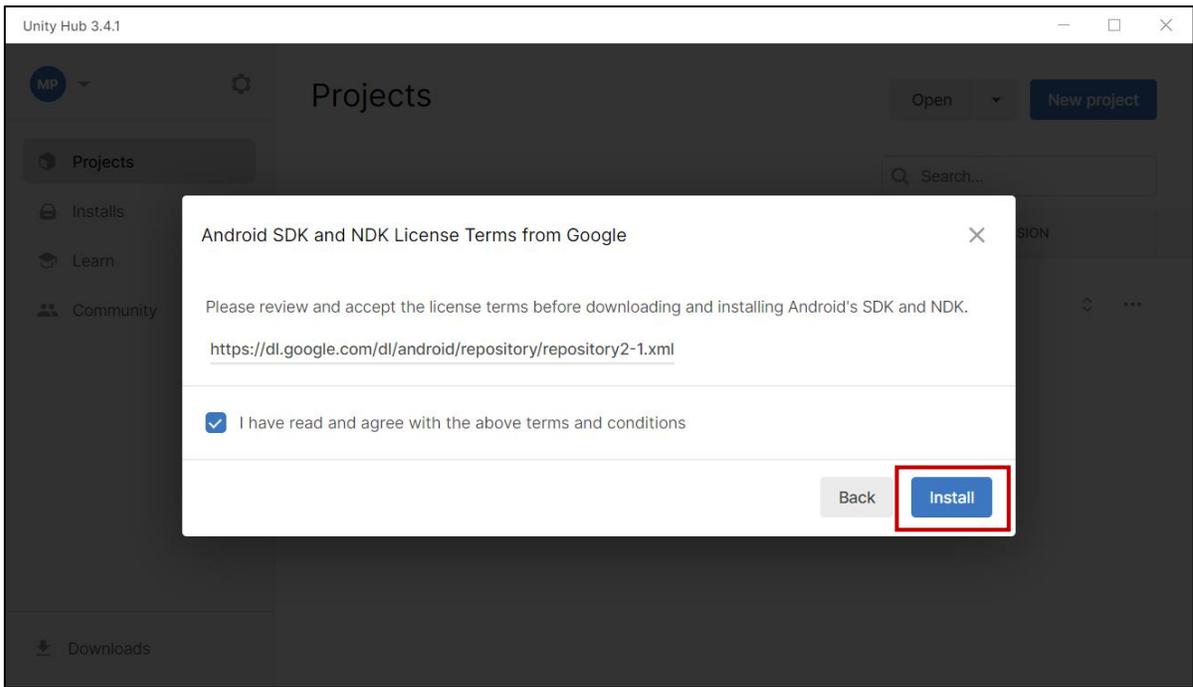


Пока в списке **Platform** не выбрана (не отмечена справа значком ) или может даже отсутствовать нужная нам платформа – **Android**, под управлением которой должно работать устройство ДР – Мобильное Устройство.

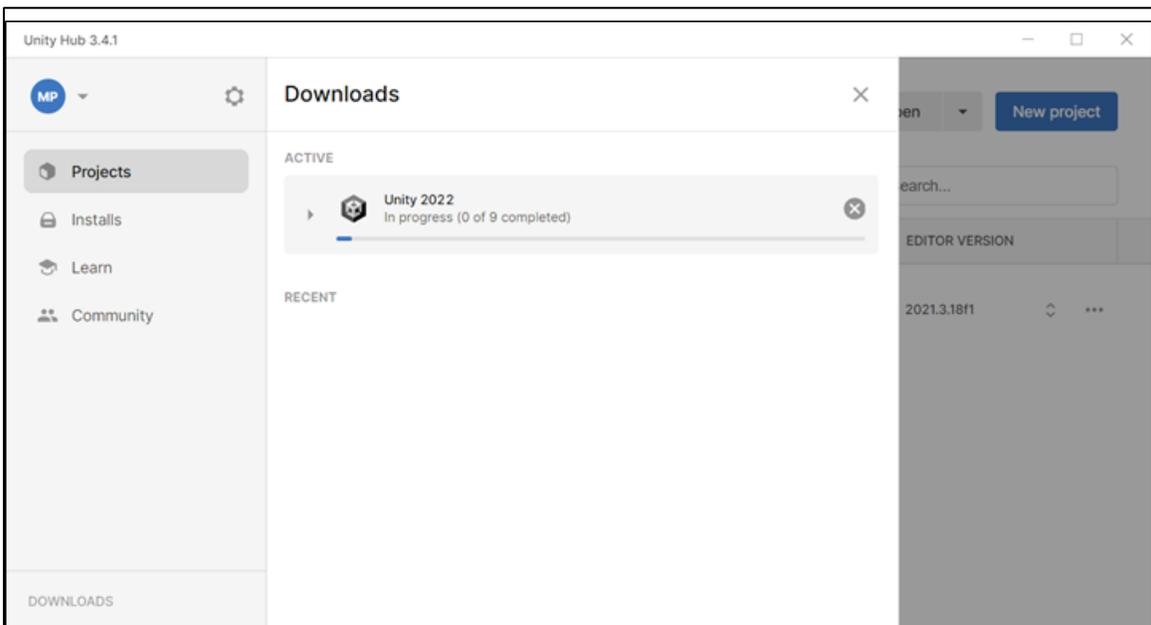
Если платформа – Android присутствует в списке, но не выбрана – переходим к пункту 5.2.

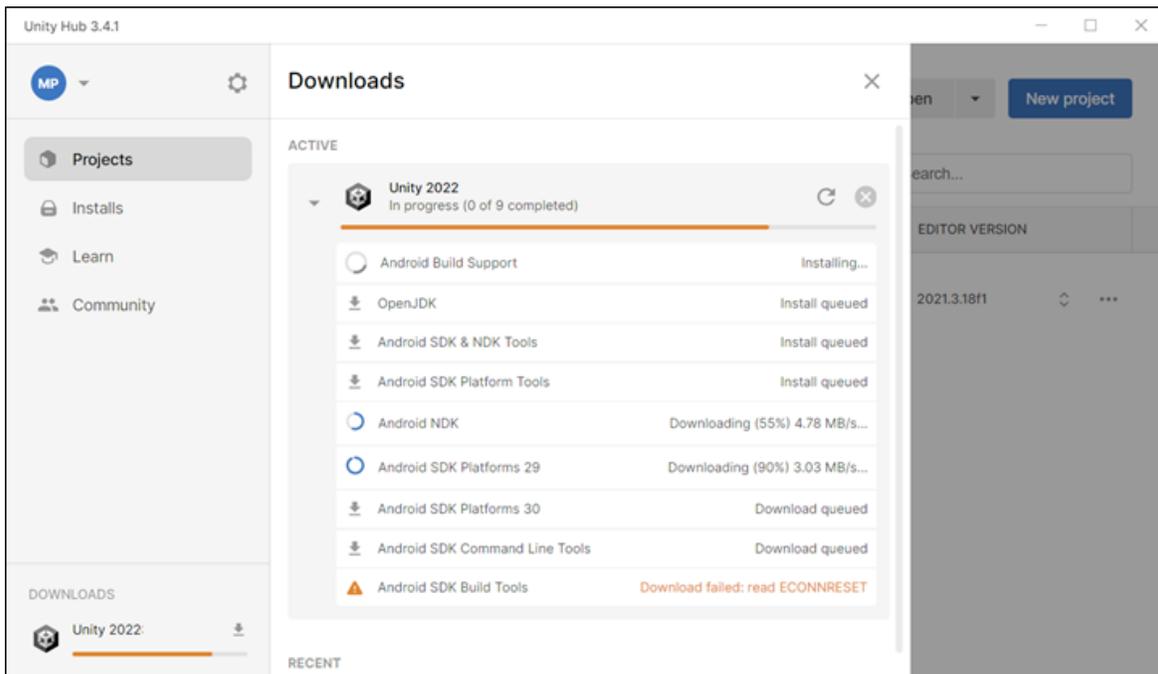
Если платформа **Android** отсутствует в списке доступных («погашена» в списке), то для загрузки необходимого для **Builder'a** ПО для **Android** выбираем в списке **Platform** «погашенную» позицию **Android** и осуществляем инсталляцию необходимого ПО указанным способом – через уже проинсталлированный на локальном компьютере **Unity Hub**:



