

**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:**

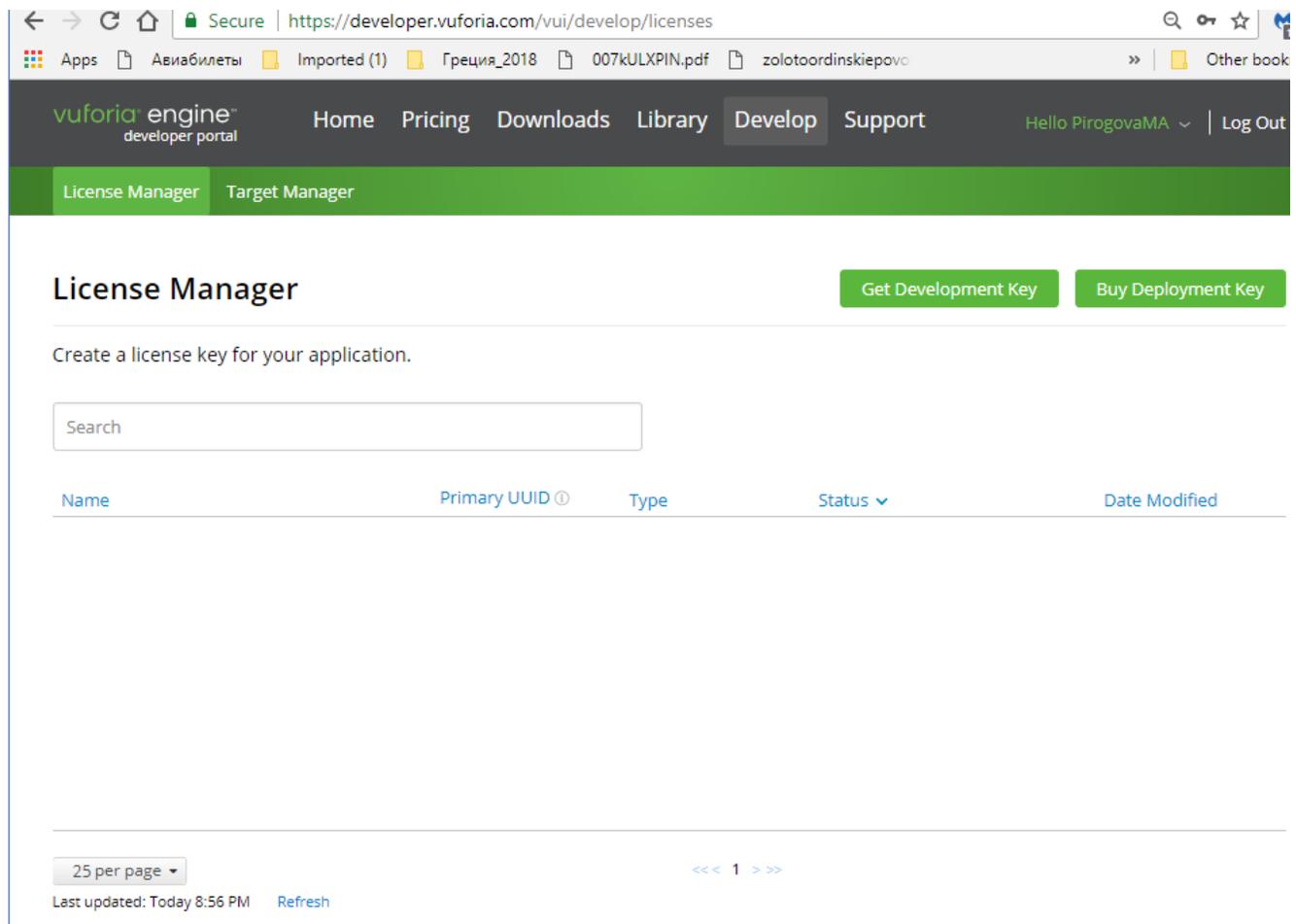
- Интернет-соединение локального компьютера;
- Наличие аккаунта пользователя **Vuforia** (результат успешного выполнения ЛР №1);
- Установленная на компьютере разработчика система **Unity 3D** (результат успешного выполнения ЛР №1);
- Заранее подготовленные изображения для метки (таргета) и контент (видеоклип);
- Кроме того, для формирования собственно приложения ДР на машине разработчика (**MSWindows 10/8.1/7**) необходимо установить **Android SDK** – набор инструментальных средств для сборки **Android-Приложений** в чужой операционной системе. Для корректной установки **Android SDK** надо предварительно установить (если это еще не сделано) **Java Development Kit – JDK**, предпочтительно последней версии; адреса для скачивания:

Android SDK – <https://developer.android.com/about/versions/10/setup-sdk>

JDK - <https://www.java.com/ru/download/manual.jsp>

\*Дополнительные пояснения по особенностям установки **Android SDK** под **Windows** – см. **Приложение 1:** [http://a0601.narod.ru/ADD01\\_LR\\_02.pdf](http://a0601.narod.ru/ADD01_LR_02.pdf)

На сайте <https://developer.vuforia.com/> осуществляем вход 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**).



The screenshot shows the Vuforia Developer Portal interface. At the top, there is a navigation bar with links for Home, Pricing, Downloads, Library, Develop, and Support. The user is logged in as 'Hello PirogovaMA' and can click 'Log Out'. Below the navigation bar, there are two tabs: 'License Manager' (selected) and 'Target Manager'. The main content area is titled 'License Manager' and contains two buttons: 'Get Development Key' and 'Buy Deployment Key'. Below these buttons, there is a text prompt: 'Create a license key for your application.' followed by a search input field. A table with the following columns is visible: Name, Primary UUID (with an info icon), Type, Status (with a dropdown arrow), and Date Modified. At the bottom of the page, there is a pagination control showing '25 per page' and a page indicator '<< 1 >>'. A footer note states 'Last updated: Today 8:56 PM' with a 'Refresh' link.

В открывшемся окне доступны две вкладки для последовательного выполнения этих процедур – **License Manager** и **Target Manager**. Если ни одного проекта еще не создано, то в открывшемся окне пусто.

1. Переходим в **License Manager**. Выбираем **Get Development Key**, в результате переходим в режим создания лицензионного ключа:

[Back To License Manager](#)

## Add a free Development License Key

You can change this later

### License Key

Develop

Price: No Charge

Reco Usage: 1,000 per month

Cloud Targets: 1,000

VuMark Templates: 1 Active

VuMarks: 100

 By checking this box, I acknowledge that this license key is subject to the terms and conditions of the [Vuforia Developer Agreement](#).

Cancel

Confirm

Обратите внимание на перечисленные характерные для студенческого режима использования **Vuforia** ограничения. Для нас единственно важным является: сколько актов распознавания метки (**Reco**) можно выполнить в течение месяца. 1000 для некоммерческого использования – более чем до статочно.

В открывшемся поле необходимо ввести уникальное имя. В рамках данной ЛР 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**.

[Back To License Manager](#)

## Add a free Development License Key

License Name \*

AR-001

You can change this later

### License Key

Develop

Price: No Charge

Reco Usage: 1,000 per month

Cloud Targets: 1,000

VuMark Templates: 1 Active

VuMarks: 100

By checking this box, I acknowledge that this license key is subject to the terms and conditions of the [Vuforia Developer Agreement](#).

Cancel

Confirm

В результате в списке лицензионного менеджера появится новая строка:

## License Manager

[Get Development Key](#)[Buy Deployment Key](#)

Create a license key for your application.

Name	Primary UUID ⓘ	Type	Status ▾	Date Modified
<a href="#">AR-001</a>	N/A	Develop	Active	Mar 14, 2020

25 per page ▾

Showing 1-1 of 1

&lt;&lt;&lt; 1 &gt;&gt;&gt;

Last updated: Today 9:09 PM [Refresh](#)

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

License Manager &gt; AR-001

**AR-001**[Edit Name](#) [Delete License Key](#)

License Key

Usage

Please copy the license key below into your app

```
Af4Rv8L/////AAABmSMOa3eEo0a6n4Q2GFaHxg4M8TaIfDax32AruVjjXt2Fkk4hAp8G8xrhfNnUQzDC3XKBLbzF/  
1P/UgmczJeM/odaFPhbX8/Mb015Irtc7cTQBdYguMfB68UoQBnKU08c4qciolqQXhBU4UfjLZA1c=OE0502onL0w+  
LgeCKDwLtsmXFJ3=49/u2NaG0jaEVI6dZ4IxsKj1T1tGzqmkBazODFmVmUDn40cdTe18UUxgM+N+NeEshvUxr4MI4  
H1iOUHCauB+soHP/#JqA6ge4HYiGF37oHB07ujR1pHnTOnAC8Kf02YYU12oRpeDDJ29Vh8Y3/NYPOHGPGXYY3BF+c  
kWdxDcN+zEjRpQG8fyQ1mkU2
```

Plan Type: Develop

Status: Active

Created: Mar 14, 2020 21:08

License UUID: baf03d11bc2c4578b6bdc3a511739e94

## Permissions:

- Advanced Camera
- External Camera
- Model Targets
- Watermark

## History:

License Created - Today 21:08

**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.

## 2. Переходим в менеджер меток – **Target Manager**:

vuforia engine<sup>™</sup>  
developer portal

Hello PirogovaMA | Log Out

Home Pricing Downloads Library **Develop** Support

License Manager **Target Manager**

## Target Manager

[Add Database](#)

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

Search

Database	Type	Targets	Date Modified
----------	------	---------	---------------

25 per page <<< 1 >>>

Last updated: Today 9:14 PM [Refresh](#)

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

### Create Database

Database Name \*

Type:

- Device
- Cloud
- VuMark

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

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.)**.

## Create Database

Database Name \*  
AR-001

Type:

- Device
- Cloud
- VuMark

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

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

vuforia engine  
developer portal

Hello PirogovaMA | Log Out

Home Pricing Downloads Library **Develop** Support

License Manager **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	Mar 14, 2020

25 per page Showing 1-1 of 1 <<< 1 >>>

Last updated: Today 9:20 PM [Refresh](#)

Для этого переходим по ссылке **AR-001** в поле **Database**:

Target Manager &gt; AR-001

## AR-001 [Edit Name](#)

Type: Device

Targets (0)

Add Target

Download Database (All)

<input type="checkbox"/>	Target Name	Type	Rating <a href="#">?</a>	Status <a href="#">v</a>	Date Modified
--------------------------	-------------	------	--------------------------	--------------------------	---------------

Last updated: Today 09:34 PM [Refresh](#)

Для задач данной ЛР в качестве метки будут использоваться плоские, хорошо распознаваемые изображения, например: фотографии, картинки, логотипы, баркоды и т.д. Такие объекты необходимо подготовить заранее: и в электронном формате (**.jpg**, **.png**), и в виде твердой копии.

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

### Add Target

Type:



Single Image



Cuboid



Cylinder



3D Object

File:

.jpg or .png (max file 2mb)

Width:

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:

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

В открывшемся окне:

- выбираем тип метки – **Single Image**,
- загружаем подготовленный заранее файл, содержащий метку (**File**),
- указываем реальный физический размер твердой копии метки в метрах (**Width**).

Поле Name заполняется автоматически.

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

Результат анализа файла метки приведен в окне:

vuforia engine<sup>™</sup>  
developer portal
Hello PirogovaMA | Log Out

[Home](#) [Pricing](#) [Downloads](#) [Library](#) [Develop](#) [Support](#)

[License Manager](#) [Target Manager](#)

Target Manager > AR-001

## AR-001 [Edit Name](#)

Type: Device

Targets (1)

Target Name	Type	Rating <sup>ⓘ</sup>	Status <sup>▼</sup>	Date Modified
<input type="checkbox"/>  AR-001Video	Single Image	★★★★★	Active	Mar 14, 2020 22:08

Last updated: Today 10:08 PM [Refresh](#)

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:**

The screenshot shows the vuforia engine developer portal. The top navigation bar includes 'Home', 'Pricing', 'Downloads', 'Library', 'Develop', and 'Support'. Below this, there are tabs for 'License Manager' and 'Target Manager'. The main content area is titled 'Target Manager > AR-001'. Underneath, it shows 'AR-001' with an 'Edit Name' link and 'Type: Device'. A 'Targets (2)' box is present. Below this are 'Add Target' and 'Download Database (All)' buttons. A table lists the targets:

<input type="checkbox"/>	Target Name	Type	Rating ⓘ	Status ▾	Date Modified
<input type="checkbox"/>	 AR-001Image	Single Image	★★★★★	Active	Mar 14, 2020 22:20
<input type="checkbox"/>	 AR-001Video	Single Image	★★★★★	Active	Mar 14, 2020 22:08

At the bottom, it says 'Last updated: Today 10:20 PM' with a 'Refresh' link.

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

Target Manager &gt; AR-001

## AR-001 [Edit Name](#)

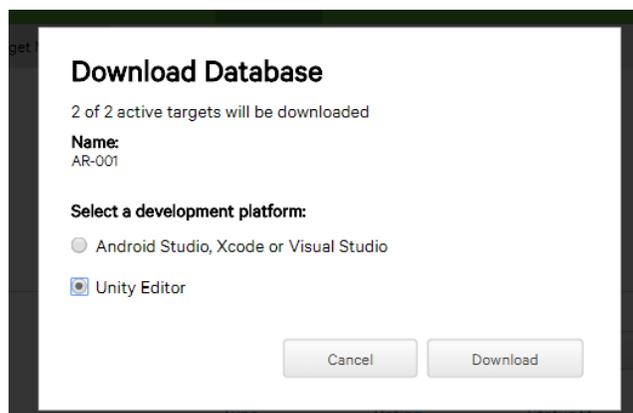
Type: Device

Targets (2)

Add Target

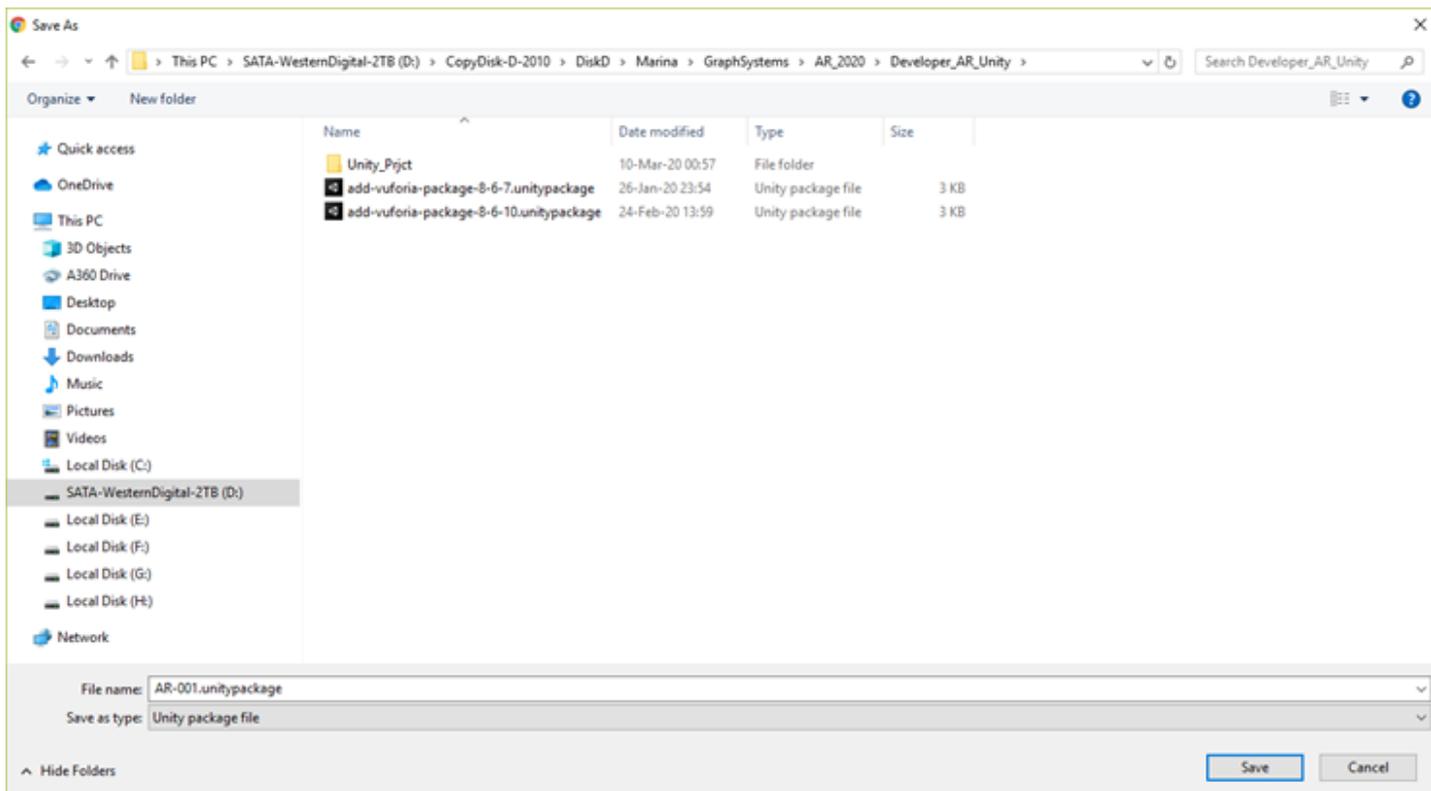
Download Database (2)

<input type="checkbox"/>	Target Name	Type	Rating <a href="#">i</a>	Status <a href="#">v</a>	Date Modified
2 selected <a href="#">Delete</a>					
<input checked="" type="checkbox"/>	 AR-001Image	Single Image	★★★★☆	Active	Mar 14, 2020 22:20
<input checked="" type="checkbox"/>	 AR-001Video	Single Image	★★★★★	Active	Mar 14, 2020 22:08

Last updated: Today 10:20 PM [Refresh](#)

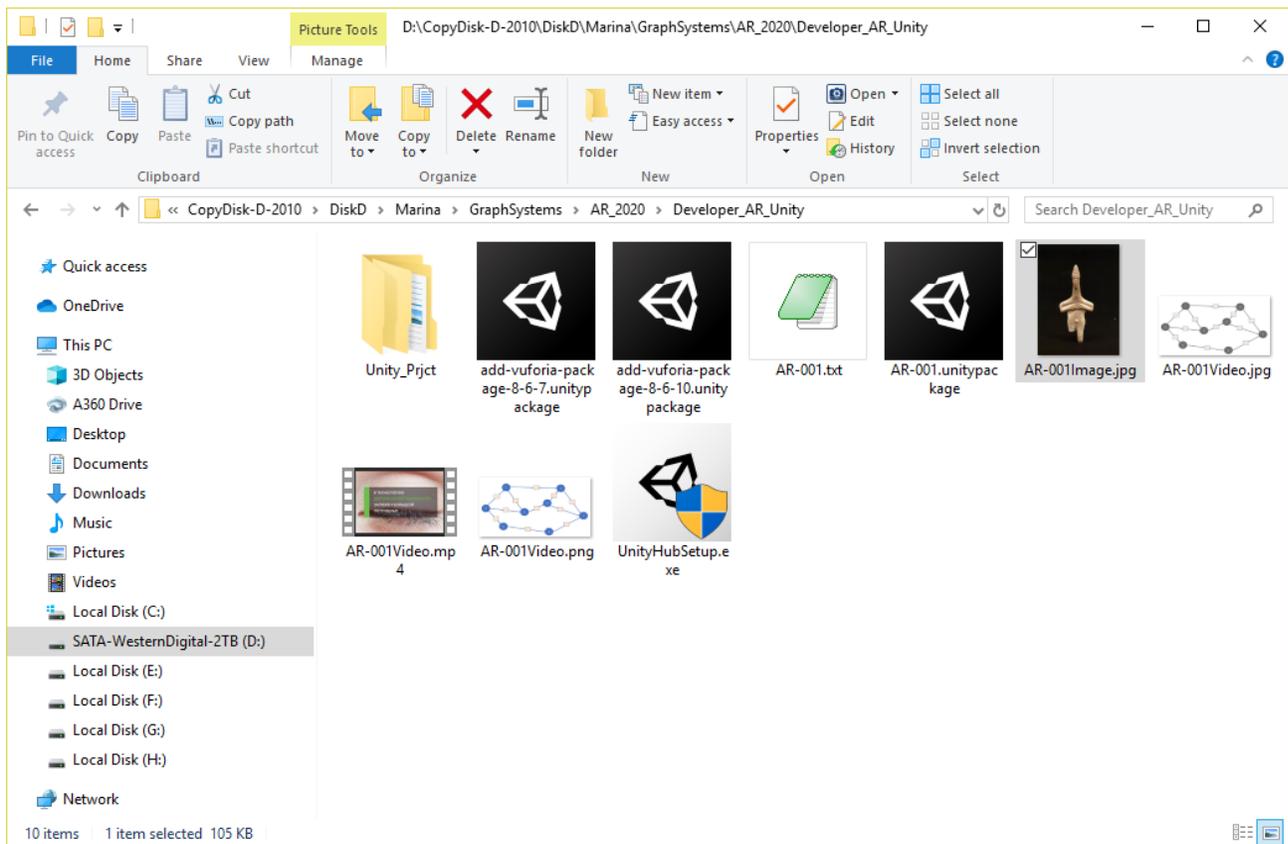
**ВАЖНО!!** Не забудьте выбрать платформу работы с объектами ДР – в нашем случае это **Unity Editor**. Для подтверждения правильности выбора и выполнения загрузки – нажмите **Download**.

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:



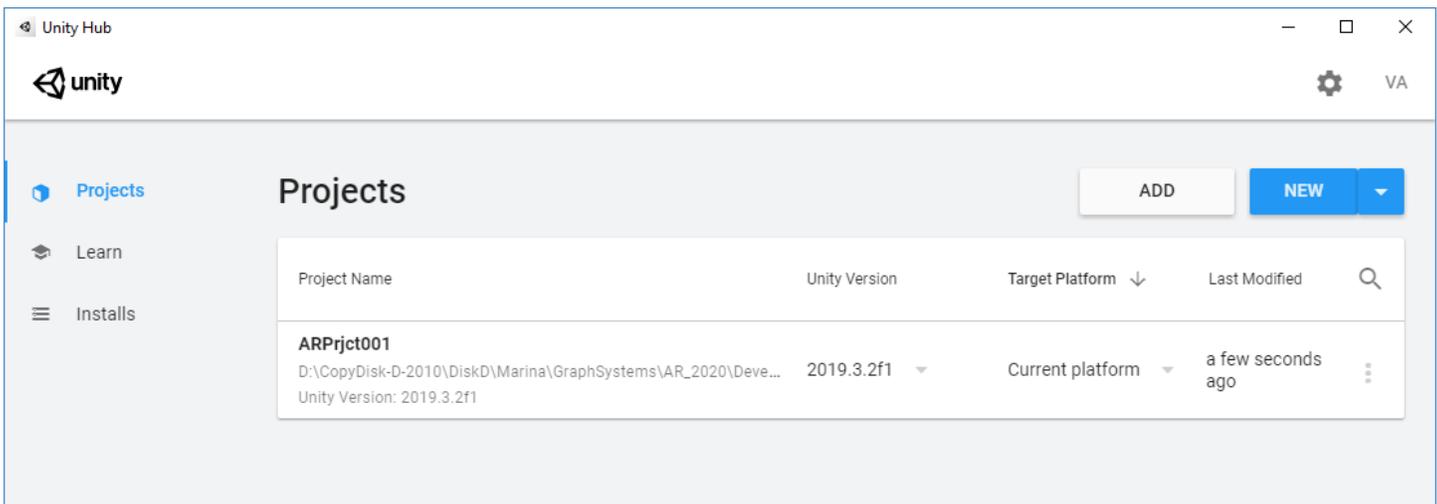
В результате выполнения двух описанных процедур мы получили из облака **Vuforia** для локальной работы в **Unity 3D** следующие ресурсы:

- 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: на данном этапе это видеоклип в формате **.mp4** – файл **AR-001Video.mp4**.

