

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 3D-Модели из открытой библиотеки моделей.

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

Введение.

Работа по созданию приложений ДР заключается в заведении проекта и объектов проекта 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 3D-объекта (например, 3D-модели) будет осуществляться через определяемую в **Vuforia** метку (**Target**). Допустимые в **Vuforia** типы таргетов будут подробно рассмотрены ниже.

Следует обратить внимание, что 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** и дополнения сцены 2D-изображениями или 3D-моделями.

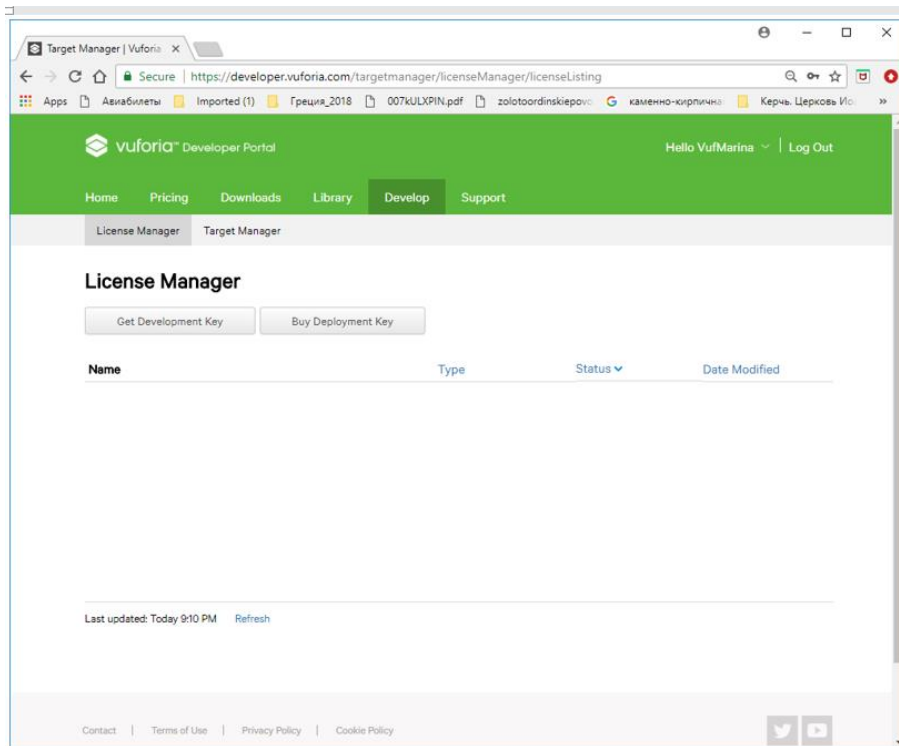
Рассмотрим типовую процедуру создания простого (игрового) **AR Приложения**.

Предлагается разработать приложение ДР для Android-устройств, в котором при наведении камеры устройства на реальную метку (изображение на бумаге) пользователь на экране устройства в области воспроизведения вместо метки увидит другое, заранее подготовленное 2D – изображение (лист инструкции по эксплуатации, пояснение, другую картинку, видеоклип и т.п.), или 3D-модель.

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**

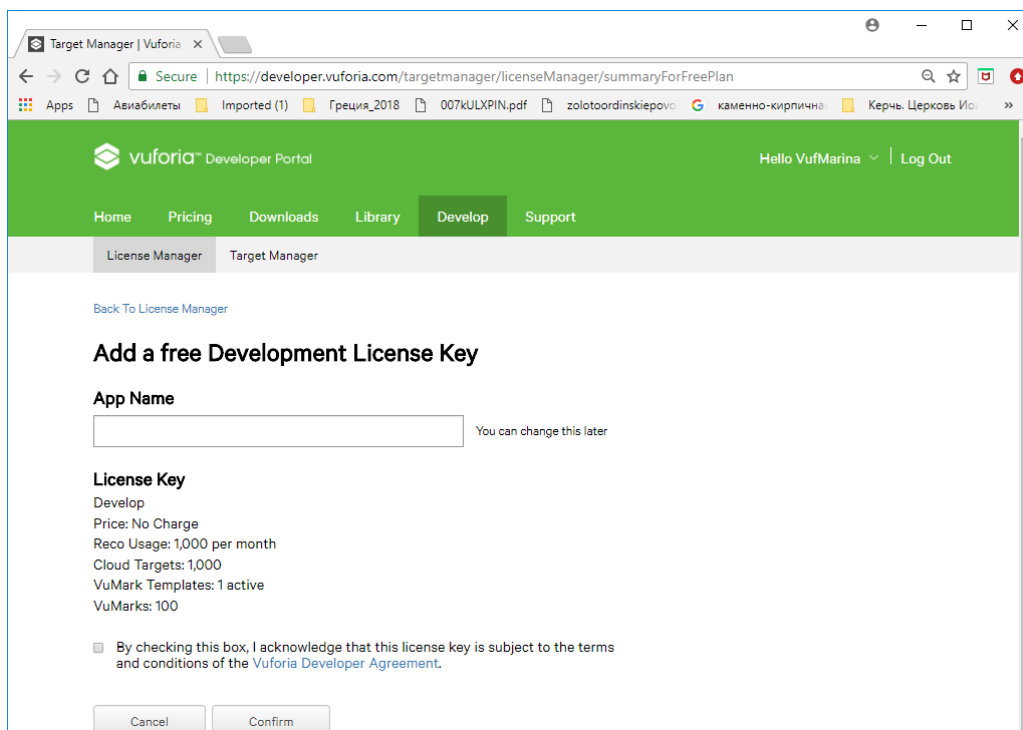
На сайте <https://developer.vuforia.com/> осуществляем вход с заведенными ранее логином/паролем. В результате получаем доступ к среде разработки **Vuforia** (облачное решение). Для ведения проекта средствами **Vuforia** необходимо выполнить две процедуры: получить **лицензионный код** на проект и сгенерировать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



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.

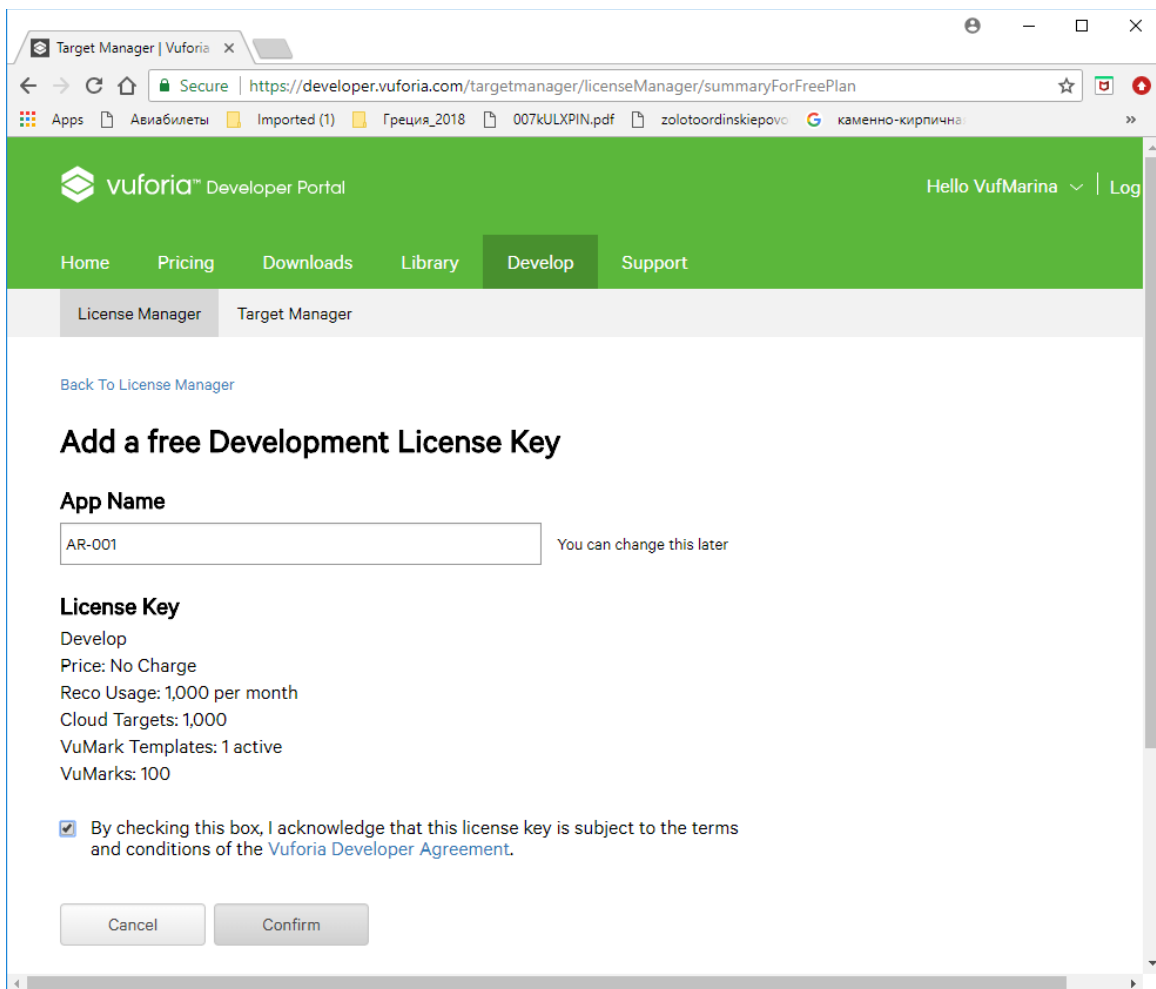
Если вы уже выполнили пункт 1. «Создание проекта и получение лицензии» и пункт 2. «Создание Базы данных таргетов», → переходите к пункту 6.

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

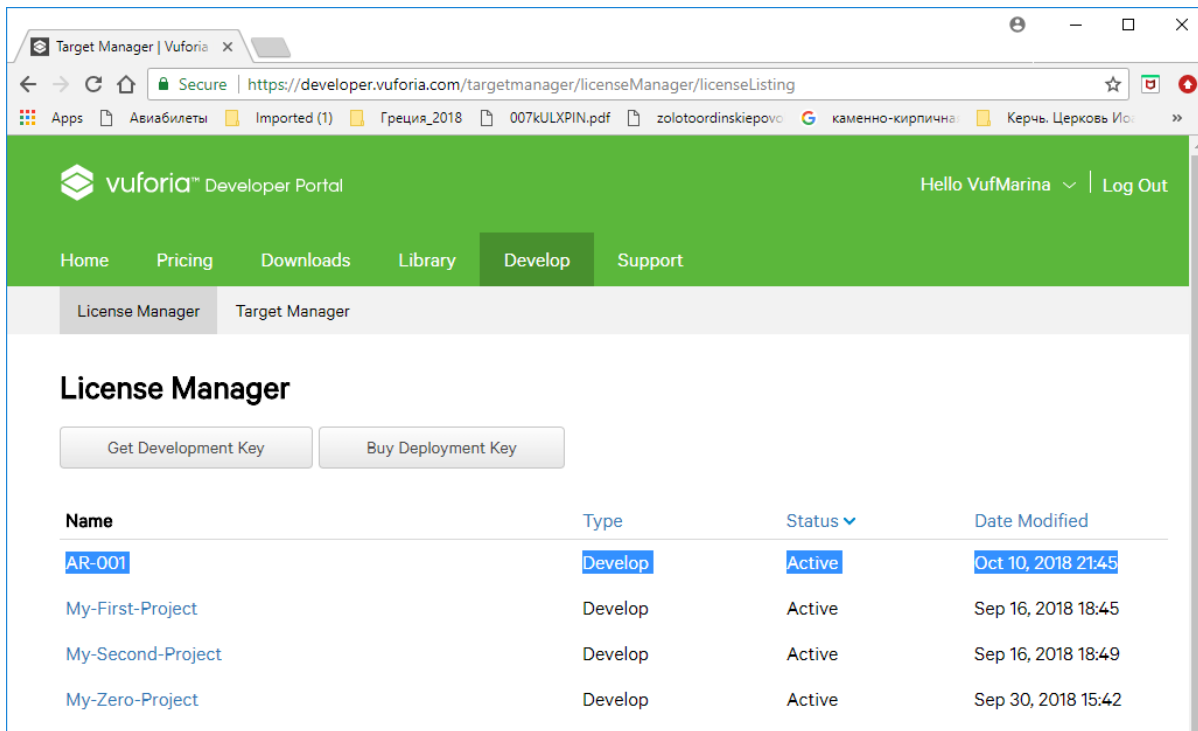


Обратите внимание на перечисленные характерные для студенческого режима использования **Vuforia** ограничения. Для нас единственно важным является: сколько актов распознавания метки (**Reco**) можно выполнить в течение месяца. 1000 для некоммерческого использования – более чем до статочно.

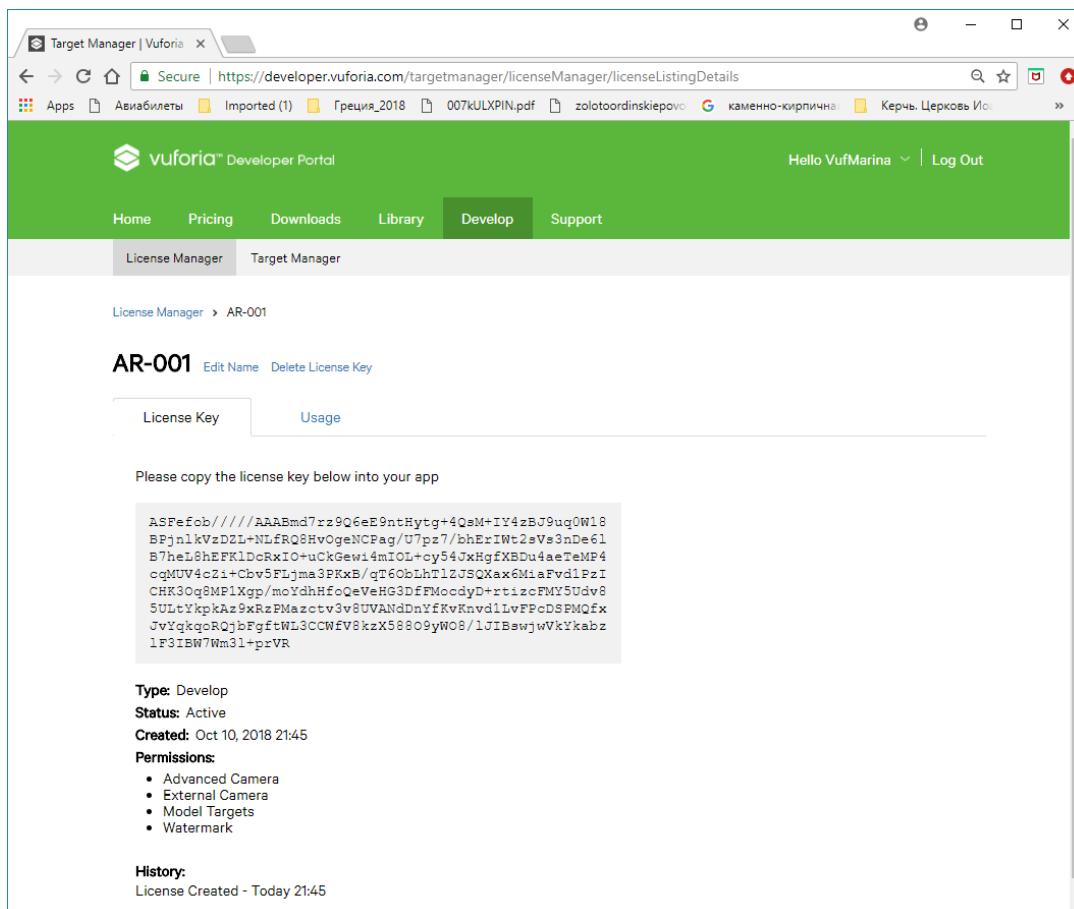
В поле **App Name** необходимо ввести уникальное имя. В рамках данной ЛР вам достаточно одной лицензии (**App Name**), одной Базы данных меток (**Target Database**), одного проекта в **Unity 3D**. Исходя из этого для всех трех объектов предлагается выбрать одно и то же имя. Например – **AR-001**. Нажмите клавишу **Confirm**.



В результате в списке лицензионного менеджера появится новая строка:

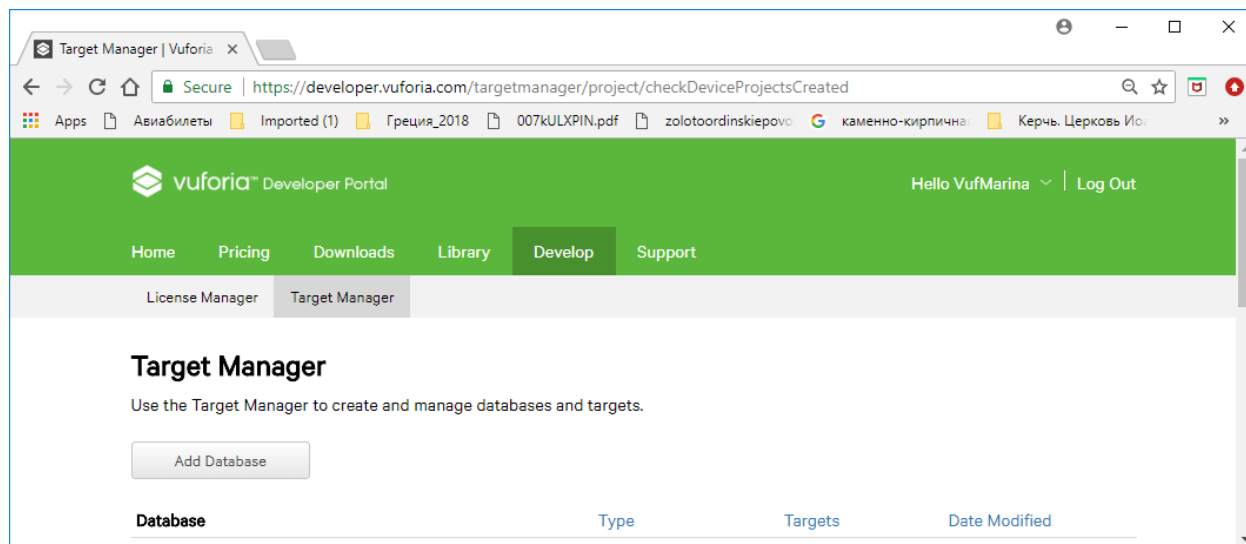


Для получения лицензионного ключа необходимо кликнуть по ссылке **AR-001** в поле **Name**. В результате вы получите **license key**:

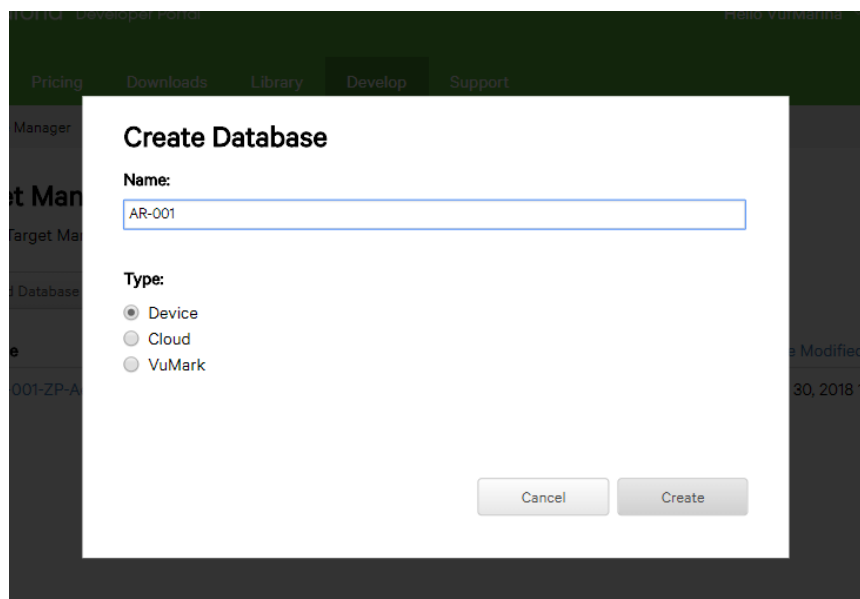


ВАЖНО!! Полученный лицензионный ключ необходимо скопировать и сохранить в любом текстовом редакторе (хороший стиль - не **Word!!**) для дальнейшего его использования в среде разработки (редакторе) **Unity 3D** на вашем локальном компьютере.