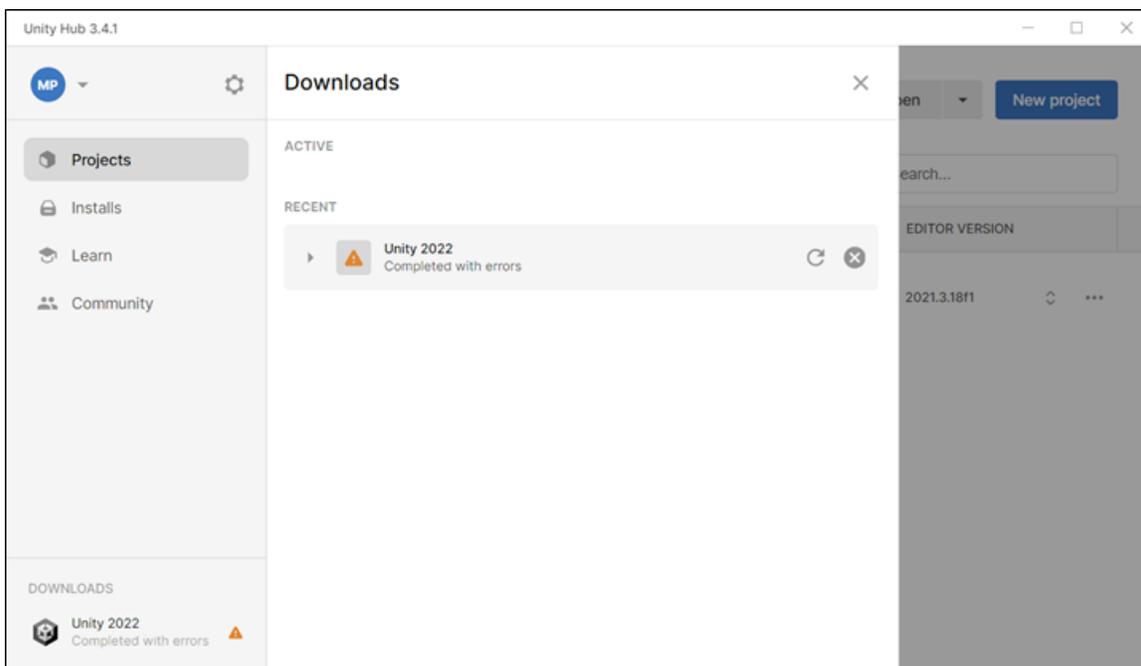


Начинаем инсталляцию программных компонент для поддержки **Android** →

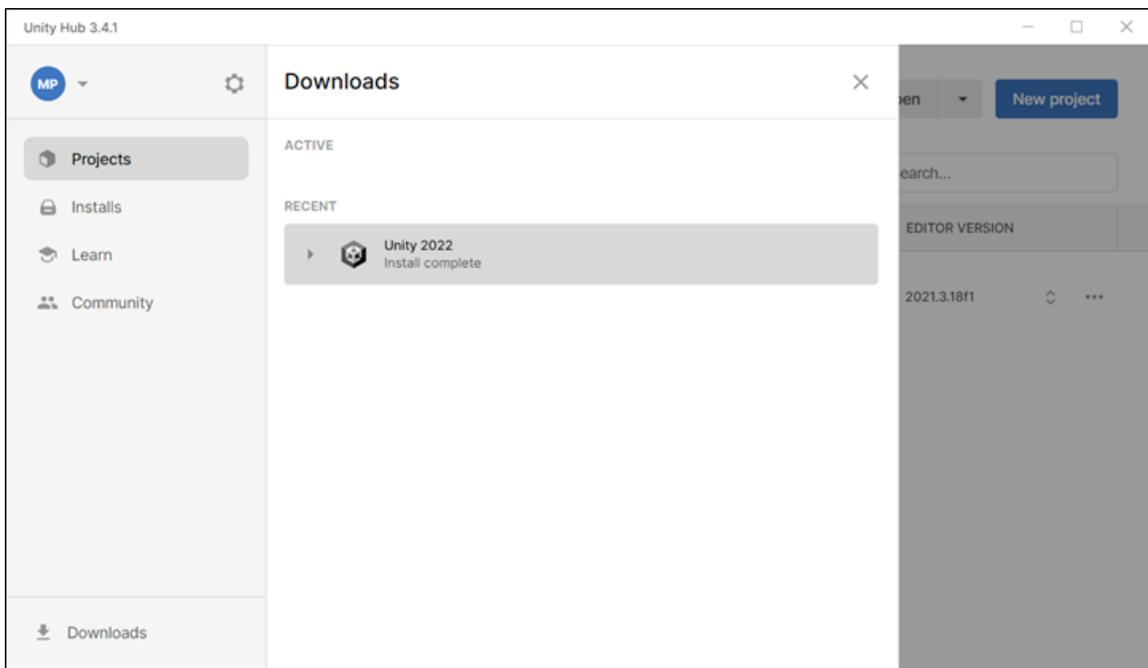
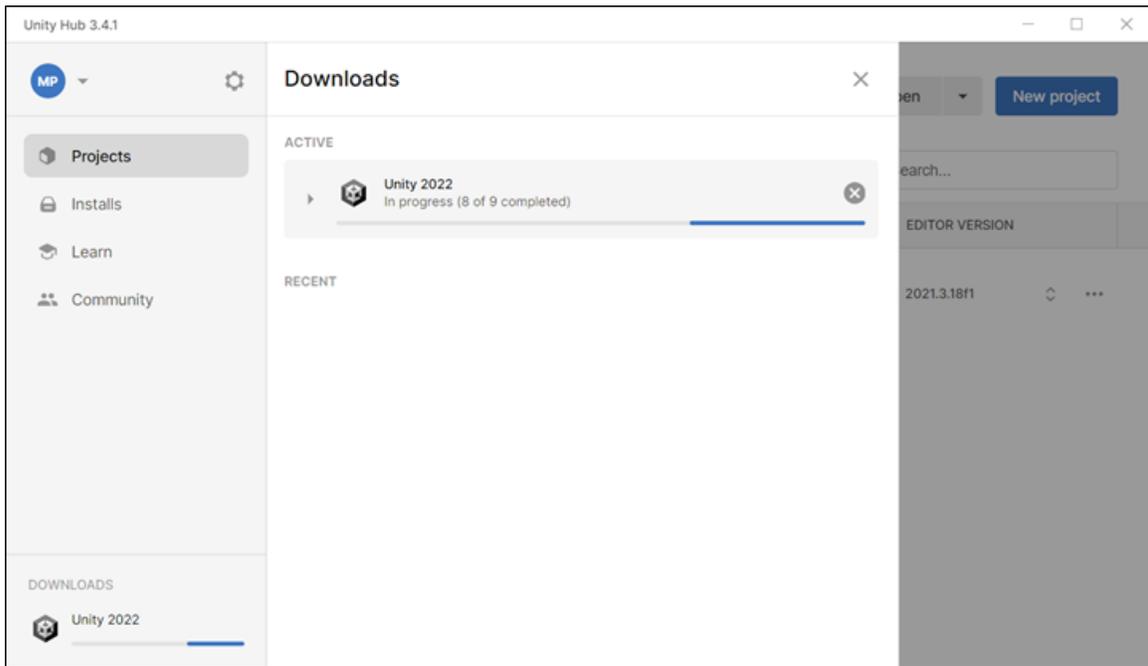




Процесс инсталляции занимает определенное время. Может случиться, что загрузка всех модулей с первого раза не получается:



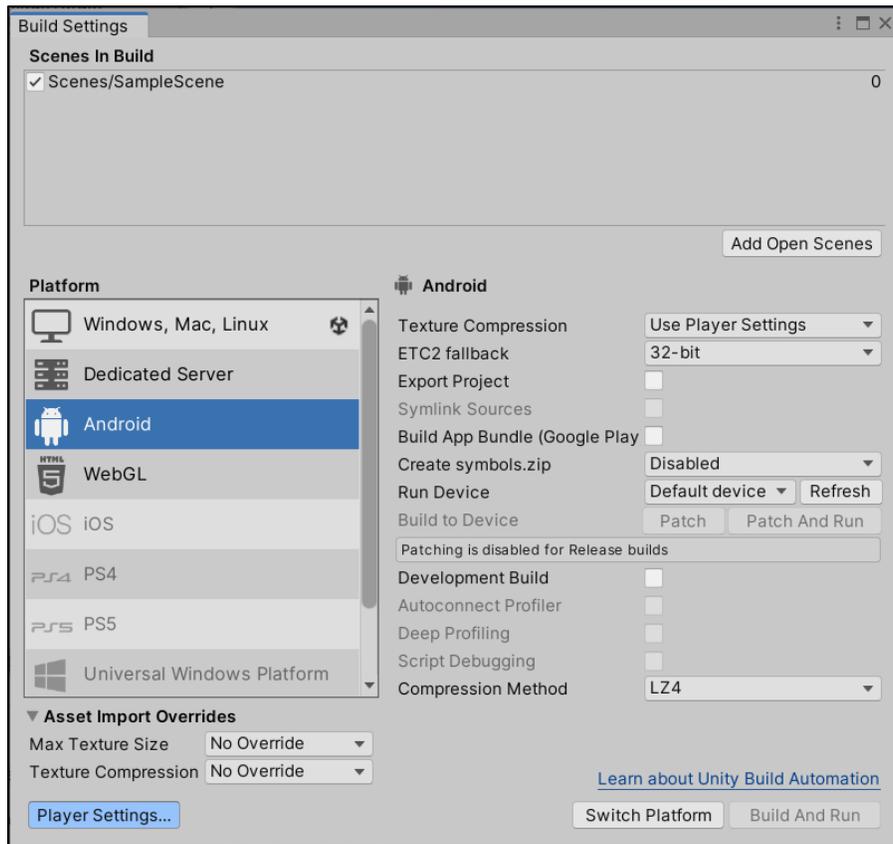
В этом случае необходимо кликнуть на рестарт и надеяться на успех:



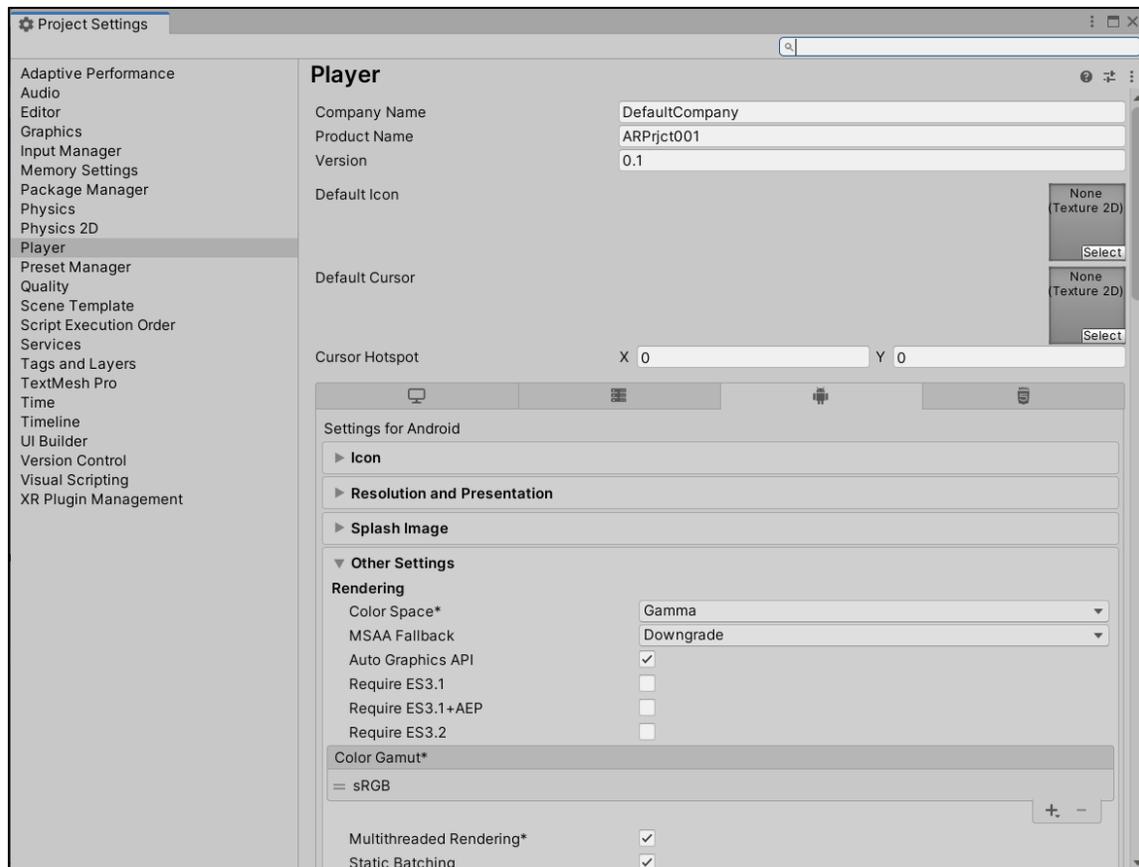
Инсталляция необходимого для **Builder'a** ПО для **Android** – **Install complete** – успешно завершена. позиция **Android** в окне **Build Settings** «отмерзает» и можно переходить к настройке работы оборудования в данной ОС → клавиша **Player Settings**.

5.2. Настройка **Builder'a** для платформы **Android**.

Позиция **Android** в окне **Build Settings** «отмерзает» и можно переходить к настройке работы оборудования в данной ОС → клавиша **Player Settings**.

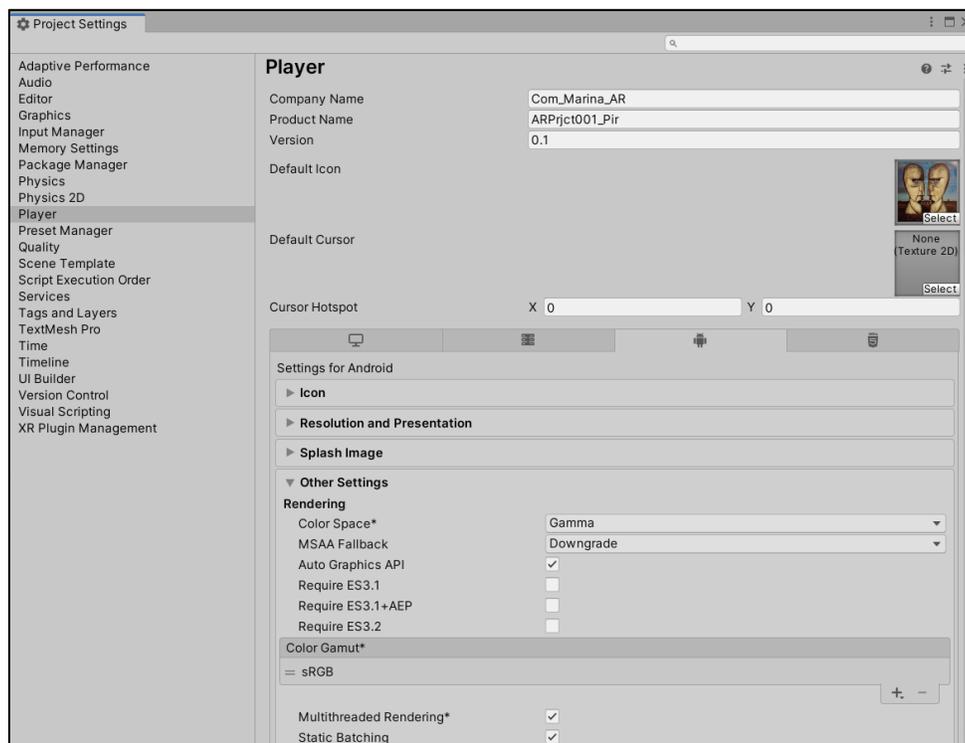


В результате в открывшемся окне **Project Settings** появляются настройки для плеера под **Android**:



Выполняем необходимые настройки. Заполните позицию **Company Name** (пофантазируйте), убедитесь в том, что остальные позиции заполняются по умолчанию (**рекомендуется изменить имя Product Name для удобства проверки ваших Проектов преподавателем, добавить фамилию латинскими буквами**), добавьте иконку для вашего Проекта → Режим **Select** в позиции **Default Icon**. Если поле для пиктограммы останется незаполненным, то для вашего Приложения ДР пиктограммой всегда будет логотип **Unity**.

