



# ЛАБОРАТОРНАЯ РАБОТА №2

## Vuforia Studio: Создание приложения дополненной реальности (часть I из III)

### Оглавление

1. Введение.....	2
2. Интерфейс Vuforia Studio .....	4
3. Разработка простого проекта в Vuforia Studio .....	6
4. Практическое задание .....	75

Дата	Автор	Версия	Описание
13.10.2020	Пирогова М.А.	1	Разработка Описания Лабораторной работы №2

## 1. Введение

Для работы в интернете вещей (IoT) необходимо иметь эффективные возможности по созданию взаимодействия пользователей (людей) и информационных моделей «умных подсоединенных вещей» (**smart connected thing**). Реальная работа с объектами интернета вещей может потребовать большей гибкости использования, большей наглядности представления результатов взаимодействия и более активного использования носимых средств мобильной связи. Для этой цели более всего подходят аппаратные и программные средства, реализующие концепцию **«Дополненной Реальности» (Augmented Reality, AR)**.

В составе программного обеспечения интернета вещей компания **PTC** предлагает платформу для разработки и публикации решений дополненной реальности, не требующее работы с программным кодом - **Vuforia Studio Enterprise Suite**. **Vuforia Studio Enterprise Suite** интегрирован для визуализации и иллюстрирования с платформой Интернета вещей (IoT) **ThingWorx**. Это помогает встраивать компоненты дополненной реальности в подключенные к сети устройства. **Vuforia Studio Enterprise Suite** помогает разрабатывать приложения для создания, эксплуатации и обслуживания интеллектуальных сетевых изделий. ПО содержит все компоненты, необходимые для создания дополненной реальности.

Ядром платформы **Vuforia Studio Enterprise Suite** является рабочее место разработчика – редактор - отладчик решений дополненной реальности **Vuforia Studio (ThingWorx Studio)**. **Vuforia Studio (ThingWorx Studio)** позволяет создавать сценарии работы с виртуальными информационными объектами 3D и 2D в реальной среде индустриального применения. Результат разработки – приложение дополненной реальности – публикуется на «облачном» сервере **ThingWorx Experience Service**, который, в свою очередь, может быть интегрирован в базовый **IoT-сервер** предприятия **ThingWorx**. За счет такой архитектуры любое приложение дополненной реальности, разработанное на платформе **Vuforia Studio**, получает непосредственный доступ ко всем информационным моделям и, таким образом, дает пользователям возможность работать в приложении дополненной реальности с **цифровыми двойниками «умных вещей», обслуживаемых сервисами IoT/IIoT сервера ThingWorx**.

Разработчик имеет возможность использовать **Vuforia Studio** для создания **AR-приложений (проектов, Experience)** как для широкого круга устройств – смартфоны, планшеты, очки дополненной реальности, ноутбуки-трансформеры, так и для широкого круга операционных систем таких устройств – **MS Windows, Android, iOS**. Сама разработка в **Vuforia Studio** основана на применении графического интерфейса пользователя вместе с методами **объектно-ориентированного проектирования программного обеспечения**. При этом от самого разработчика практически не требуется глубоких знаний ни в языках программирования, ни в особенностях интеграции **AR-разработок** с серверами **IoT/IIoT**, поскольку в самой **Vuforia Studio** уже содержится большой **набор настраиваемых шаблонов 3D и 2D компонент AR-разработки («виджеты»)** и присутствует интуитивно-понятное дружественное оформление рабочего пространства, выполненное на основе **Web-интерфейса**. Результат разработки в **Vuforia Studio**, опубликованный на **ThingWorx**-сервере, просматривается пользователями на мобильных устройствах (смартфонах, планшетах, очках дополненной реальности и т.д.), на которые предварительно загружается **AR-просмотрщик Vuforia View**. **Vuforia View** – это бесплатное, доступное для загрузки через **Google Play Market, AppStore или MS Application Store** приложение, объемом ~4 МБ и не требующее особенных дополнительных технических и программных компонент мобильного устройства, кроме базовых.

Применение **Vuforia Studio Enterprise Suite** в качестве платформы разработки **AR-приложений Интернета Вещей (IoT) и промышленного Интернета Вещей (IIoT)** значительно упрощает и сокращает сам процесс разработки **AR-приложения**, а за счёт облачного размещения результата разработки – облегчает доступ к самому **AR-приложению** в любом месте при наличии интернет-соединения.