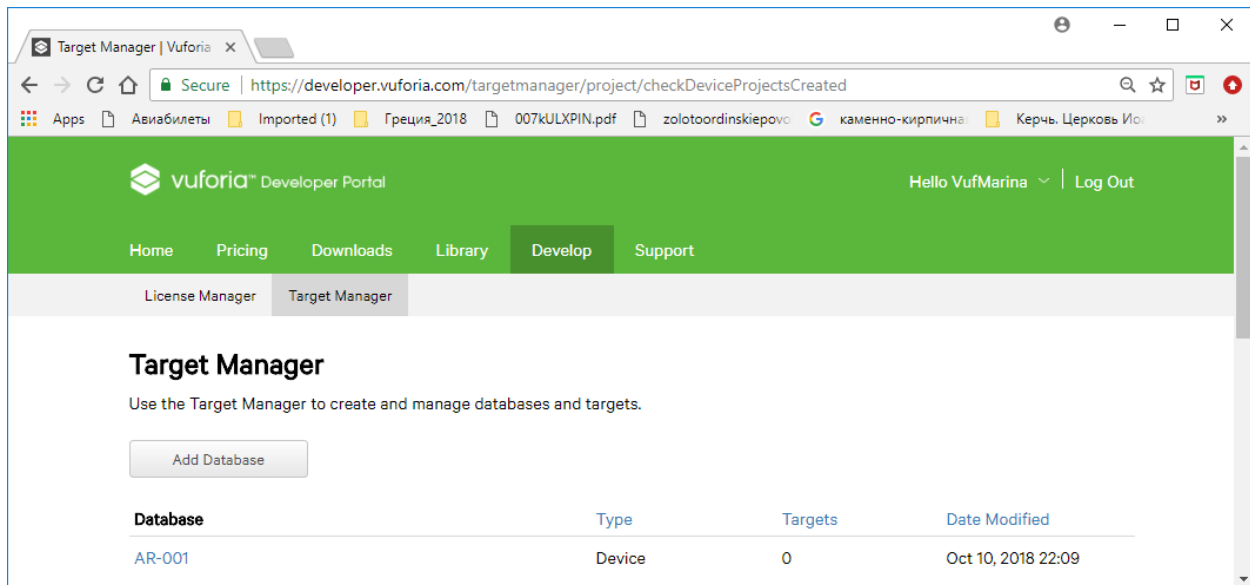
2. Переходим в менеджер меток – Target Manager:



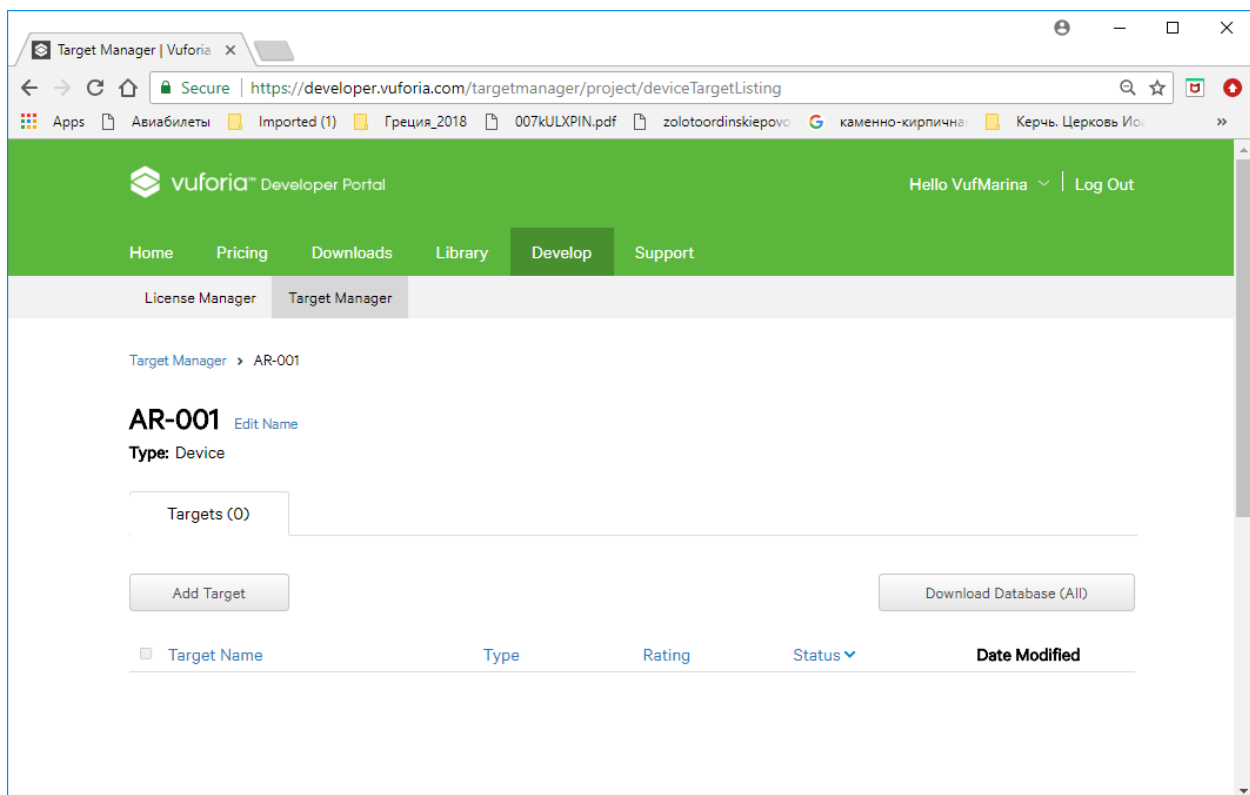
Фактически на данном этапе начинается создание базы данных меток → **Add Database**:



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.)**. После выбора кнопки **Create** можно переходить к созданию БД **AR-001**:



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



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: 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:

VUF-001.jpg

.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:

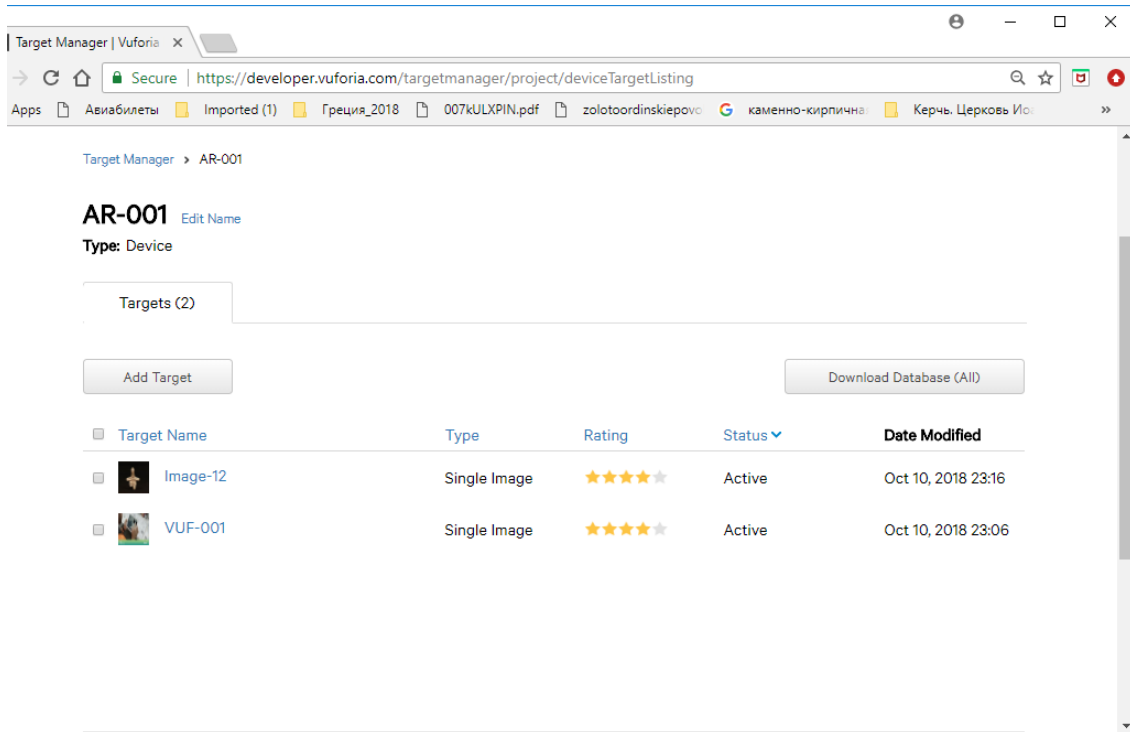
The screenshot shows the Vuforia Developer Portal interface. The page title is "Target Manager | Vuforia". The URL is "https://developer.vuforia.com/targetmanager/project/deviceTargetListing". The page has a green header with the Vuforia logo and "vuforia™ Developer Portal". The user is logged in as "Hello VufMarina" and can click "Log Out". The navigation menu includes "Home", "Pricing", "Downloads", "Library", "Develop", and "Support". The "Target Manager" tab is active, showing "License Manager" and "Target Manager" sub-tabs. The main content area shows "Target Manager > AR-001". The target name is "AR-001" with an "Edit Name" link. The type is "Device". There is a "Targets (1)" section with an "Add Target" button and a "Download Database (All)" button. Below this is a table with columns: Target Name, Type, Rating, Status, and Date Modified. The table contains one entry: VUF-001, Single Image, 5 stars, Active, Oct 10, 2018 23:06.

Target Name	Type	Rating	Status	Date Modified
 VUF-001	Single Image	★★★★★	Active	Oct 10, 2018 23:06

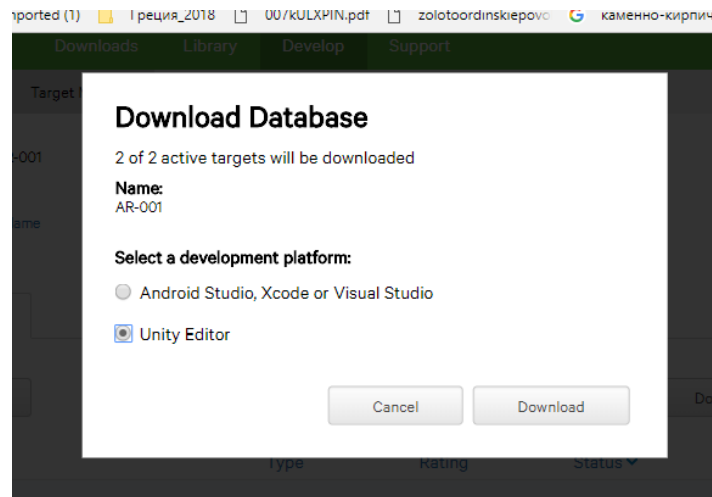
Выбранный нами файл изображения помещен в базу. Качество файла – достаточное для дальнейшей работы и распознавания.

ВАЖНО! Для разных приложений ДР можно использовать одну метку! Если для каждого Приложения ДР в одном проекте д.б. использована своя метка, то добавить ее необходимо на данном шаге.

Добавим в формируемую БД меток еще одну → **Add Target:**

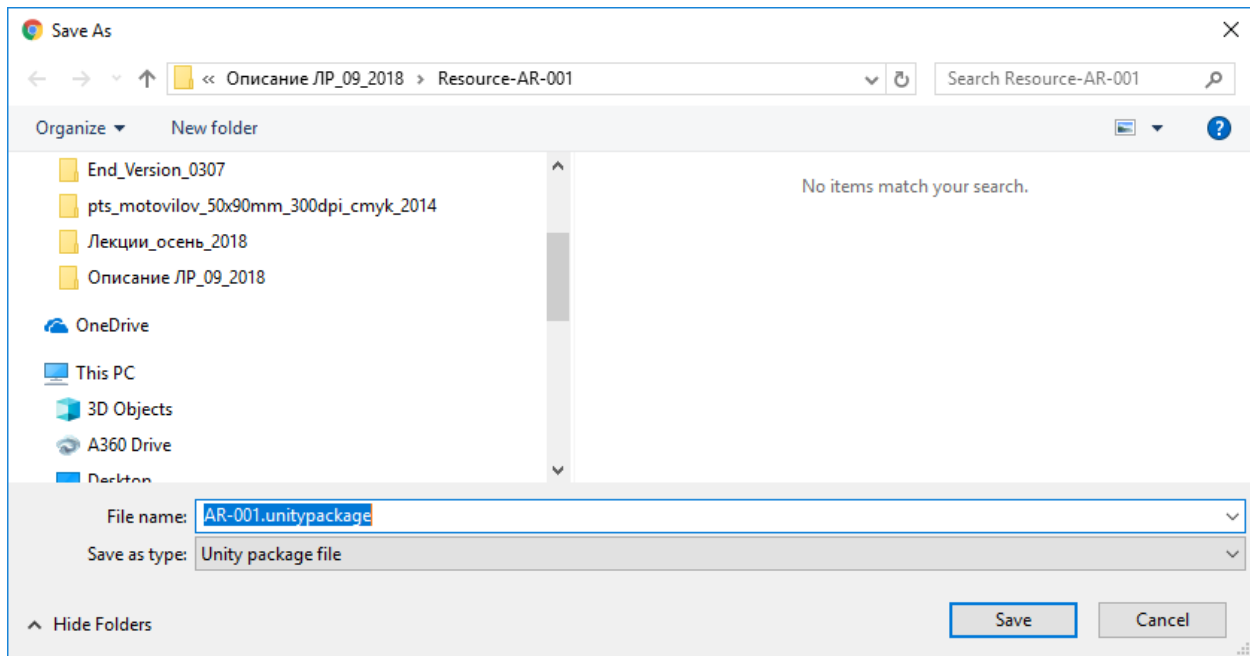


Для размещения этих меток в БД **AR-001** выбираем их и загружаем → **Download Database (2).**



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

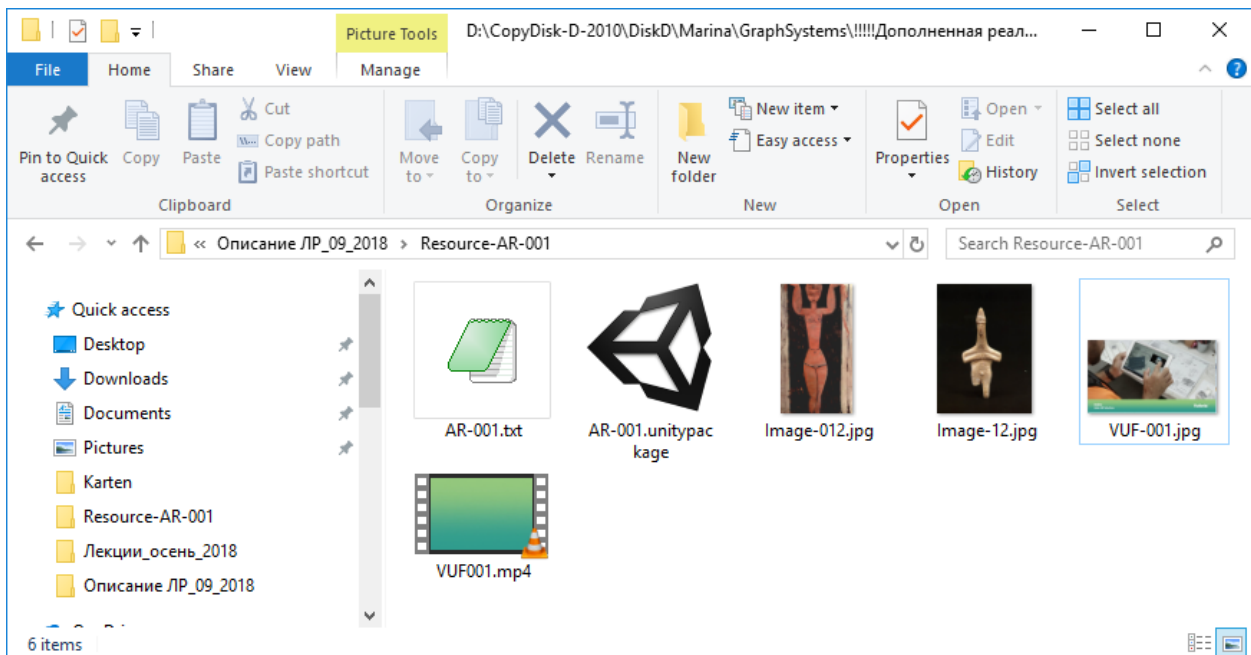
В результате формируется образ Базы Данных **AR-001** для работы в среде **Unity 3D** в специальном формате - **.unitypackage**, который выгружается из облака **Vuforia** для сохранения в локальной файловой системе:



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

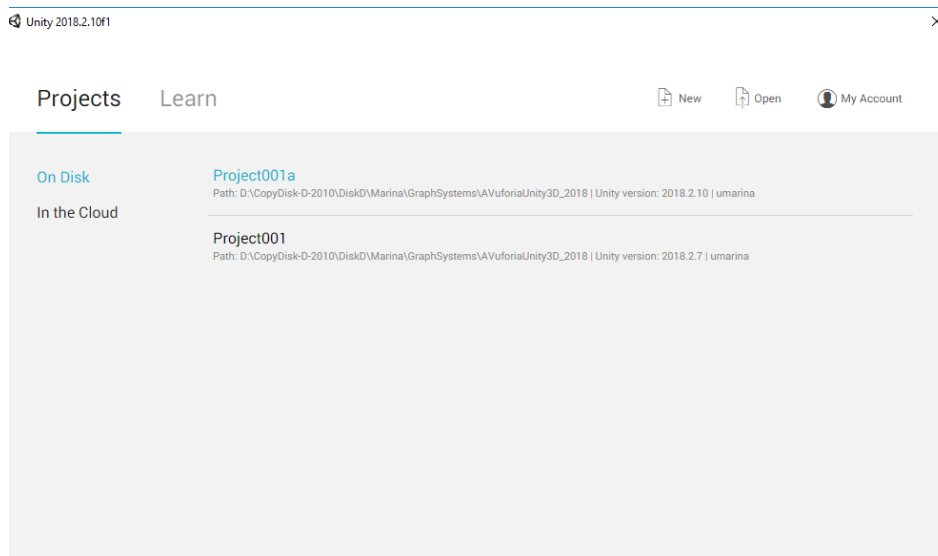
- **Лицензионный ключ**
- **Базу данных меток.**

Кроме того, подготовим объекты для замены метки на экране **Android**-устройства: видео в формате **.mp4** и изображение в формате **.jpg**.

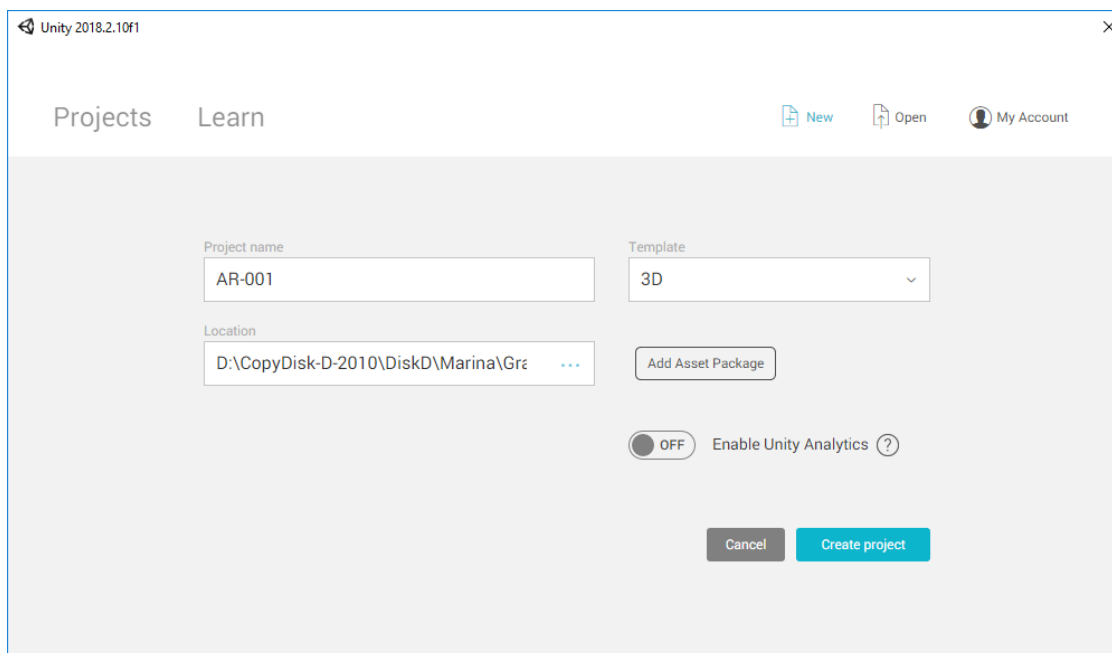


ВАЖНО!! Для Варианта № 2 ЛР №2 (**3D-Модель**) загрузка контента будет показана ниже.

3. Стартуем приложение **Unity 3D**.