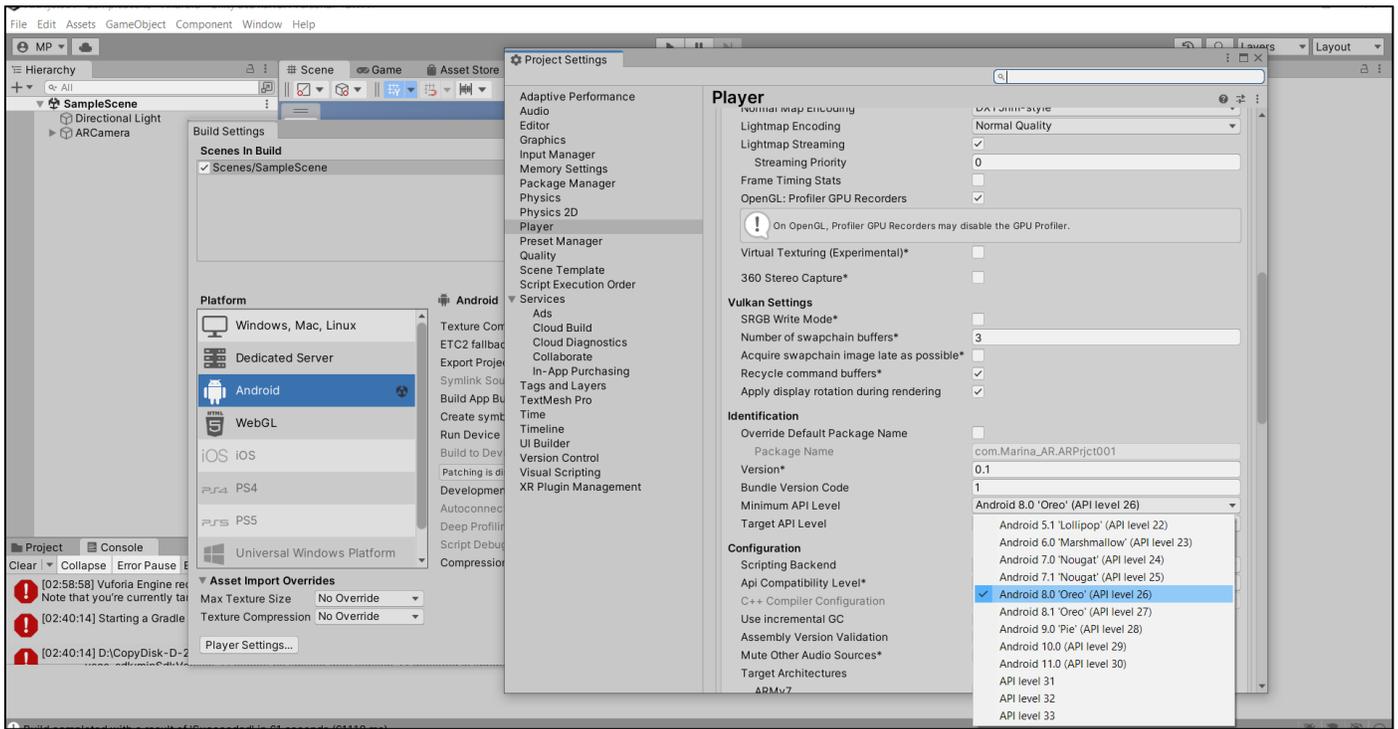
И **Company Name** и **Product Name** выбираются разработчиком произвольно, с учетом следующих ограничений: можно использовать только буквы латинского алфавита, цифры и знак «_». Выбранные **Company Name** и **Product Name** размещаются в соответствующих полях в верхней части **Player Settings**.



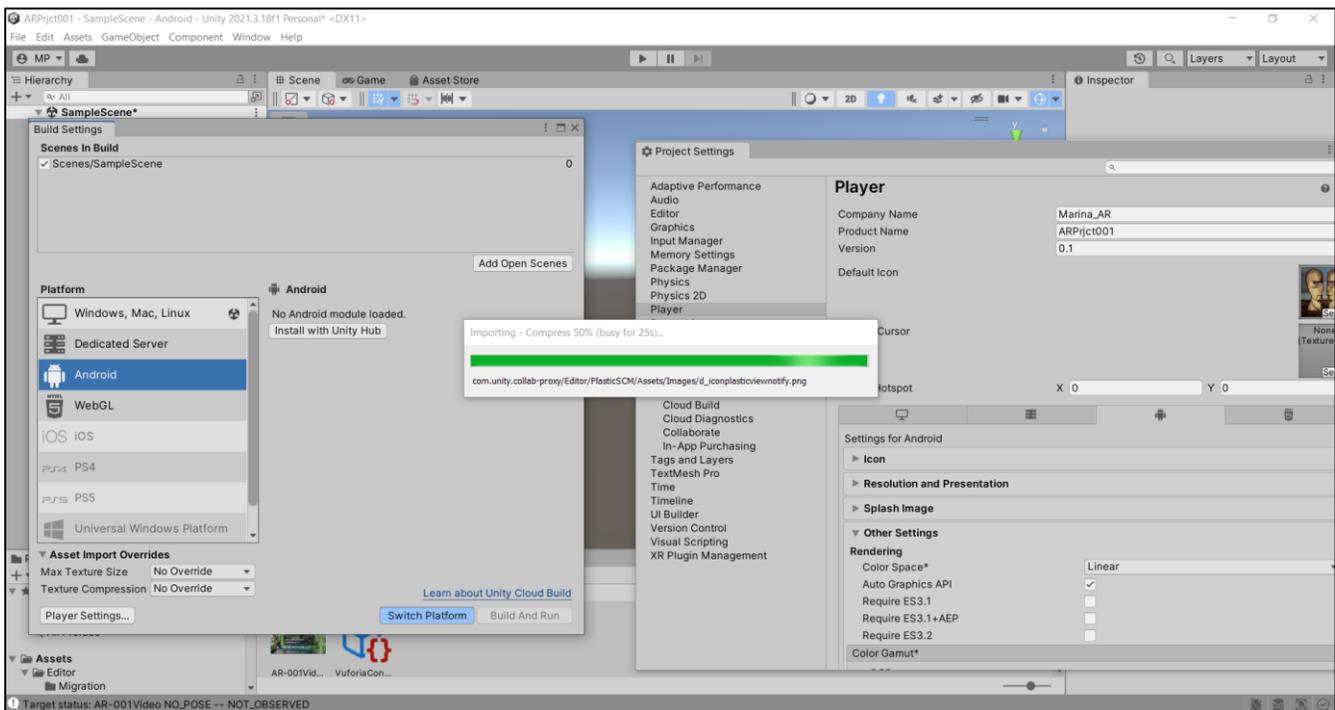
Познакомьтесь с остальными позициями **Project Settings**. Изменять заполненные позиции по умолчанию не следует.

ВАЖНО!!! Обратите внимание, что для текущей версии **Unity** минимальная версия **Android – 8**. Это нужно **ОБЯЗАТЕЛЬНО** прописать в

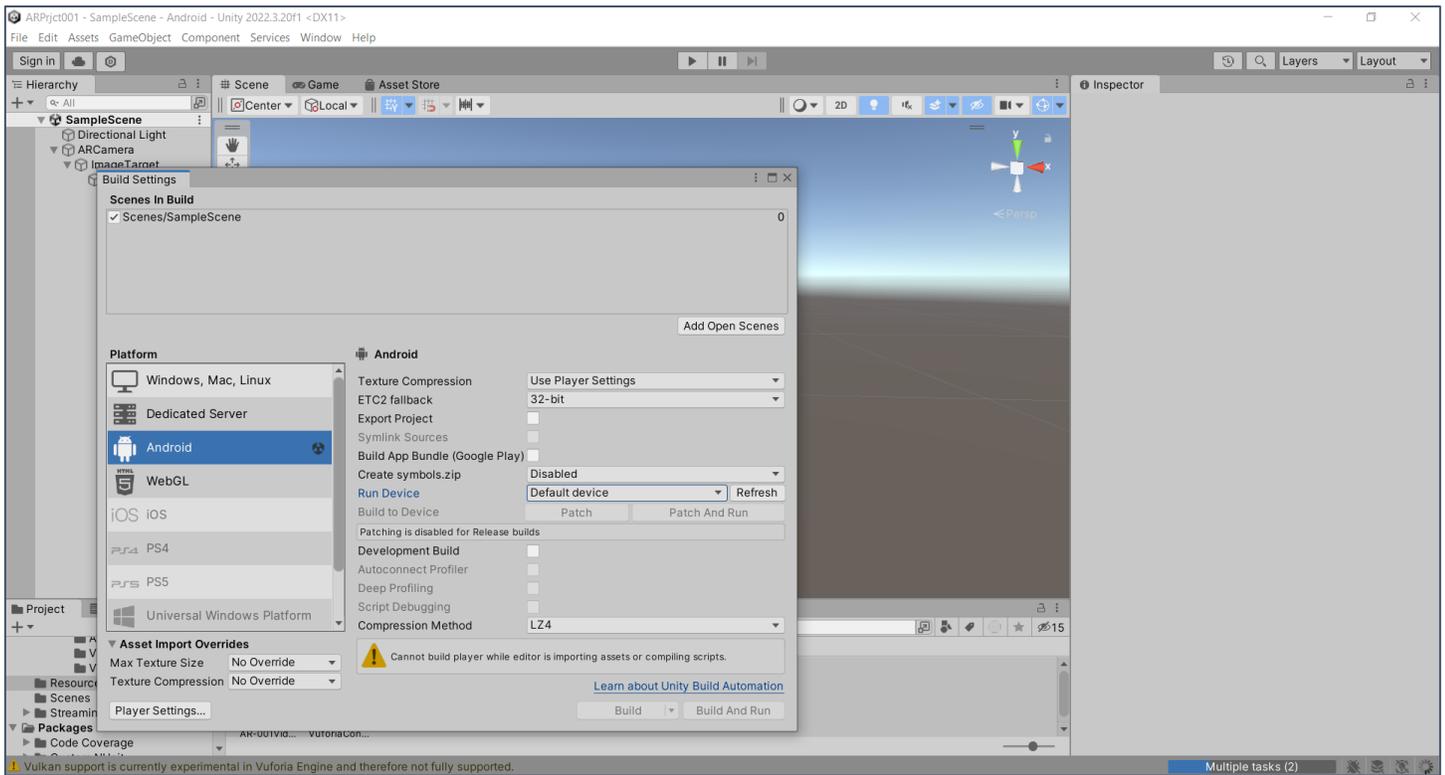
Project Settings → Player → Identification → Minimum API Level