В данной лабораторной работе обучающимся предлагается:

1. Разработать **AR – проект (Experience)** для варианта привязки («таргетинга») **ThingMark** дополненной реальности и реальной обстановки, с использованием **3D- и 2D-виджетов** (базовый набор)
2. Созданный **AR – проект (Experience)** должен содержать объекты, предусматривающие связывание с ним информационной модели «умная вещь», трансляцию свойств «умной вещи» в качестве информационного наполнения дополненной реальности.
3. Для освоения интерфейса и практического изучения имеющегося в распоряжении у разработчика набора настраиваемых шаблонов **3D и 2D компонент AR-разработки («виджетов»), приемов настройки рабочего (координатного) пространства и графического программирования,** в создаваемом **AR – проекте (Experience)** необходимо разработать:
  - **2D Интерфейс** для визуализации отдельных элементов контента, изменения их видимости/прозрачности, масштабирования и вращения и т.д.
  - **Кроме того, рядом** с объектом должны располагаться плоские «экраны», на которых будет выводиться информация об объекте, например, спецификация принтера и его рекламное представление – изображение, логотип компании и пр.
4. Разработка ведется обучающимися в соответствии с индивидуальным набором вариантов контента (объектов) ДР для заранее подготовленной **3D-модели** технического устройства **3D-Принтер MakerBot Replicator 2**. Набор вариантов для индивидуальных заданий представлен в Разделе «**Задания на Лабораторную работу**».
5. Разработанный **AR – Проект** в рамках данной ЛР **не публикуется на сервере**, а лишь сохраняется в созданном Проекте. Его работоспособность оценивается преподавателем в режиме предварительного просмотра средствами редактора **Vuforia Studio**.