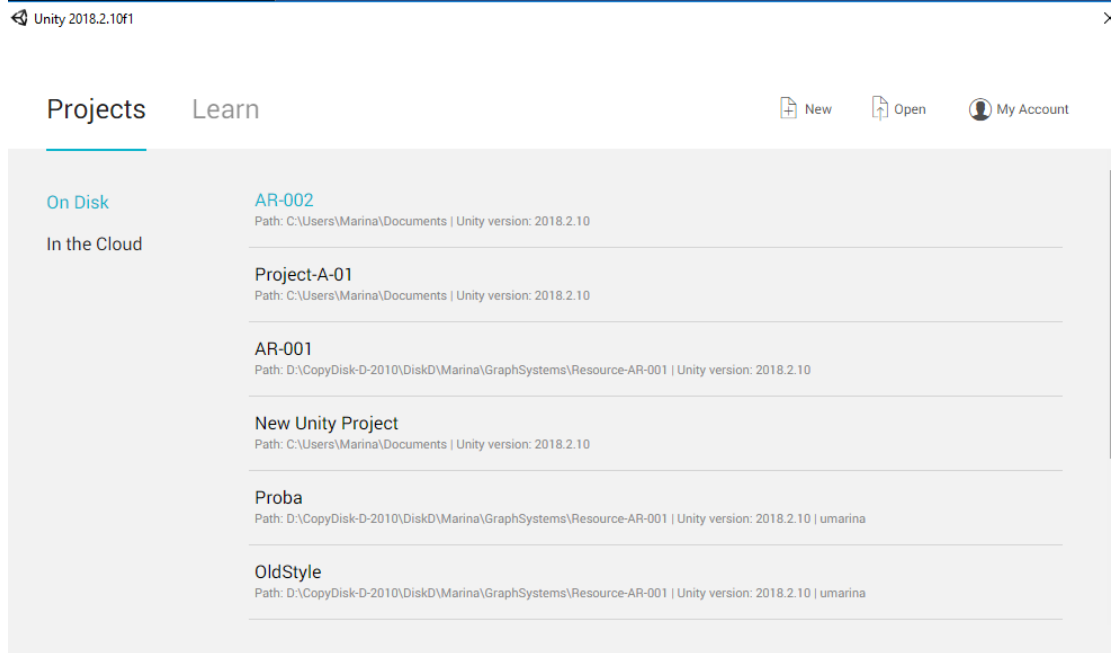
Выбираем, а при первом старте – создаем проект. Для этого нажимаем **New** в правом верхнем углу экрана и заполняем регистрационную карточку проекта (например, **AR-001**):



ВАЖНО!! Имя проекта – как мы и договаривались – например, **AR-001**; шаблон проекта (**Template**) д.б. **3D**, **Location** – это каталог, куда будут размещаться результаты работы в **Unity 3D** по данному проекту. Переключатель – в положении **OFF**. И только после этого → **Create project**.

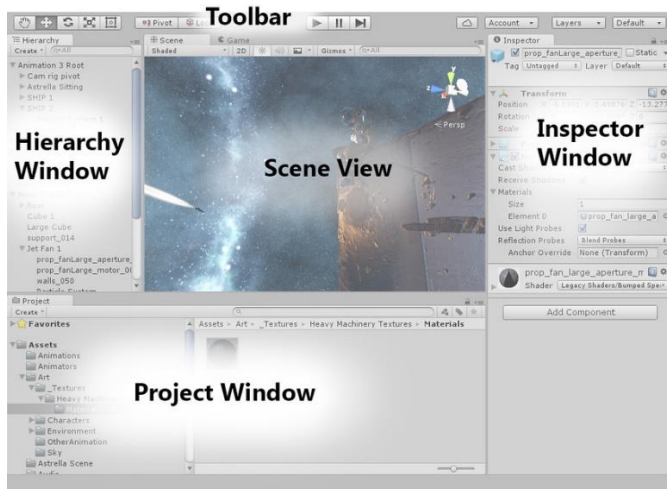
ВАЖНО!!: для имен объектов проекта, файлов и каталогов, включая родительские, допускается использование букв ТОЛЬКО латинского алфавита, цифр и двух спецсимволов: «-» и «_».

Если проект уже был однажды создан, то на этом этапе его можно выбрать из списка имеющихся, например – **AR-002** (или любое имя из пространства ваших имен проектов!!):

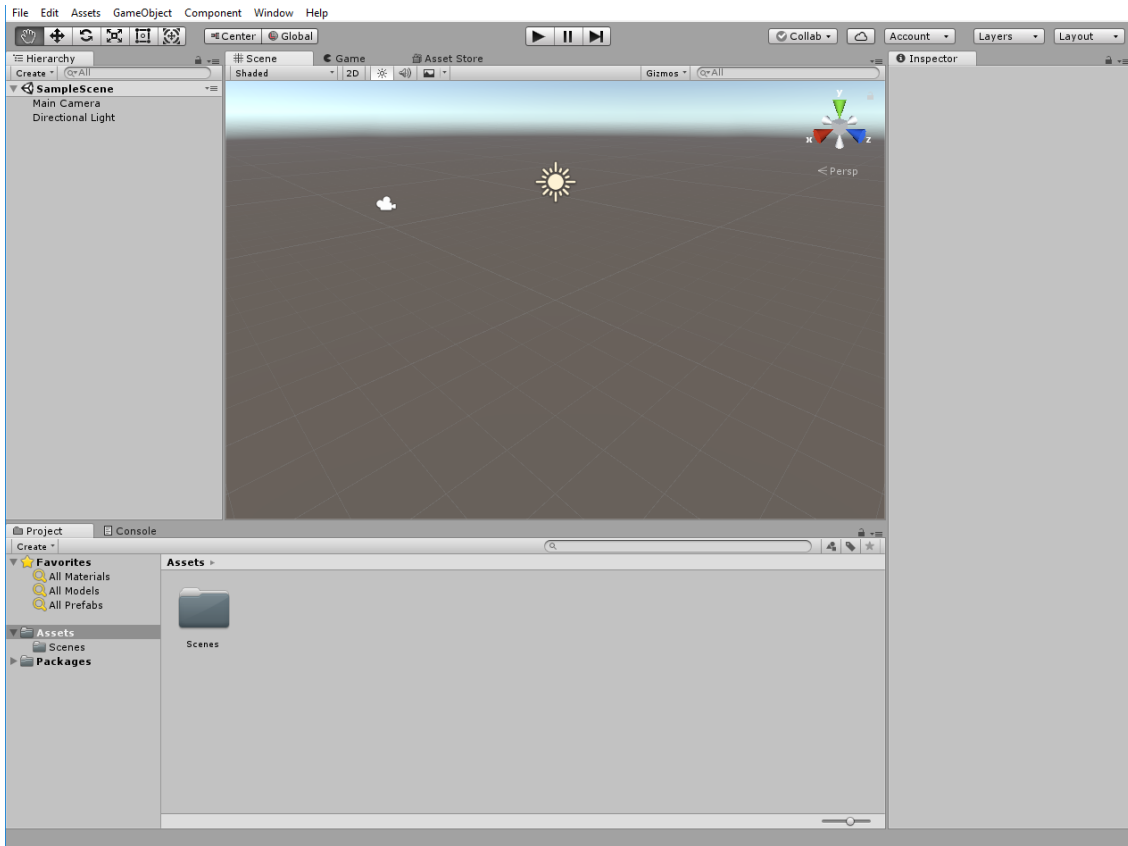


Открывается среда разработки – графический редактор **Unity 3D**. Документацию – см. по ссылке:

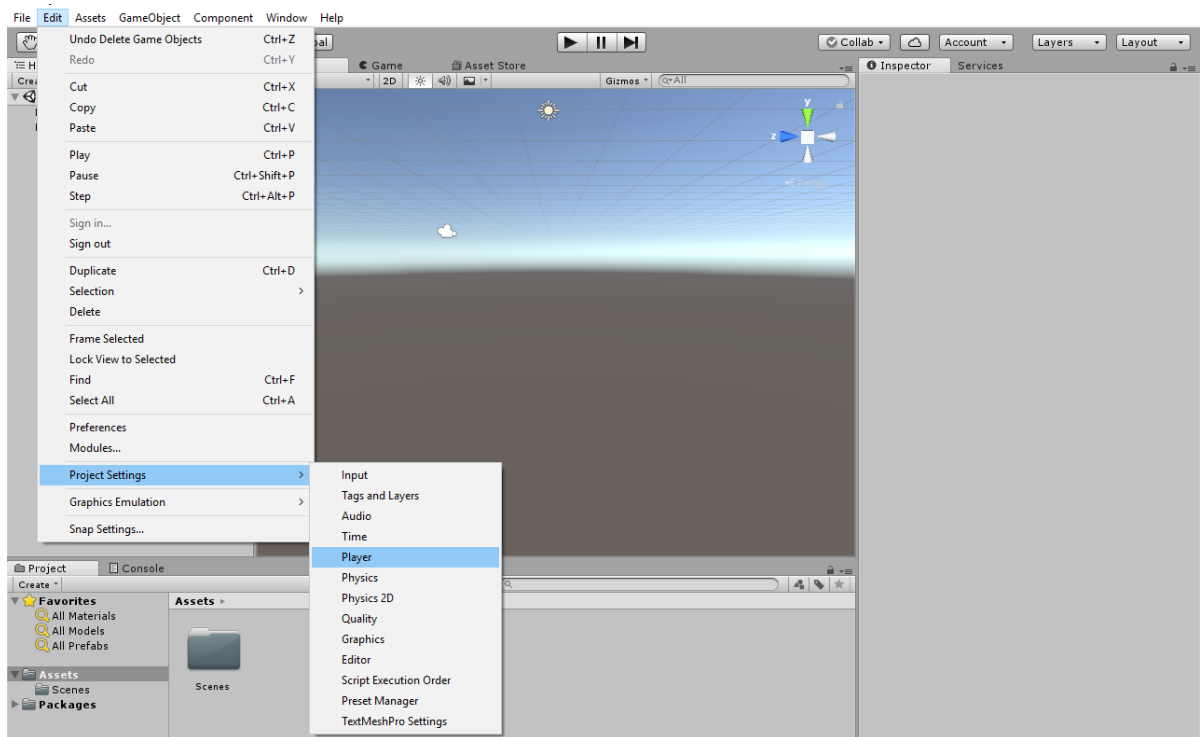
<https://docs.unity3d.com/Manual/LearningtheInterface.html>



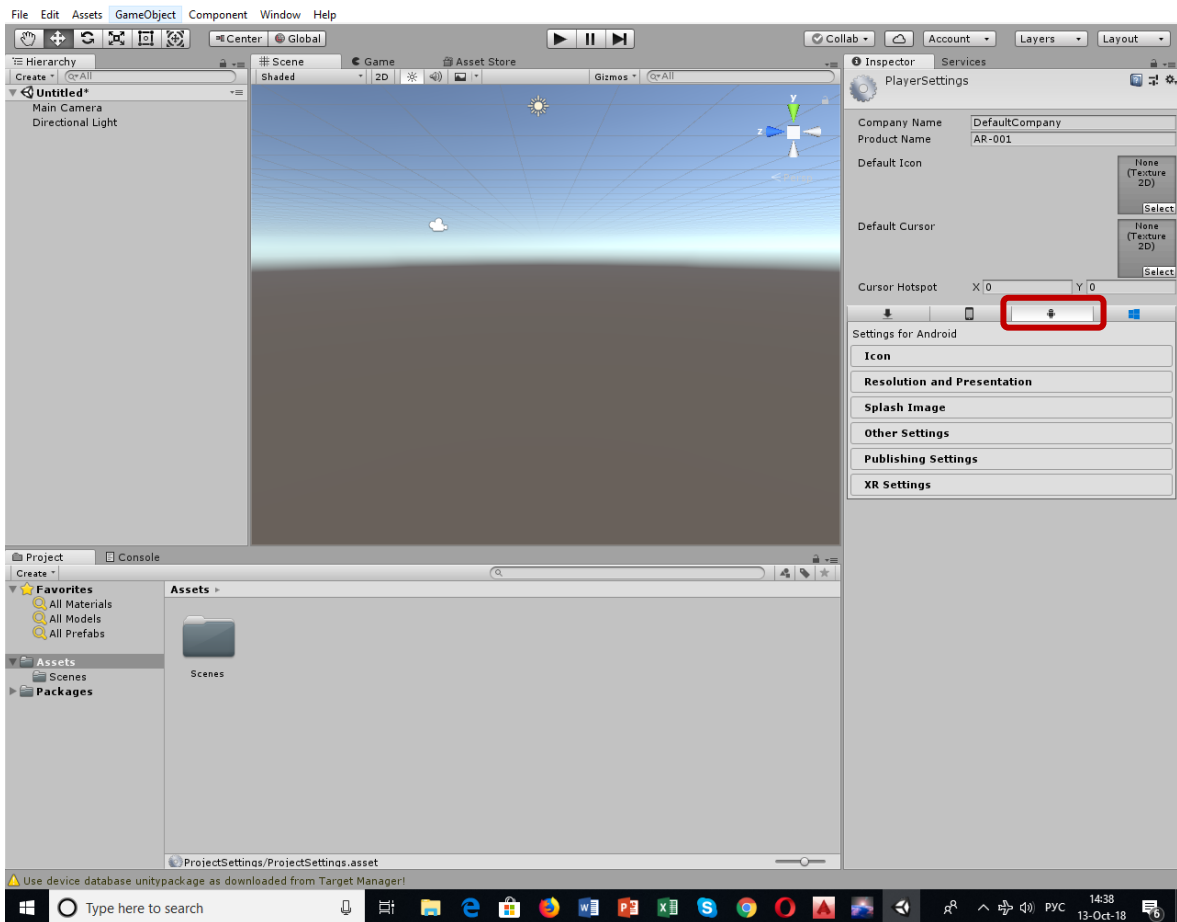
Обратите внимание - начальные действия по организации среды игрового движка **Unity 3D** для нашего проекта (Приложение ДР) происходят с использованием закладки **Scene** (область **Scene View**) и группы элементов линейки **Toolbar**, объединенных в группу с названием **GameObject**.



Без дополнительных настроек редактор **Unity 3D** работает в режиме виртуальной реальности. По условиям Задания ЛР наше **Android**-устройство в Проекте должно работать одновременно в режиме камеры и в режиме плеера (проигрывателя изображения ДР). Для этого необходимо перед началом работы со сценой выполнить некоторые настройки – т.е. выставить режим такой работы камеры редактора **Unity 3D**:

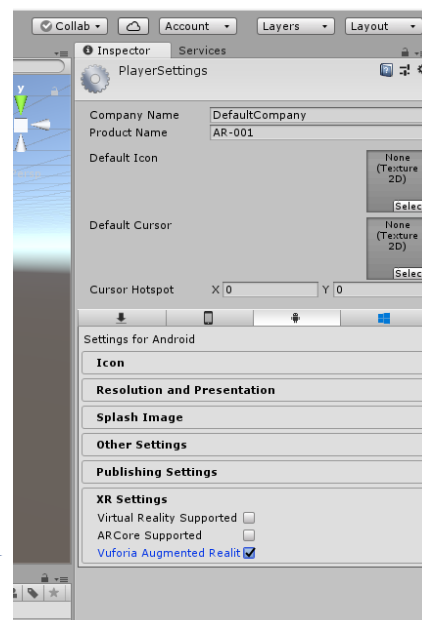


В результате в окне **Inspector**'а появятся настройки плеера (**Player Settings**). Использовать будем настройки плеера для **Android**-устройства:



Включим режим дополнительных настроек

XR Settings→**Vuforia Augmented Reality**:



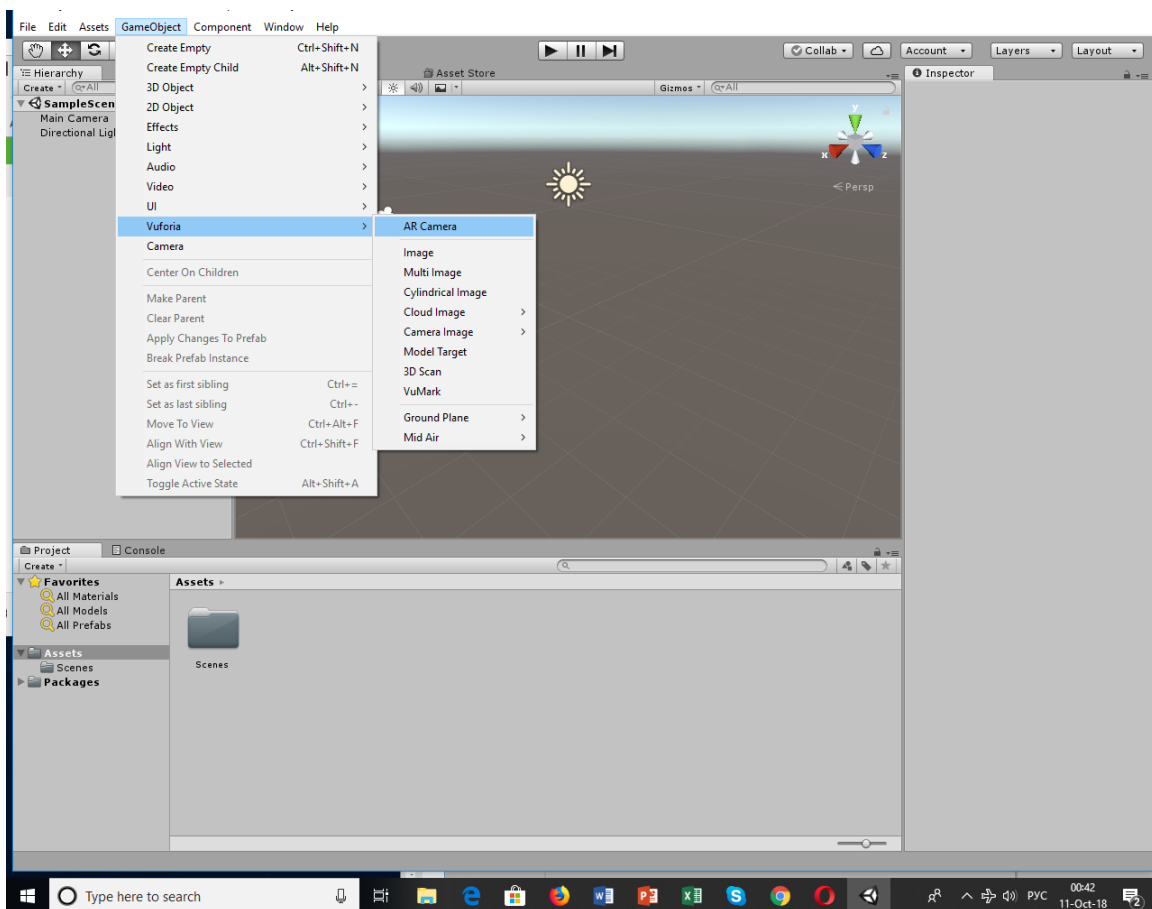
Использование настроек и дополнительных возможностей **Vuforia** позволяет перевести работу редактора в режим **дополненной реальности**.

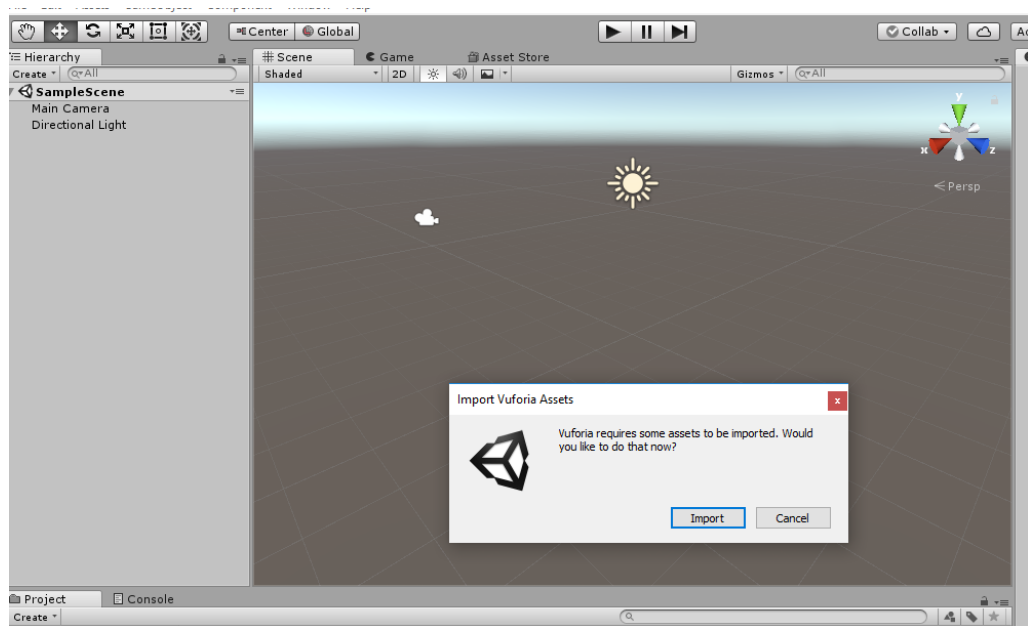
Main Camera редактора **Unity 3D** не позволяет выполнять визуализацию реального окружения, а работает только с **виртуальными объектами сцены в виртуальном пространстве**.