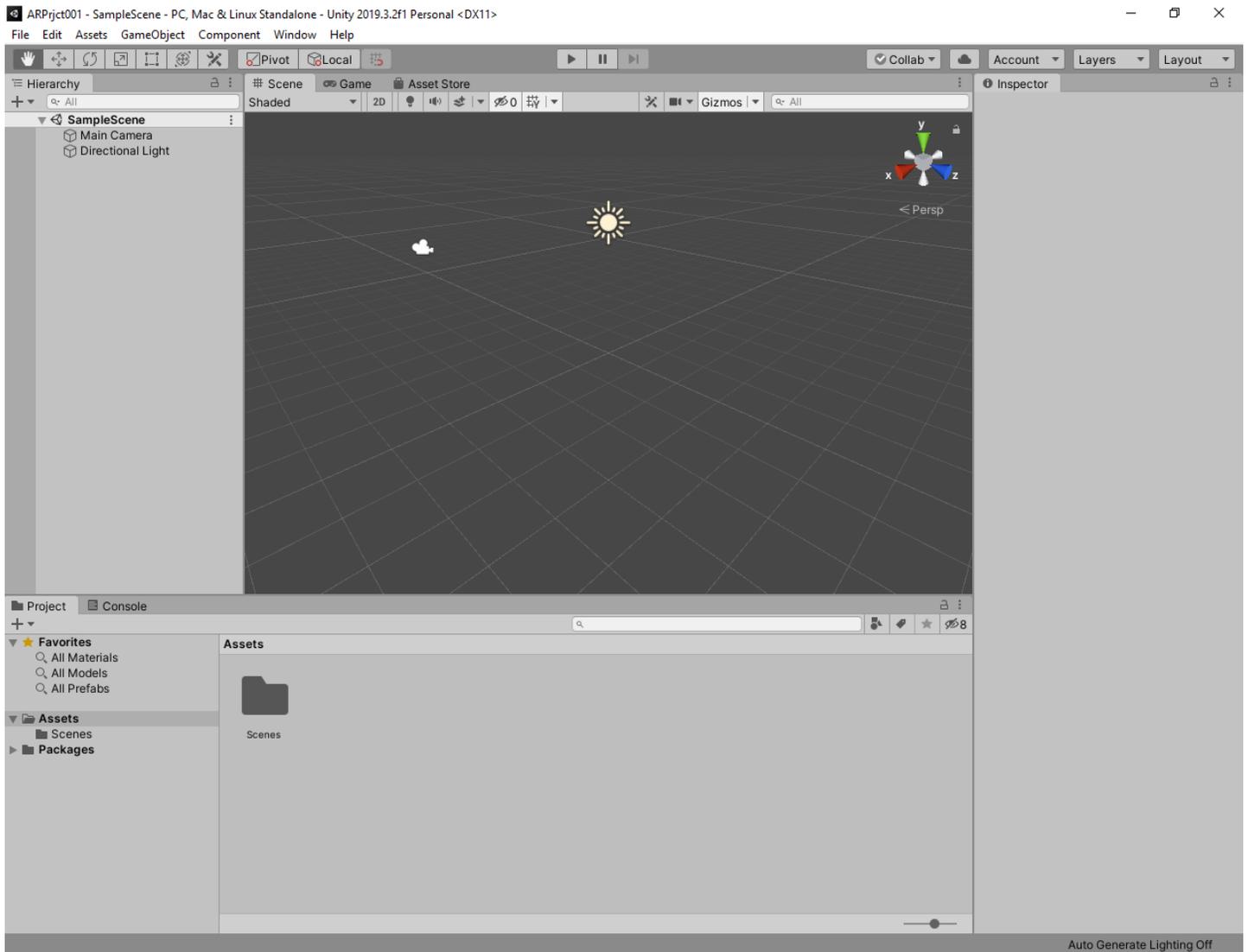


3. Стартуем приложение **Unity 3D** и выполняем настройки, необходимые для начала работы с **Vuforia**.

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

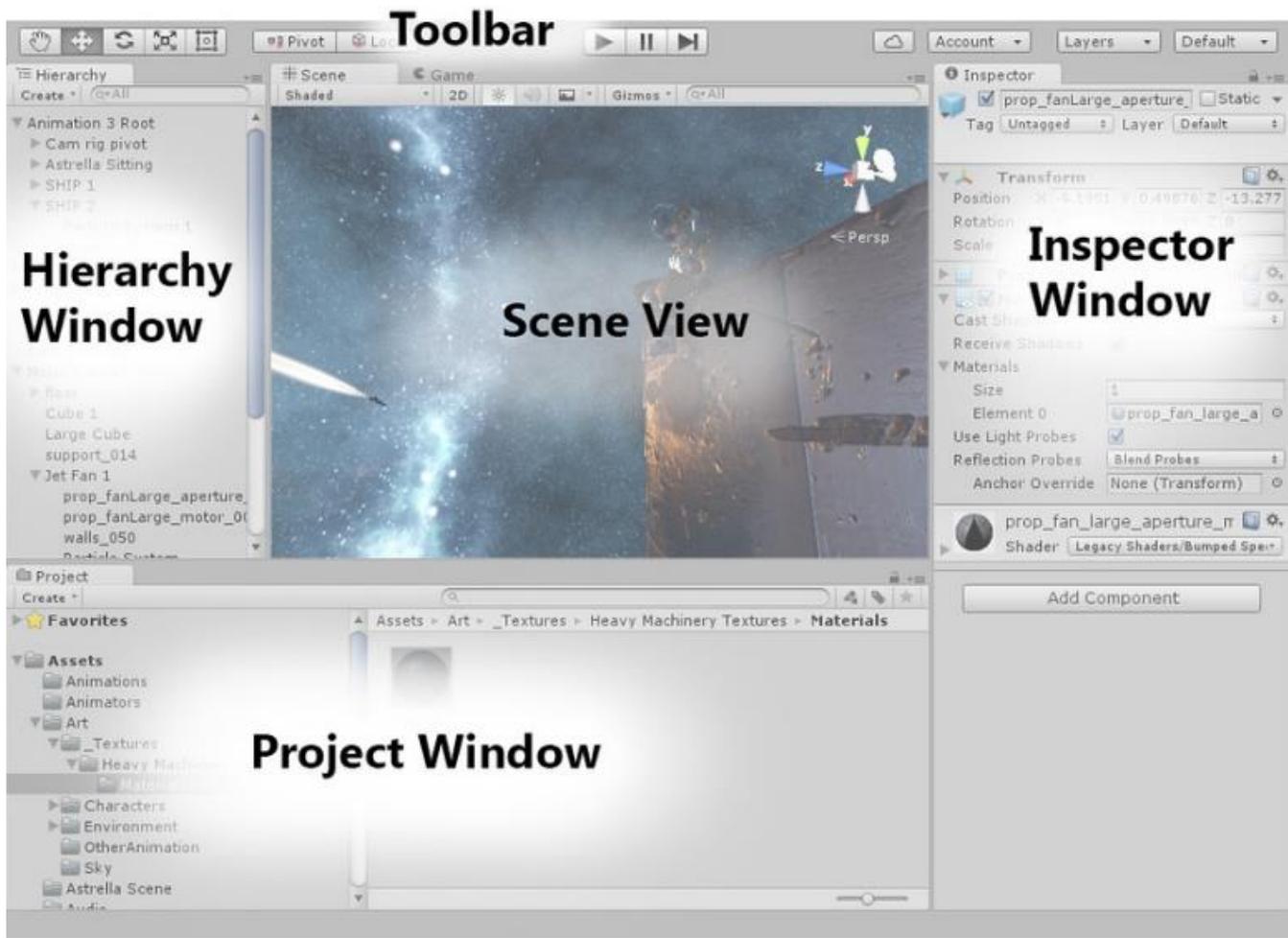


Стартуем проект (клик левой клавишей по имени в списке). Открывается редактор **Unity 3D**:



Документацию по работе в редакторе – см. по ссылке:

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



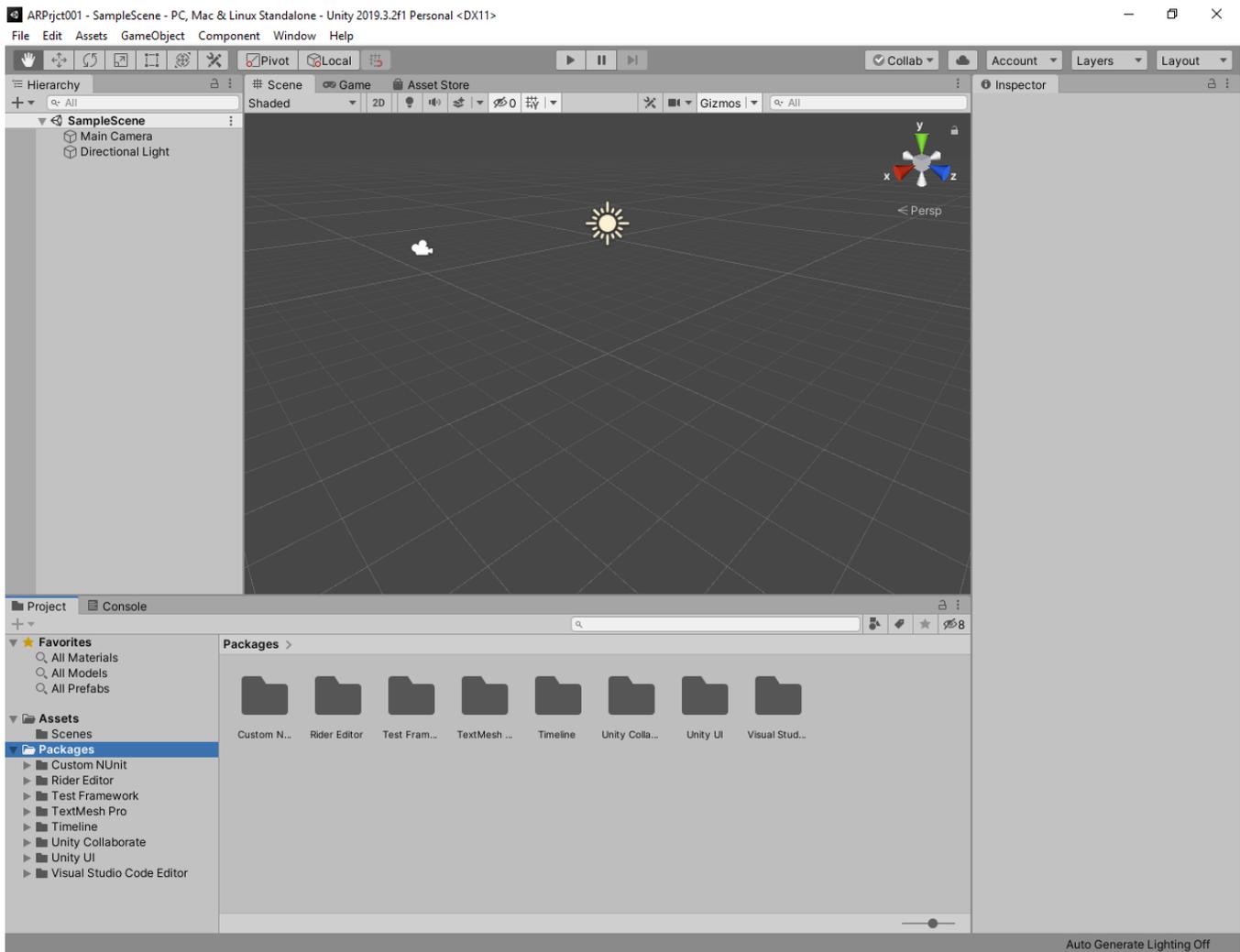
**Обратите внимание** - начальные действия по организации среды игрового движка **Unity 3D** для нашего проекта (Приложение ДР) происходят с использованием закладки **Scene** (область **Scene View**) и группы элементов линейки **Toolbar**, объединенных в группу с названием **GameObject**.

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** необходимо добавить специфические объекты **Vuforia**. А работу с ними поддержать действующими лицензиями **Vuforia**.

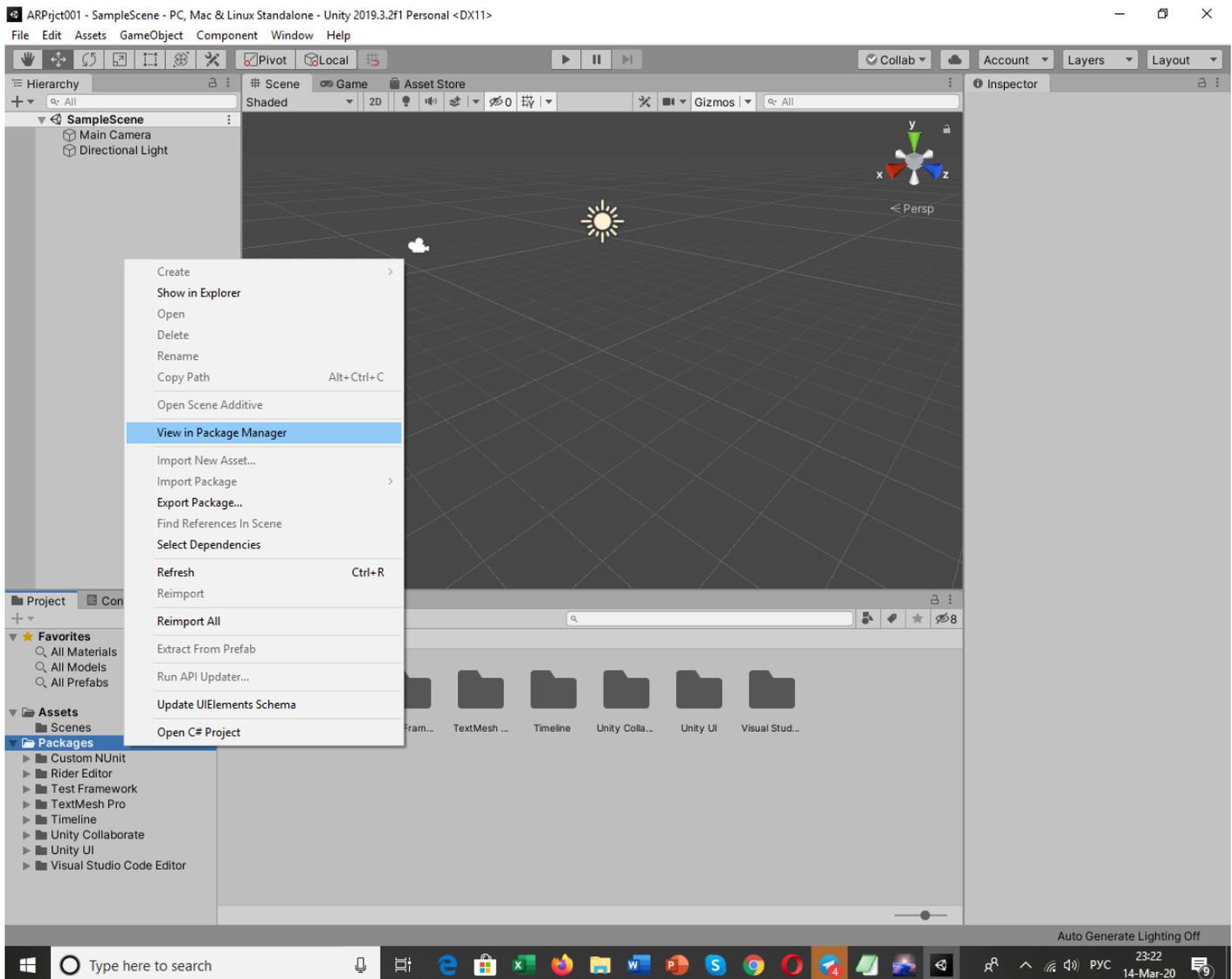
Для этого необходимо файлы типа **.unitypackage** загрузить в **Unity 3D** с помощью входящего в ее состав средства **Package Manager**. Эти файлы вы заранее подготовили (в ЛР2) и загрузили из облака **Vuforia** на локальную машину (см. ЛР1).

Использование **Package Manager** для подключения **Vuforia** предполагает выполнение следующих шагов:

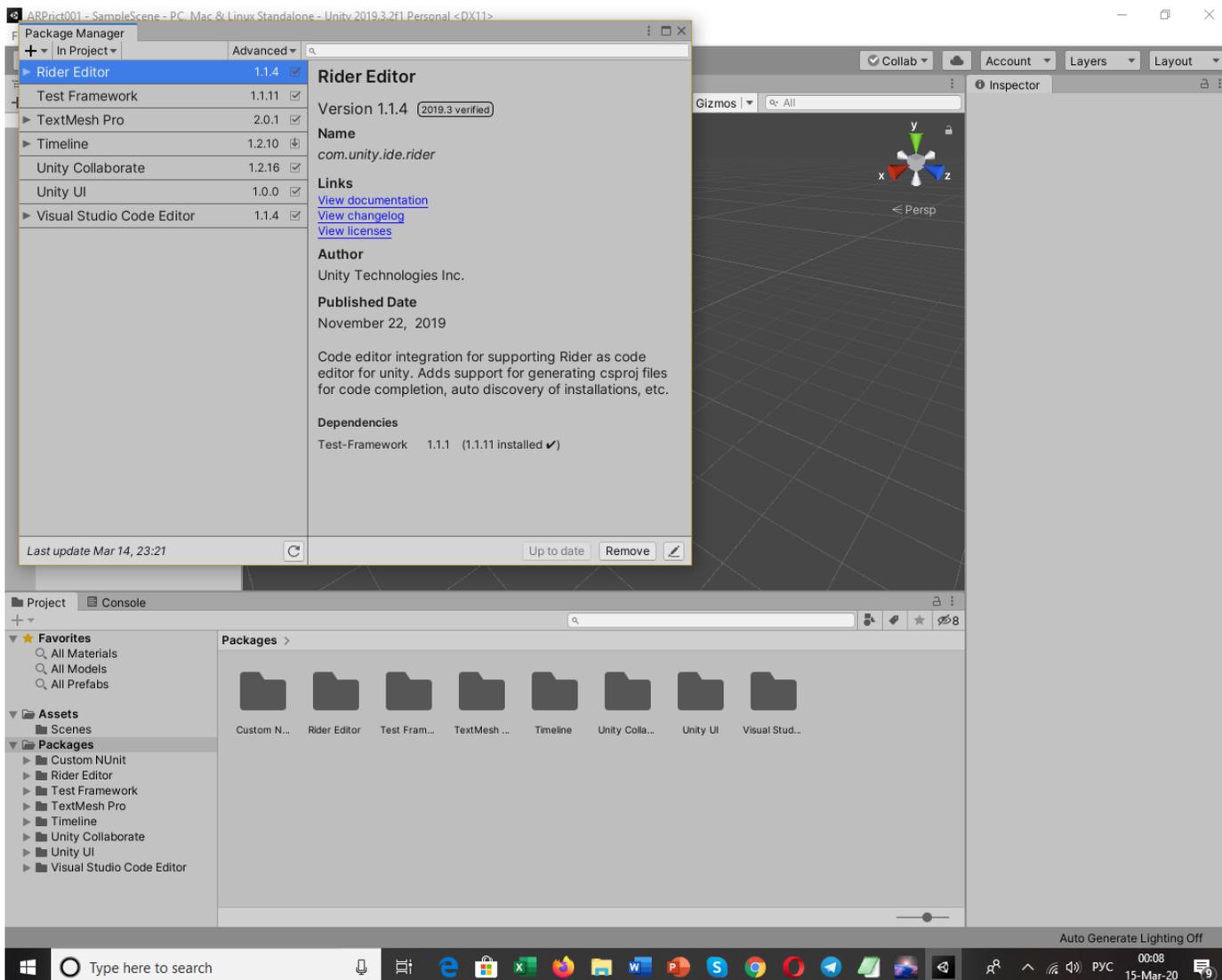
- В области **Project Window** в строке **Packages** → кликаем правой клавишей мыши (**RMB**)



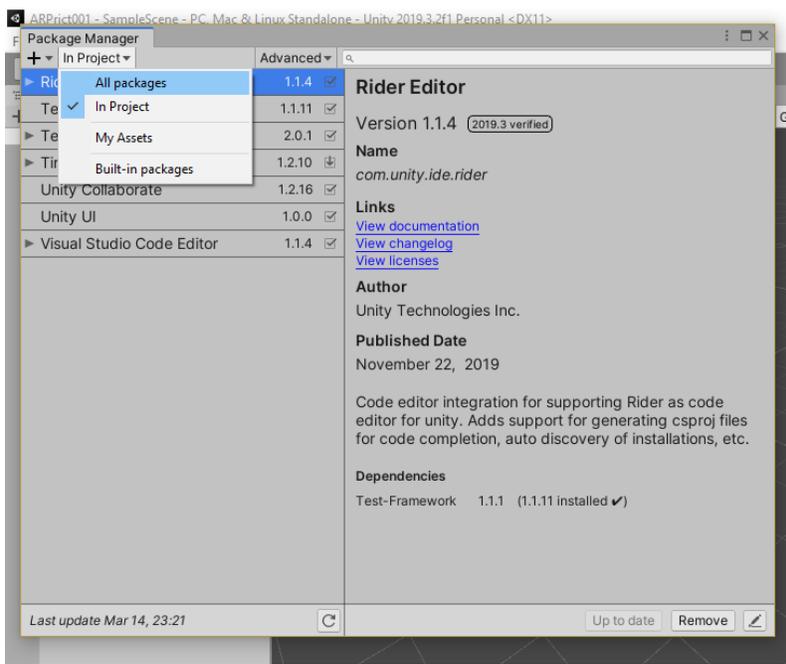
- В списке возможных действий выбираем **View in Package Manager**:



- В результате, спустя некоторое время появится окно **Package Manager** с полным списком доступных пакетов (см. далее на стр. 20); Если же у вас появляется окно **Package Manager**, так, как отображено ниже (на стр. 19), то для получения полного списка необходимо выполнить промежуточные действия, описанные на стр. 19 :



- В окне **Package Manager** в падающем меню необходимо выбрать режим **All packages**, в котором возможно подключение пакетов Vuforia:



- В результате в списке **All packages** выбираем инсталляцию (кнопка **Install**) **Vuforia Engine AR**

The screenshot shows the Unity Package Manager window. On the left, a list of packages is displayed with columns for package name and version. The package 'Vuforia Engine AR' with version '8.5.9' is highlighted in blue. On the right, the details for 'Vuforia Engine AR' are shown, including its version (8.5.9), name, author (PTC Inc.), published date (October 29, 2019), a description, and its dependencies (Unity UI 1.0.0, which is installed).

Package Name	Version
Oculus Desktop	2.38.4
Oculus XR Plugin	1.1.5
OpenVR Desktop	2.0.5
Polybrush	1.0.1
Post Processing	2.3.0
ProBuilder	4.2.3
Quick Search	1.5.2
Remote Config	1.0.9
Rider Editor	1.1.4 <input checked="" type="checkbox"/>
Scriptable Build Pipeline	1.5.6
Searcher	4.0.9
Settings Manager	1.0.2
Shader Graph	7.1.8
Subsystem Registration	1.0.6
Test Framework	1.1.11 <input checked="" type="checkbox"/>
TextMesh Pro	2.0.1 <input checked="" type="checkbox"/>
Timeline	1.2.10 <input type="checkbox"/>
Unity Collaborate	1.2.16 <input checked="" type="checkbox"/>
Unity Distribution Portal	1.3.0
Unity UI	1.0.0 <input checked="" type="checkbox"/>
Universal RP	7.1.8
Visual Effect Graph	7.1.8
Visual Studio Code Editor	1.1.4 <input checked="" type="checkbox"/>
<b>Vuforia Engine AR</b>	<b>8.5.9</b>
Windows Mixed Reality	4.2.1
Windows XR Plugin	2.0.3
Xiaomi SDK	1.0.3
XR Interaction Subsystems	1.0.1
XR Legacy Input Helpers	1.3.11

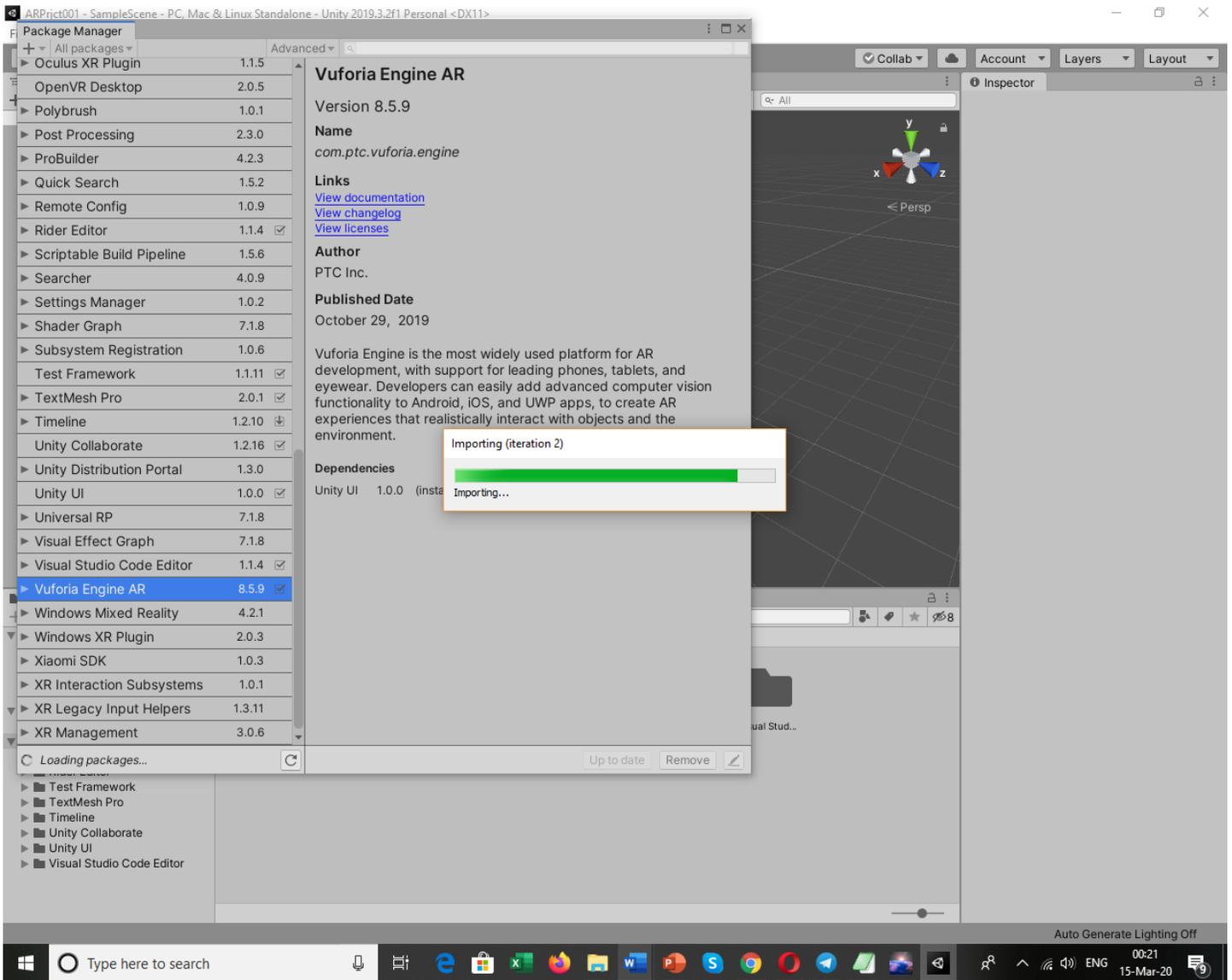
**Vuforia Engine AR**  
Version 8.5.9  
**Name**  
*com.ptc.vuforia.engine*  
**Links**  
[View documentation](#)  
[View changelog](#)  
[View licenses](#)  
**Author**  
PTC Inc.  
**Published Date**  
October 29, 2019  

Vuforia Engine is the most widely used platform for AR development, with support for leading phones, tablets, and eyewear. Developers can easily add advanced computer vision functionality to Android, iOS, and UWP apps, to create AR experiences that realistically interact with objects and the environment.