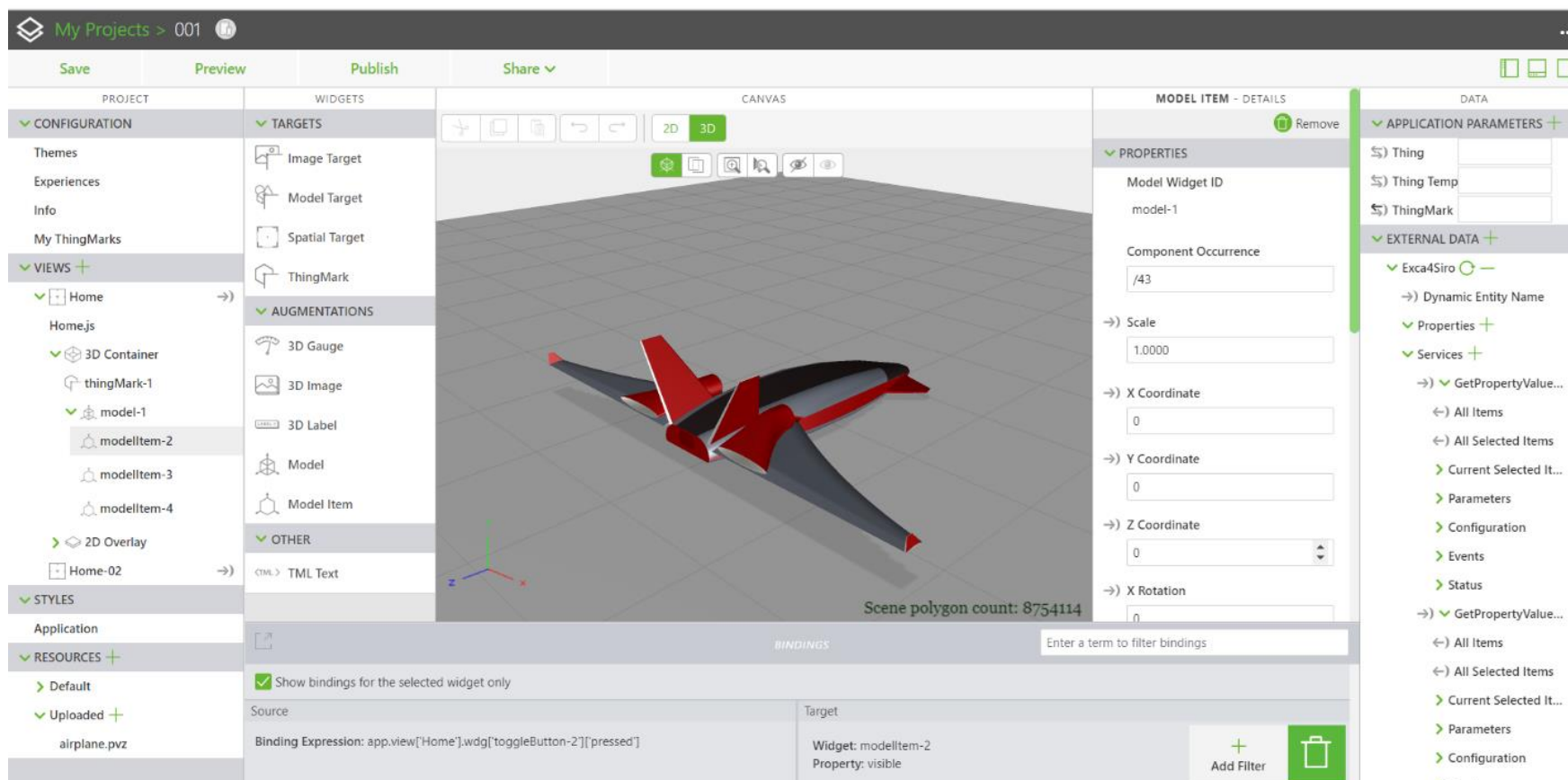
## 2. Интерфейс Vuforia Studio

Интерфейс **Vuforia Studio** построен на основе принципов оконного интерфейса **Web-приложений**.

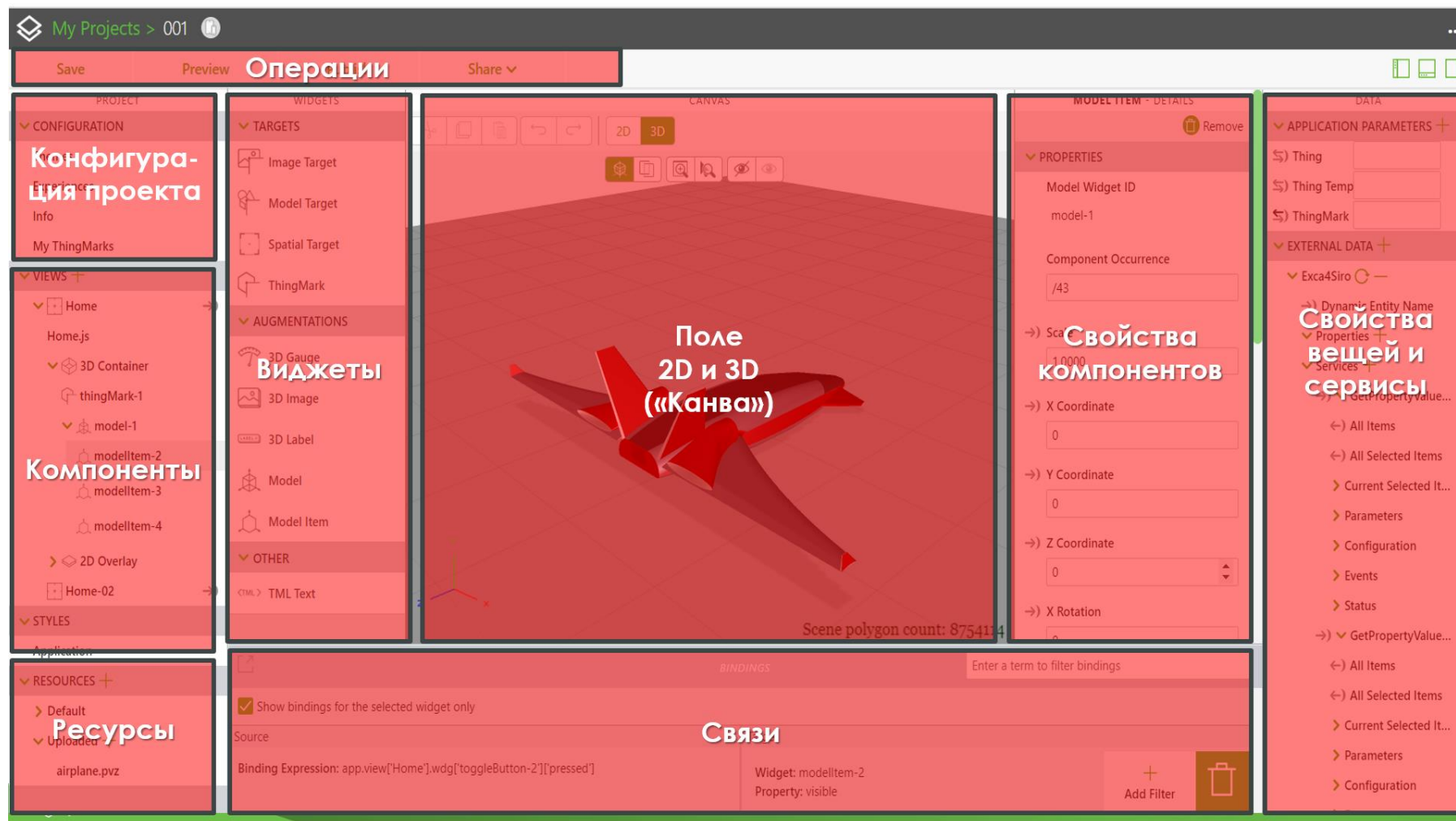
Для получения доступа к интерфейсу **Vuforia Studio** необходимо в окне **URL (URI)** стандартного **Web-просмотрщика** (**PTC** рекомендует преимущественное использование **Google Chrome**) ввести адрес и порт сервера **Vuforia Studio**. Если **Vuforia Studio** установлена на локальную машину разработчика, то в поле **URL (URI)** просмотрщика вводится →