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**.

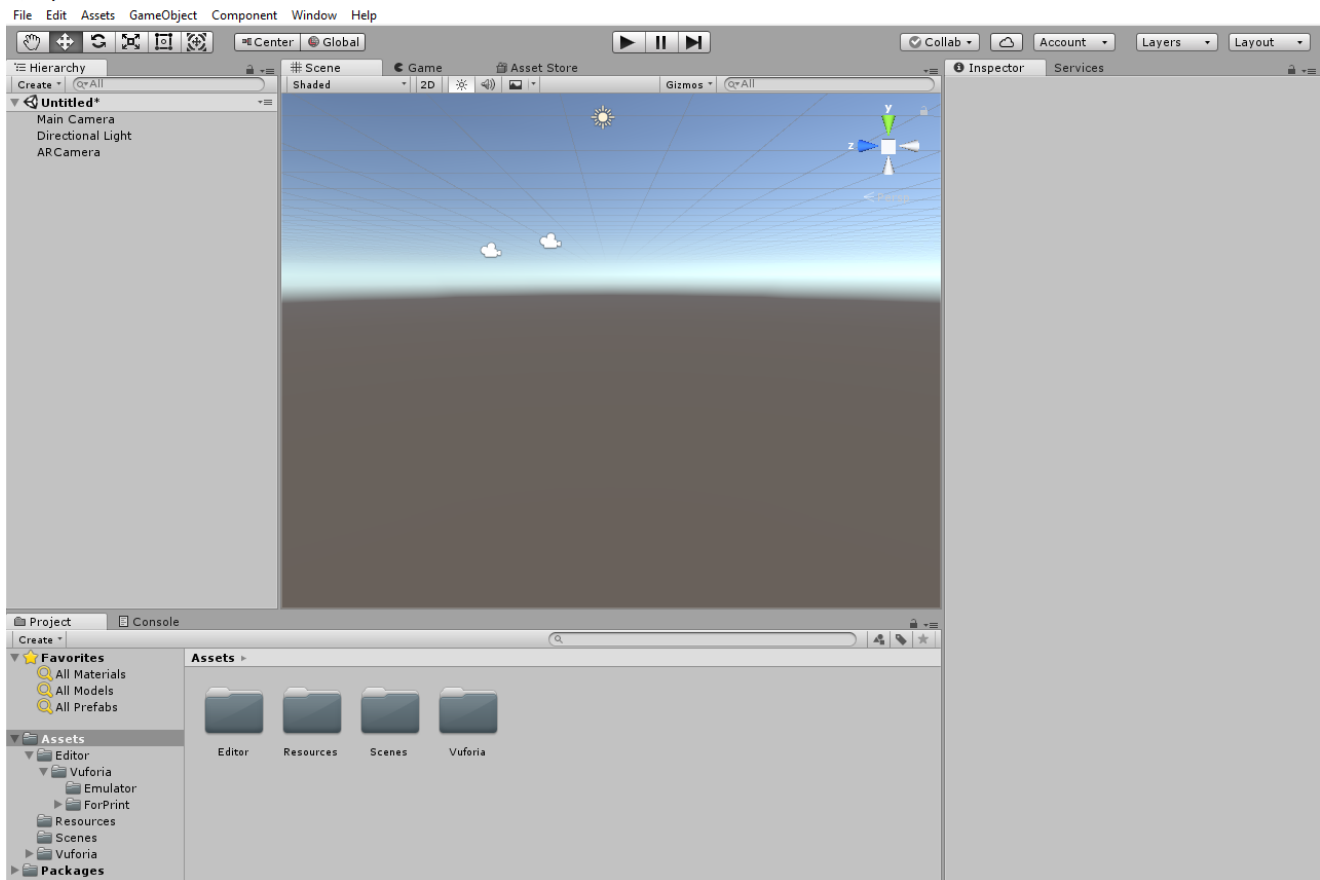
Для этого выполним следующие вызовы:

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

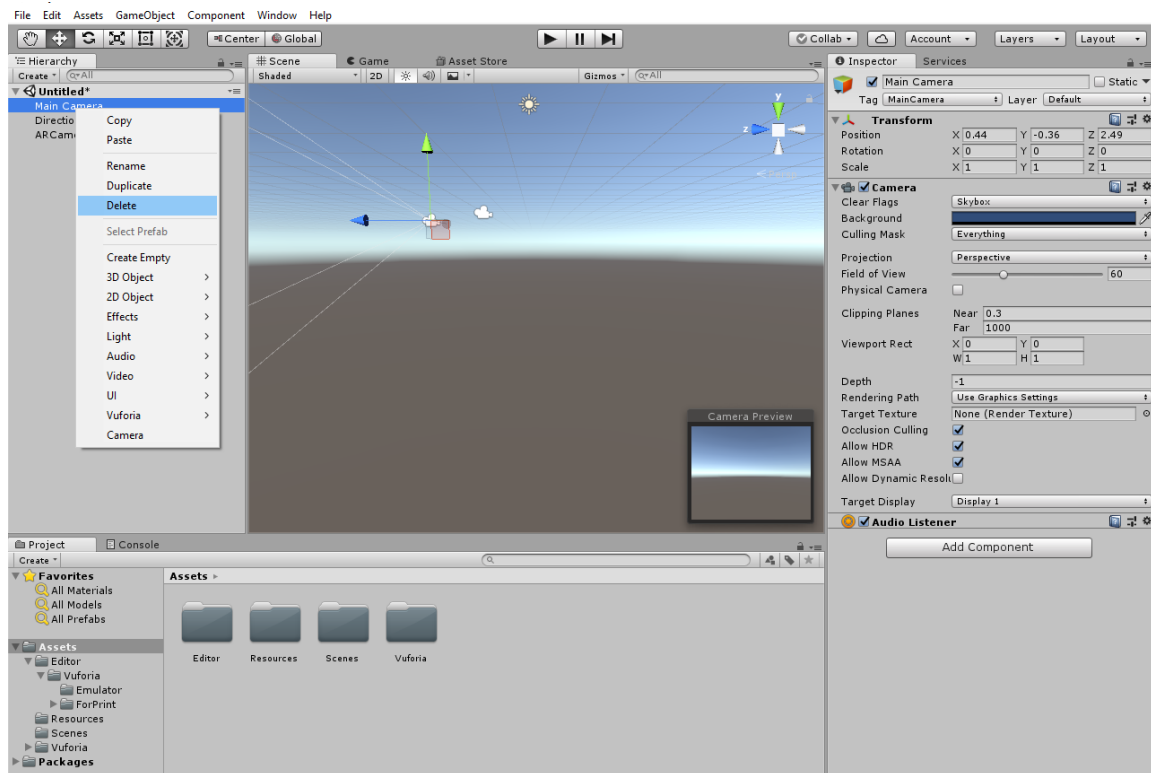




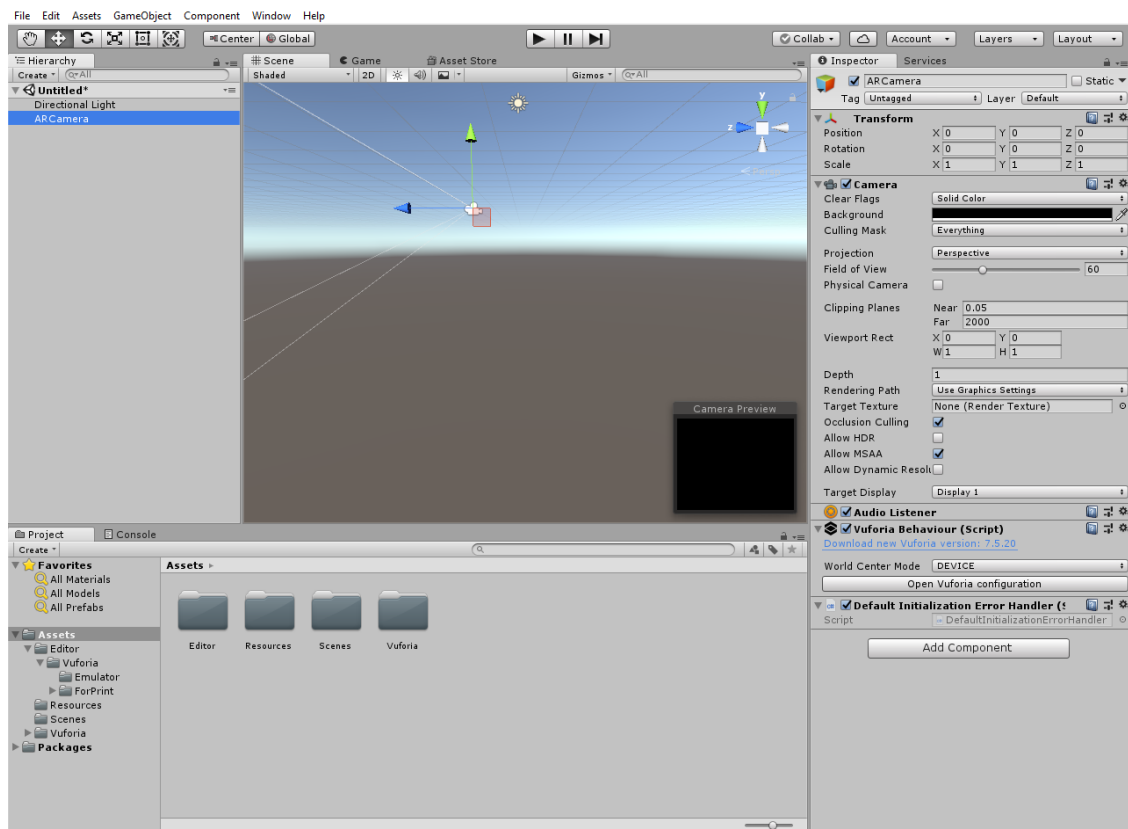
Нажимаем **Import**. Результат работы приведен на следующем рисунке. Обратите внимание, что в области **Project** экрана среды **Unity 3D** в папке **Assets (активны)** появились новые объекты, ответственные за создание приложений ДР с помощью **Vuforia**.



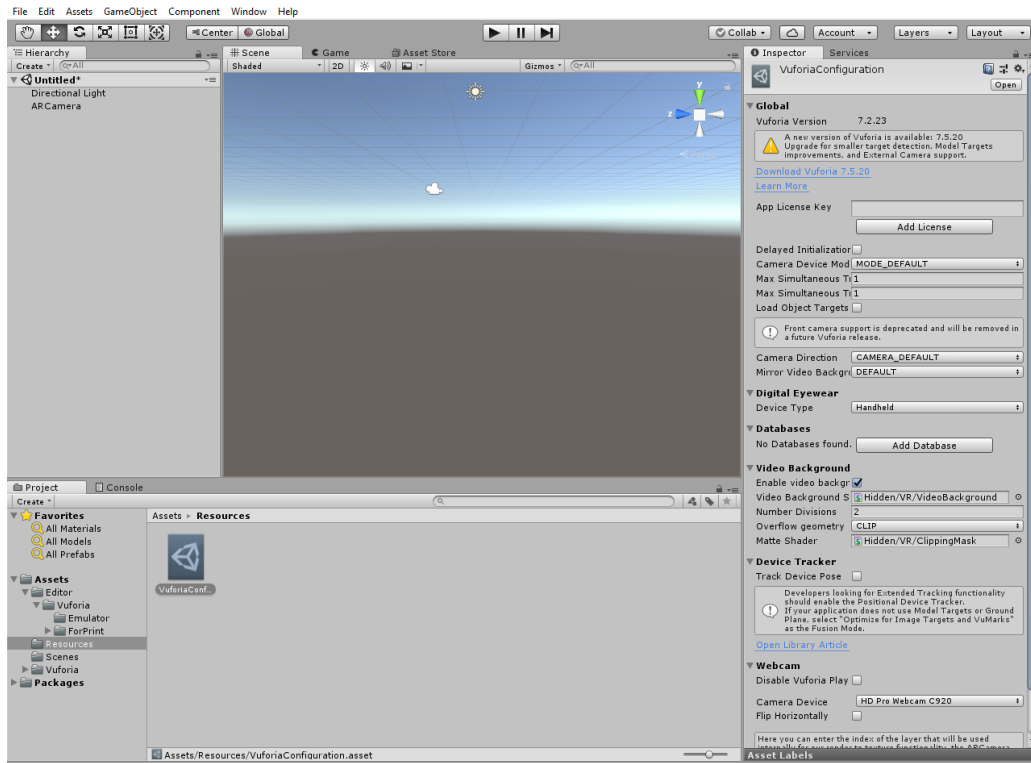
Теперь выполняем замену камеры: оставляем **AR Camera** и удаляем **Main Camera**. **Main Camera** должна быть удалена обычным способом – маркируем **Main Camera** в меню иерархии и выбираем в падающем контекстном меню (по правой клавише) функцию **Delete**.



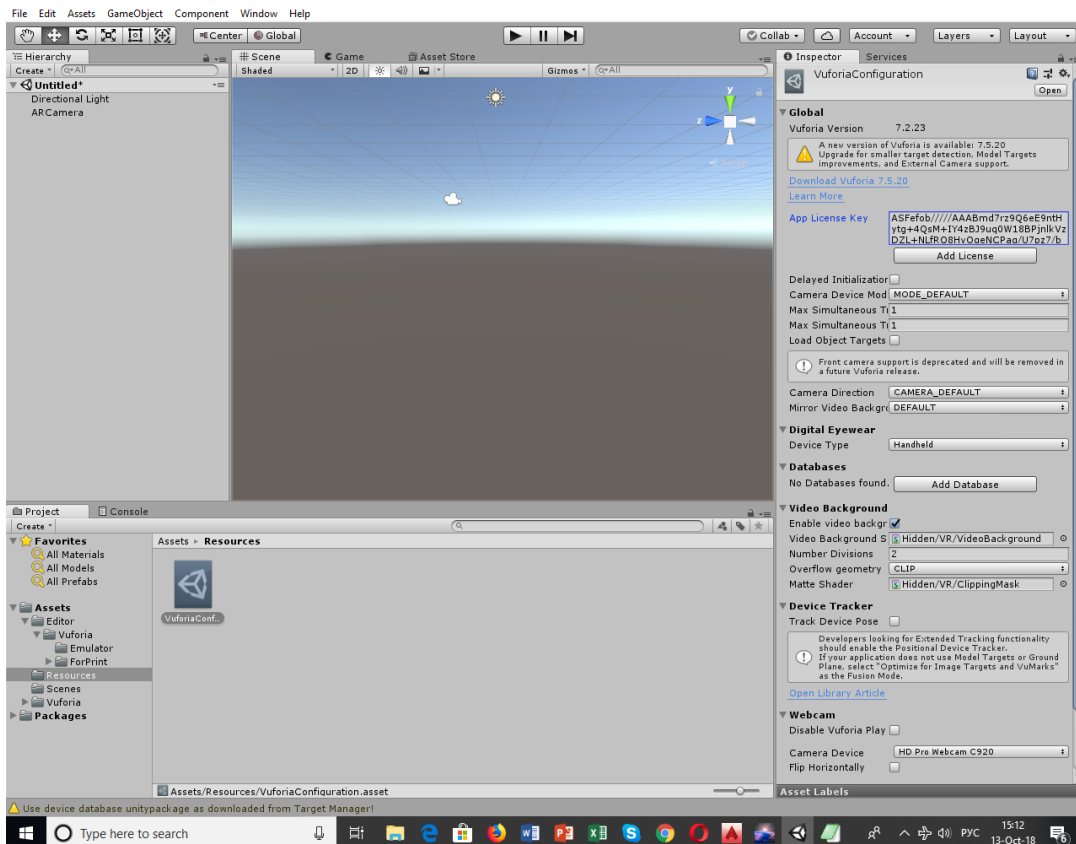
Теперь в области **Inspector** можно посмотреть параметры и свойства загруженной **AR Camera**:



4. На этом этапе можем начать процесс конфигурирования **Vuforia** для работы с **Unity 3D** в нашем проекте → выбираем **Open Vuforia Configuration** в области **Inspector** и получаем возможность установки **лицензионного ключа**, сгенерированного на предыдущем этапе.



В появившемся окне **Inspector**'а необходимо в поле **App License Key** ввести (например, через **Copy/Paste**) текст лицензии, сохраненный ранее.



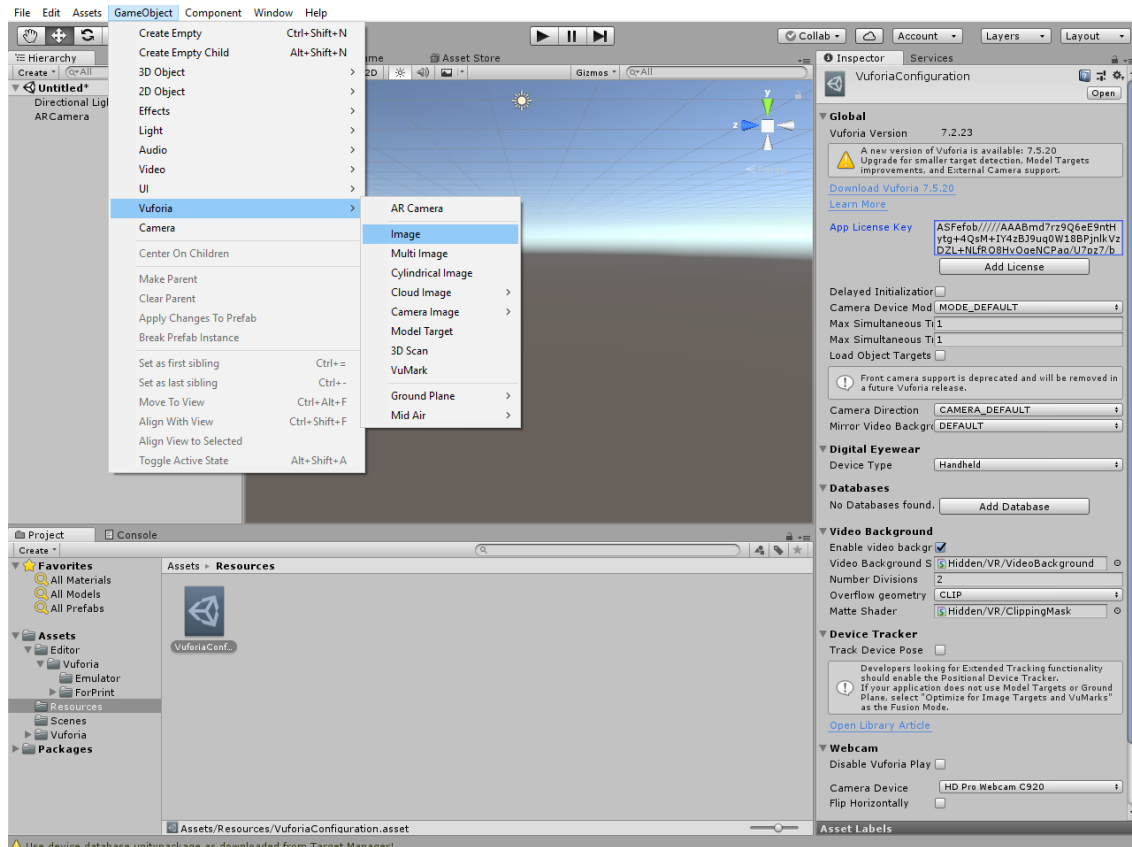
ВАЖНО!! Если Вы выберете на этом этапе поле **Add License**, то вновь попадете на этап ее генерации, который в данной ЛР был выполнен в первую очередь. Иными словами, лицензия автоматом не устанавливается при конфигурировании, ее необходимо скопировать в поле **App License Key**, предварительно сгенерировав.

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

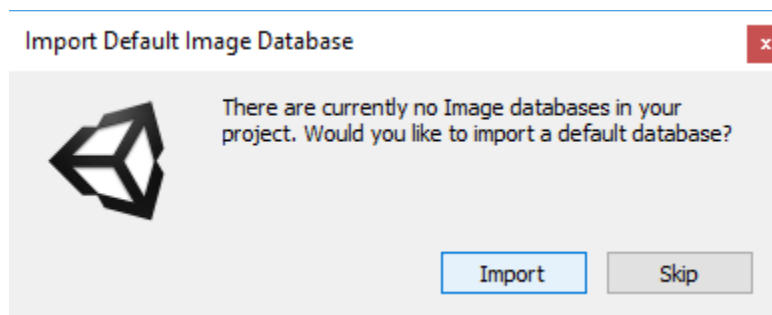
5. Далее необходимо загрузить Базу данных таргетов.

Сначала выполняем настройки **Unity 3D** на загрузку БД **Vuforia** - таргетов.