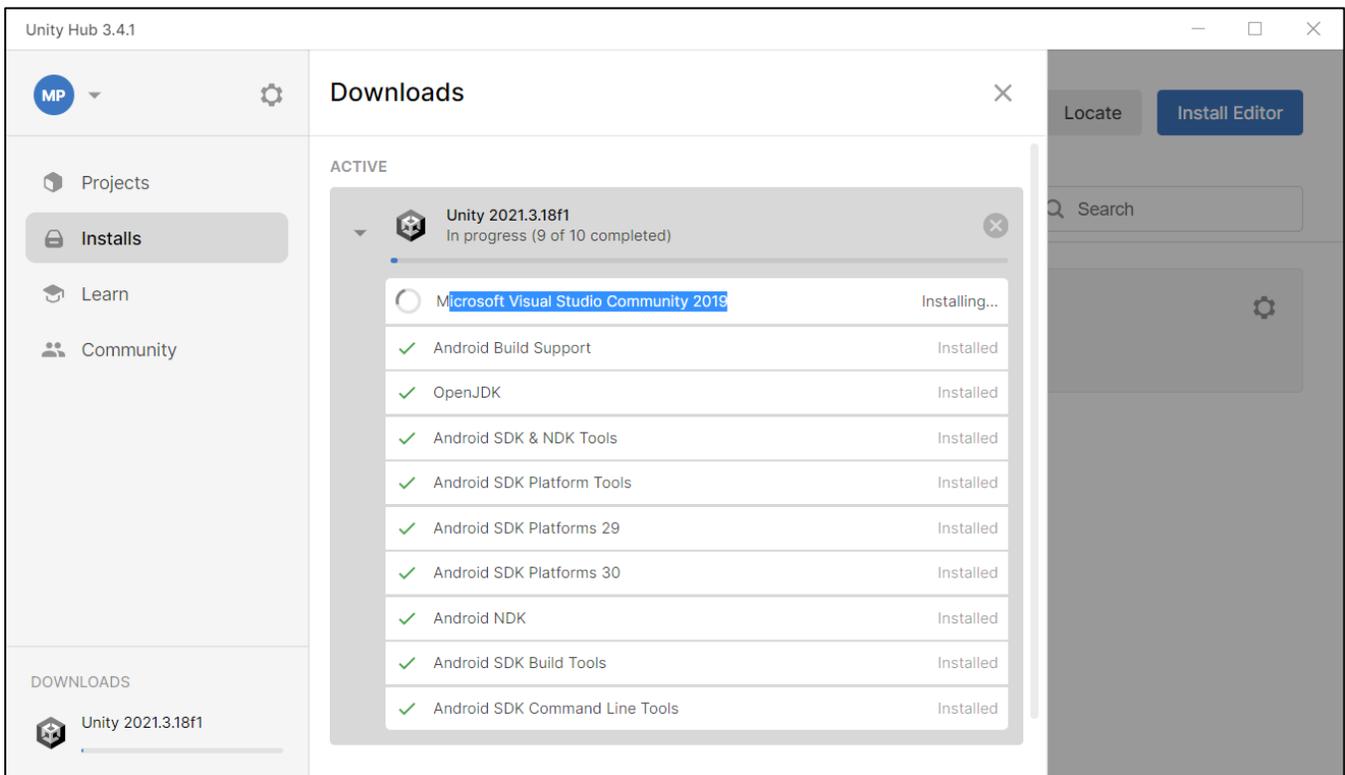
После выполнения всех этих настроек можно выполнить операцию **Build**:



На этом этапе, после выбора опции **Switch Platform**, может появиться сообщение о недоустановке необходимых программных компонент.



Необходимо устранить появившиеся ошибки. Вновь вернуться на позицию **Install with Unity Hub** и завершить установки. В нашем случае недоустановленным оказался **MS Visual Studio Community**.



Далее следует стандартная последовательность шагов для инсталляции **MS Visual Studio Community** →

Visual Studio Installer ×

Before you get started, we need to set up a few things so that you can configure your installation.

To learn more about privacy, see the [Microsoft Privacy Statement](#).
By continuing, you agree to the [Microsoft Software License Terms](#).

Visual Studio Installer

Getting the Visual Studio Installer ready.

Verifying

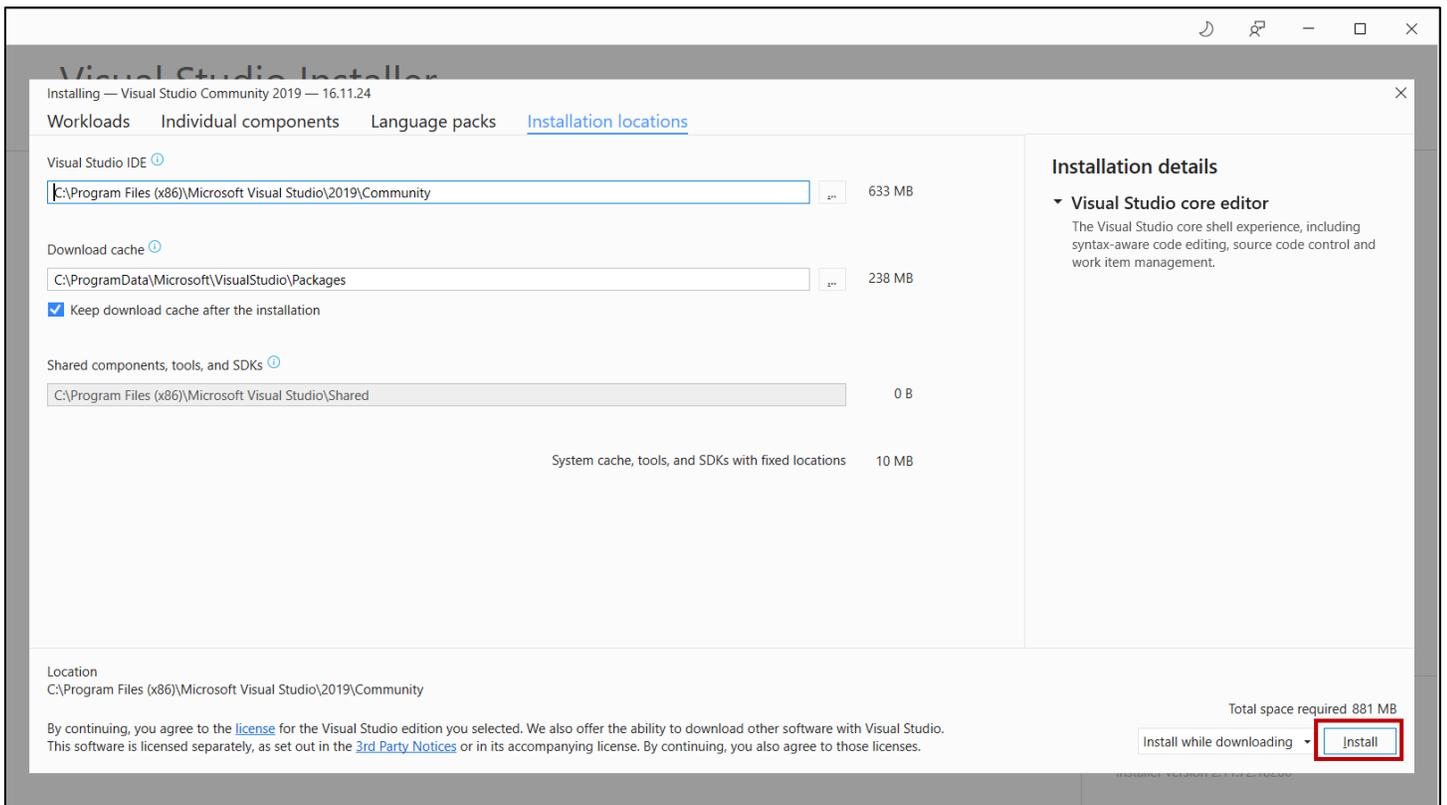
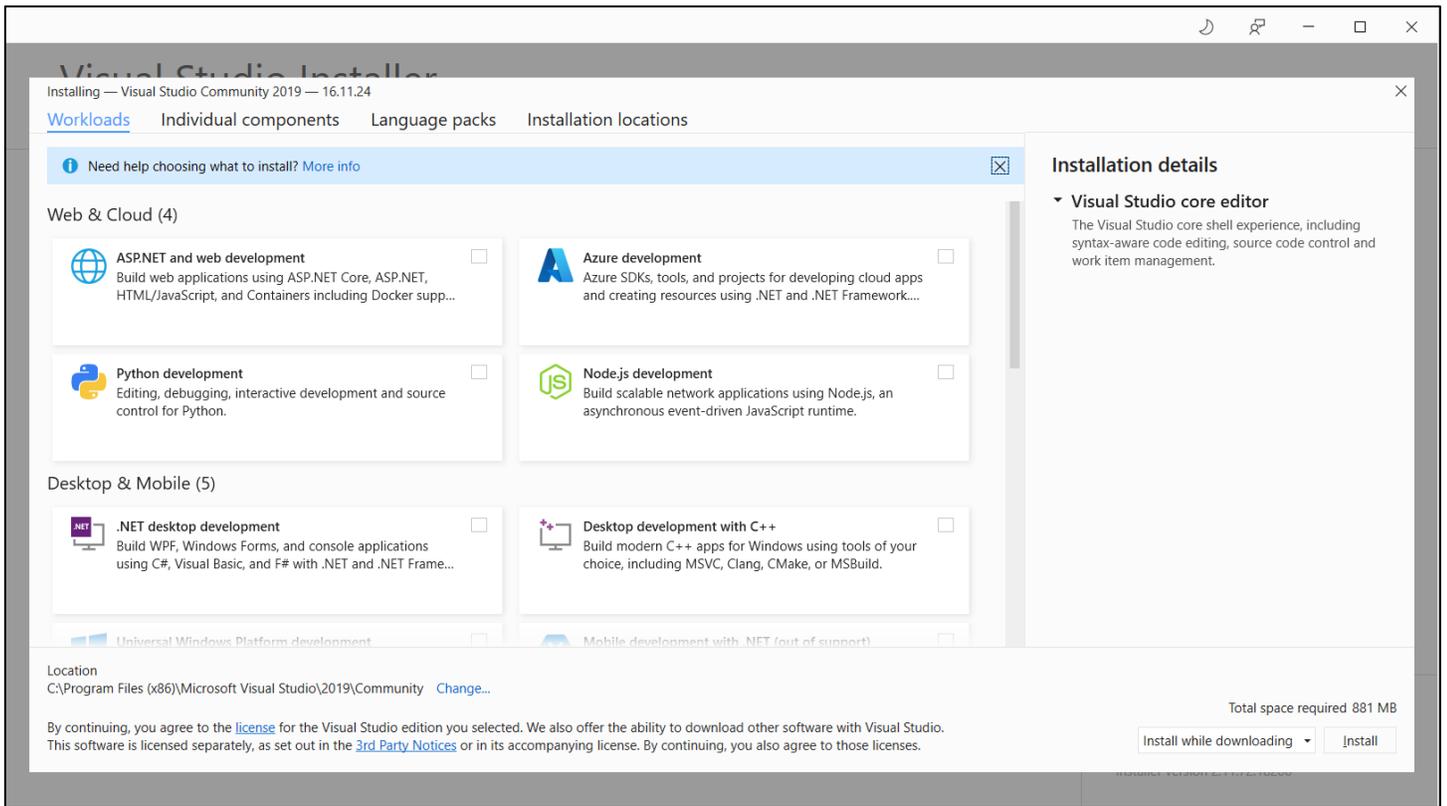