**localhost:3000**

В результате разработчик **AR-приложения** получает возможность создавать **AR-контент в интерфейсе Vuforia Studio**. Типичный вид интерфейса [редактора] **Vuforia Studio** представлен ниже:



Работа над каждым отдельным приложением оформляется в виде проекта **Vuforia Studio**, функциональные компоненты проекта пользователь выбирает из набора **3D или 2D виджетов**, помещая их в «канву» (**CANVAS**) проекта. Состав элементов интерфейса при работе с **редактором Vuforia Studio** представлен на рисунке ниже:



Боле подробно элементы интерфейса рассмотрим в процессе разработки и публикации **AR-приложения**.

### 3. Разработка простого проекта в Vuforia Studio

**Первая задача:** создать приложение дополненной реальности - **AR – проект (Experience)** - для работы по обнаружению зарегистрированного устройства: **3D-Принтера MakerBot**. Для определения того, является ли данный **3D-Принтер MakerBot** именно тем устройством, которое зарегистрировано, используем специализированный **target - ThingMark**, содержащий в том числе аналог **QR-кода** регистрации. Кроме того, для визуального сравнения будем использовать в приложении дополненной реальности **3D-модель** именно искомого устройства **MakerBot**.

#### Шаг 1.

Загружаем и запускаем **Vuforia Studio** на компьютере под управлением **MS Windows 7** или **MS Windows 10** или **iOS**.