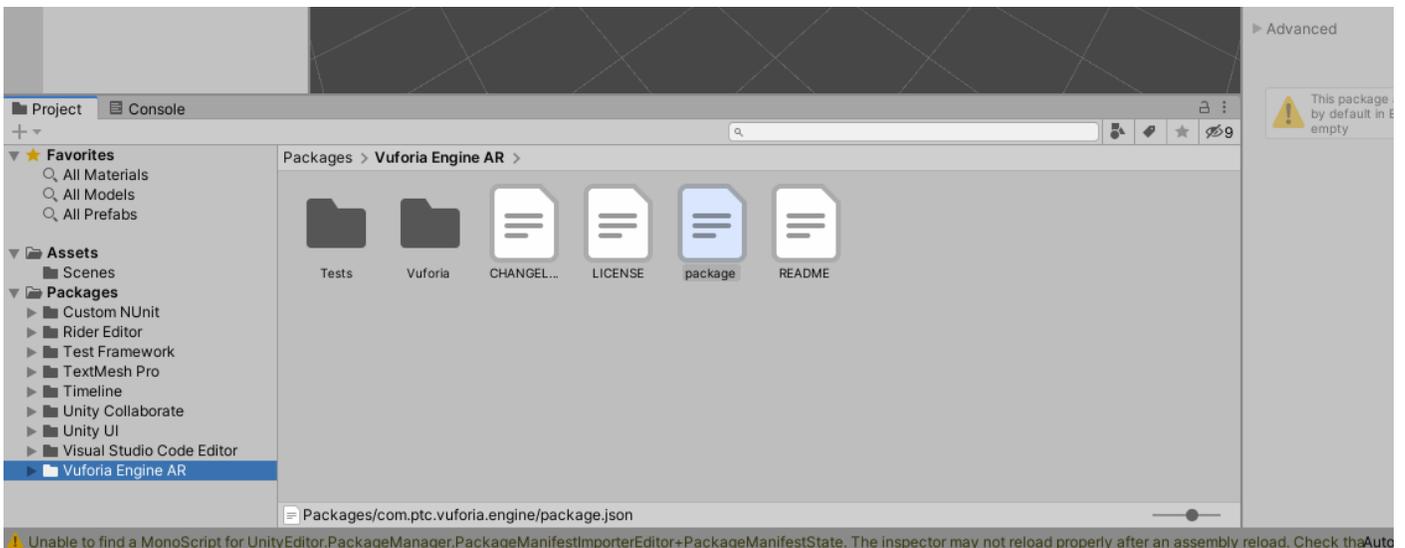
**Dependencies**  
Unity UI 1.0.0 (installed )

Last update Mar 14, 23:21

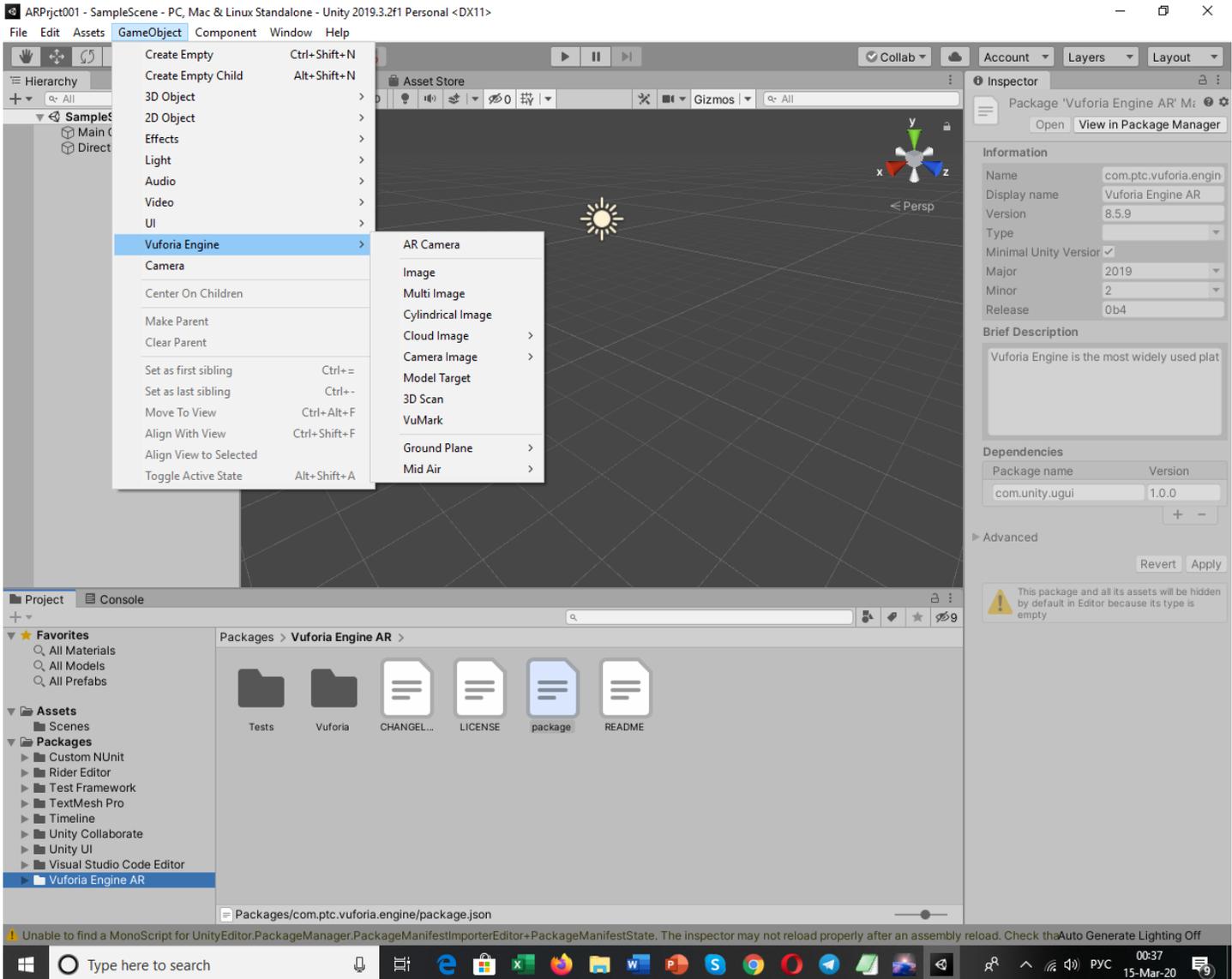
- Происходит инсталляция пакета **Vuforia**:



- 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 работа с **Package Manager** завершена. Базовые настройки для работы **Unity 3D** с **Vuforia** выполнены.

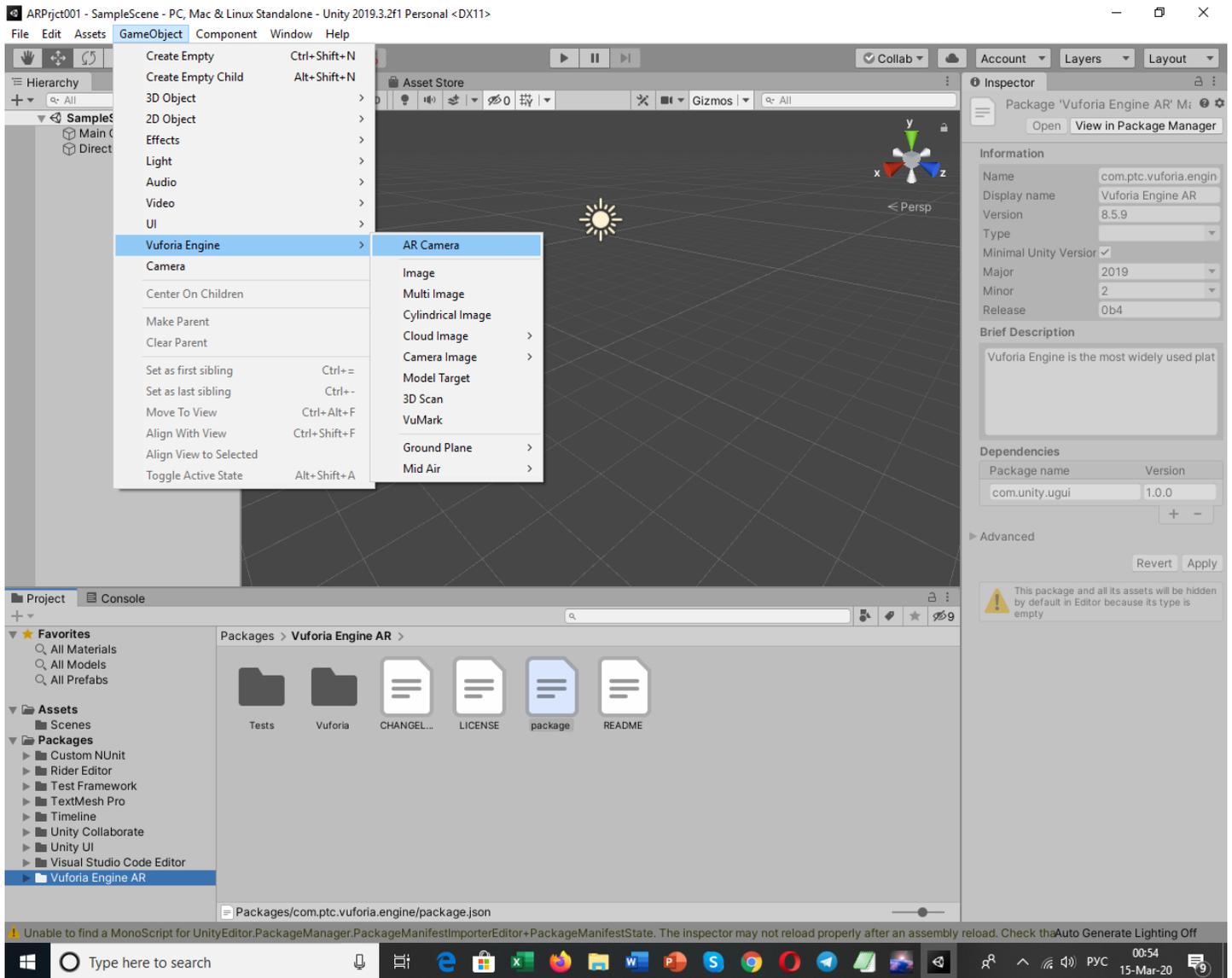
#### 4. Создание Приложения ДР в среде **Vuforia + Unity 3D** для просмотра видеоклипа.

В разрабатываемом Приложении ДР редактора **Unity 3D** не позволяет выполнять визуализацию реального окружения, а работает только с **виртуальными объектами сцены в виртуальном пространстве**.

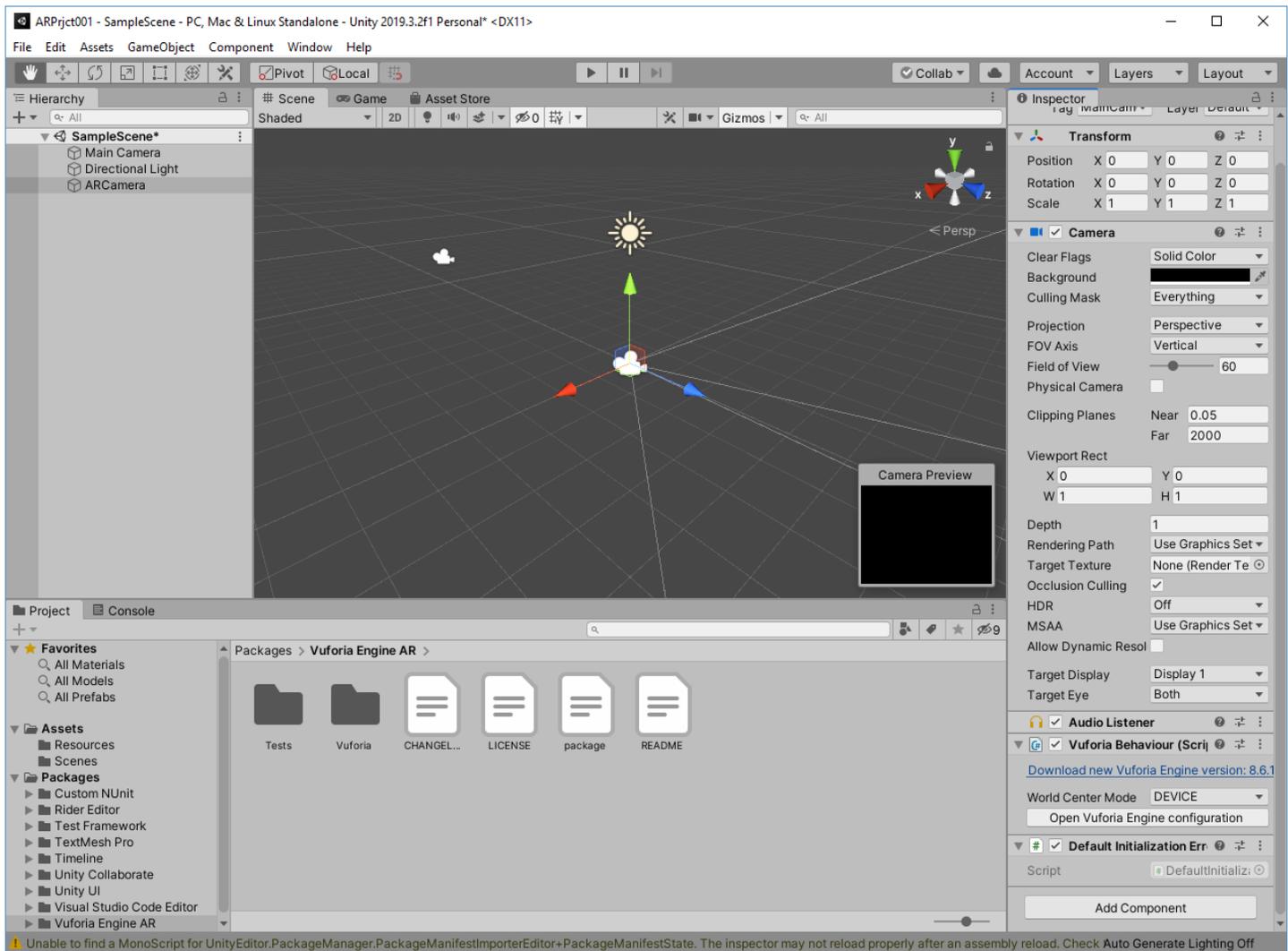
Для работы в режиме ДР необходима «камера», способная выполнять визуализацию виртуальных объектов сцены на фоне реального окружения. Для переключения нашего проекта из режима виртуальной реальности в режим Дополненной реальности заменим в наборе иерархических объектов (область экрана редактора **Hierarchy**)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

Для этого выполним следующие вызовы:

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

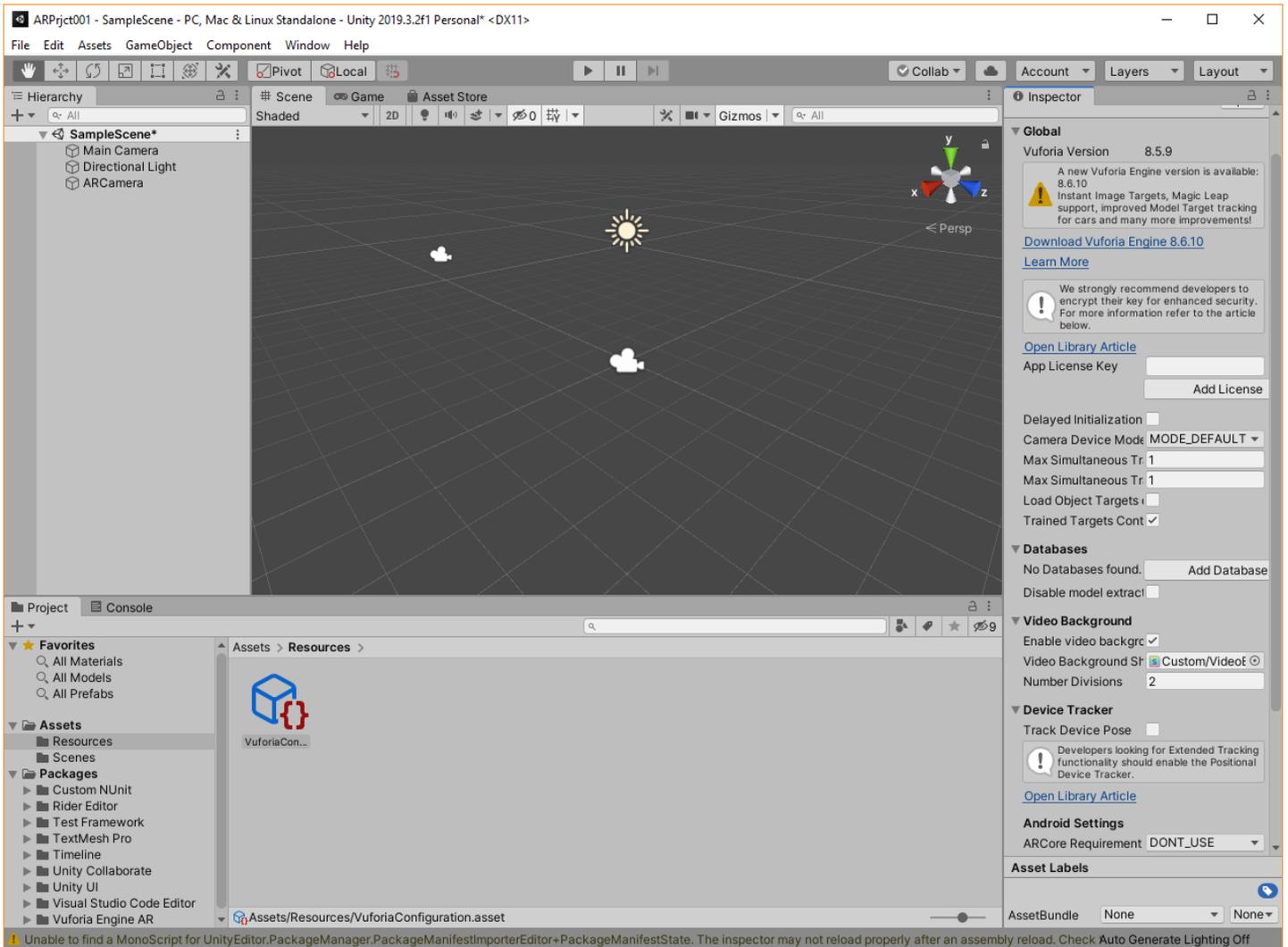


В области Hierarchy появляется Vuforia-объект ARCamera.

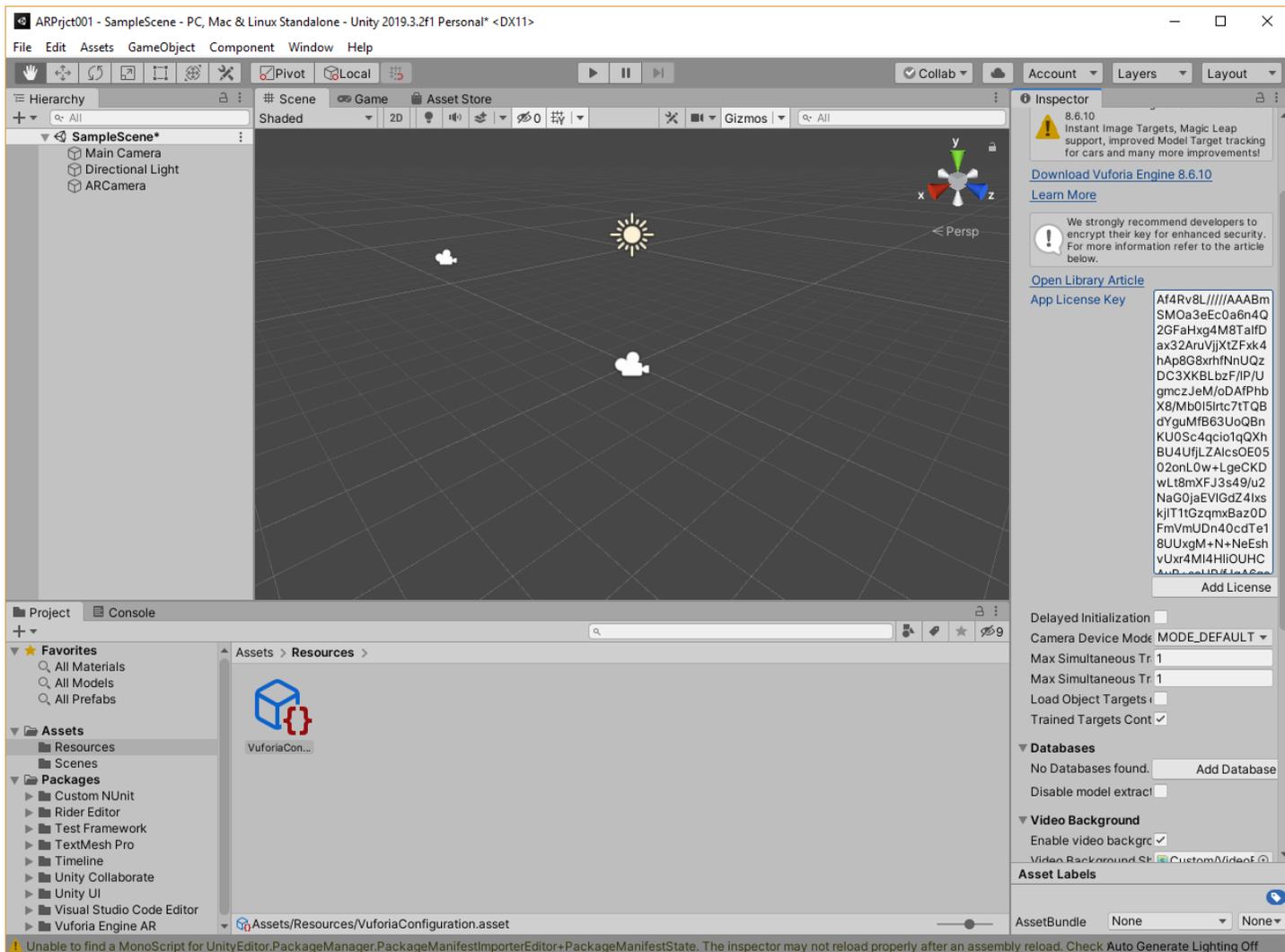


Для работы с объектами **Vuforia** необходимо перевести их под действие сгенерированной вами выше лицензии. Делать мы это будем через верхний уровень объектов **Vuforia**. Установленный объект **ARCamera** является верхним уровнем иерархии всех объектов **Vuforia** разрабатываемой сцены:

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

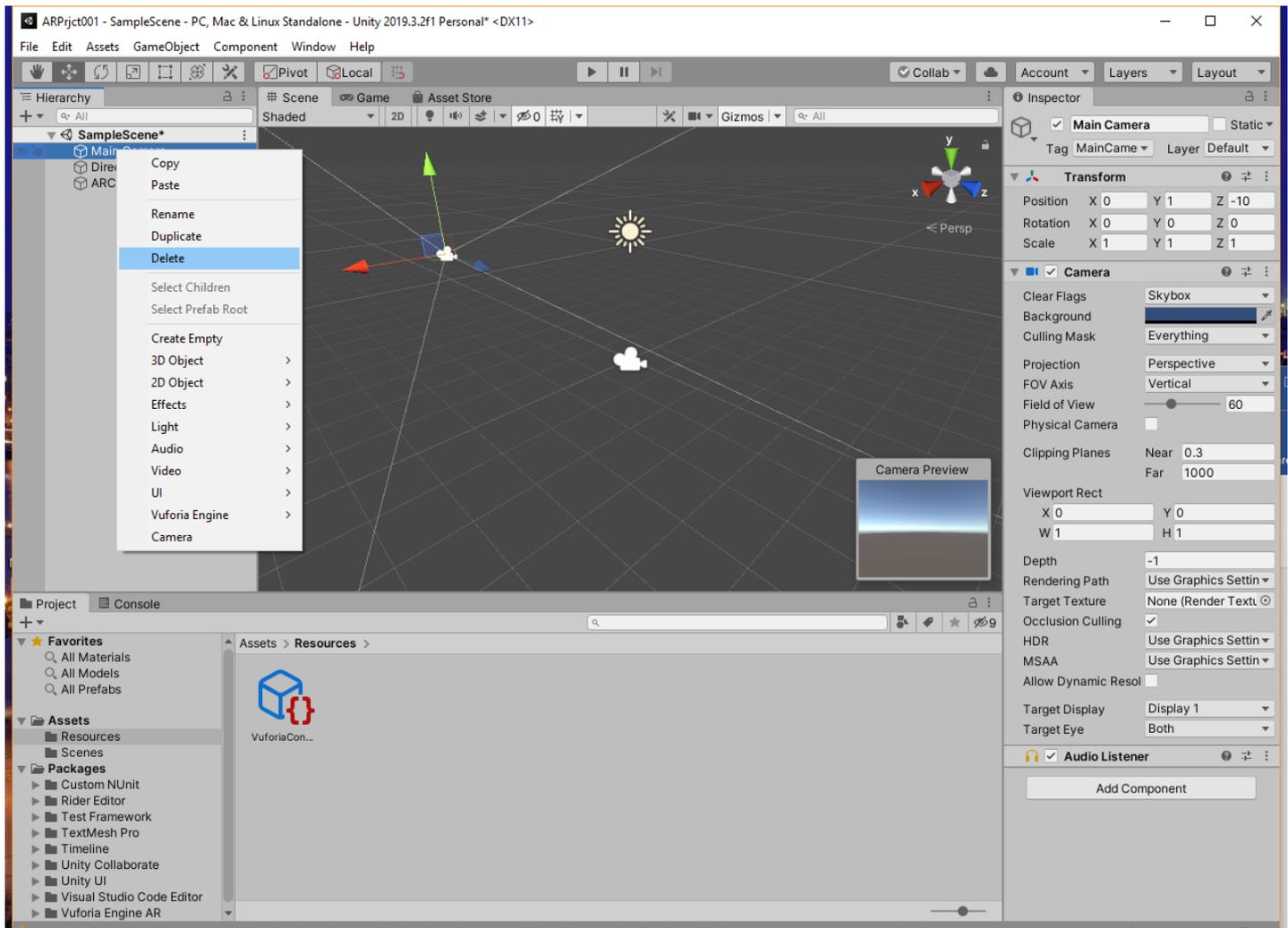


**ВАЖНО!!** Если Вы выберете на этом этапе поле **Add License**, то вновь попадете на этап ее генерации, который в данной ЛР был выполнен в первую очередь. Иными словами, лицензия автоматом не устанавливается при конфигурировании, ее необходимо скопировать в поле **App License Key**, предварительно сгенерив.