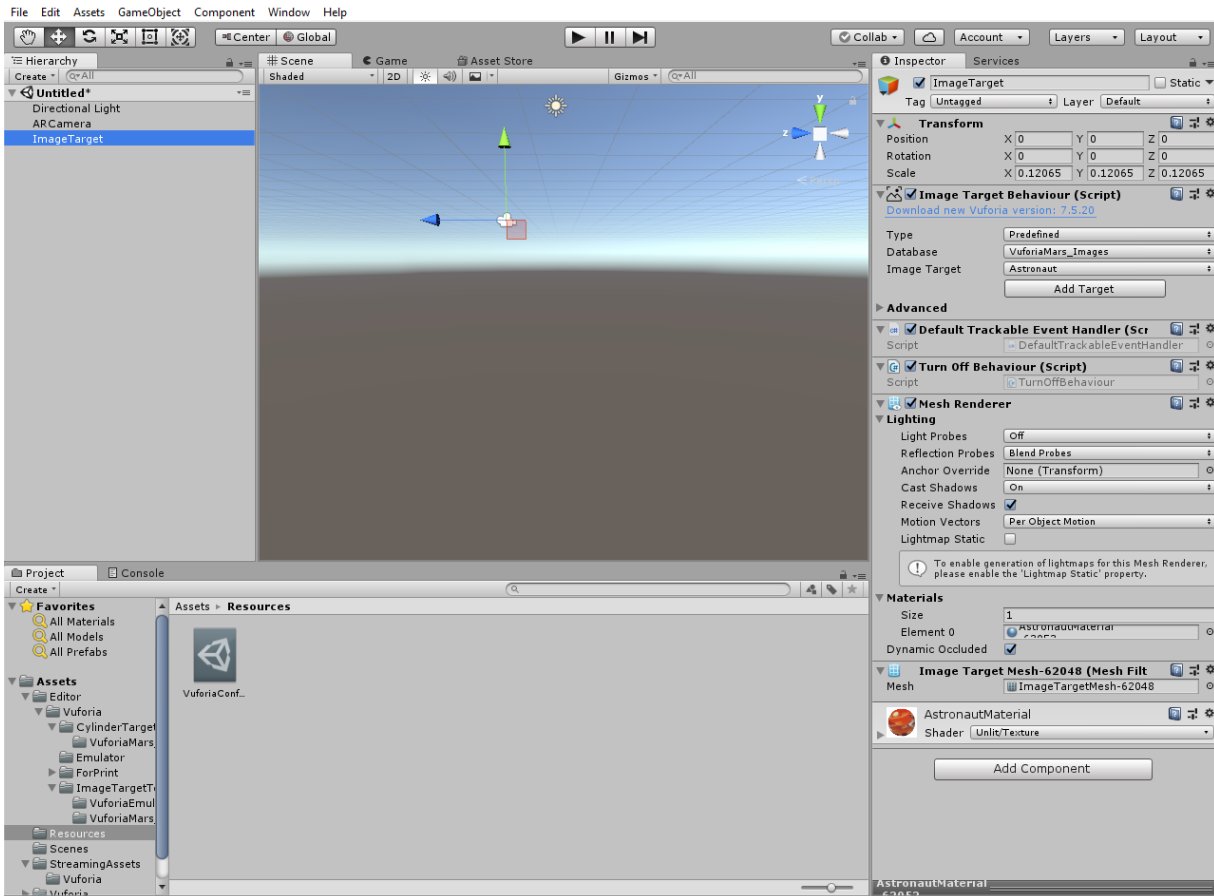
Для этого осуществляем вызовы в закладке **Game Object** → **Vuforia** → **Image**:



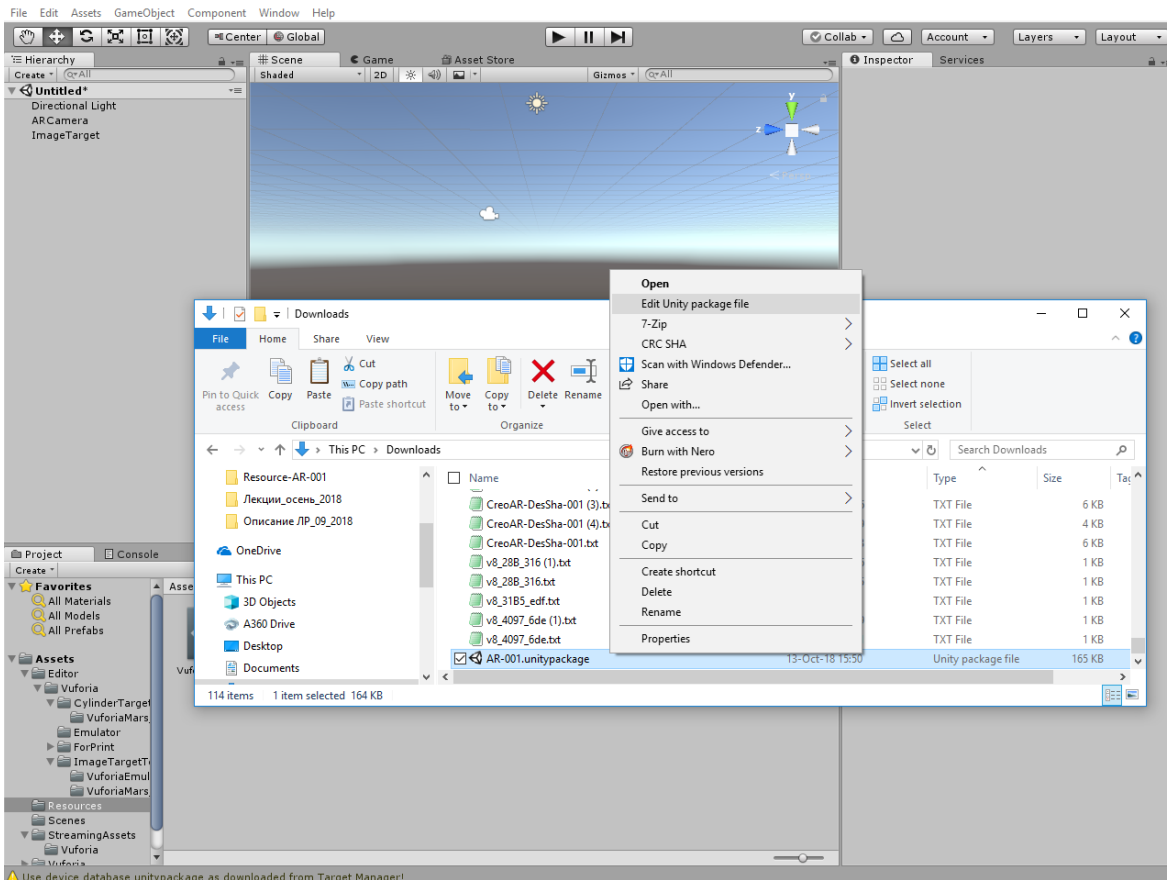
В появившемся запросе нажмите кнопку **Import**



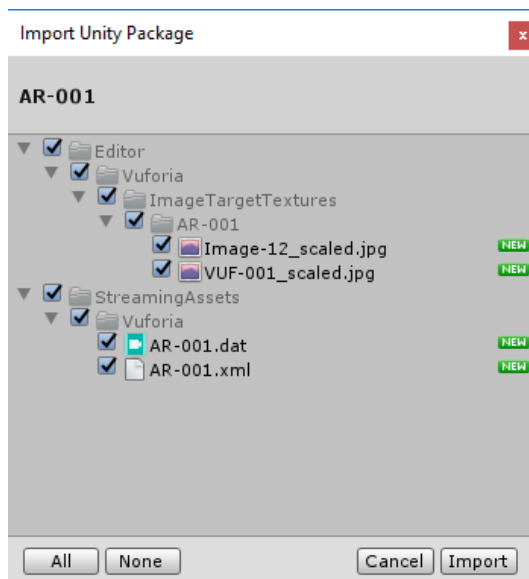
В результате загружается БД по умолчанию («Астронавт»). Добавляем подготовленную заранее БД таргетов в сцену нашего проекта.



Для этого перейдем в локальной файловой системе в тот каталог, где была сохранена БД таргетов (файл *.unitypackage). Подведите курсор к сохраненному *.unitypackage и по правой клавише выберите действие **Edit Unity Package File**.

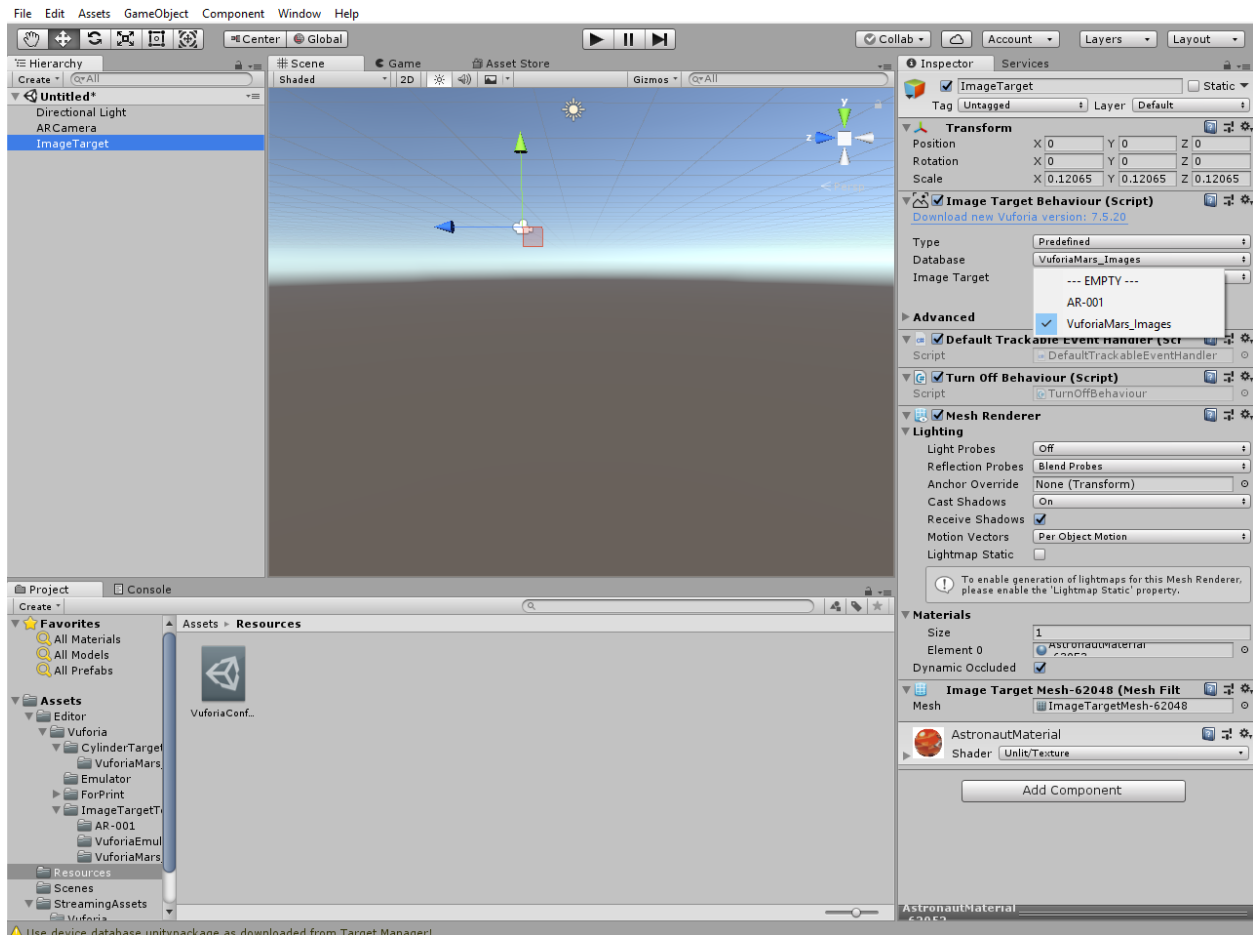


В результате в окне среды **Unity 3D** появляется диалоговое окно импорта БД таргетов:

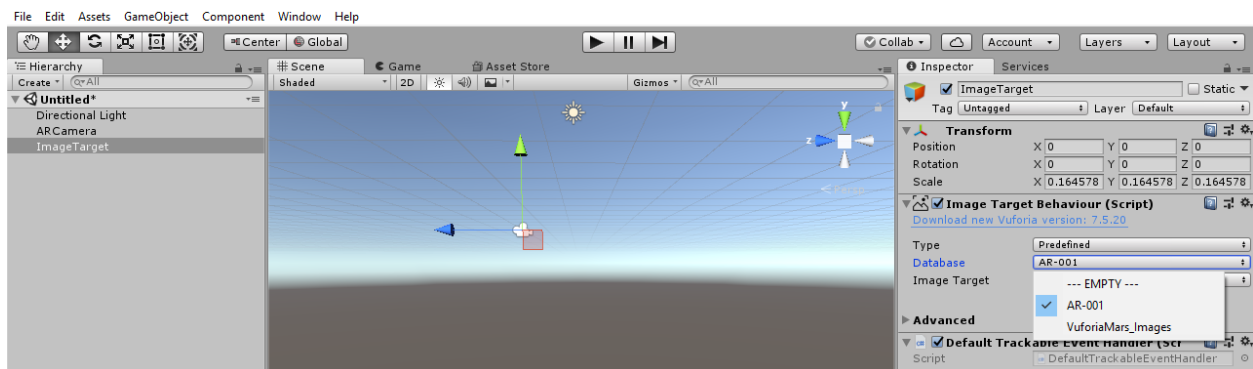


Выполняем импорт нашей БД таргетов нашего проекта.

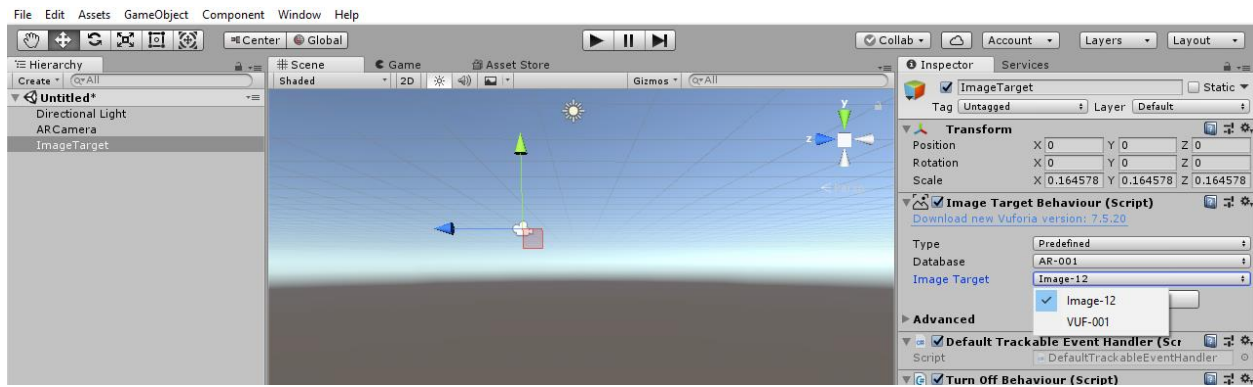
Проверить результат данного действия можно, выбрав в окне **Hierarchy** позицию **ImageTarget**. Далее в проявившемся окне **Inspector** можно убедиться в наличии БД таргетов нашего проекта, выбрав выпадающий список в области **Image Target Behaviour** → **Database**:



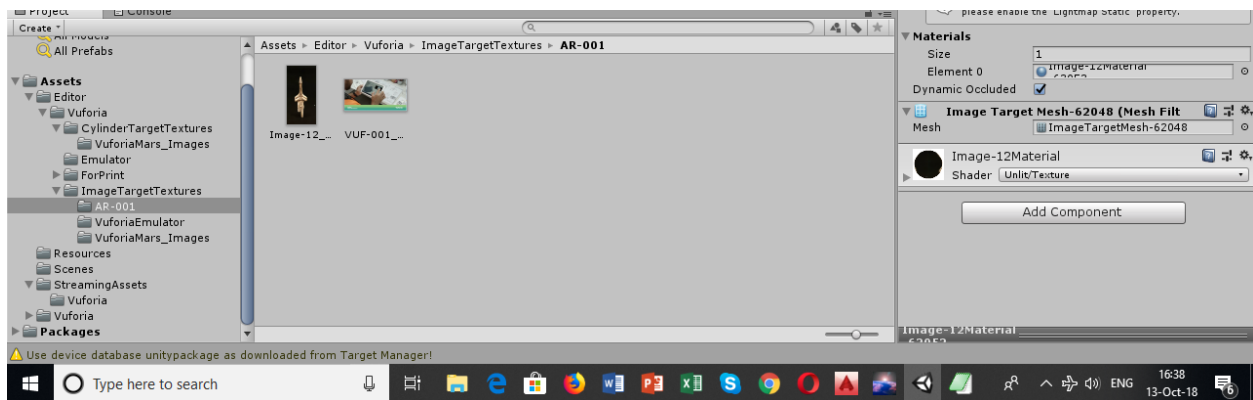
Теперь базу данных по умолчанию («Астронавт») надо заменить на БД таргетов выбранного **вами** проекта.



Таргетов может быть несколько, что отслеживается в соответствующих полях рассматриваемого меню:



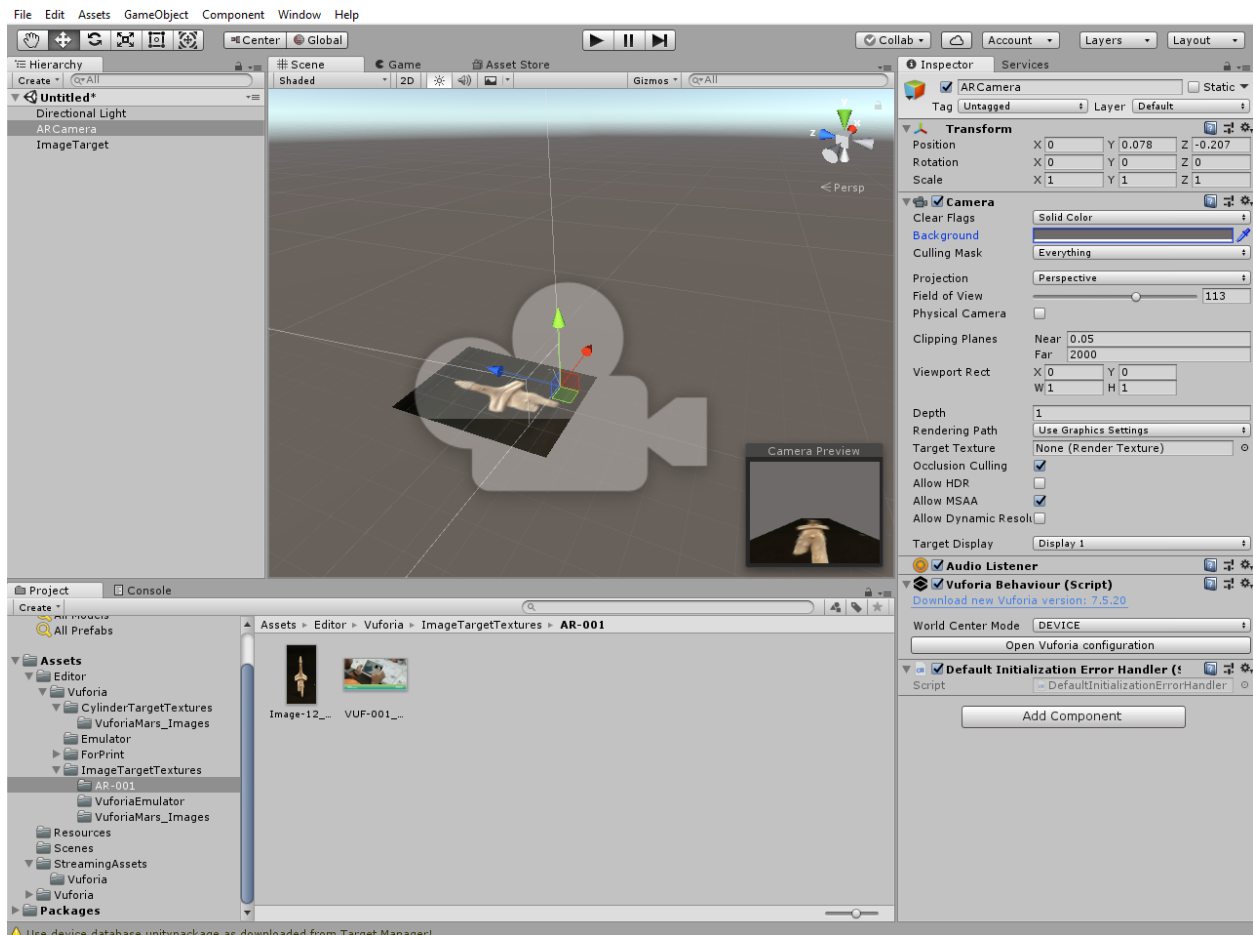
По мере выполнения всех этих действий следите за изменением наполнения области экрана **Unity 3D Project** и содержанием появляющихся там папок, в том числе и тех, которые имеют отношение к **выбранному вами** проекту:



Теперь в поле сцены имеются два интересующих нас объекта:

- **AR Camera** и
- **Image Target**.