Installing


Visual Studio Installer

 Getting the Visual Studio Installer ready.

Downloaded
 

Installed
 



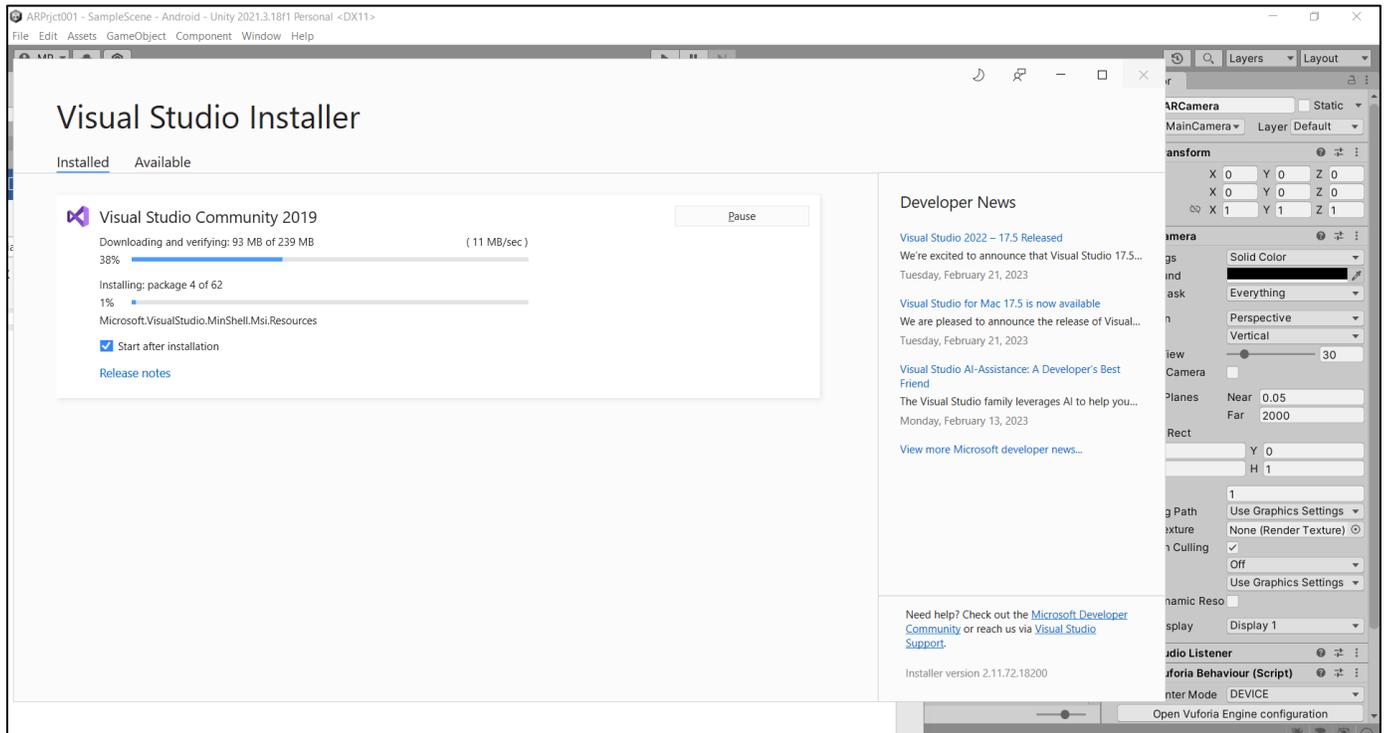
Do you want to continue without workloads?

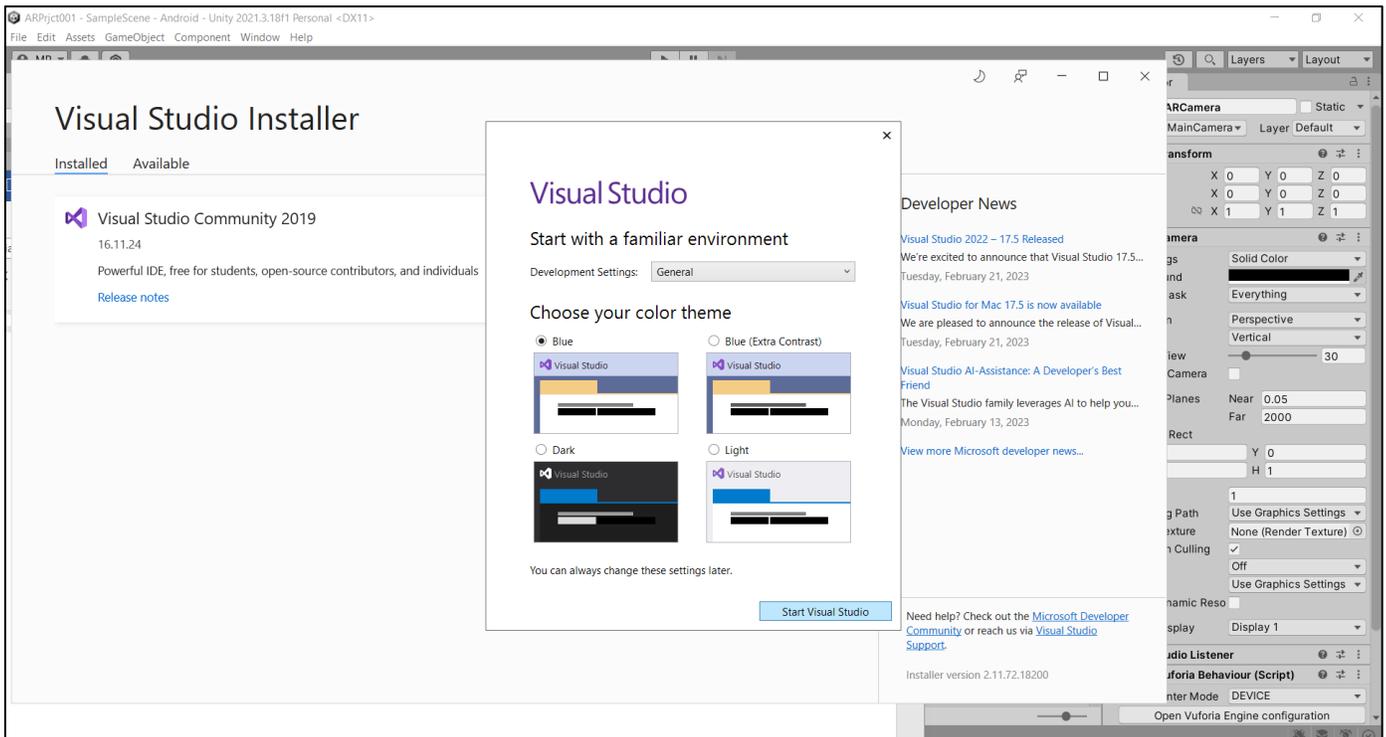
The core Visual Studio editor can open folders, edit files, and manage work items. Visual Studio needs to install additional components to create, open, and debug projects for desktop, mobile, and cloud apps.