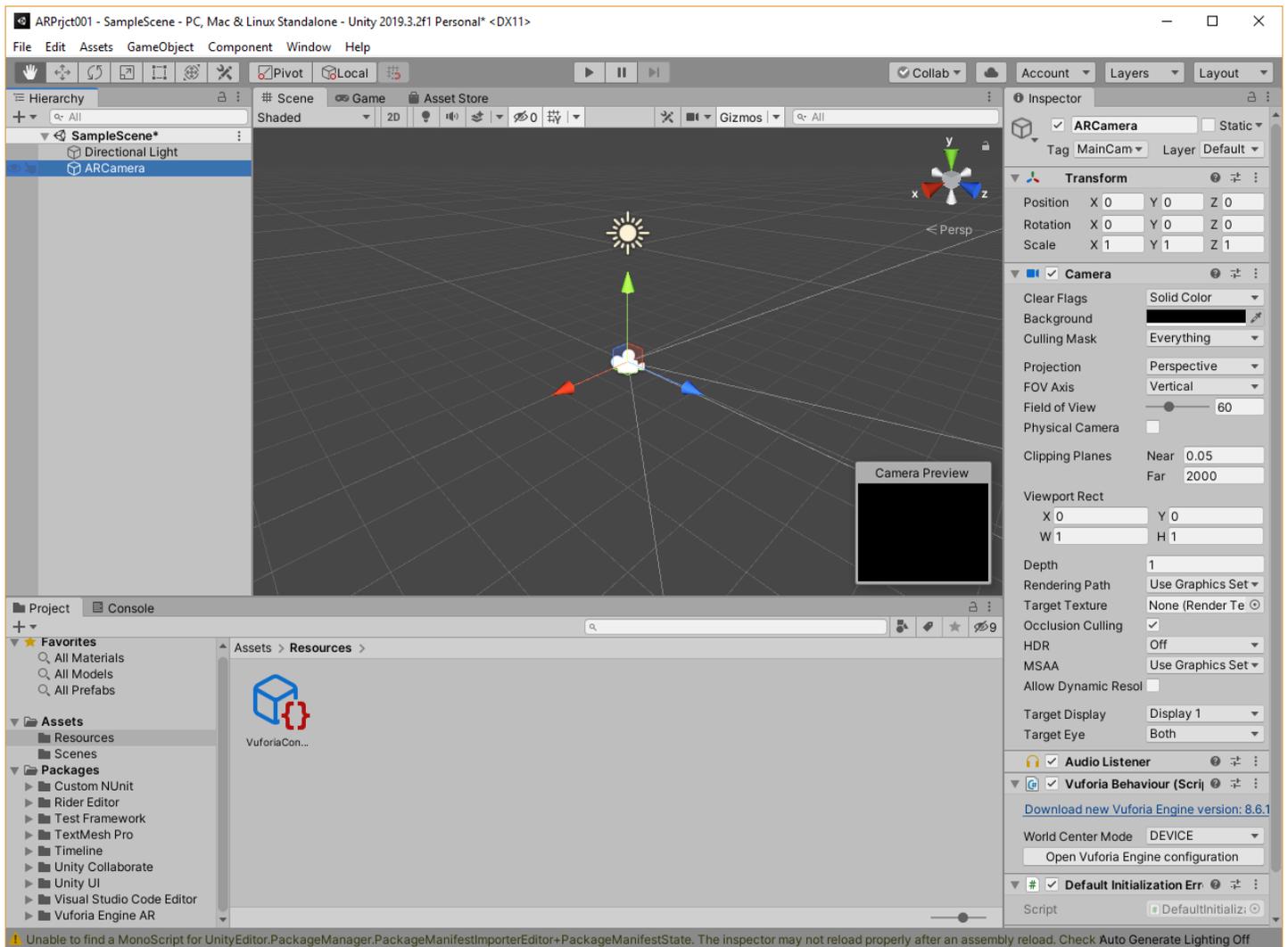
Но если вы забыли это сделать в свое время – здесь у вас есть возможность исправить эту ошибку.

**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**.

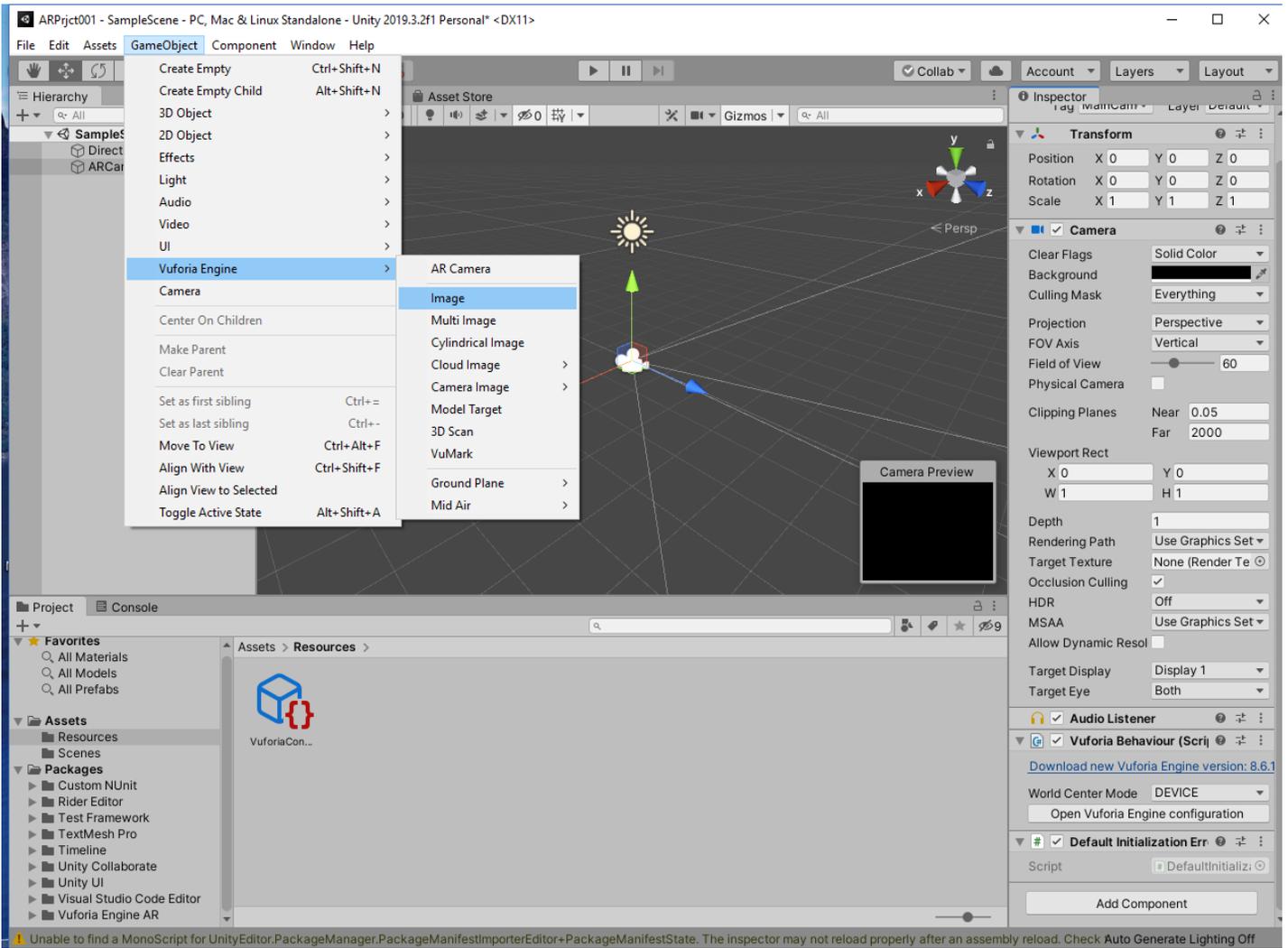


В результате в **Hierarchy** остается только одна камера, свойства которой отображены в области **Inspector**.

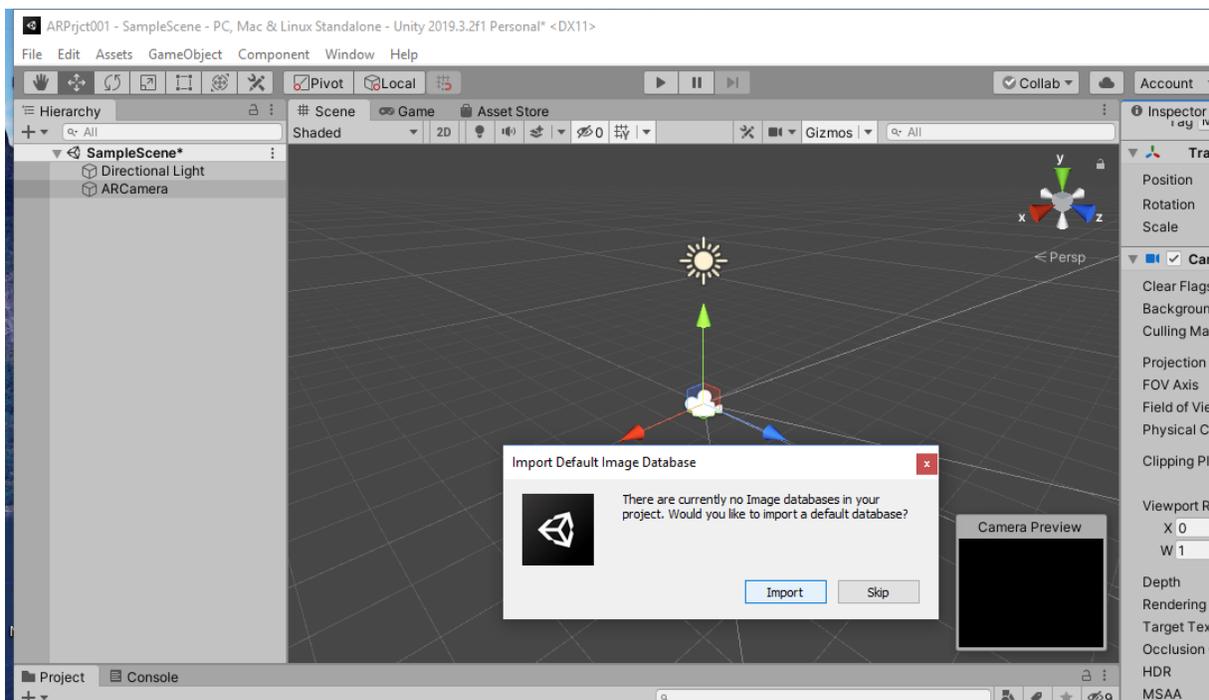


Далее необходимо загрузить Базу данных таргетов.

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

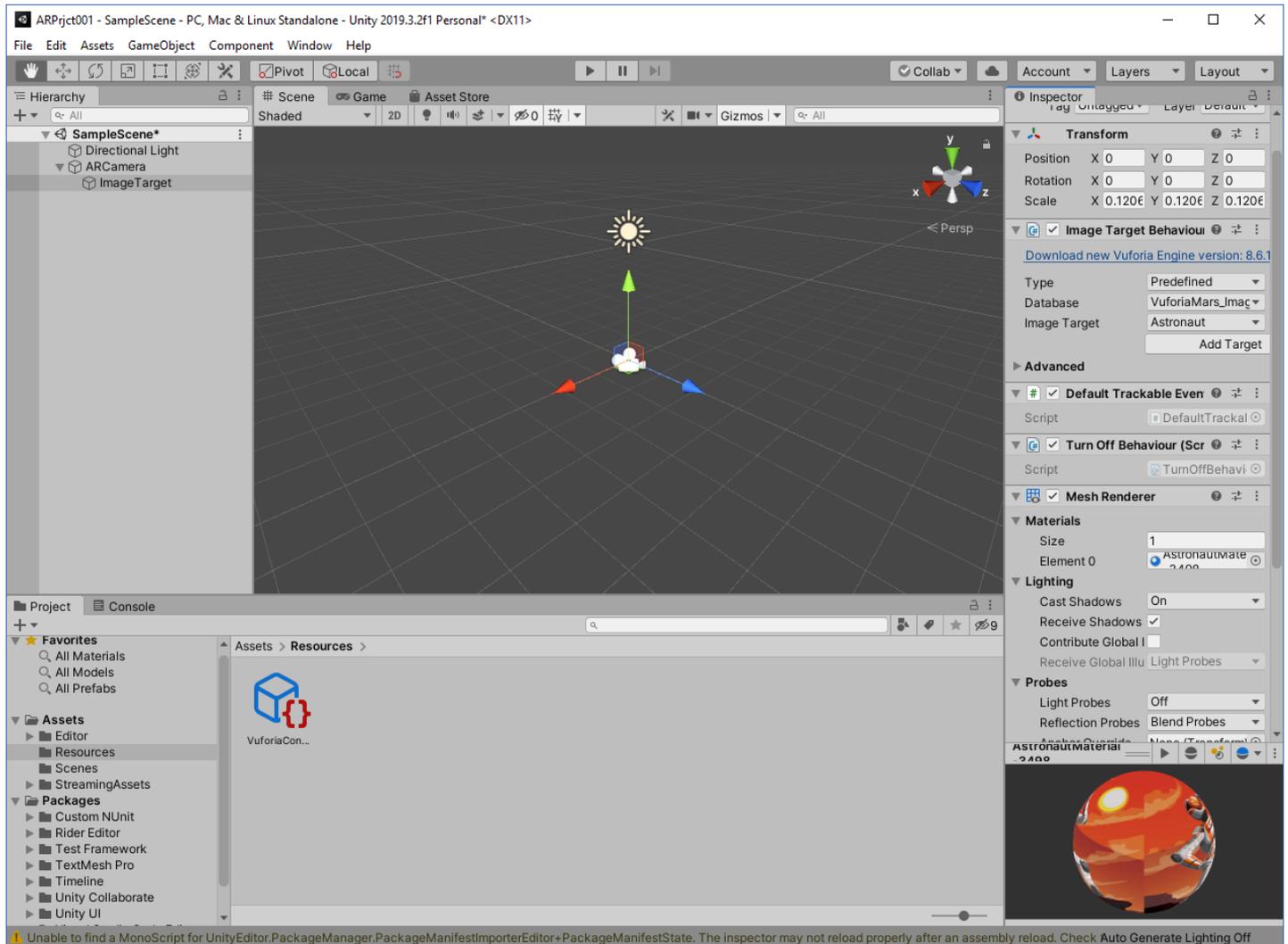


Получаем сообщение:



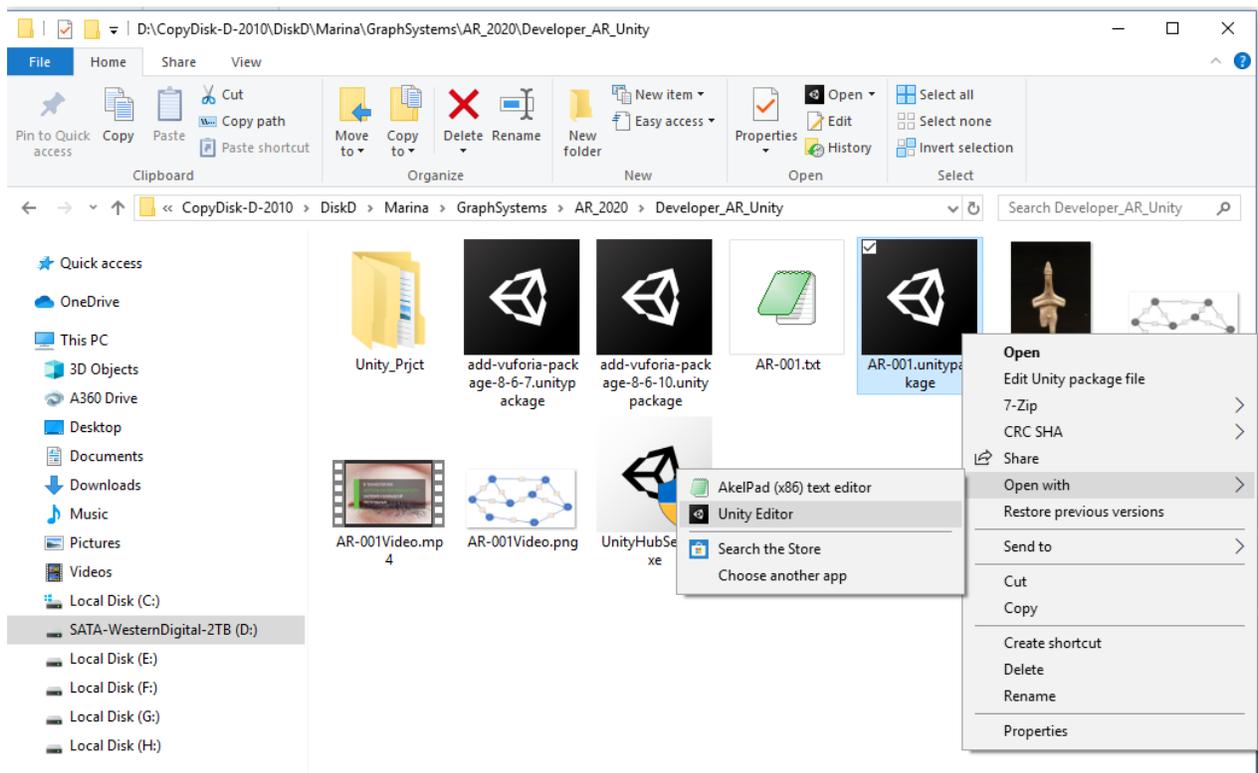
В появившемся запросе нажмите кнопку **Import**.

В результате загружается БД таргетов по умолчанию («**Астронавт**»).

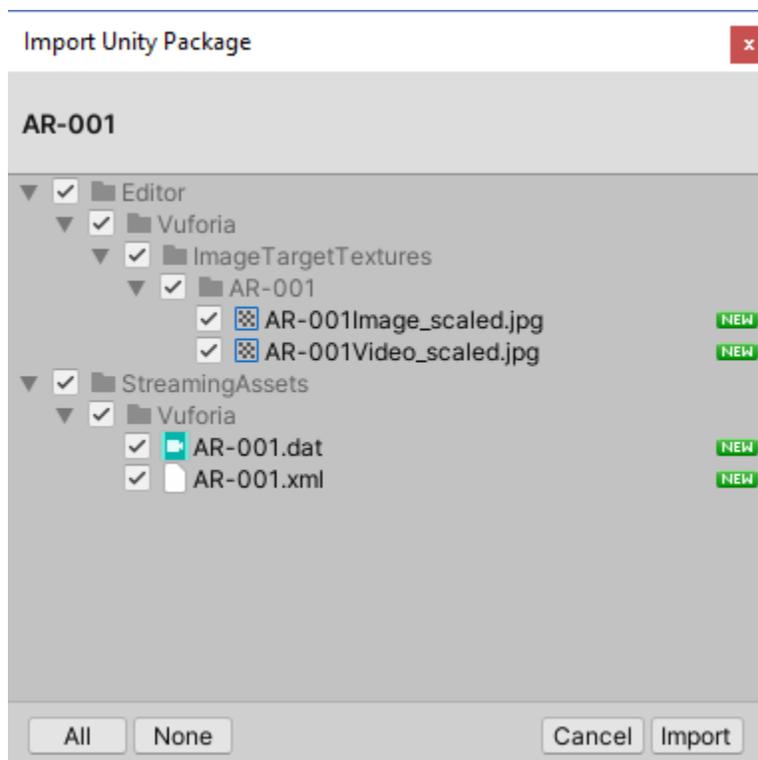


Добавляем подготовленную заранее БД таргетов в сцену нашего проекта **AR-001**.

Для этого в локальной файловой структуре находим эту ранее подготовленную БД таргетов (в конце шага 2), по правой клавише мыши вызываем контекстное меню работы с БД и выбираем действие (**ВОЛШЕБСТВО!!**) **Open with** → **Unity Editor**:

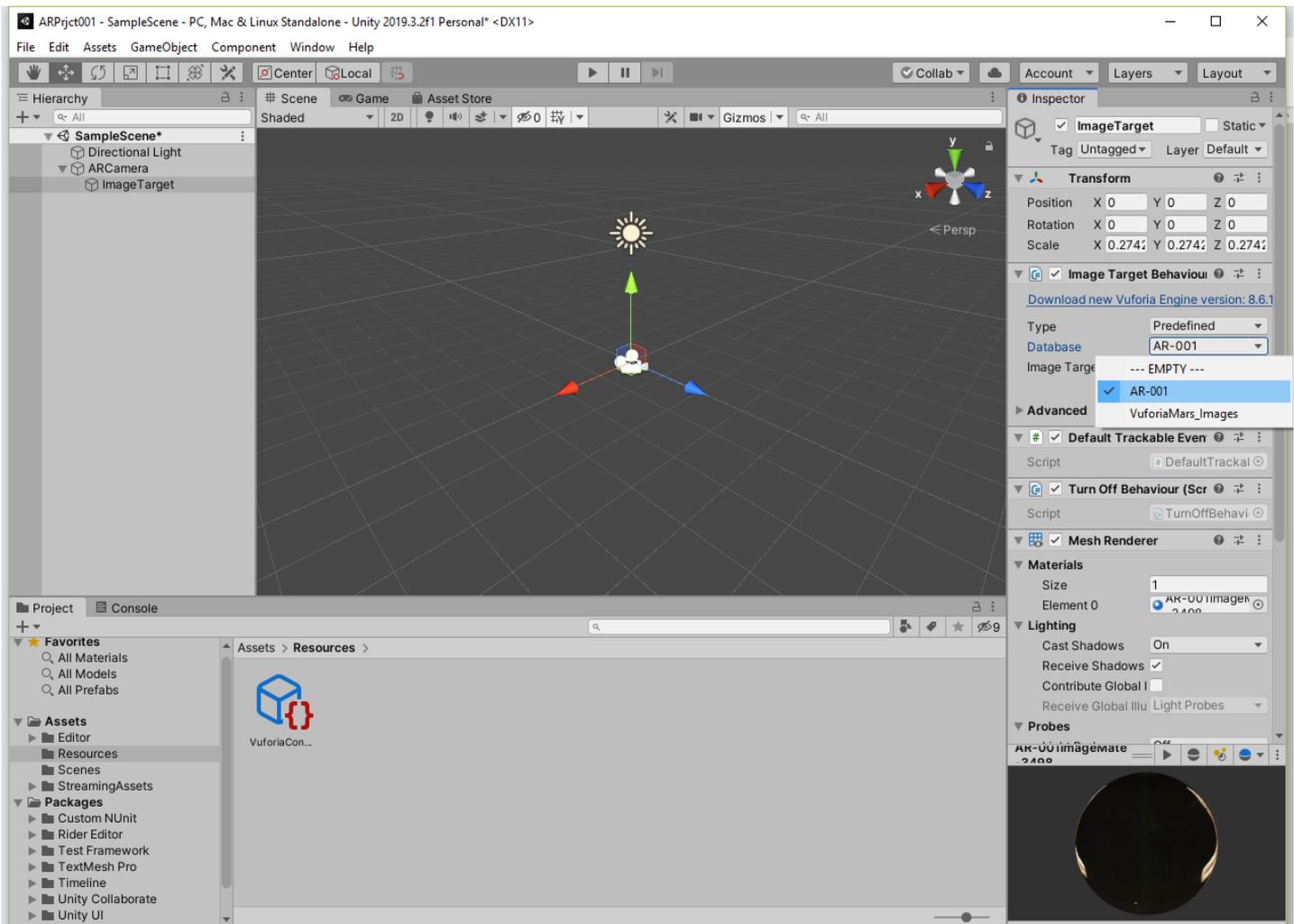


В результате в окне среды **Unity 3D** появляется диалоговое окно импорта БД таргетов:

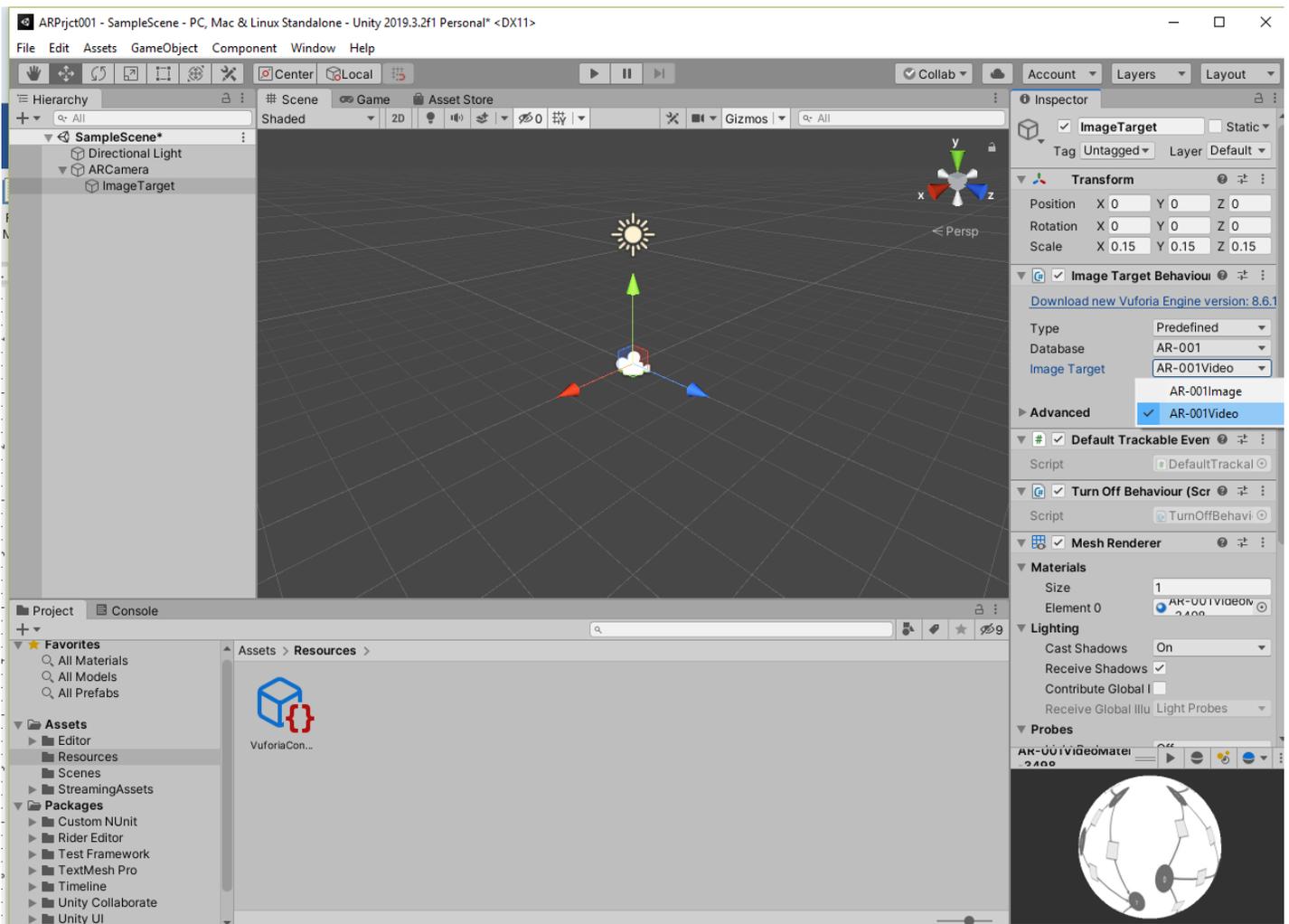


Выполняем импорт нашей БД таргетов проекта **AR-001** путем нажатия клавиши **Import**.

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**.

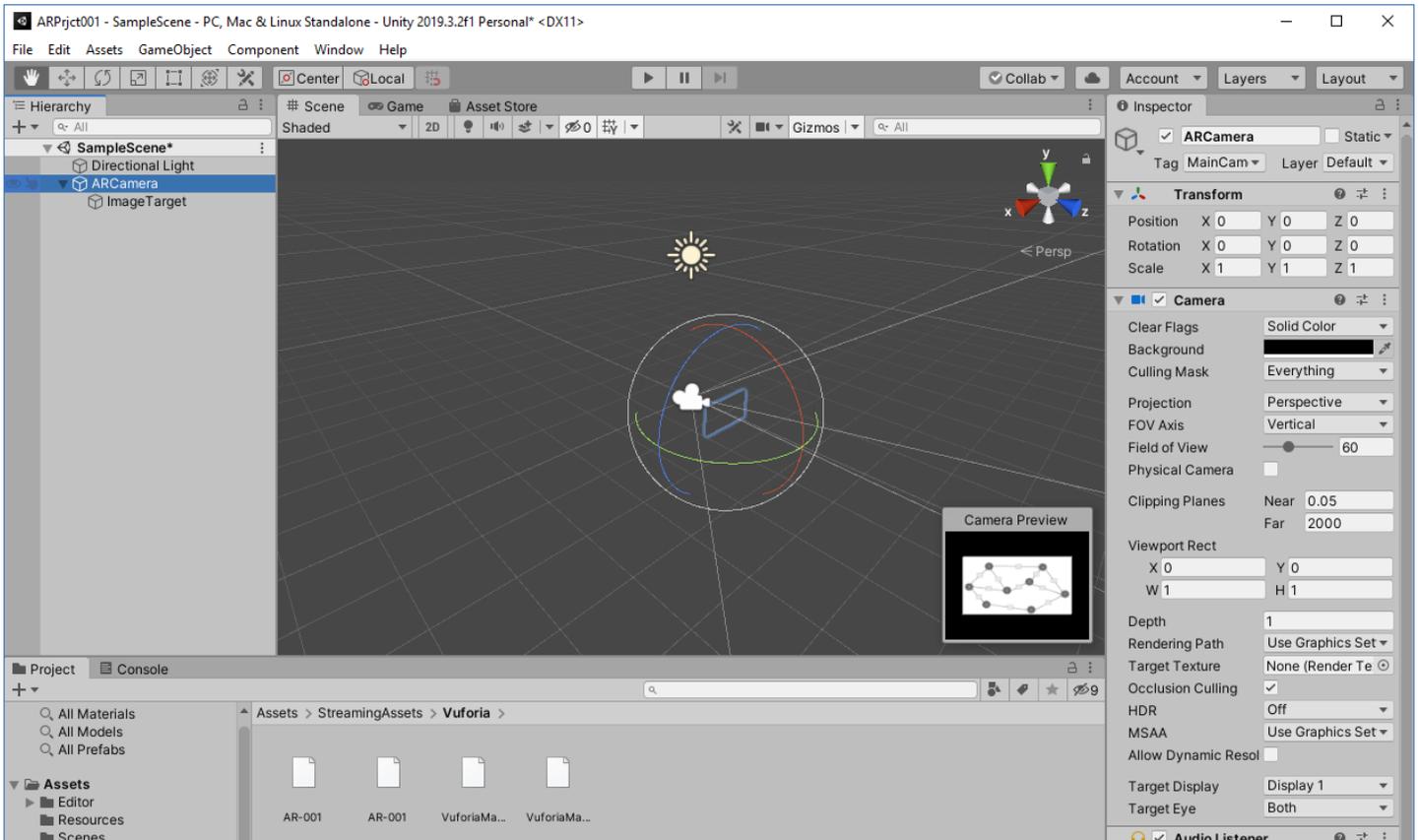
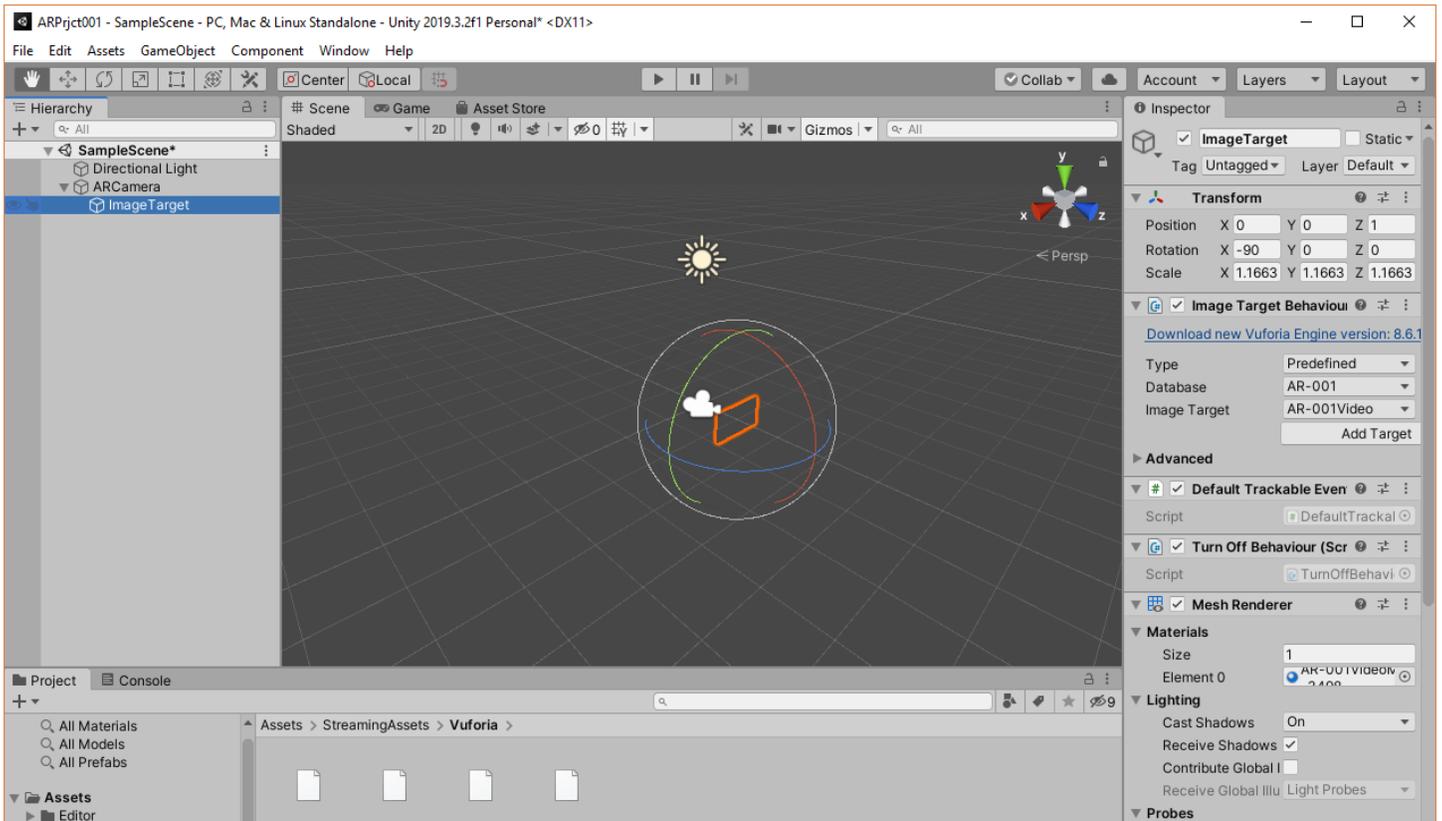


Теперь в поле сцены имеются два интересующих нас объекта **Vuforia**:

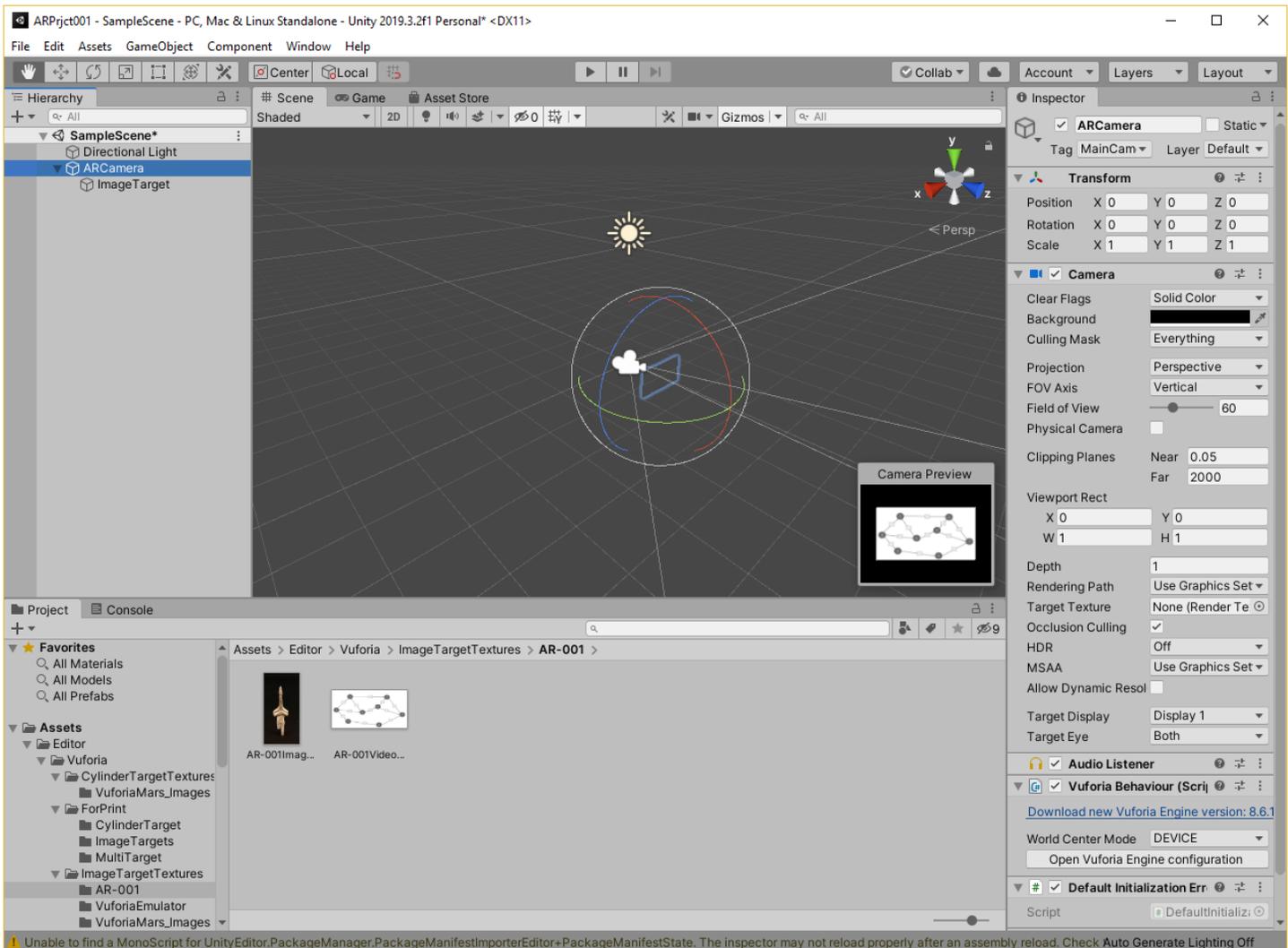
- **AR Camera** и
- **Image Target**.

Нам необходимо добиться правильного взаимного расположения камеры и таргета. 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** или аппарат **Gismo** или **Toolbar** как для объекта **AR Camera** так и для объекта **Image Target** (выбираются в **Hierarchy**). Хорошим результатом является тот, когда изображение **Image Target** занимает от 30% до 60% окна **Camera Preview**. Добейтесь этого.

Ниже приведен вариант взаимного расположения камеры и таргета, когда **ARCamera** смотрит фронтально на таргет, удаленный на расстояние «1» от камеры. Обратите внимание на значения **Transform** в области **Inspector** для обоих объектов (**ARCamera** и **Image target**).



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.

Для размещения контента в **Unity 3D** воспользуемся функционалом **Unity 3D**. Для этого в системе существует большое количество собственных объектов. В частности, 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, это – **Quad**.

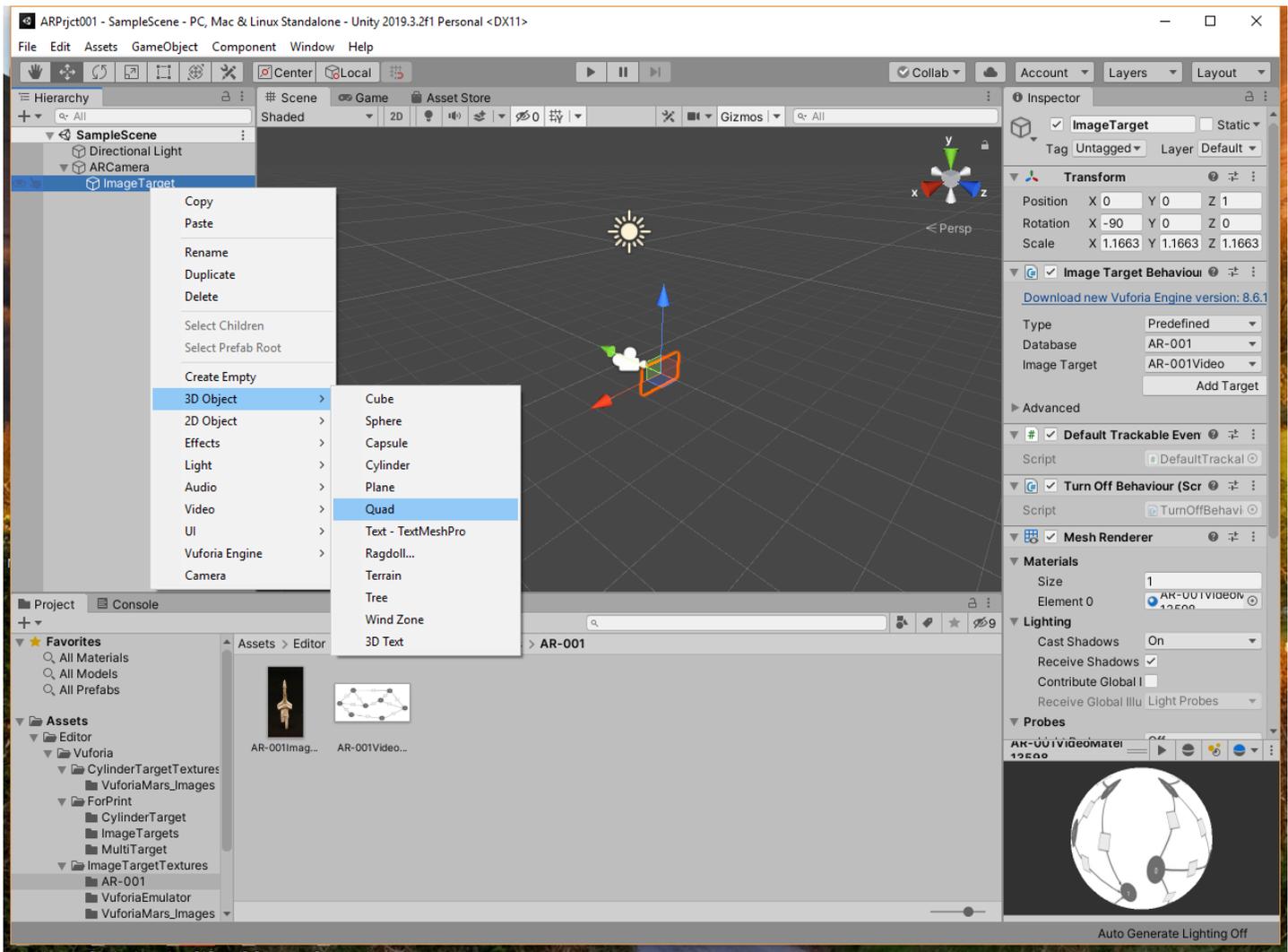
Т.о. в **Unity 3D** связываем **Image Target** с размещаемым контентом через объект (шаблон, контейнер) **Quad**. Для такого связывания в иерархии (**Hierarchy**) **Quad** должен располагаться под **Image Target**.

Последовательность операций для получения результата:

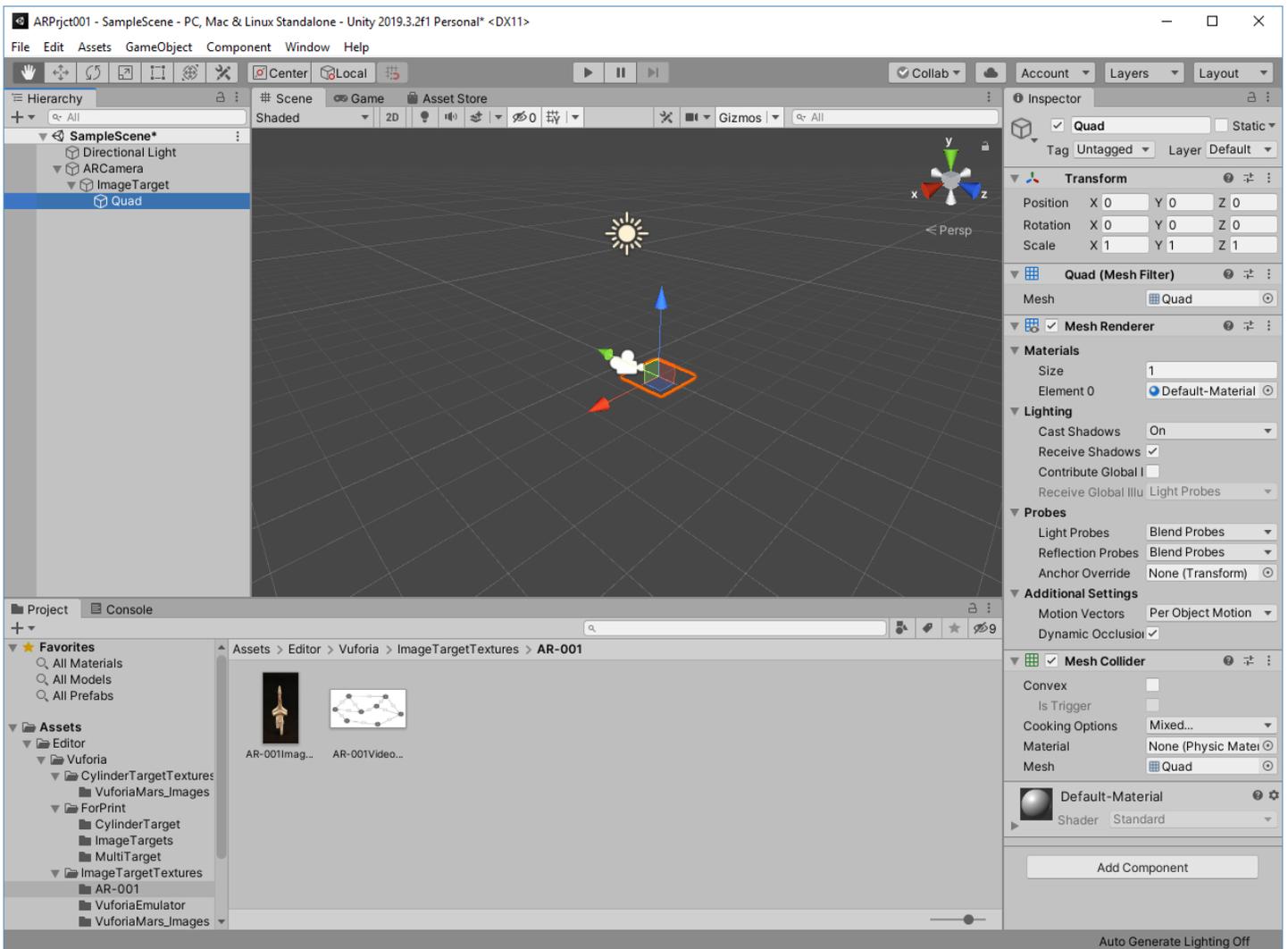
- В **Hierarchy** выбираем **Image Target**;