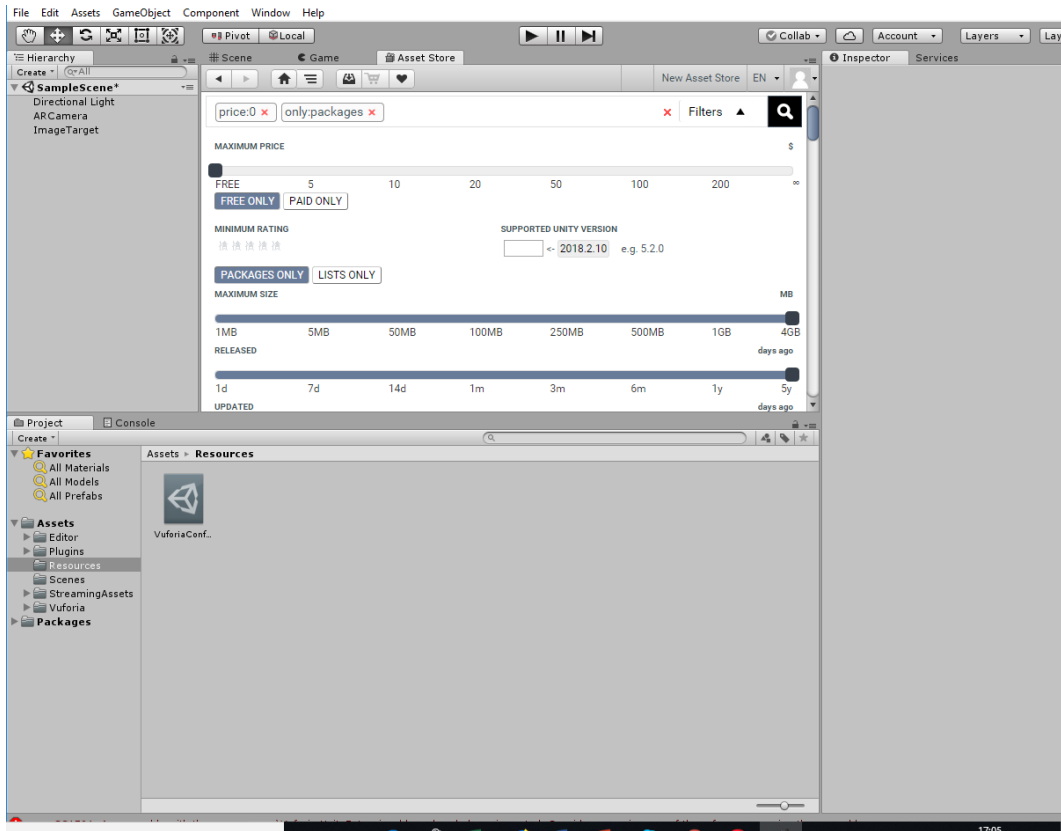
Нам необходимо добиться правильного взаимного расположения камеры и таргета. Контролировать правильность выполнения этого действия можно в окне предварительного просмотра (выбрать **AR Camera** в области **Hierarchy**) в нижней правой части области **Scene View** – окно **Camera Preview**. Используйте ручки **Transform (Position, Rotation, Scale)** в области **Inspector** 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.



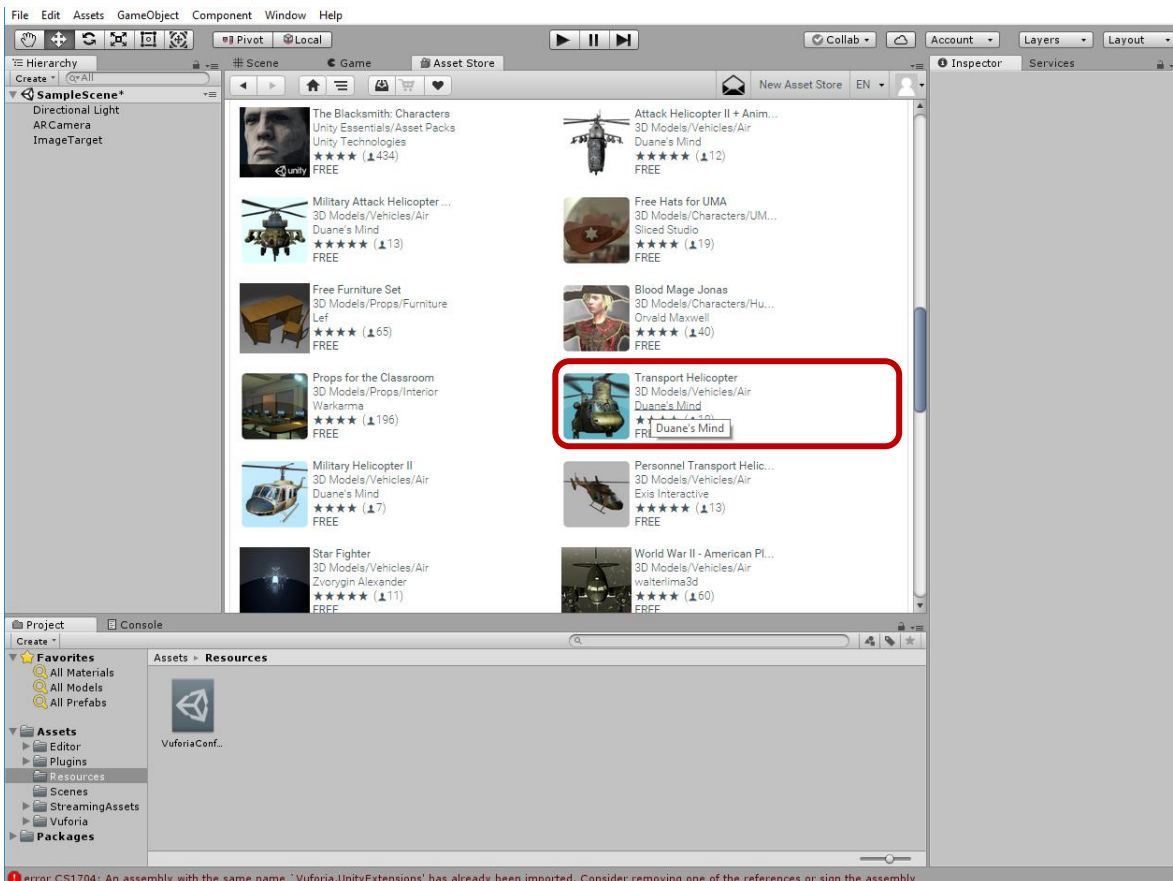
6. Теперь необходимо загрузить **3D - Модель**, которая должна появиться на экране устройства поверх изображения транслируемой реальности на месте, определяемом таргетом. – <https://docs.unity3d.com/ru/current/Manual/HOWTO-importObject.html>

Сами **3D-Модели** могут быть импортированы в среду проектирования Приложения ДР из разных источников – внешних приложений (например, САПР); из библиотек готовых моделей, доступ к которым реализуется непосредственно из редактора **Unity 3D**. В нашем случае воспользуемся именно этим вариантом.

В нашем случае, в рамках данного варианта ЛР №2, модели в среду **Unity Editor** будем добавлять из библиотек **3D-моделей Asset Store** («магазин»), доступных через интернет. Для этого переходим на одноименную вкладку **Asset Store** редактора **Unity 3D** и выбираем требуемый объект из открывшегося интерфейса «магазина».

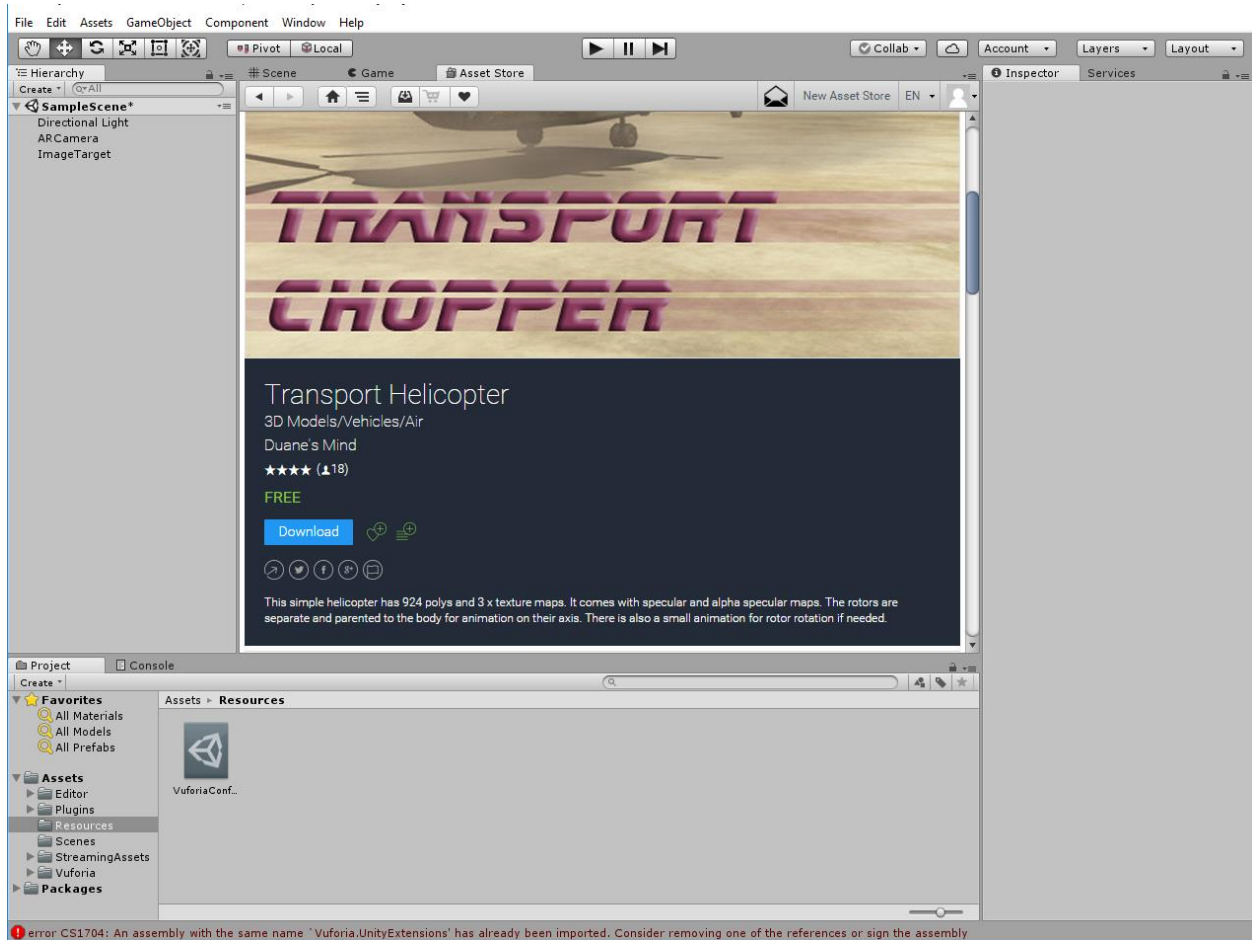


Предлагается выбирать бесплатные модели из какой-либо технической области, воспользовавшись доступными (стоимость) и формируемыми пользователем фильтрами, например, область применения. В описанном ниже примере была выбрана нулевая стоимость и область применения **Air**.

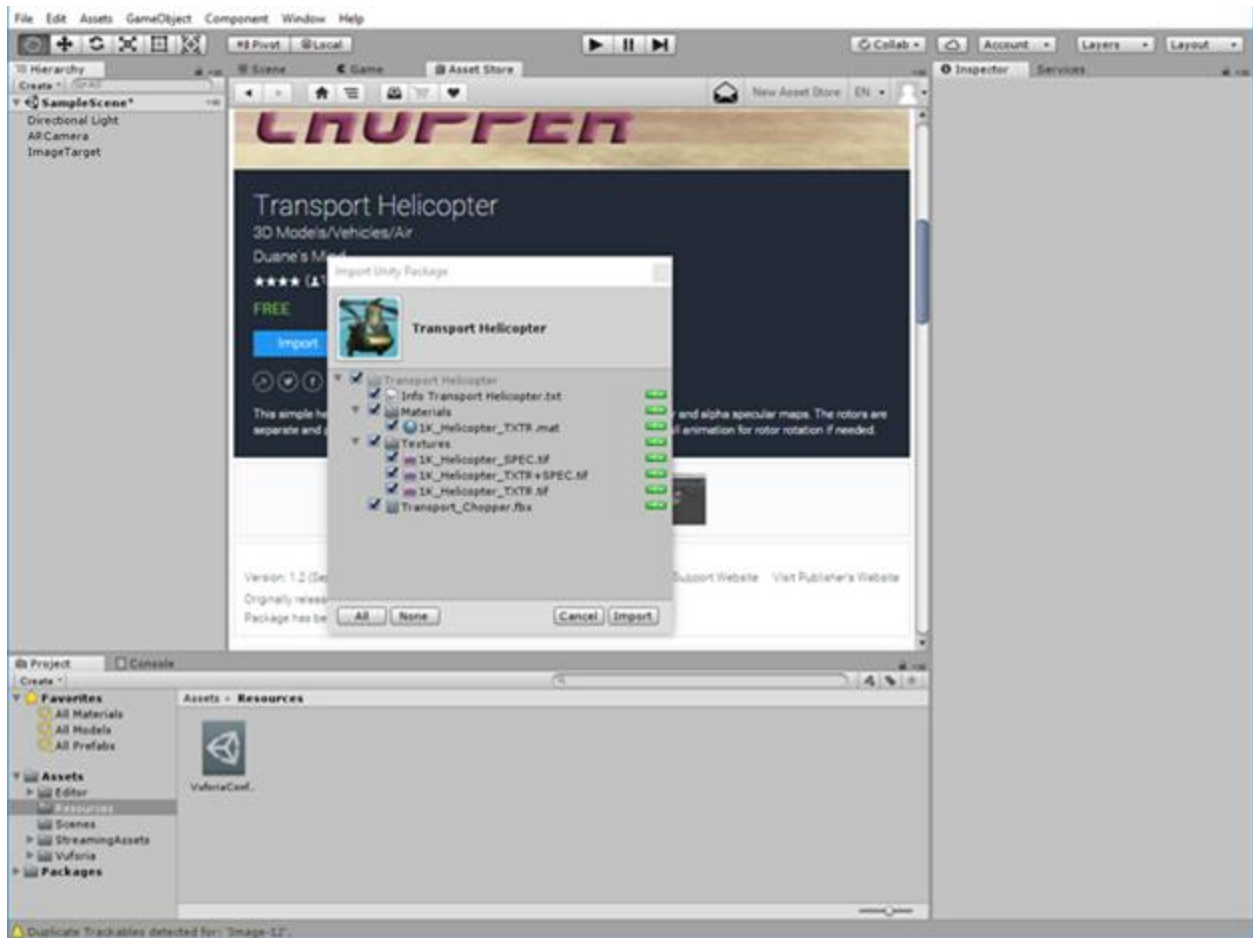


error CS1704: An assembly with the same name 'Vuforia.UnityExtensions' has already been imported. Consider removing one of the references or sign the assembly

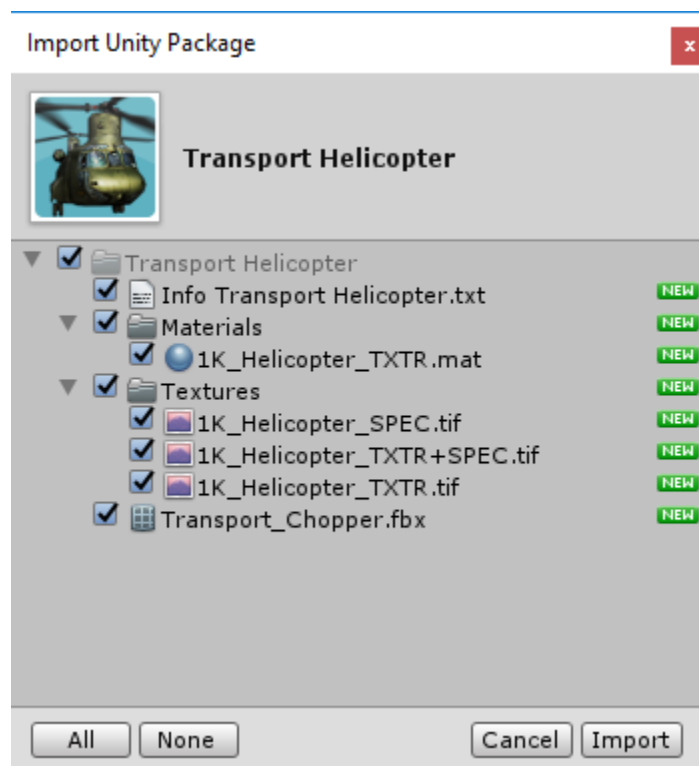
Перейдем к «покупке» выбранной нами модели, кликнув на выбранной в списке пиктограмме. При этом мы попадаем в меню разрешенной загрузки выбранной «покупки» (модели):



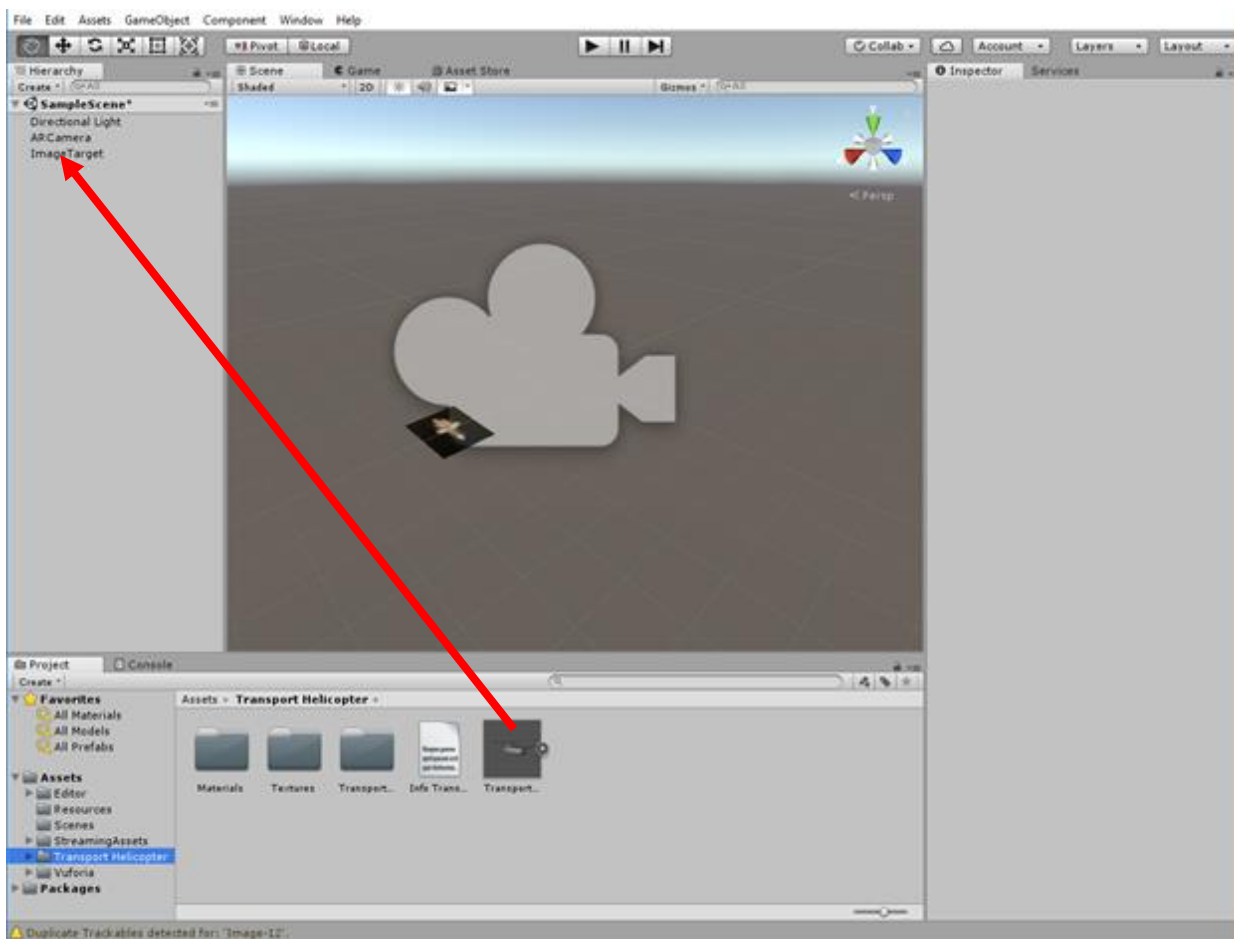
Нажатие клавиши **Download/Import** (могут быть варианты названия этой очевидной по предназначению клавиши) приводит к загрузке информационного набора нашей «покупки» (**3D-Модель** + вся сопутствующая информация – назовем их «Файлы моделей») в раздел **Assets** локального приложения – Редактор **Unity 3D**. Файлы моделей, размещаемые в папке **Assets** внутри проекта **Unity**, автоматически импортируются и сохраняются как ассеты **Unity**. В окне **Project** (проект) главный импортированный объект представляется в виде **Model Prefab** («префаб» модели) – особый вид ассета для работы с **3D-Моделями**.