[Learn more about workloads and components](#)

Continue

Add Workloads

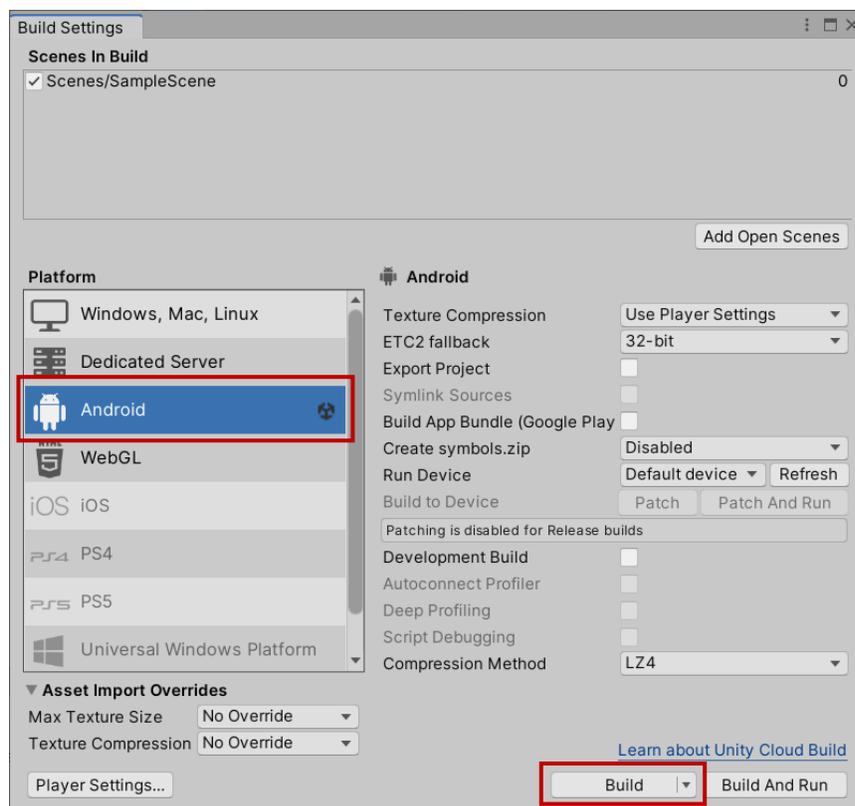




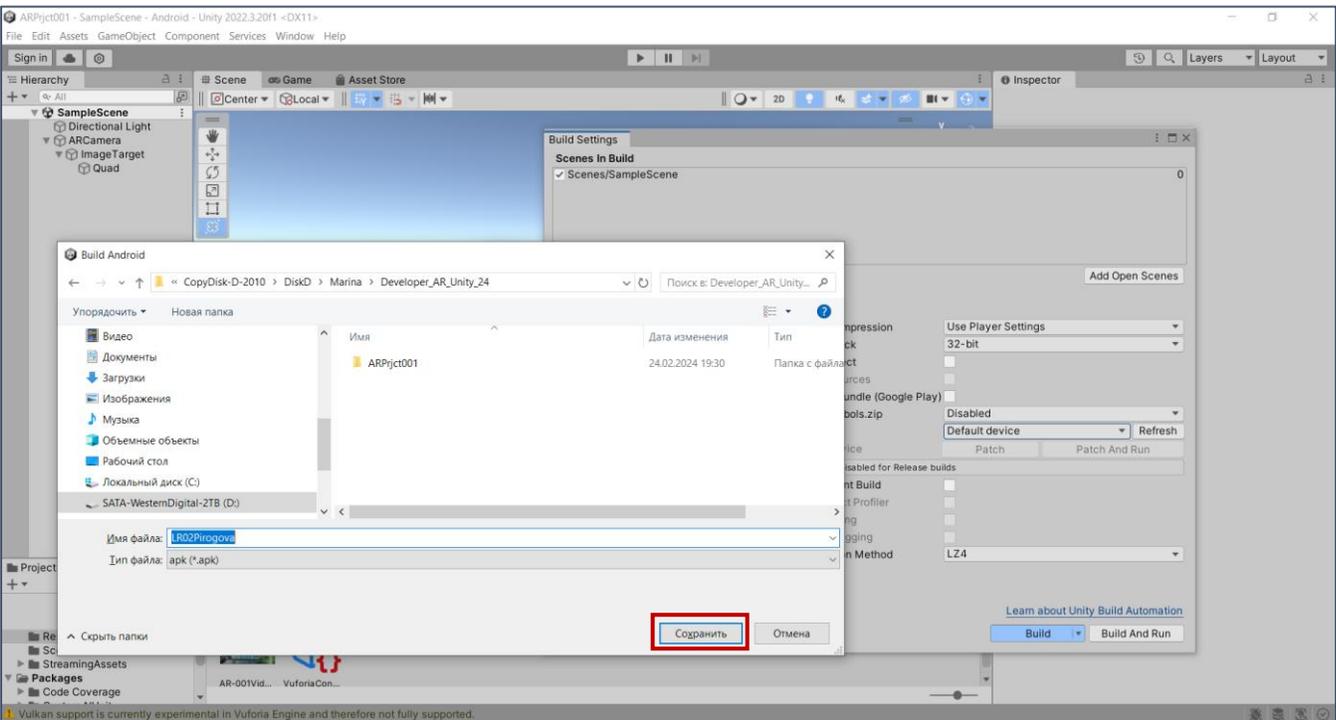
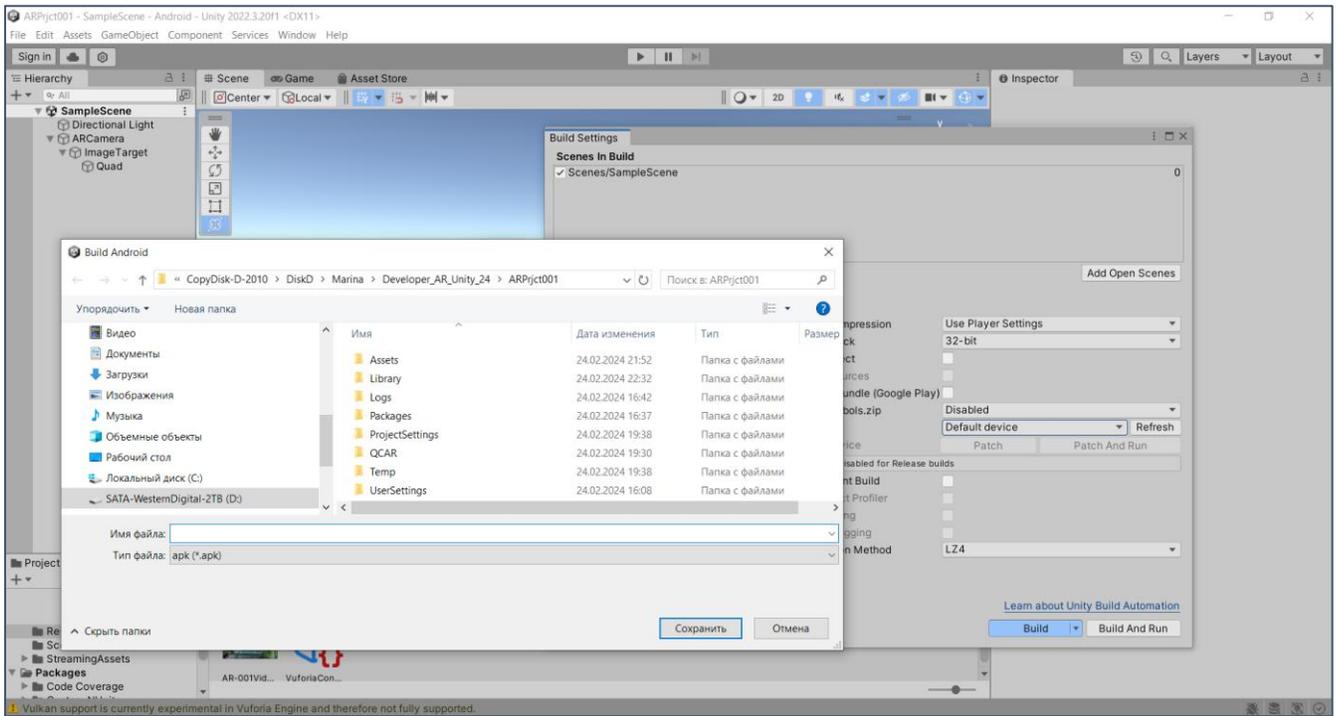
После всех этих установок рекомендуется выйти из Проекта **Unity** и войти в него заново.