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

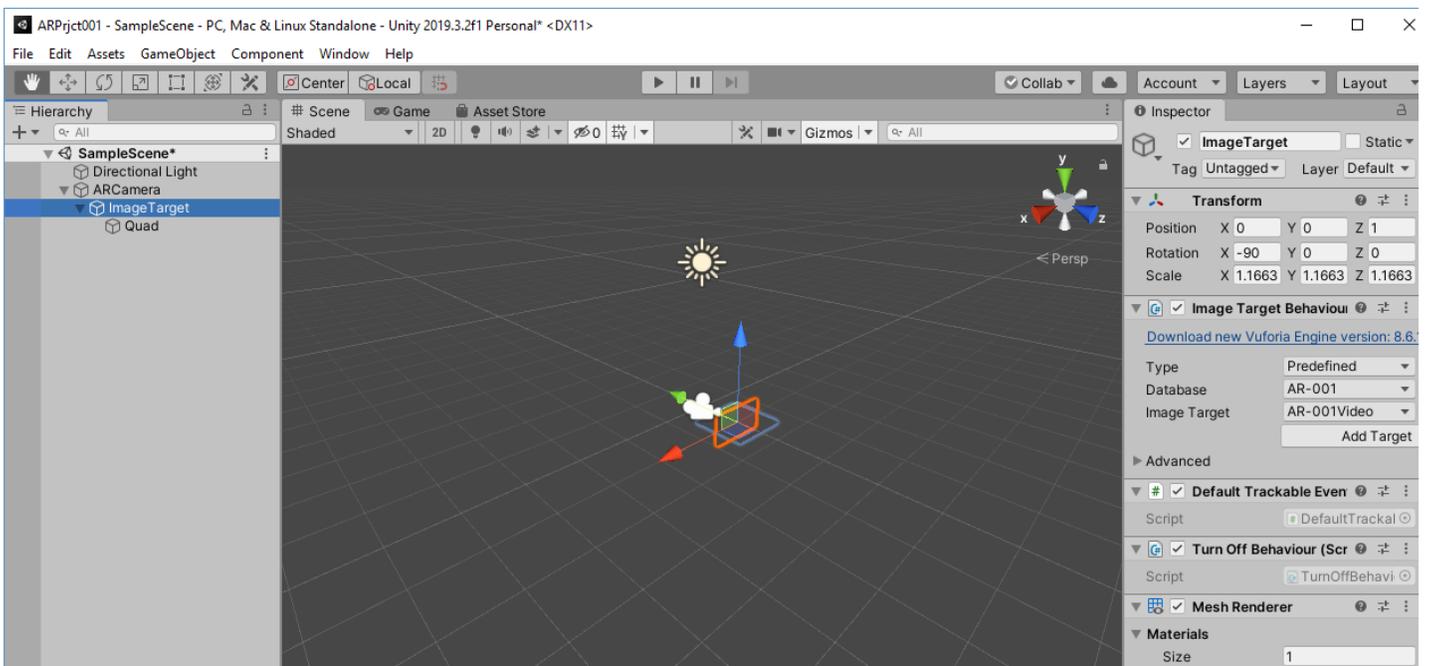
Unity 3D: 3D Object → Quad, контейнер 2D-объекта:



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

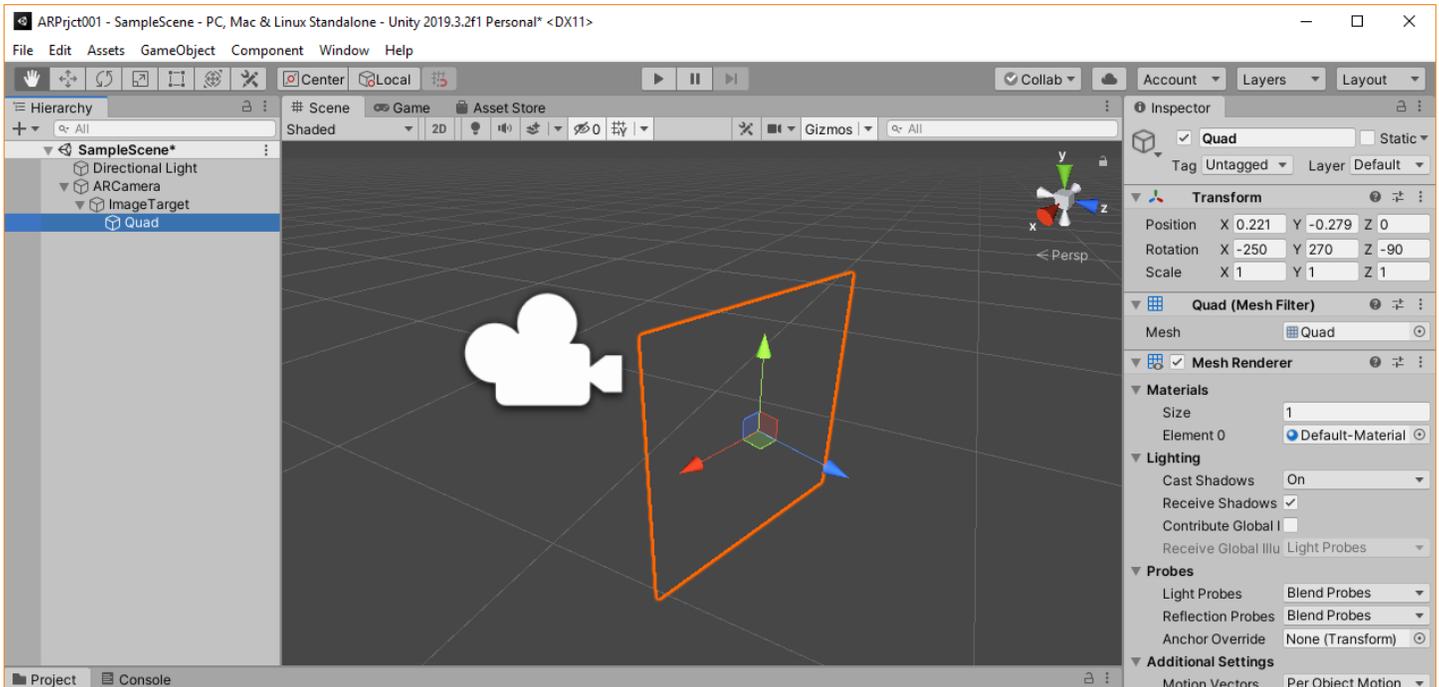


Исходное взаимное расположение таргета (выбран в **Hierarchy**) и контейнера для видеоклипа видно из нижеприведенного вида сцены:

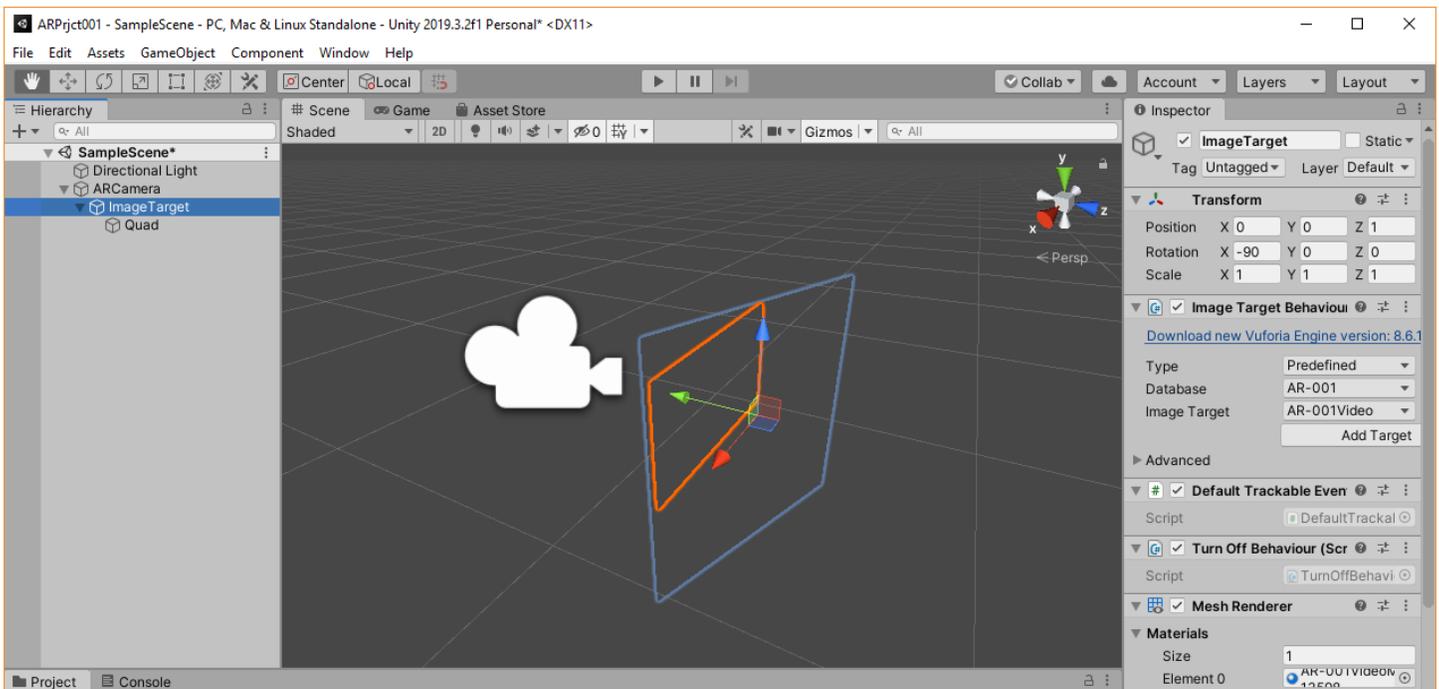


Не меняя расположения камеры и таргета, найдем наиболее удобное положение для контейнера видеоклипа – **Quad'a**. Например, экран с видео (**Quad** с видео) будет расположен на таком же расстоянии от камеры, что и таргет, но с небольшим разворотом по вертикали (ось «Y»). Вы можете выбрать расположение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

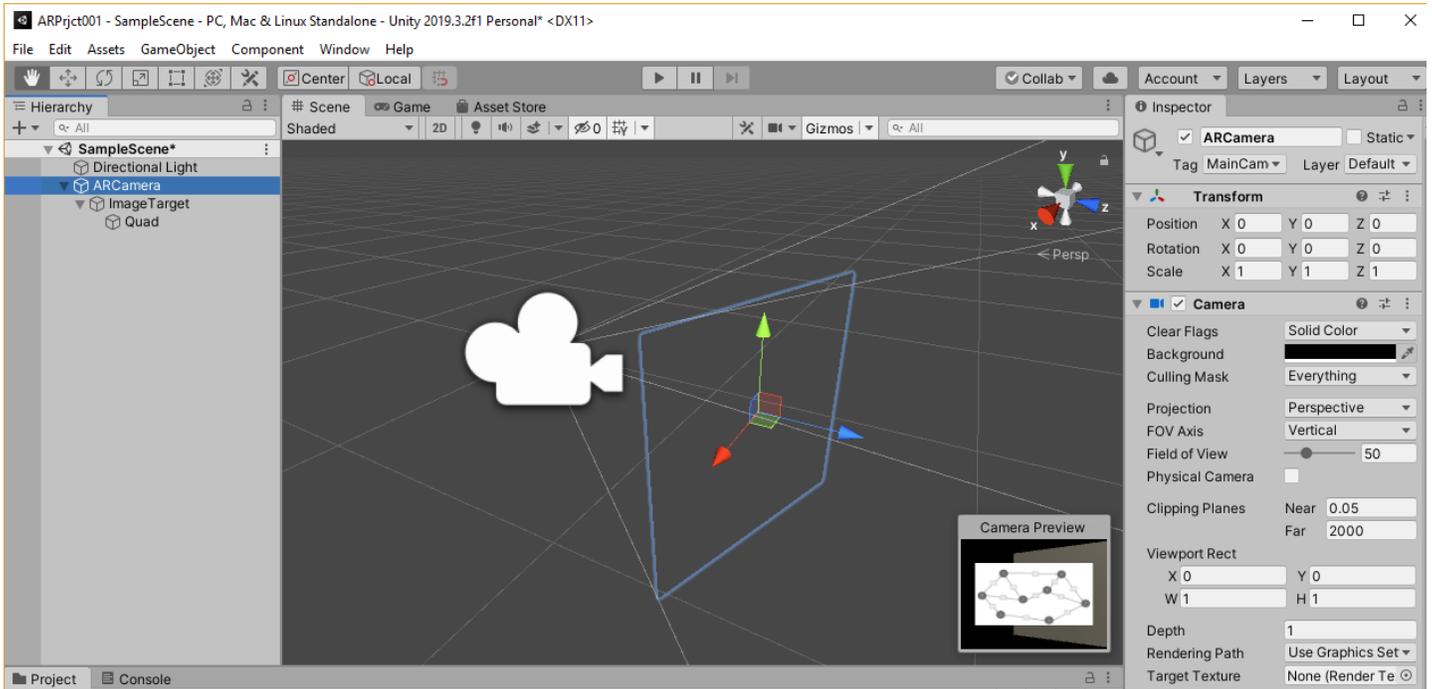
- **Quad:**



- **Image Target и Quad:**



- То, что видит ARCamera:

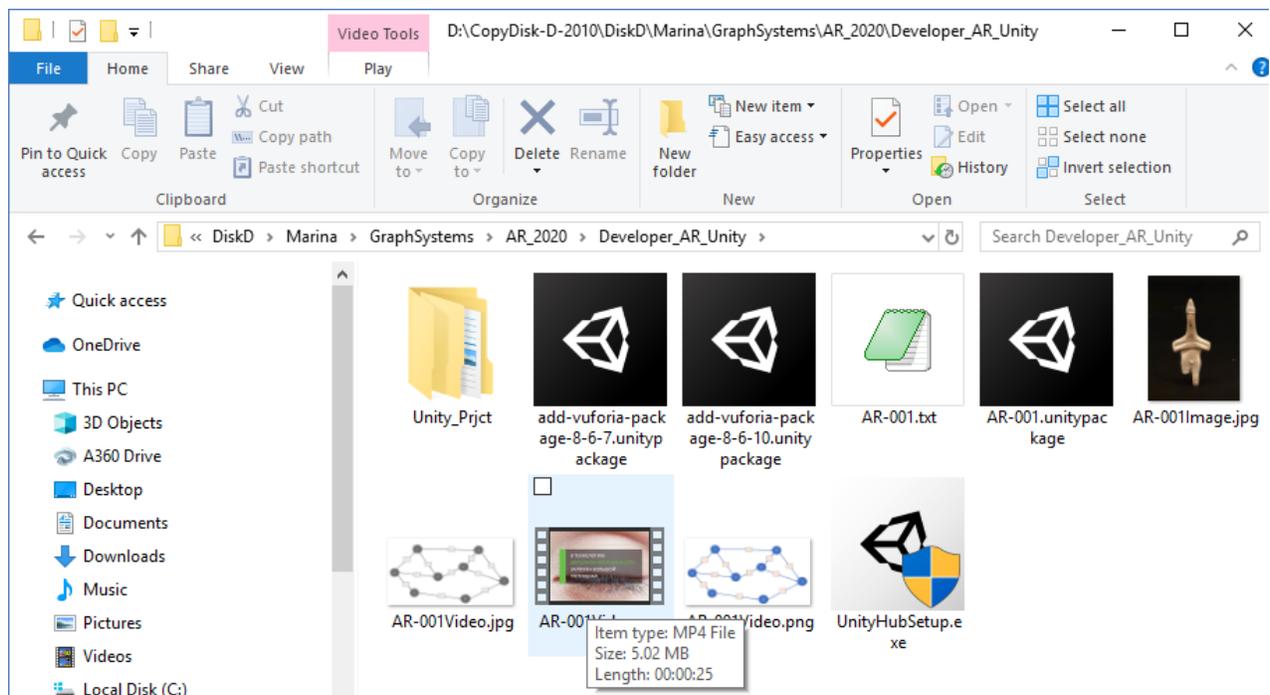


Таким образом, в сцене размещены **ARCamera**, таргет и контейнер для 2D-контента – видеоклипа.

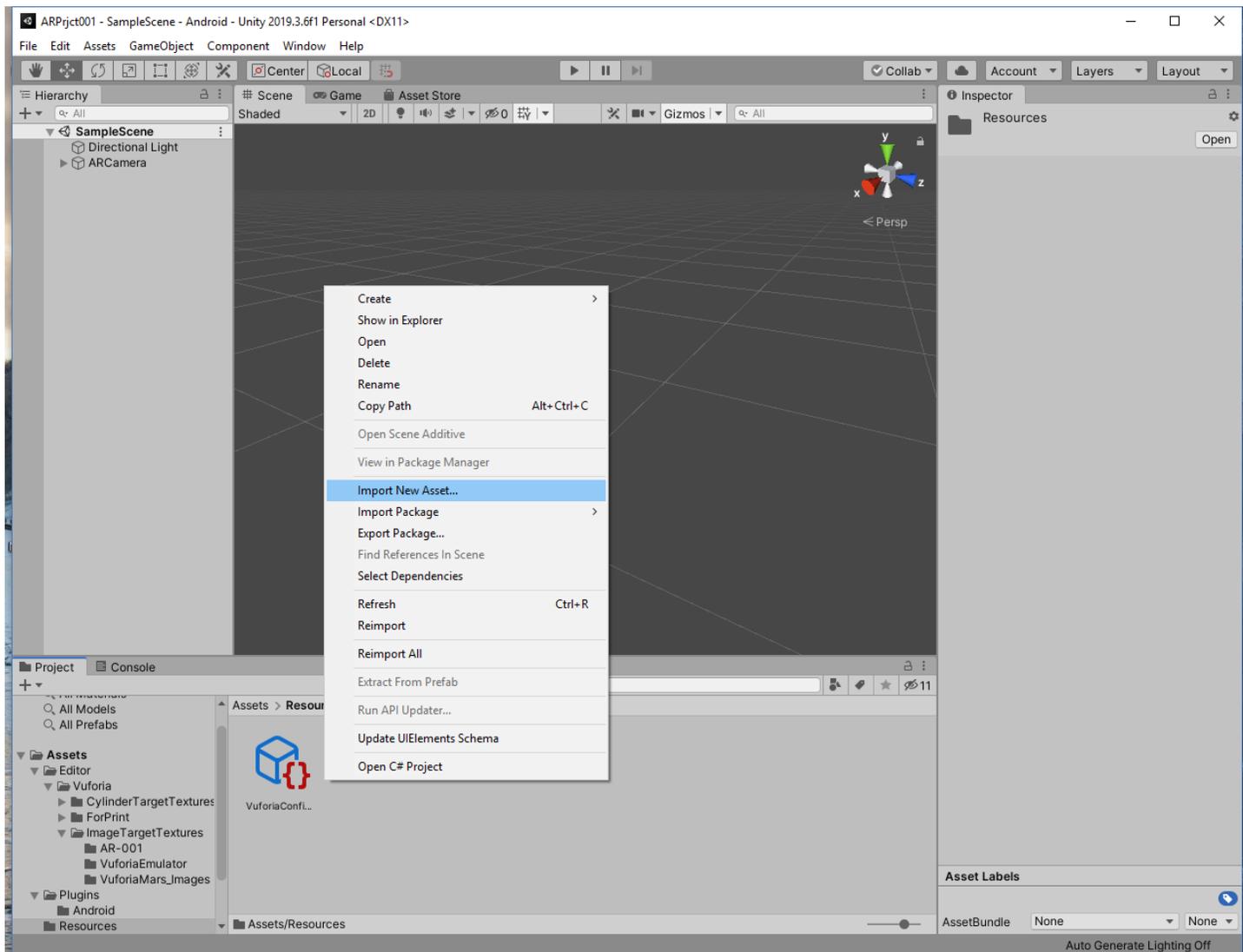
### ВАЖНО!!

Элементы контента такого типа (пользовательские видеоклипы, изображение и пр.) нашего Приложения ДР при работе с **Unity**, должны быть помещены в раздел ресурсы (**Resources**) библиотеки активов (**Assets**).

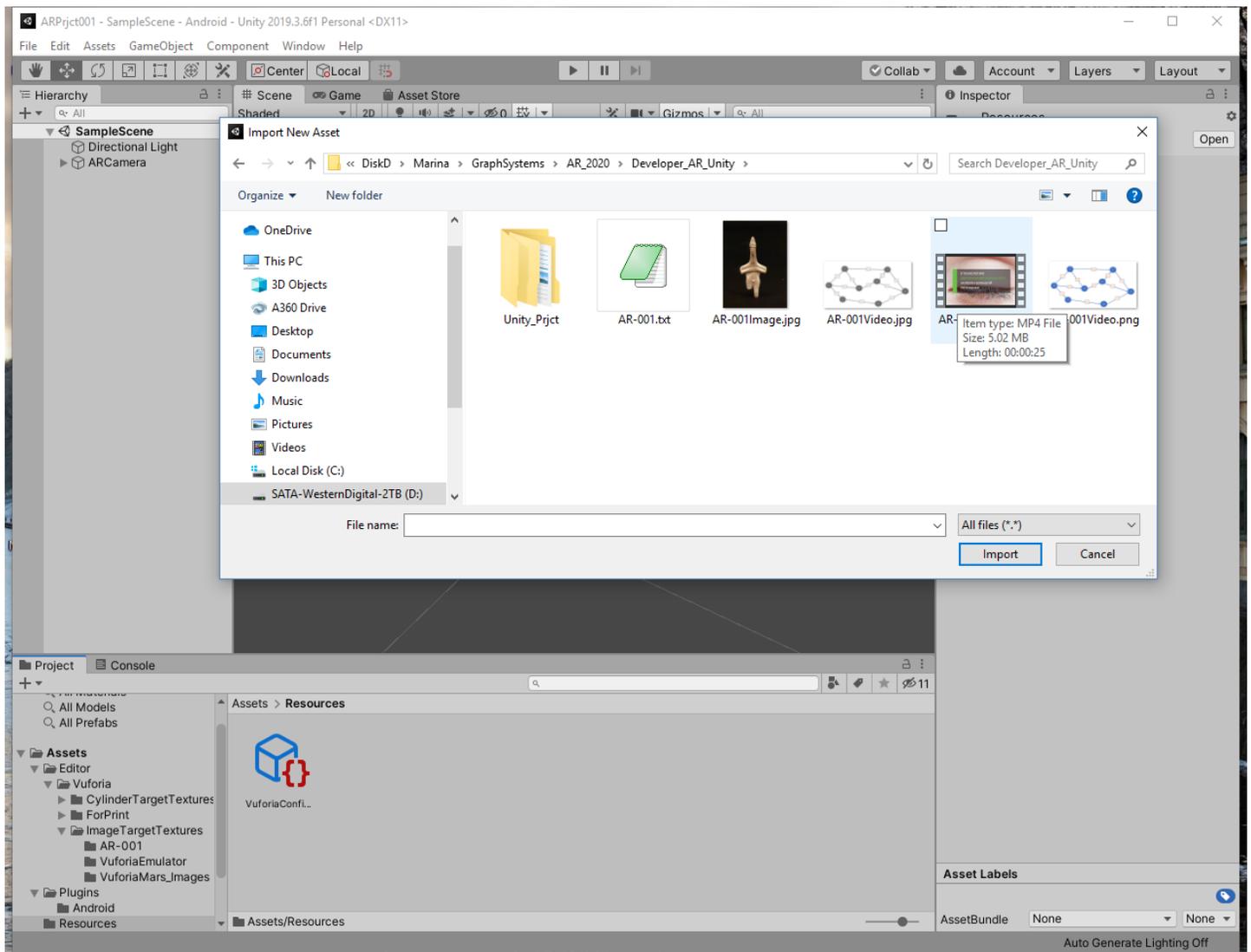
Заранее подготовим файл в формате **.mp4** для использования его в Приложении ДР: **AR-001Video.mp4**



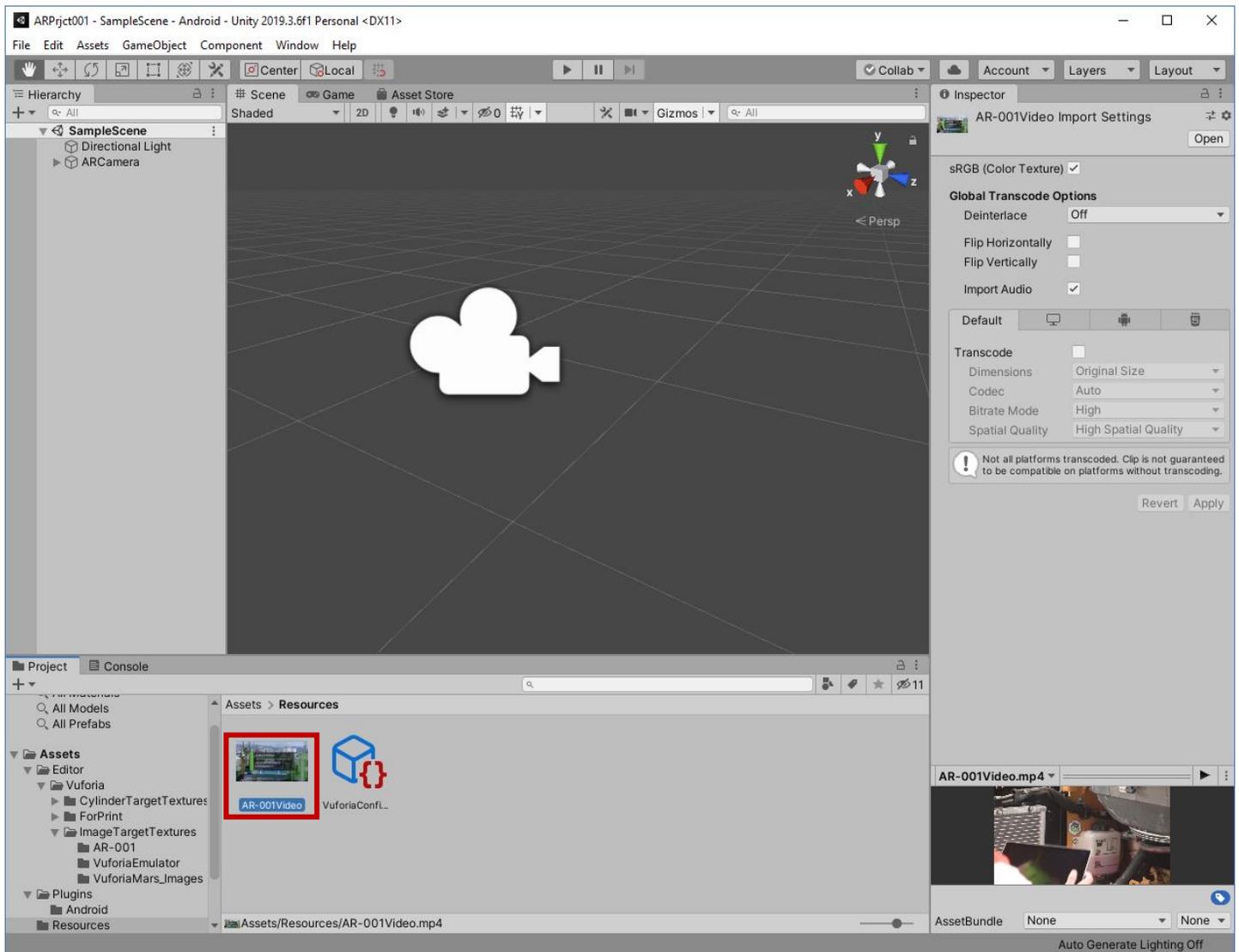
Для того, чтобы поместить наш видеоклип в подраздел **Resource** раздела **Assets** перейдем на закладку **Resource Project Window** редактора **Unity**, в ее свободной области по правой клавише мыши (**RMB**) вызываем меню, в котором выбираем позицию **Import New Asset**:



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

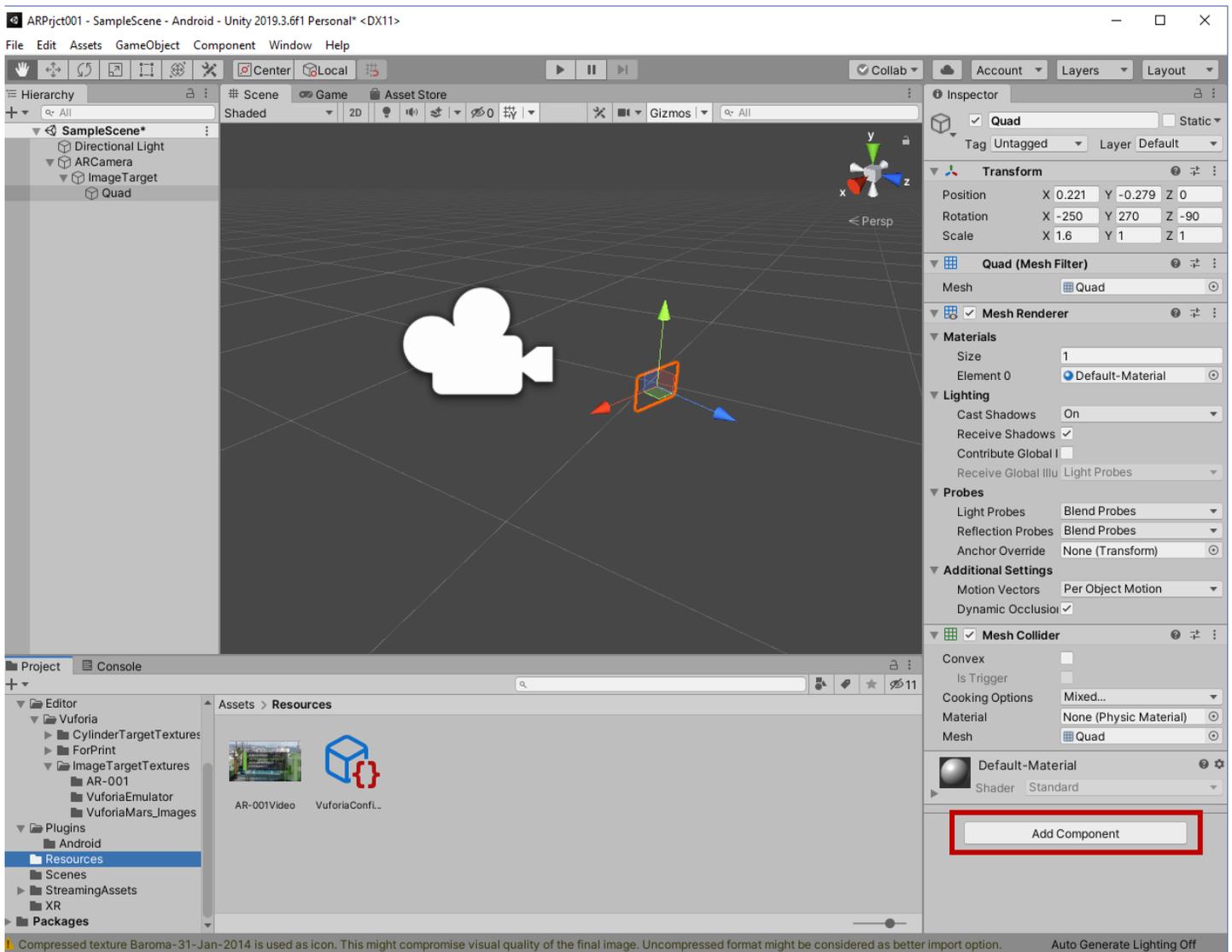


Импортируем выбранный файл, в результате чего он появляется в разделе ресурсы (**Resources**) библиотеки активов (**Assets**) в области **Project Window** редактора **Unity**:

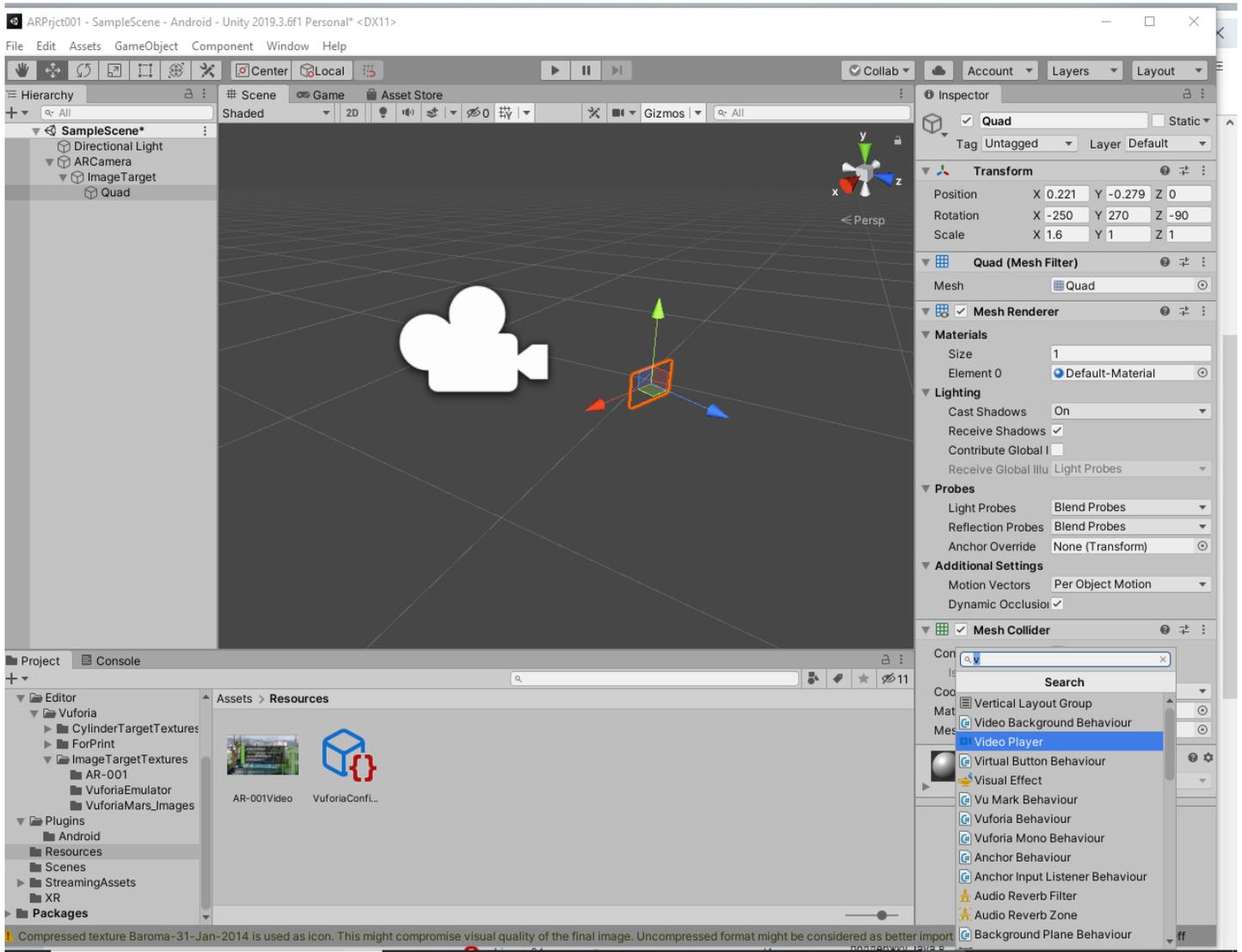


Следует иметь в виду, что по логике работы **Unity 3D** контентом является как визуализируемый объект (в нашем случае видеоклип), так и «устройство», этот объект визуализирующее. У нас это – заранее подготовленный файл с видеоклипом - **AR-001Video.mp4**, и компонент **Unity 3D**, отвечающий за визуализацию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

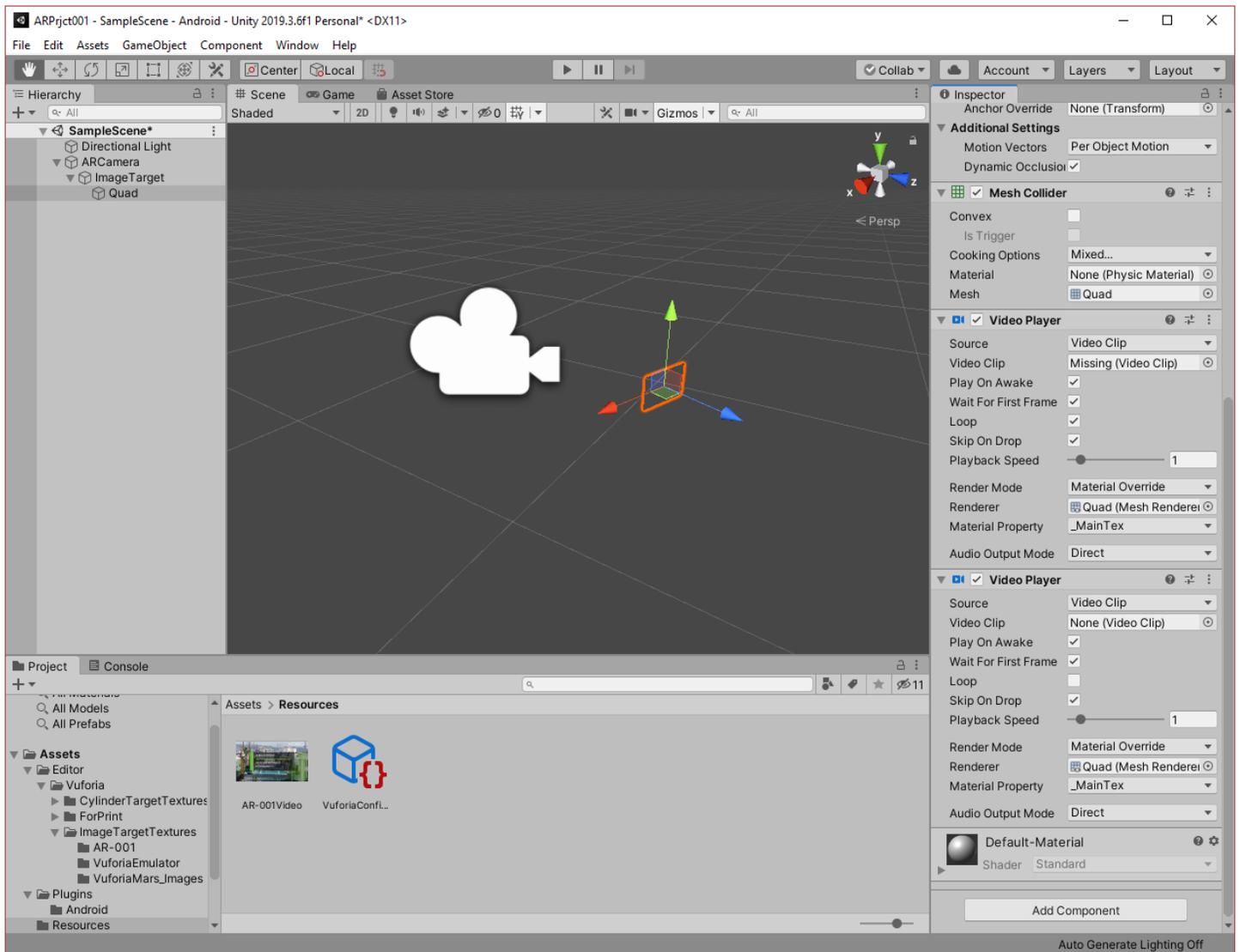
Начинаем наполнять контентом контейнер **Quad**. В первую очередь помещаем в контейнер компонент **Video Player**. Для этого в области редактора **Hierarchy** выбираем **Quad**, а в появившемся **Inspector'e** – кнопку **Add Component**:



В выпадающем списке находим компонент **Video Player** (в более старших релизах поиск **Video Player**'а может быть двухступенчатым **Add Component**→**Video**→**Video Player**):

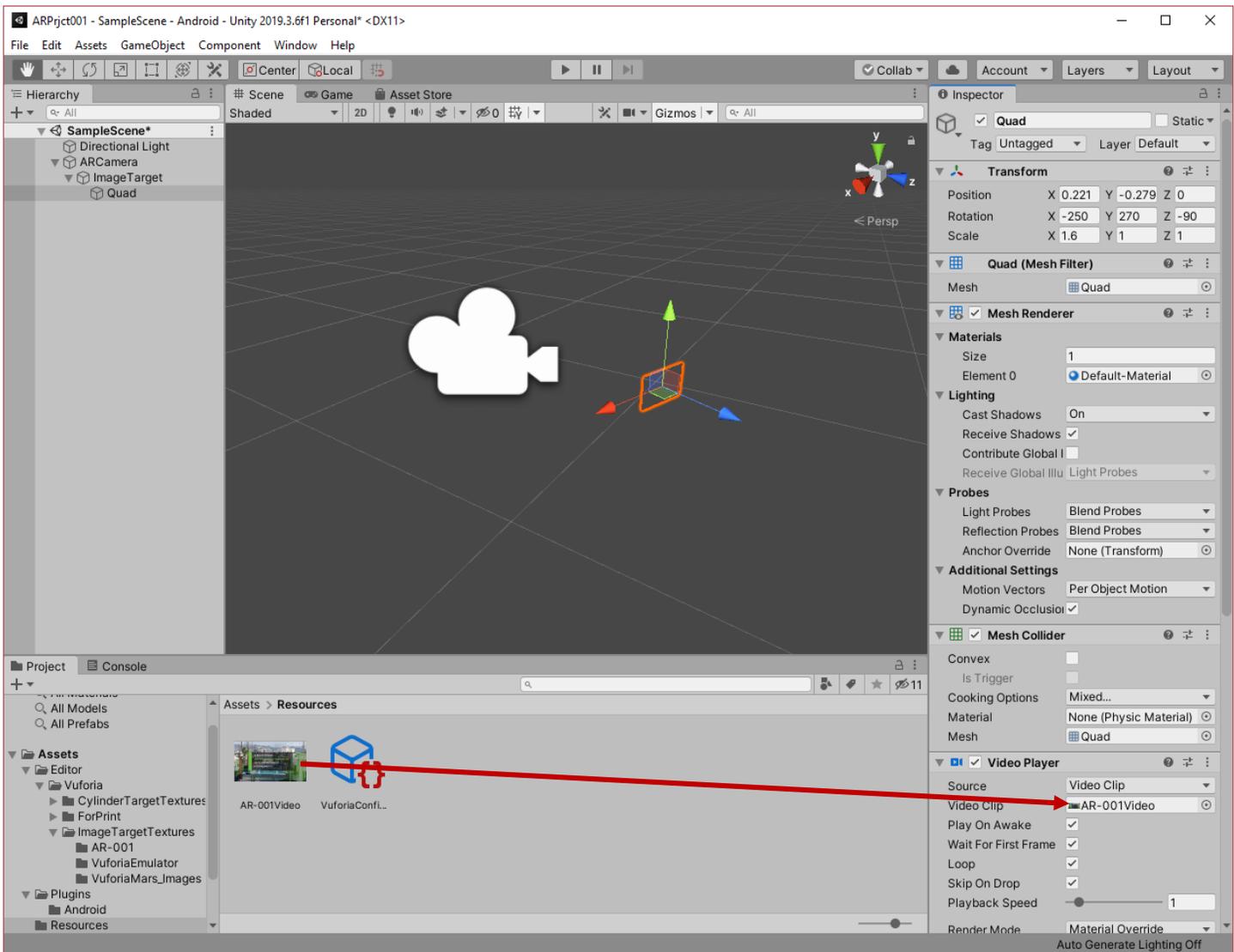


В результате в **Inspector**'е для **Quad** появится область настройки видеоплеера:



**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:



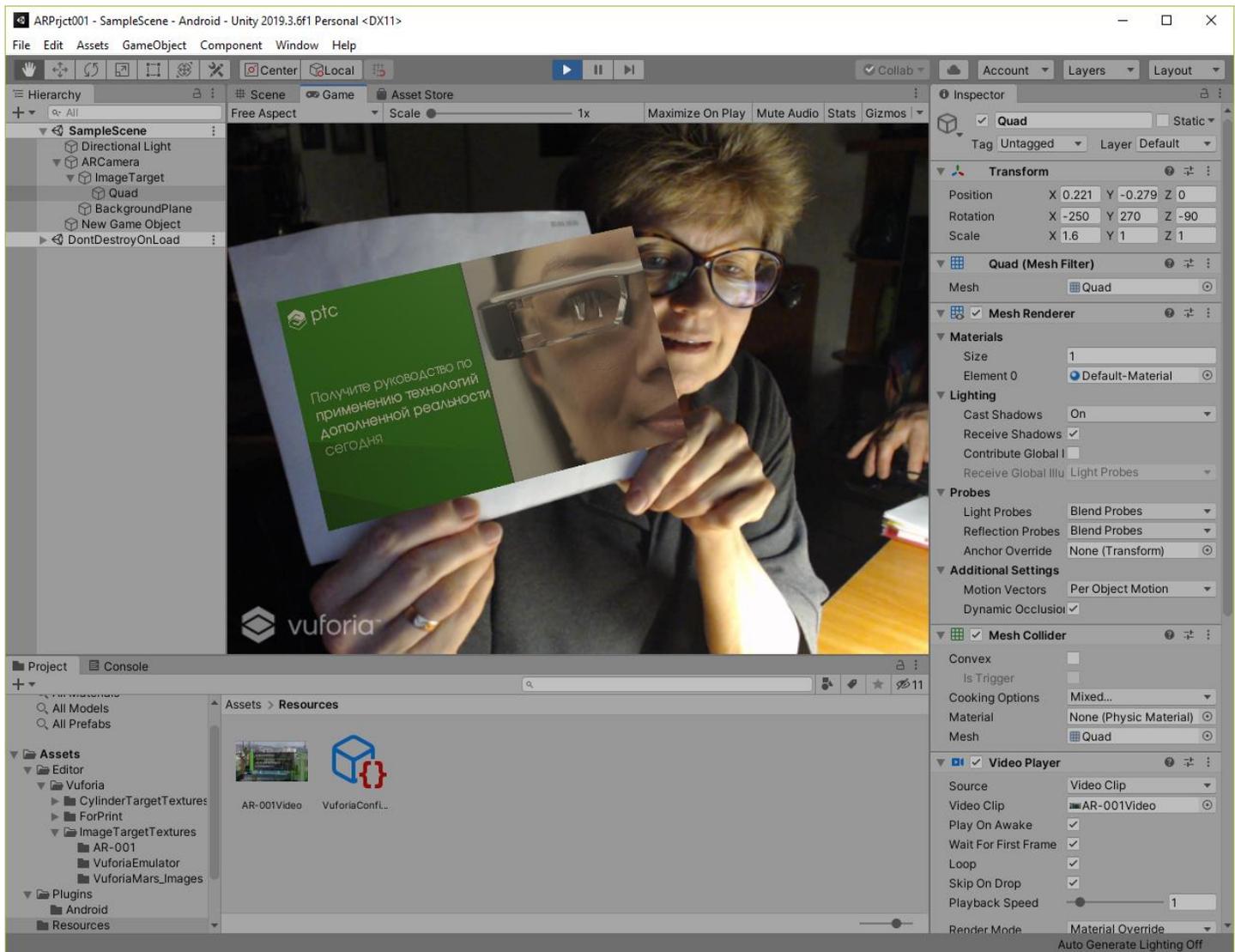
**ВАЖНО!!** Таким образом в **Unity 3D** сформирована сцена Приложения ДР:

- 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** в **Toolbar**:

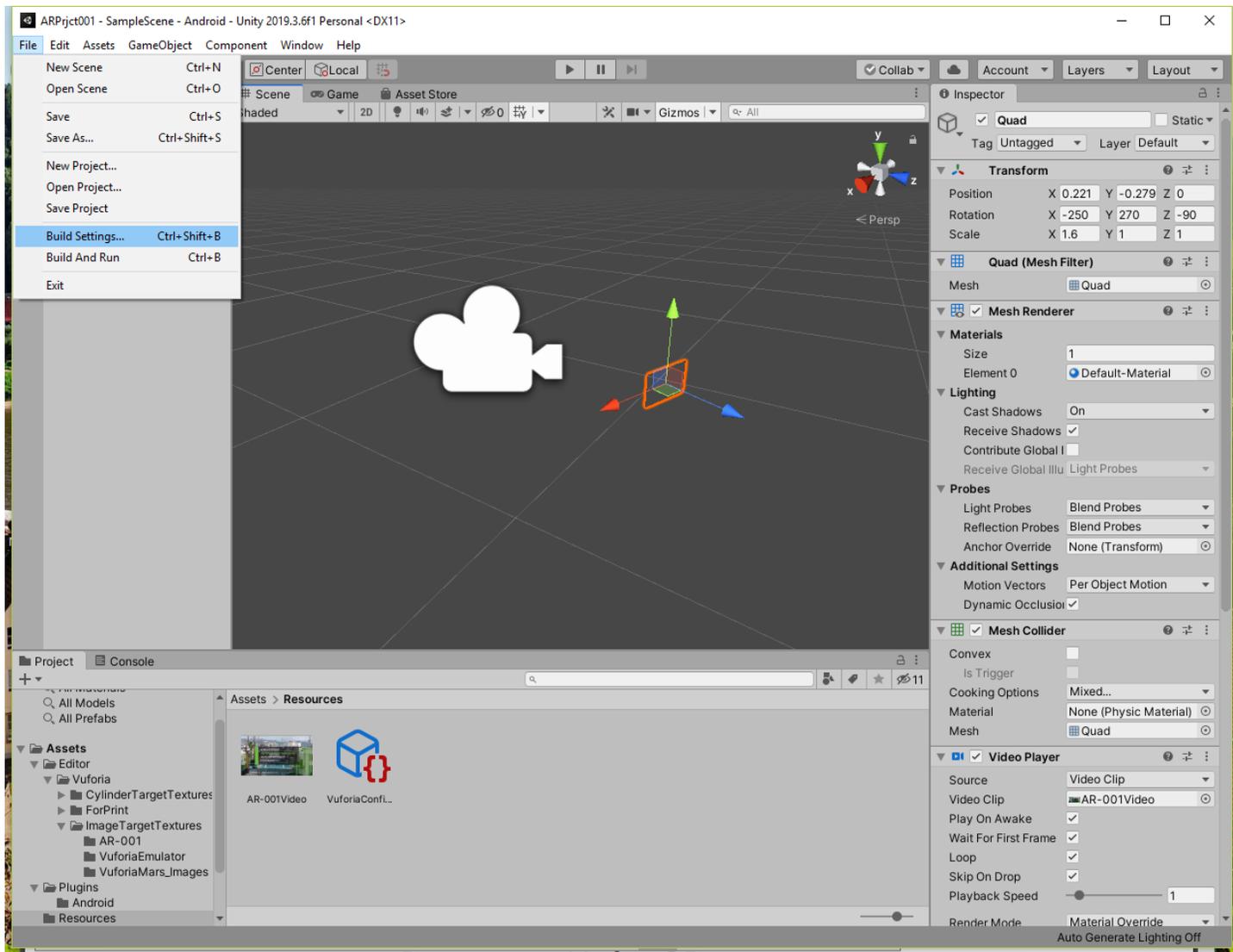


Разработка сцены закончена. Здесь можно сохранить полученный результат – сцену - не выходя из **Unity 3D**. Это может пригодиться в дальнейшем при доработке сцены, при прерывании сеанса работы в **Unity 3D** и т.д. Для сохранения сцены выполнить: **File→Save** или **File→Save as**. Сохранение производится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