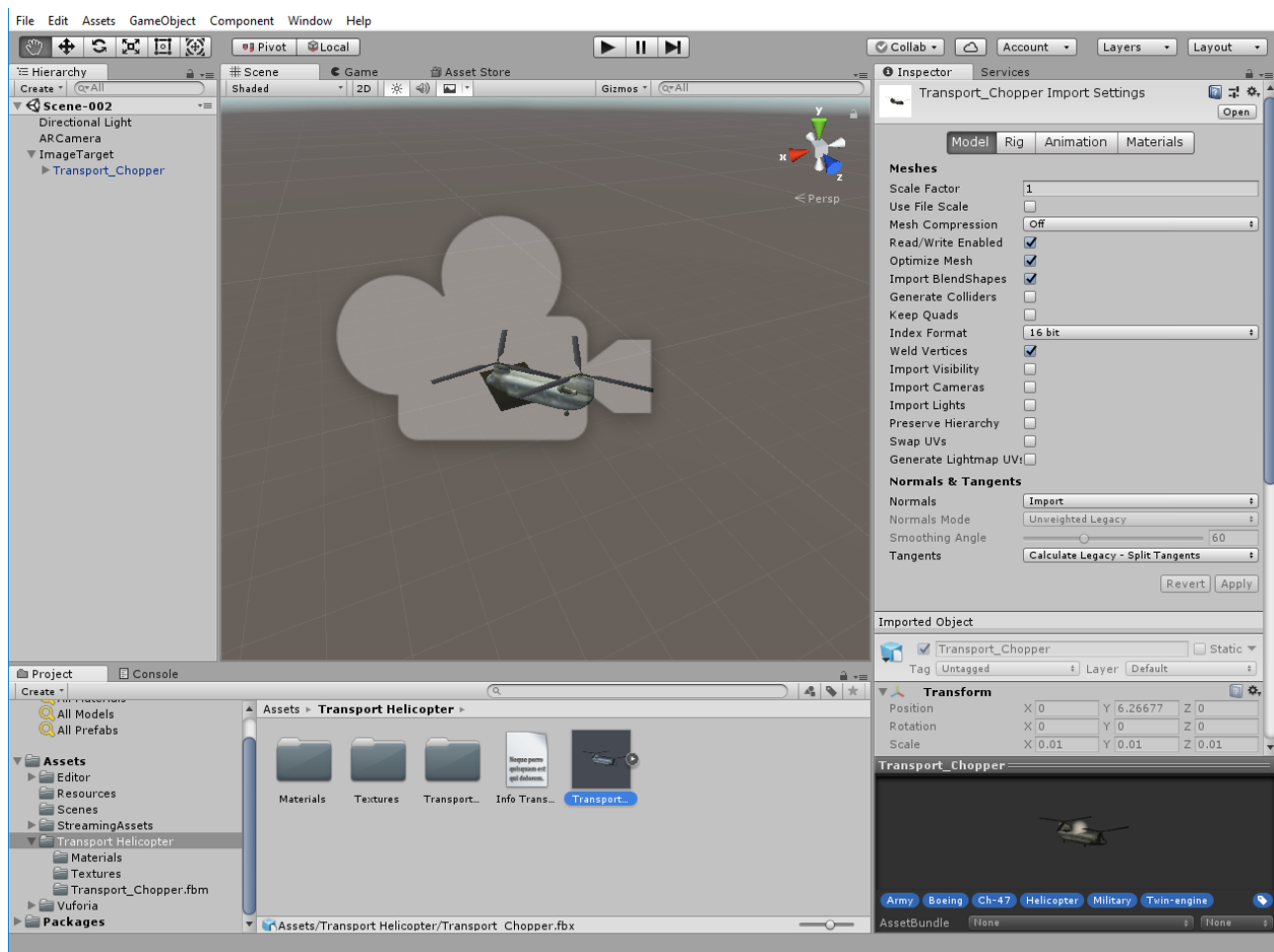
В появившемся меню нажимаем клавишу **Import**.



Результат (в нашем случае – **Transport Helicopter**) будет доступен как элемент списка **Assets**.



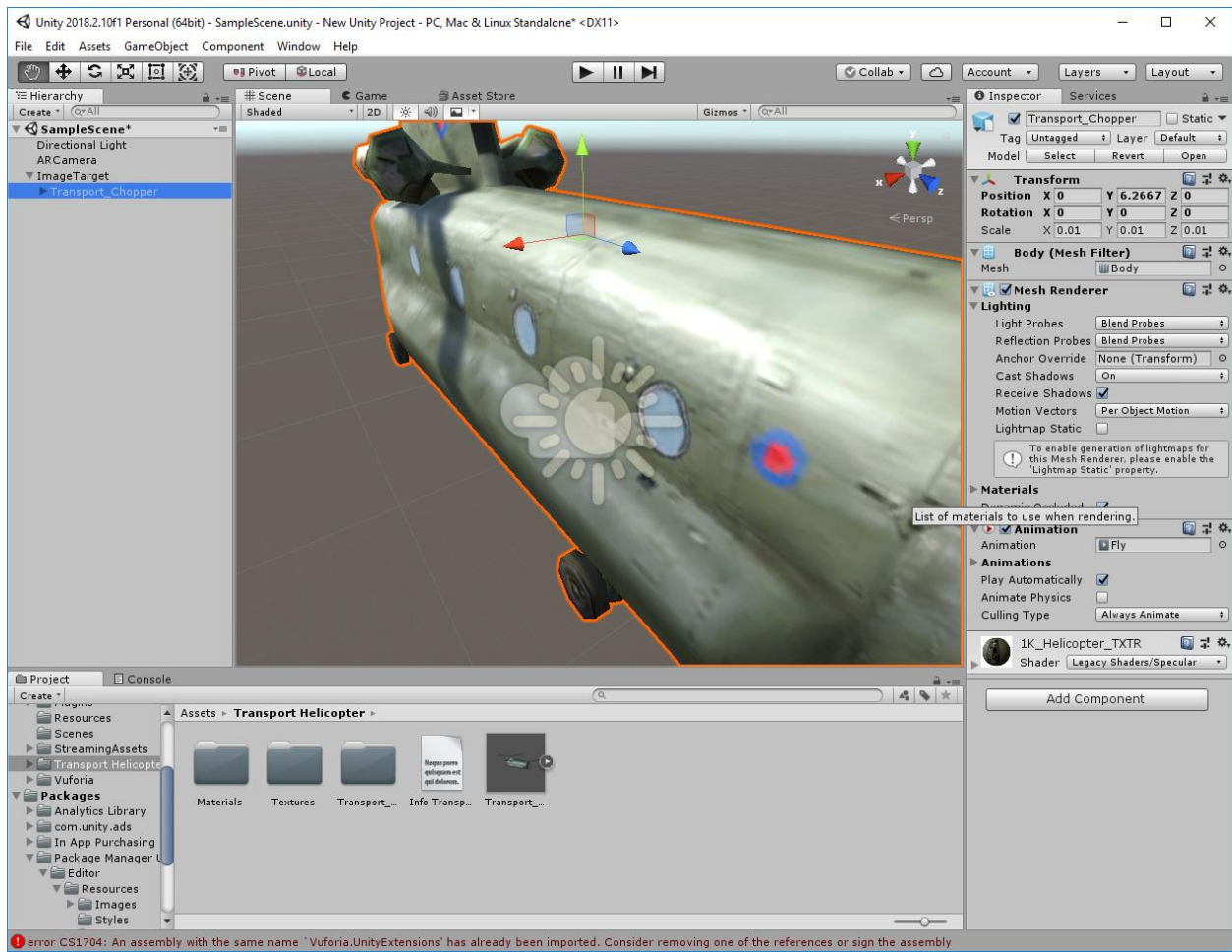
Теперь связываем полученный **3D-объект (Transport Helicopter → Transport Chopper** в формате **.fbx** в поле **Assets**) с зарезервированным таргетом нашего проекта путем перетаскивания (**Drag-n-Drop**) пиктограммы **Transport Chopper** на объект **Image Target** в редакторе сцены (левая часть экрана - **Hierarchy**).



Полученный результат: **3D-модель** из «магазина» плюс все сопутствующие файлы теперь связаны с таргетом нашего проекта.

Выполняем необходимые или желательные преобразования масштаба и взаимного расположения **3D-модели** и таргета, выбирая в сцене связанный с таргетом наш объект - **Transport-Chopper** (обратите внимание, что этот объект относится к типу **Meshes** – что еще раз подчеркивает то обстоятельство, что по умолчанию **Unity 3D** акцептирует поверхностные **3D-модели** в перечисленных в документации форматах: <https://docs.unity3d.com/ru/current/Manual/class-Mesh.html>).

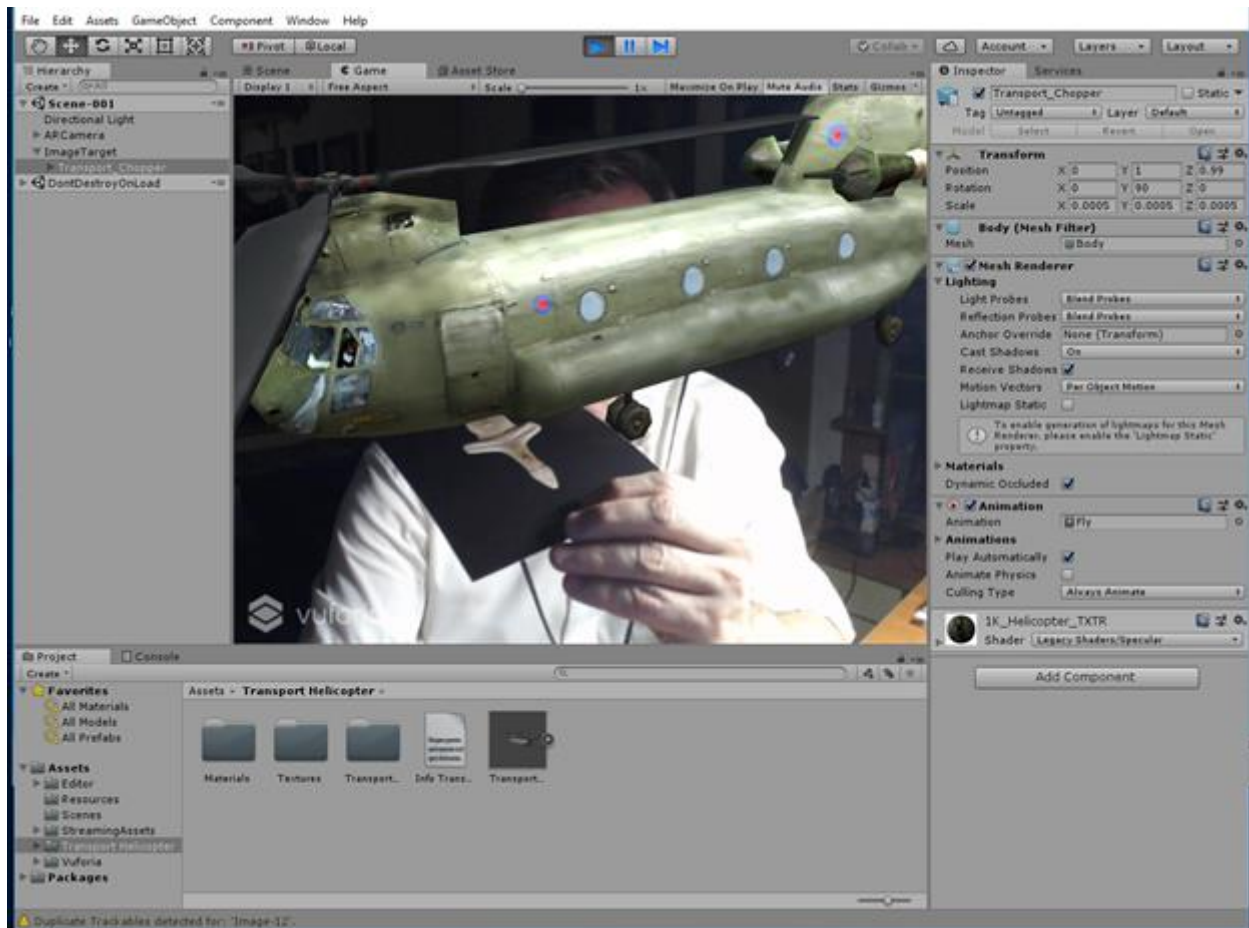
Выполните в инспекторе все нужные преобразования:



Проверка достигнутого в разрабатываемой сцене выполняется в режиме **Game (Play)** с использованием камеры локального компьютера:



Проверяем получаемый результат в режиме **Game**, добиваясь нужного нам соответствия освещённости, перспективы, масштаба и ограничений видимости.



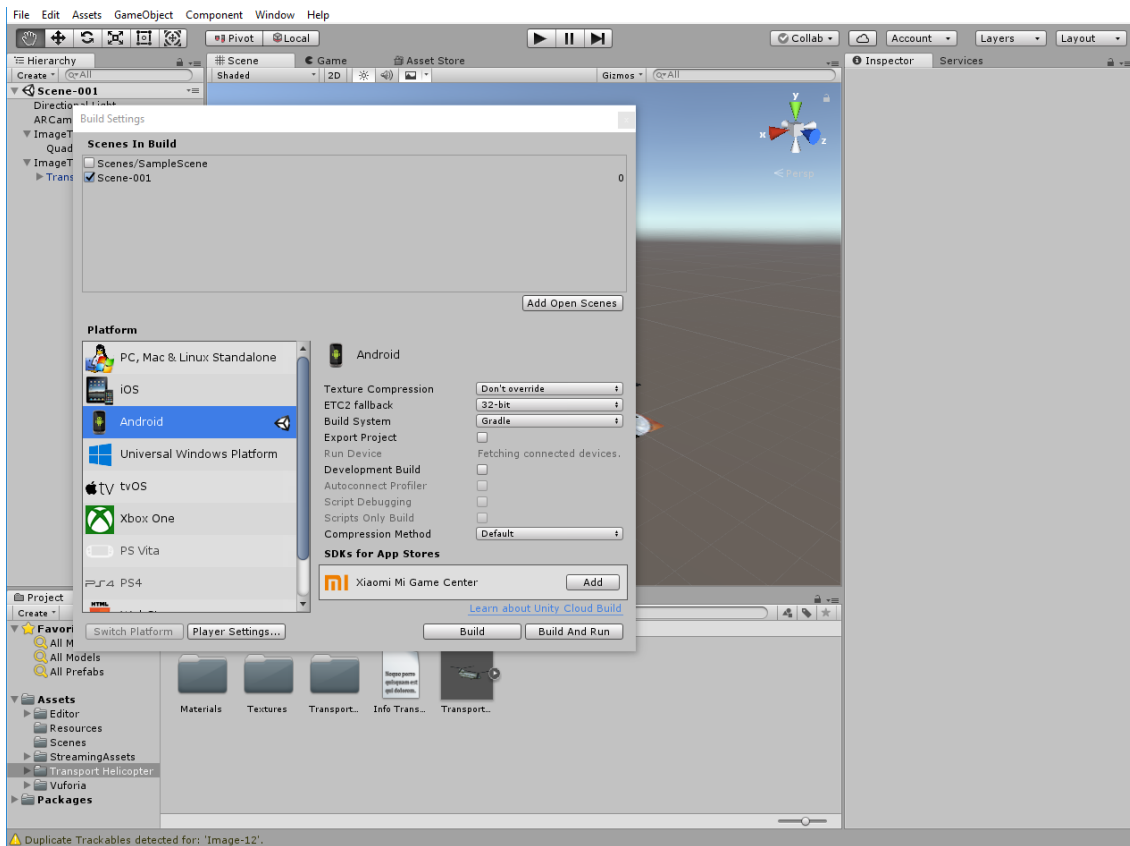
Подготовленную т.о. сцену необходимо преобразовать в загружаемое Приложение ДР для устройства пользователя (в нашем случае – мобильное Android-устройство).

Разработка сцены закончена. 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 Scene** или **File→Save Scene as**: вводим название сцены, например **Scene-001**.

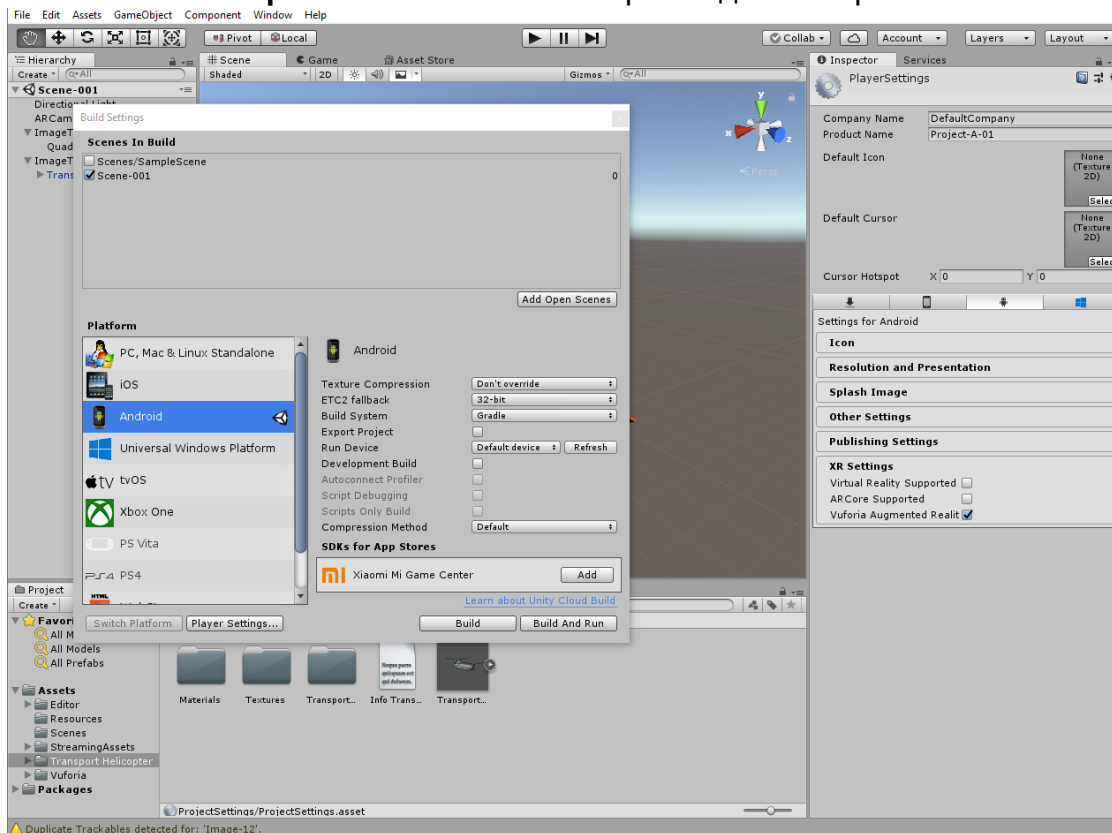
7. Последний шаг выполнения задания в данной ЛР – создание AR Приложения для Android-устройства на основе разработанной сцены в **Unity 3D**.

Для этого включаем в редакторе **Unity 3D** режим **Build**: **File→Build Settings**:

Все предварительные настройки – выбор сцены, настройки **ОС** устройства – осуществляются в открывшемся окне **Build Settings**. Выбираем в этом окне в разделе **Scenes to build** нашу сцену и переходим к настройке работы оборудования в данной ОС → клавиша **Player Settings**.