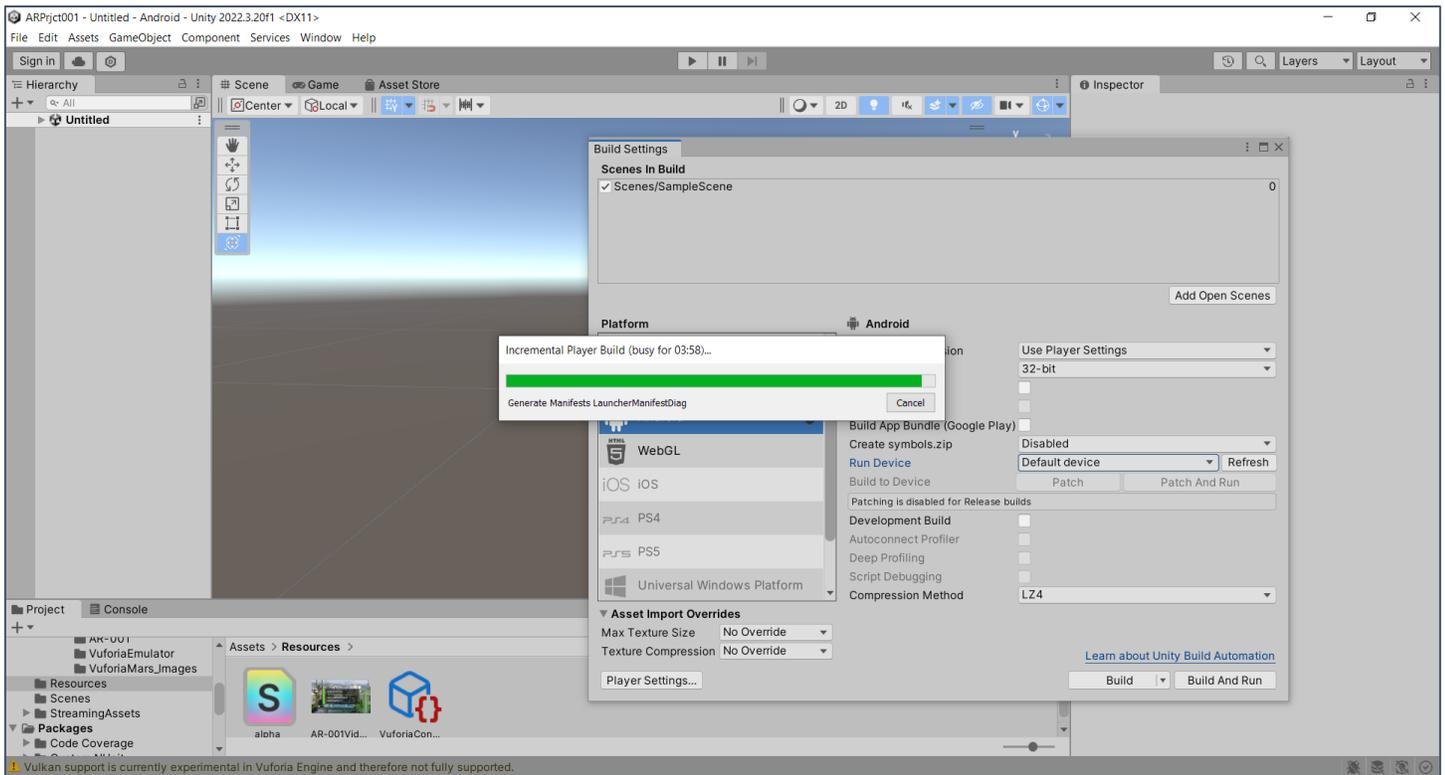
5.3. Выполнение операции **Build**.

В списке **Platform** доступна и выбрана платформа **Android**, доступна клавиша **Build**:



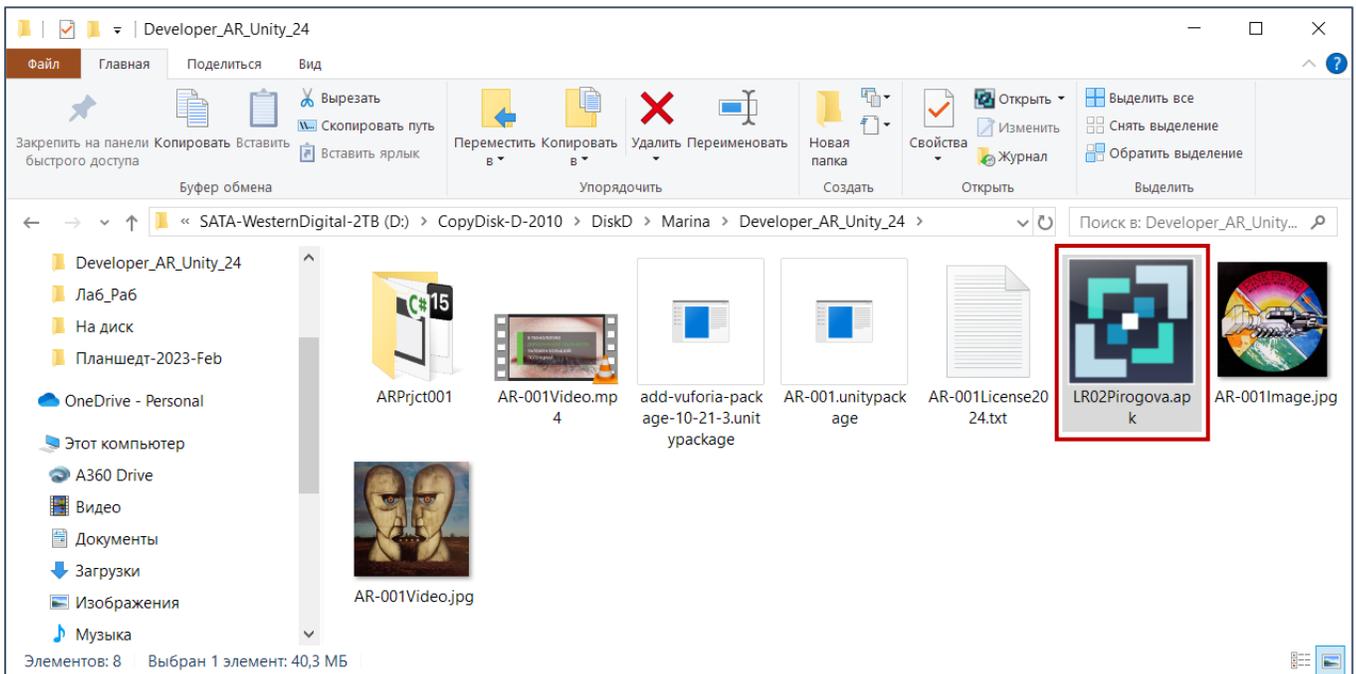
Нажимаем клавишу **Build** → выбираем рабочую директорию в локальной файловой системе разработчика для сохранения собранного файла **.apk**:





Процесс сборки файла **.apk** может длиться несколько минут.

Файл **LR02Pirogova.apk** разработанного в данном описании Приложения ДР сохранен.



Теперь его осталось загрузить (и далее – установить) в **Android - МУ** любым известным вам способом.

Отчет по ЛРН№2 должен

- быть оформлен в соответствии с правилами, принятыми в МЭИ,
- содержать цель ЛР,
- краткое описание основных этапов разработки с необходимым количеством сканов экрана,
- описание проблем, если они возникали,
- результаты тестирования в режиме Play в редакторе Unity и сканы экрана МУ с результатами работающего Приложения.

Создание коротких видеороликов и процесса тестирования и работающего Приложения ДР приветствуются.

Разработанное в рамках ЛР № 2 AR-Приложение необходимо продемонстрировать преподавателю. Для этого загрузите свой .apk на любой файлообменник или доступное облако и пришлите мне ссылку!

Не забудьте про таргет – его тоже необходимо прислать мне отдельным файлом!

Для справки: ниже – вот так выглядят в консоли Unity сообщения о возможных, возникавших в процессе работы по созданию Приложения ДР, ошибках...