## 5. Создание файла **.apk** для загрузки Приложения ДР на **Android-MY**.

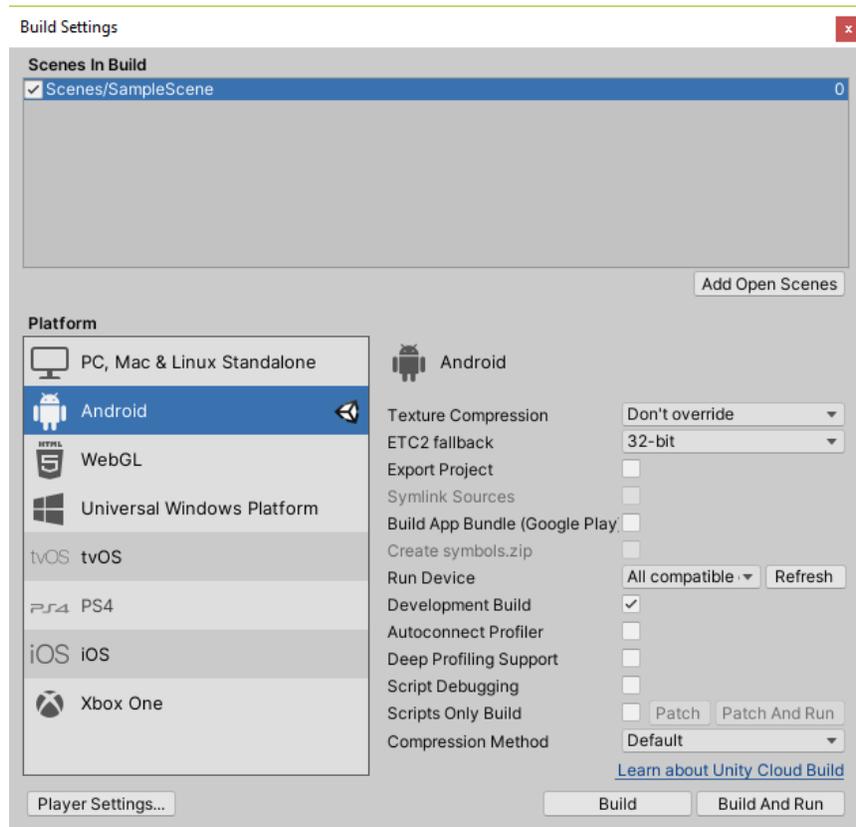
Разрабатываемое приложение должно будет использовать камеру конкретного **Android**-устройства. **Unity 3D** в нашем проекте только подготавливает универсальный драйвер **AR Camer’ы** для типового **Android**-устройства. Все особенности конкретных, передовых **Android**-устройств (стерео, 4K, и т.д.)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

Для создания файла **.apk** включаем в редакторе **Unity 3D** режим **Build: File→Build Settings**, который использует загруженный нами заранее **Android SDK** (см. **Приложение 1**):

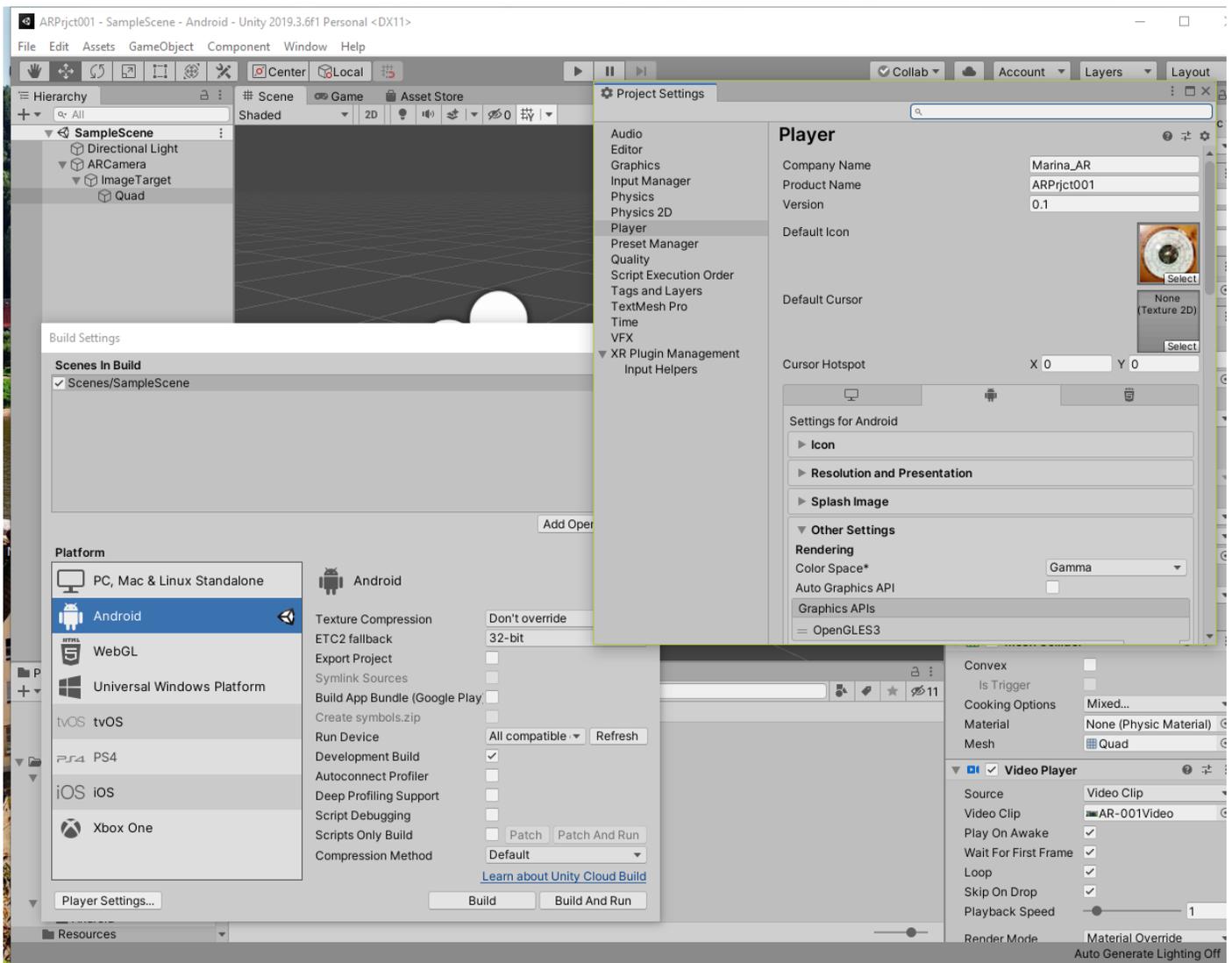


Все предварительные настройки – выбор сцены, настройки ОС устройства – осуществляются в открывшемся окне **Build Settings**.

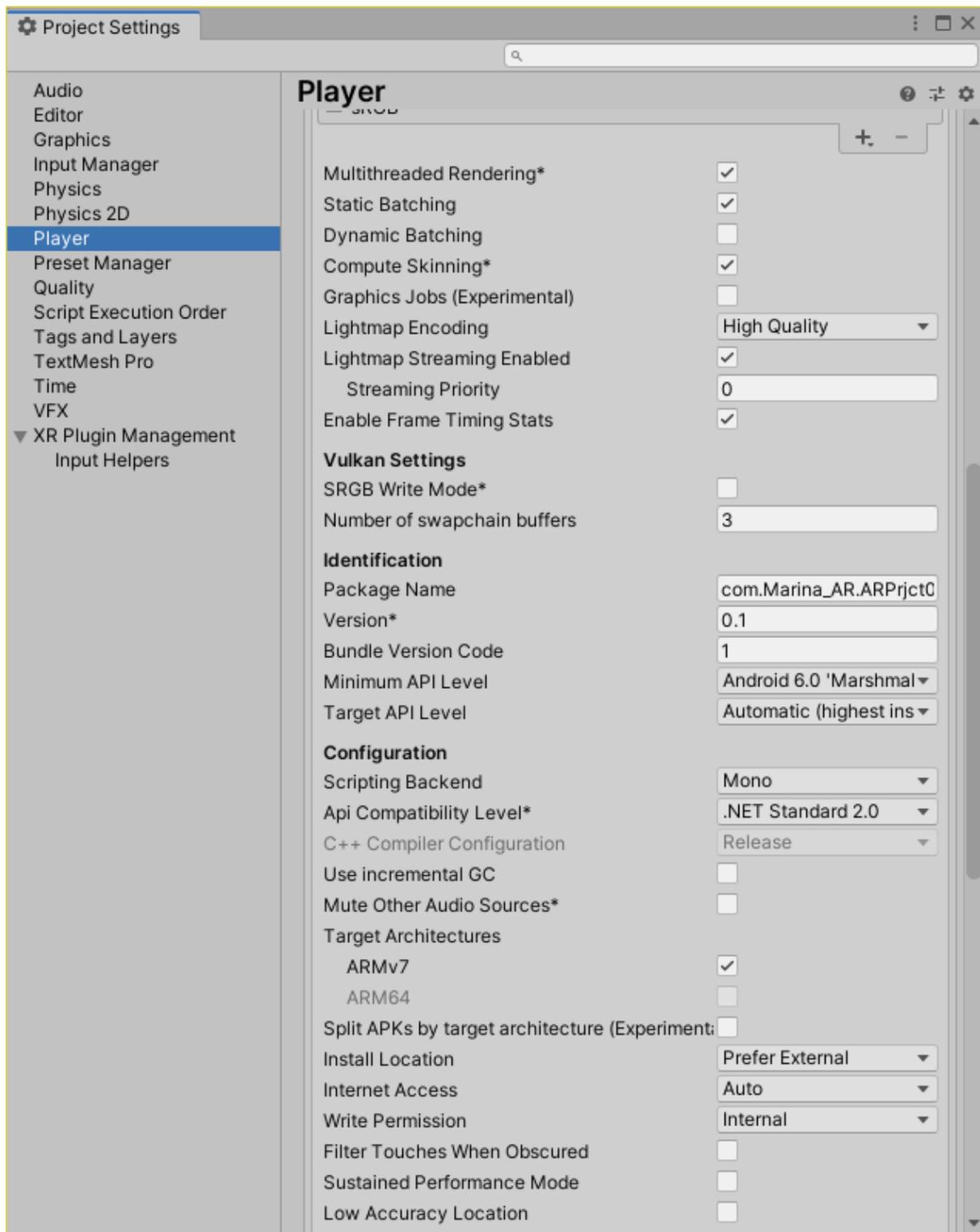
Для выбора сцены нужно использовать кнопку **Add Open Scene**, пометив в случае необходимости нужную из списка. Настройку на ОС **MY (Android)** осуществляем в меню **Platform → Switch Platform**:



Теперь переходим к настройке работы оборудования в данной ОС → клавиша **Player Settings**.  
В результате в области **Inspector** появляются настройки для плеера:



Выполняем необходимые настройки.



В поле **Package Name** введите имя вашего пакета для размещения его в облаке **Vuforia**. Имя в нашем случае формируется по правилу: **com.<Company Name>.<Product Name>**. 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**.

Project Settings

Audio  
Editor  
Graphics  
Input Manager  
Physics  
Physics 2D  
**Player**  
Preset Manager  
Quality  
Script Execution Order  
Tags and Layers  
TextMesh Pro  
Time  
VFX  
XR Plugin Management  
Input Helpers

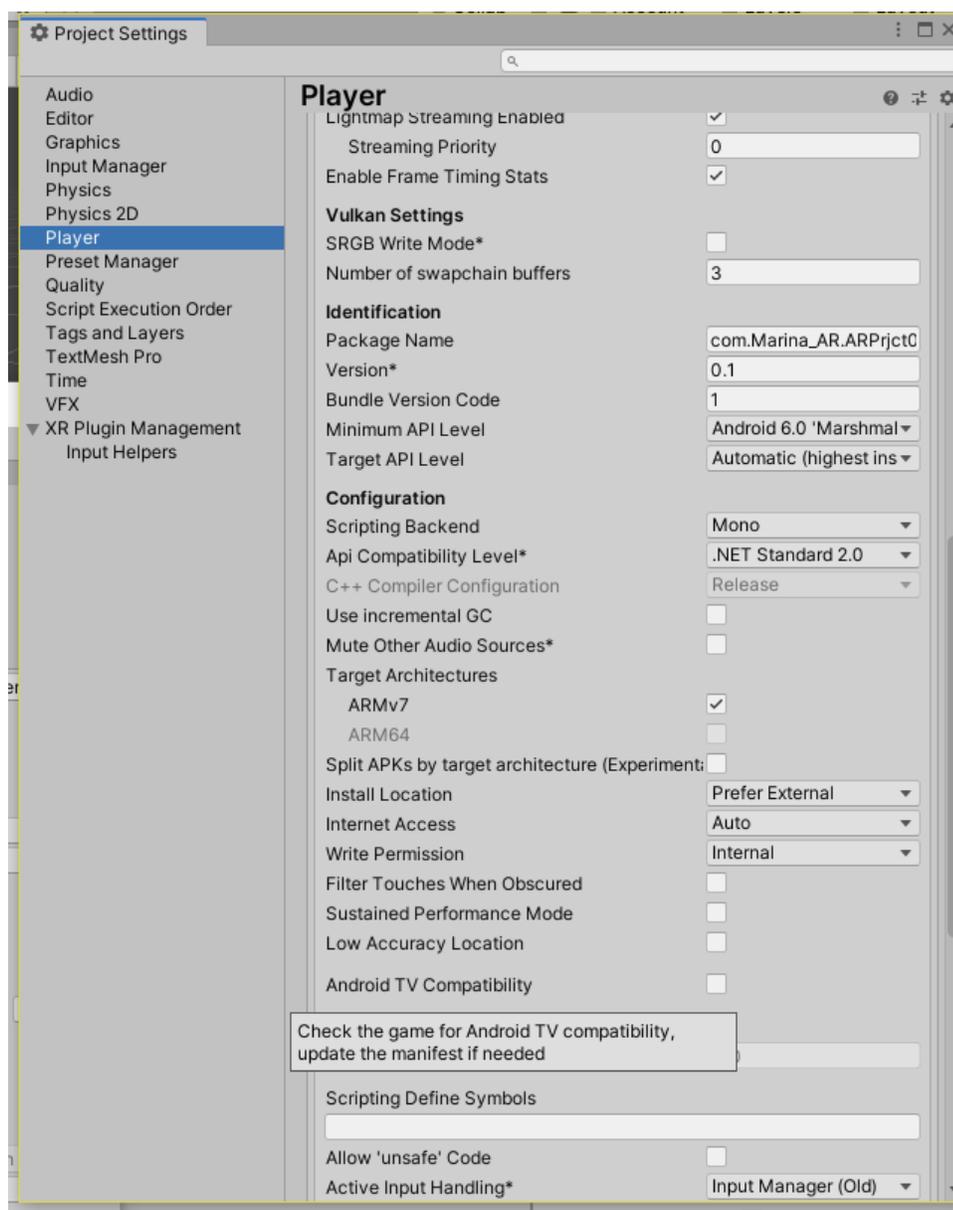
## Player

Company Name: Marina\_AR  
Product Name: ARPrjct001  
Version: 0.1  
Default Icon:  Select  
Default Cursor: None (Texture 2D) Select  
Cursor Hotspot: X 0 Y 0

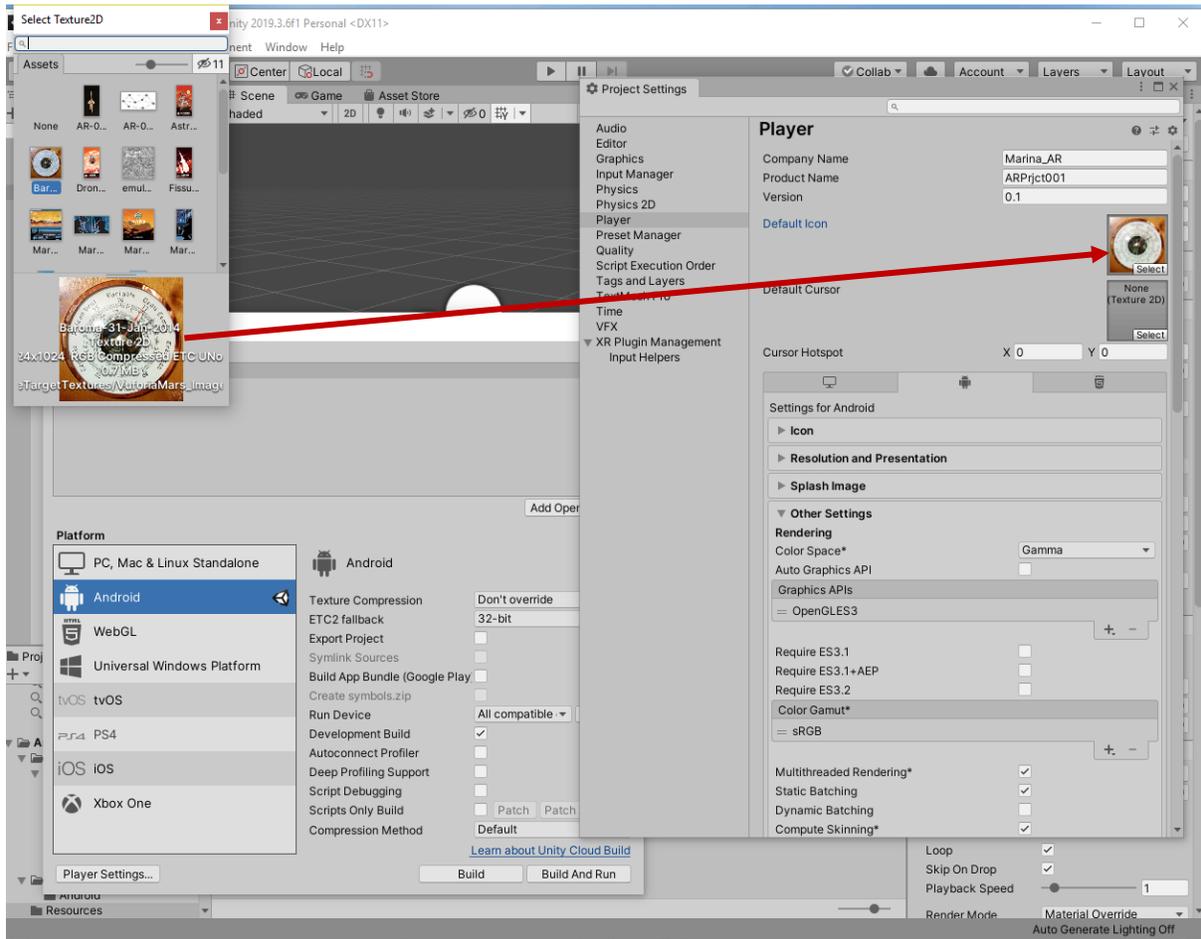
Settings for Android

- ▶ Icon
- ▶ Resolution and Presentation
- ▶ Splash Image
- ▼ Other Settings
  - Rendering**
  - Color Space\*: Gamma
  - Auto Graphics API:
  - Graphics APIs
    - OpenGL ES3
  - Require ES3.1:
  - Require ES3.1+AEP:
  - Require ES3.2:
  - Color Gamut\*
    - sRGB
  - Multithreaded Rendering\*:
  - Static Batching:
  - Dynamic Batching:
  - Compute Skinning\*:

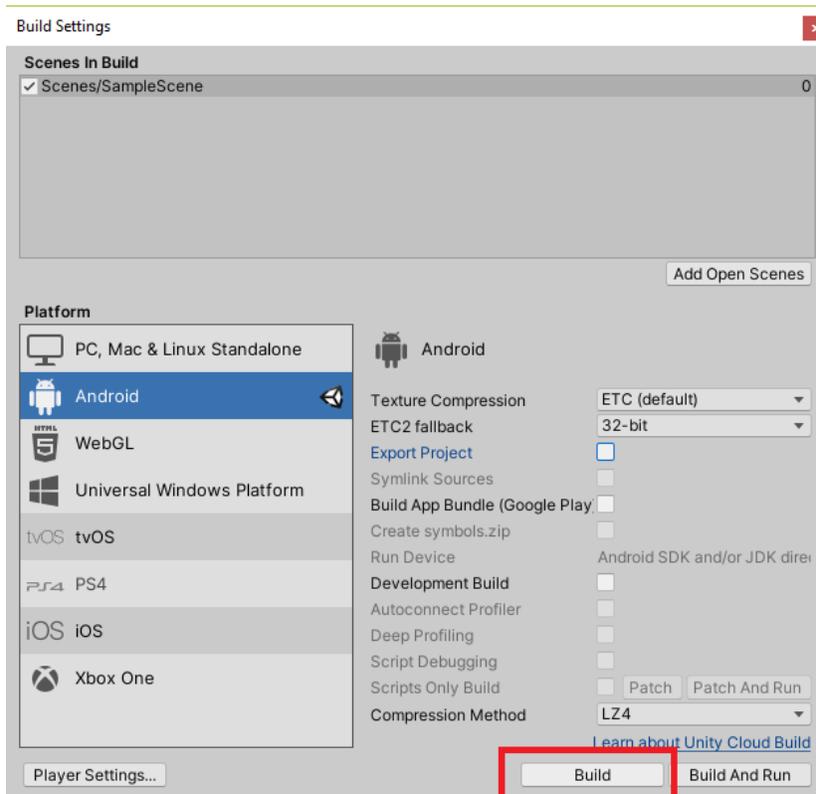
В подразделе **Configuration** обратите внимание на состояние переключателей – необходимо обязательно снять галочки в переключателях **Android TV Compatibility**:



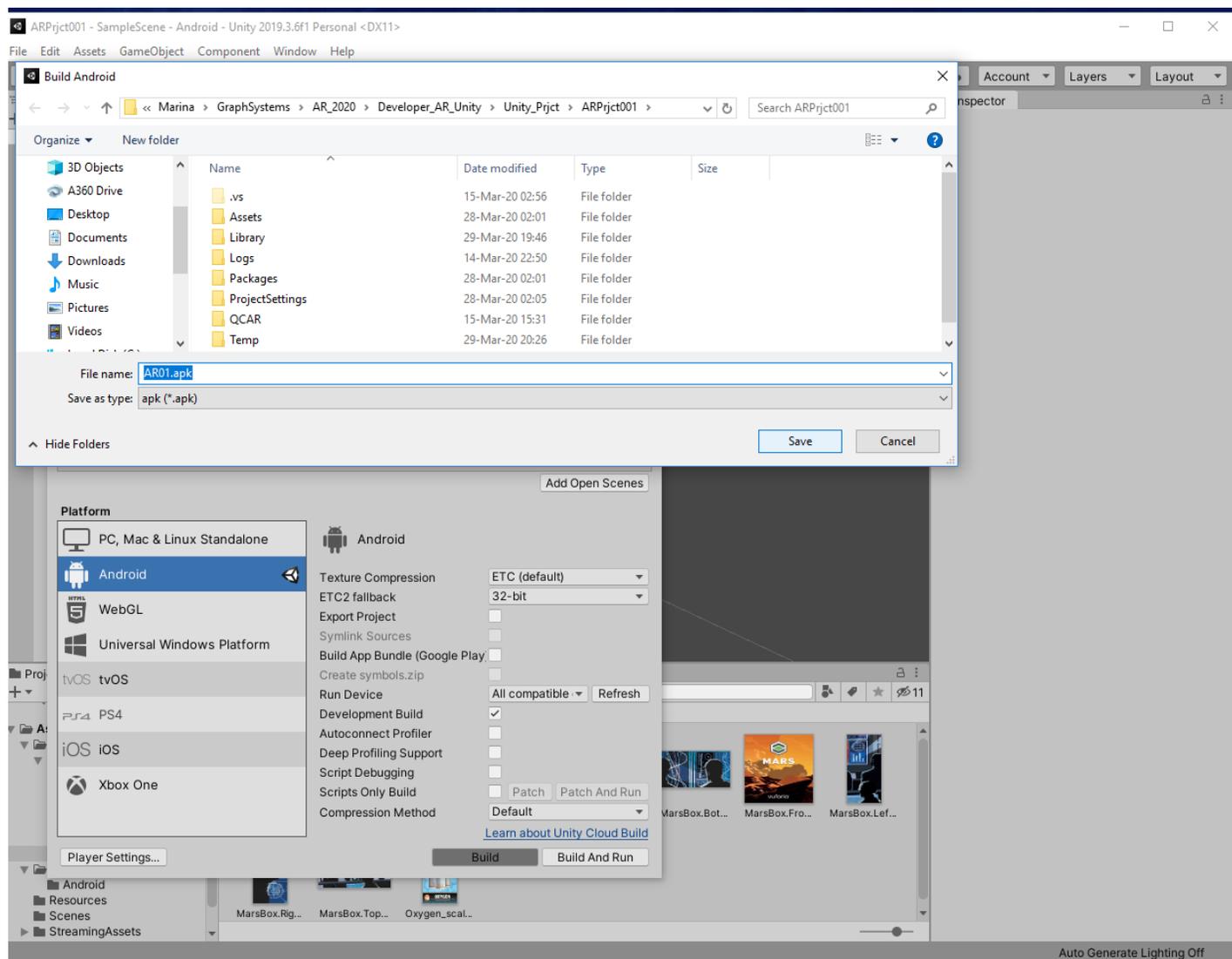
В Верхней части **Inspector'a** заполните поле для пиктограммы вашего **AR-Приложения**. Обратите внимание, что для выполнения этой операции можно использовать операцию **drag-n-drop** из **Assets→Resources** для тех объектов, которые ранее были загружены в **Resources** в разделе **Project Windows**. Или, выбрав опцию **Select** у иконки по умолчанию (**Default Icon**) в настройках плеера, также операцией **drag-n-drop** перетащить желаемое изображение для иконки из открывшегося окна **Select Texture 2D** в поле для иконки **Default Icon**. О пиктограммах надо позаботиться заранее. Если поле для пиктограммы останется незаполненным, то для вашего Приложения ДР пиктограммой всегда будет логотип **Unity**.



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



Сохраняем собранный файл **.apk** в локальной файловой системе разработчика:



Нажимаем **Save**.

Файл **.apk** разработанного в данном описании **AR** - Приложения сохранен. Теперь его осталось загрузить (и далее – установить) в **Android** - МУ любым известным вам способом.

Видео разработанного в данной ЛР Приложения ДР доступно по ссылке:

<https://youtu.be/2opzDQuNU0Y>

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!