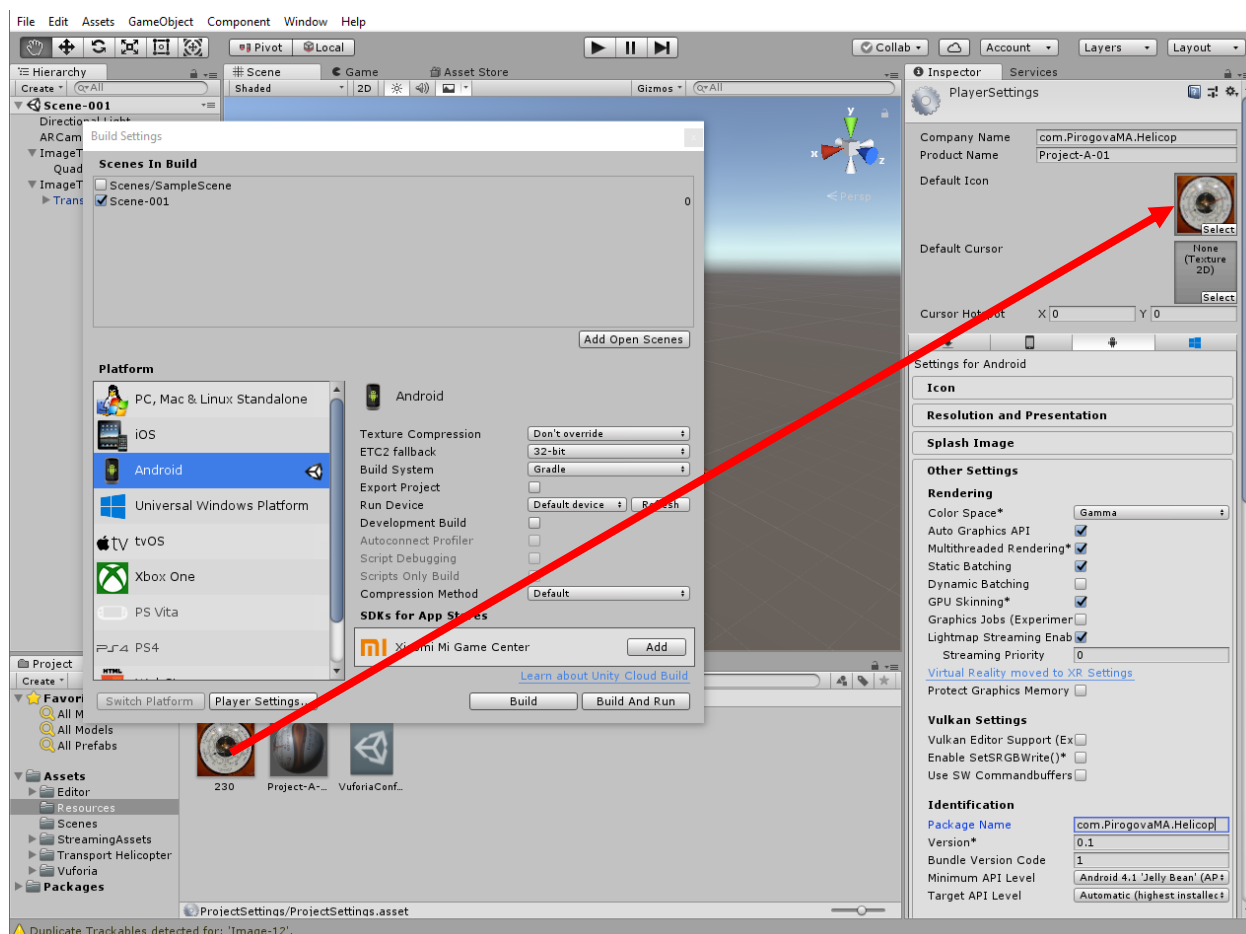


В результате в области **Inspector** появляются настройки для плеера.

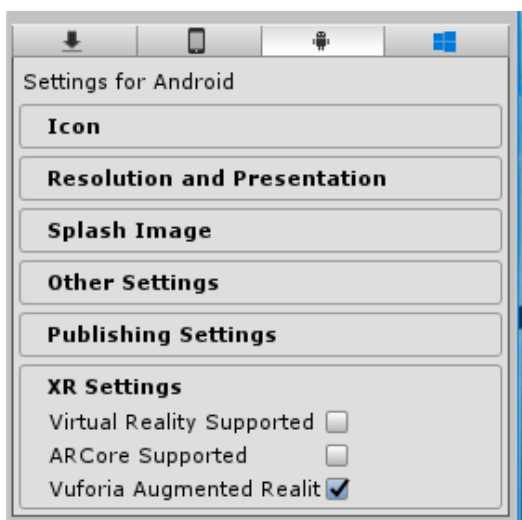


В верхней части **Inspector**'а заполните поле для пиктограммы вашего **AR** Приложения. Обратите внимание, что для выполнения этой операции можно использовать только

операцию **drag-n-drop** из **Assets**→**Resources** для тех объектов, которые ранее были загружены в **Resources** в разделе **Project**. Т.е. о пиктограммах надо позаботиться заранее:



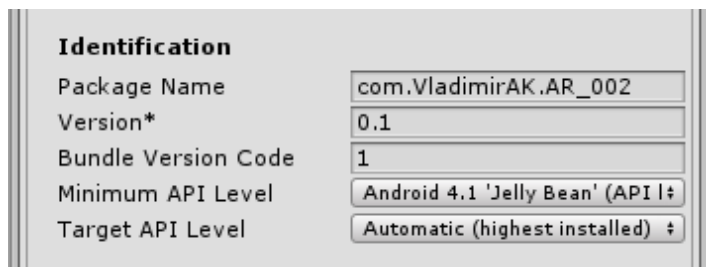
Убедитесь в том, что в **Inspector**'е выполняются настройки для выбранной **ОС Android**, перейдите в раздел **XR Settings** и установите флажок напротив **Vuforia Augmented Reality**:



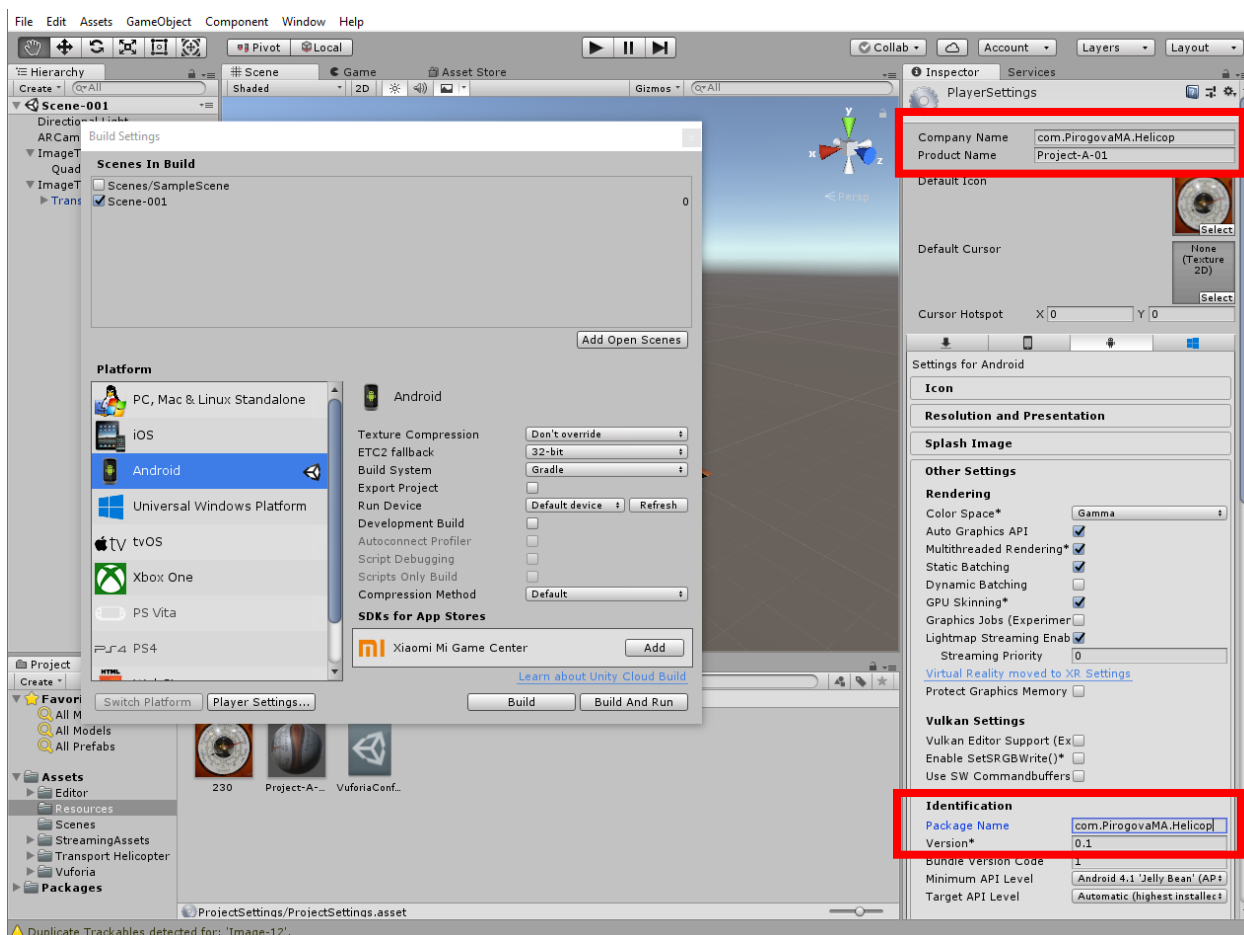
Далее открываем раздел **Other Settings**:

Здесь важно правильное заполнение следующих позиций:

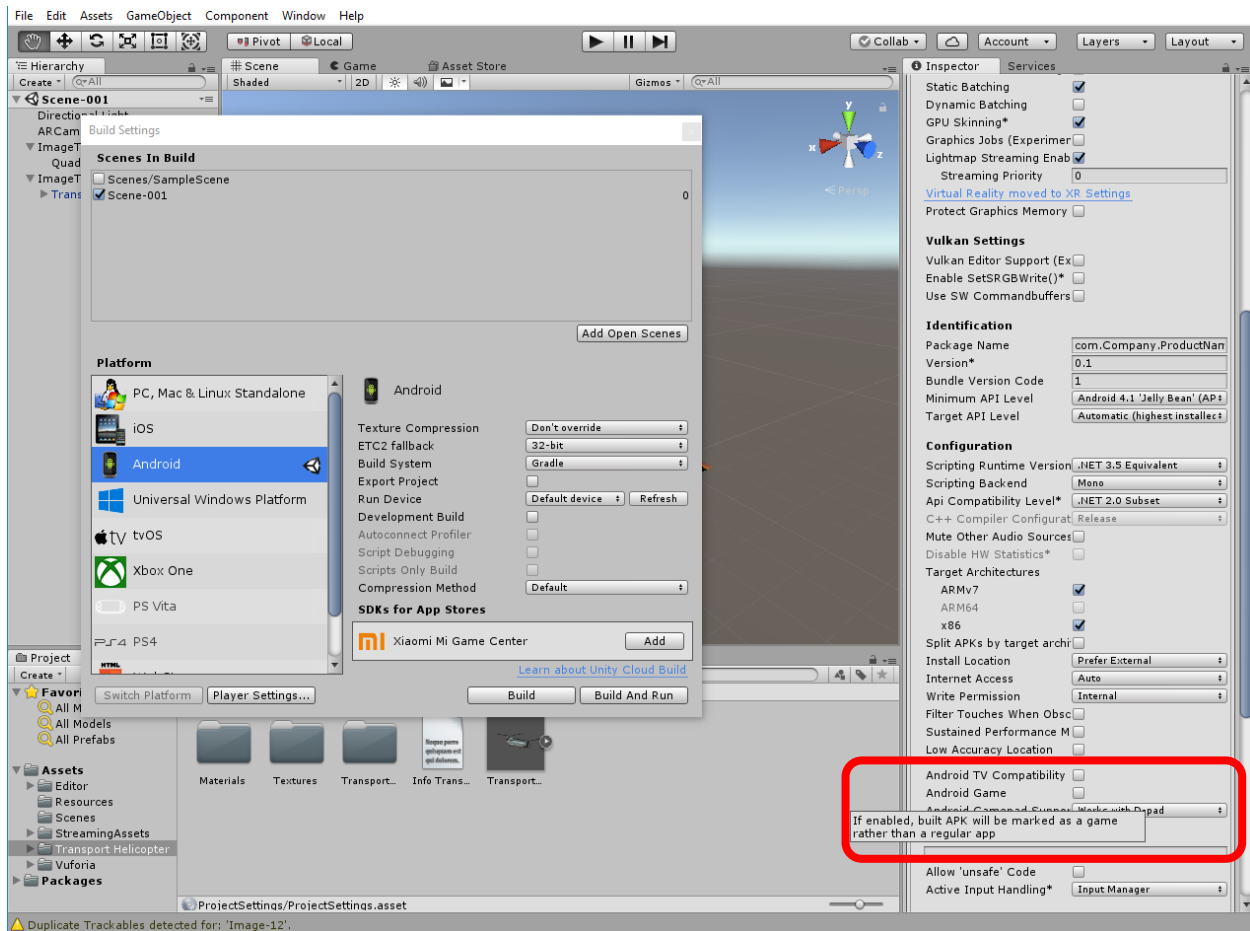
- В подразделе **Identification** –

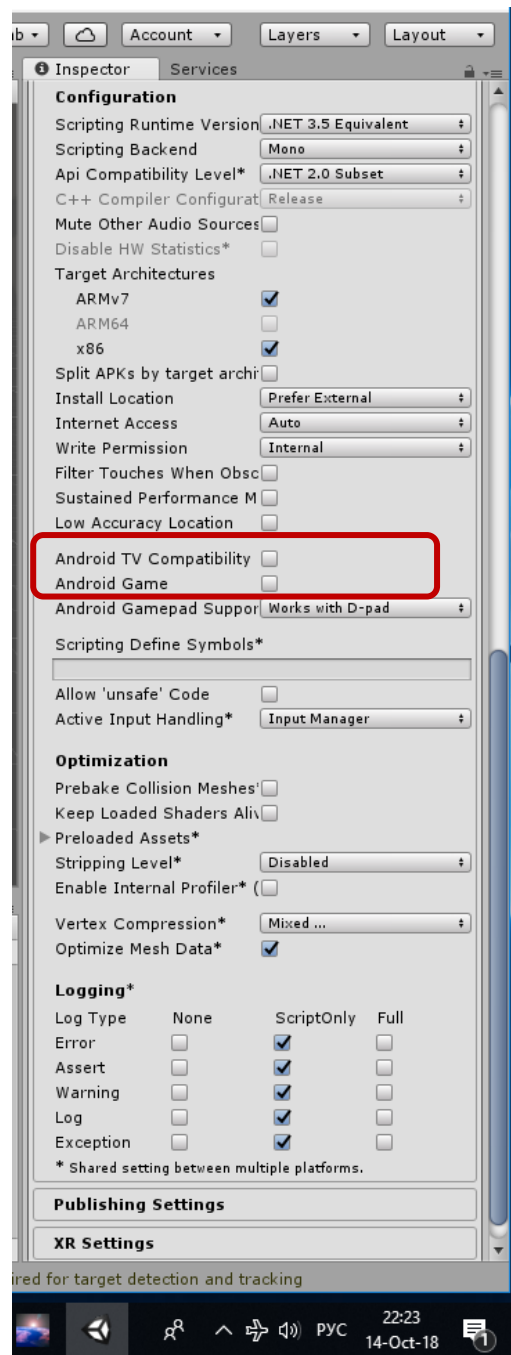


В поле **Package Name** введите имя вашего пакета для размещения его в облаке **Vuforia**. Имя в нашем случае формируется по правилу: **com.<Company Name>.<Product Name>**. И **Company Name** и **Product Name** выбираются разработчиком произвольно, с учетом следующих ограничений: можно использовать только буквы латинского алфавита, цифры и знак «_». Выбранные **Company Name** и **Product Name** размещаются в соответствующих полях в верхней части **Player Settings** в **Inspector**'е.

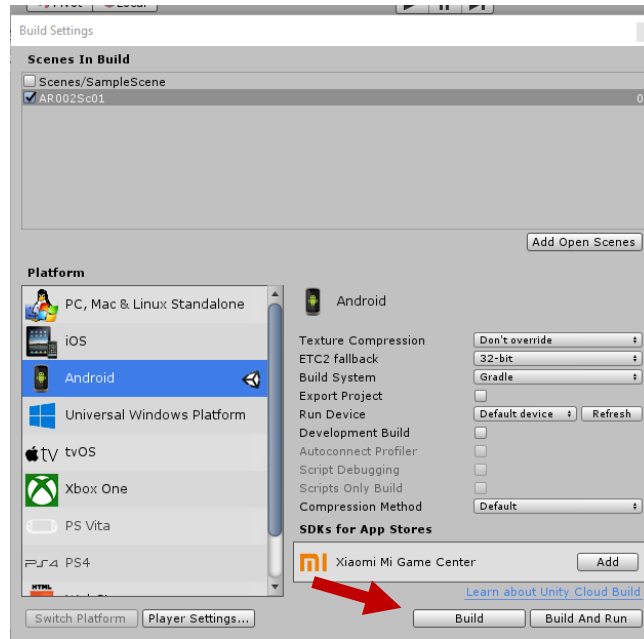


- В подразделе **Configuration** обратите внимание на состояние переключателей – необходимо обязательно снять галочки в переключателях **Android TV Compatibility** и **Android Game**:

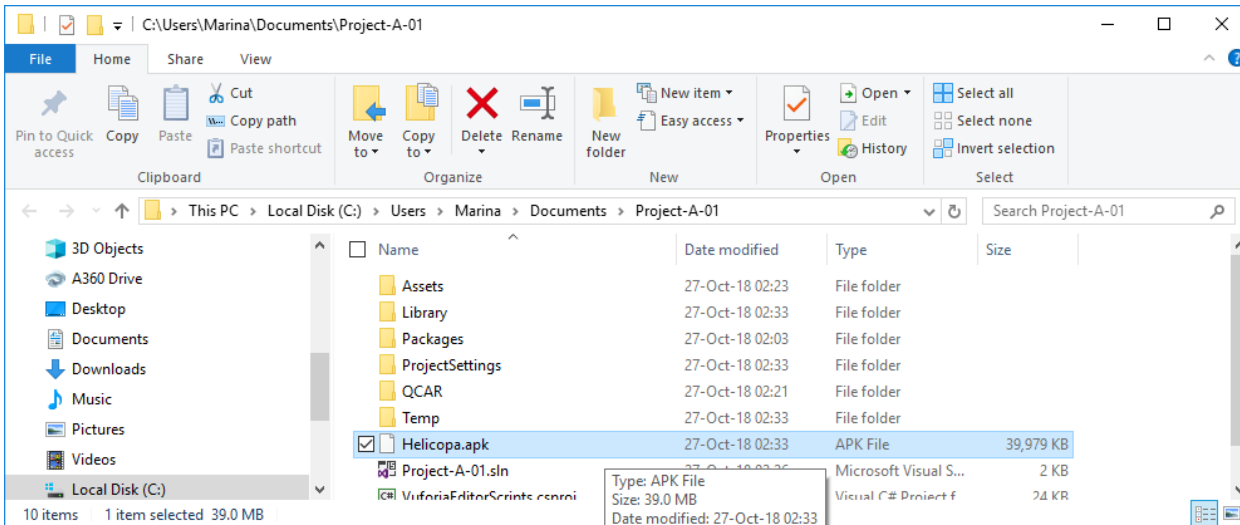
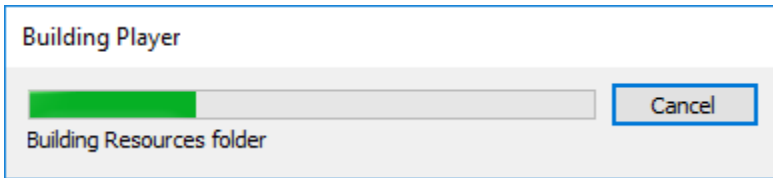




После выполнения всех этих настроек можно выполнить операцию **Build**, нажав на клавишу **Build** в нижней правой части окна **Build Settings**:



В результате вы получаете запрос на размещение получаемого файла **.apk AR** Приложения на вашей локальной машине.



8. Для компиляции **.apk-файла** необходимо заранее установить на локальной машине (Windows) **Java DK** и **Android DK**. Это бесплатные приложения, доступные на сайтах разработчиков (**Oracle** и **Android**). Если этого не было сделано заранее, то придется выполнить эту процедуру в процессе компиляции, отвечая на соответствующие запросы системы. Процесс компиляции может прерываться и по другим причинам. В этом случае необходимо еще раз выполнить **Build** в окне **Build Settings**. Все выполненные предварительно настройки на **OC Android** и **Player Settings** для нашей сцены нашего проекта сохранятся.

Файл **.apk** разработанного AR Приложения сохранен в локальной файловой системе. Теперь его осталось загрузить в **Android-устройство** любым известным вам способом.

И продемонстрировать преподавателю.