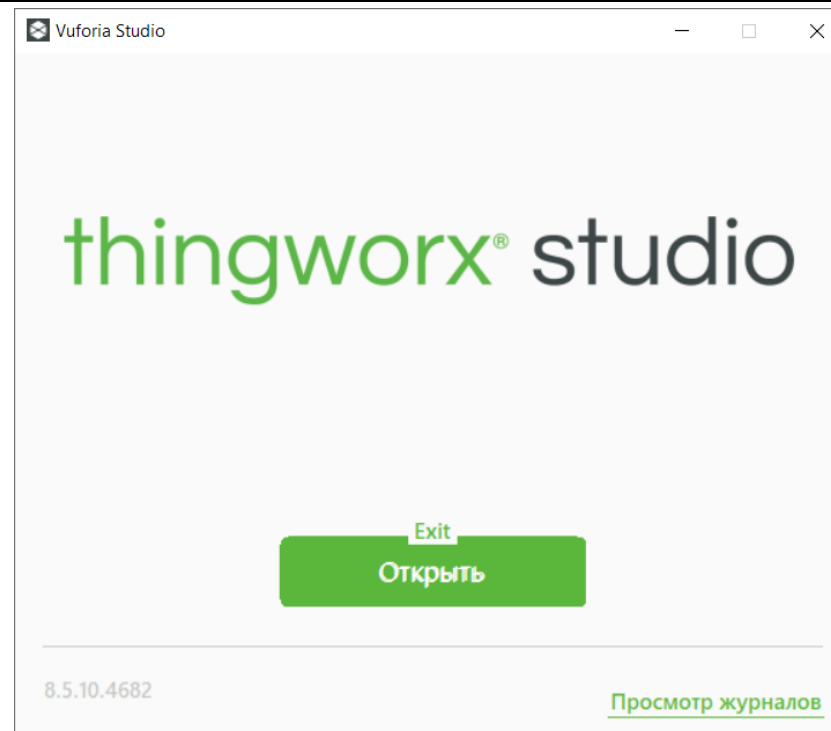
Дожидаемся появления окна входа.

При нажатии на кнопку

Открыть

Автоматически запускается **Web-просмотрщик** «по умолчанию» на этом же компьютере (желательно, по рекомендации **ПТС**, чтобы это был **Google Chrome**), в поле **URL (URI)** которого автоматически устанавливается адрес **localhost:3000**

**ВАЖНО!!** Предлагается прописать указанный адрес вручную в поле **URL(URI) Google Chrome**, не закрывая окно **thingworx studio** и не нажимая кнопку «открыть», чтобы не стартовать **Vuforia Studio** в браузере, установленном у вас по умолчанию.



## Шаг 2.

Появляется окно выбора проектов

### **Дополненной Реальности.**

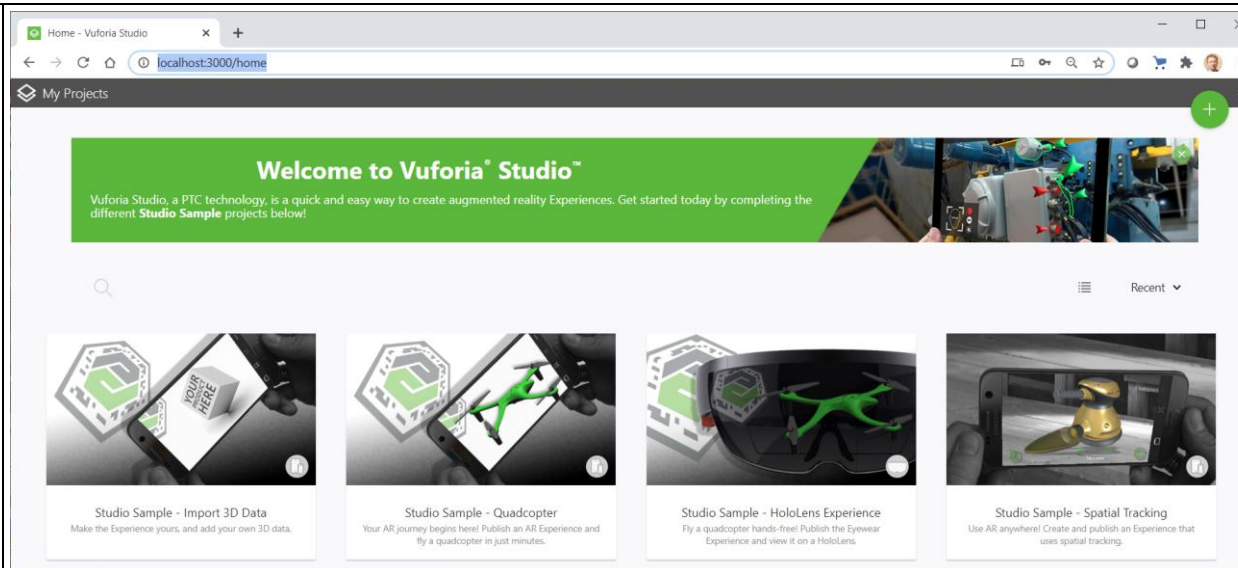
Даже при первом старте **Vuforia Studio** у разработчика имеется возможность работать с четырьмя примерами проектов – **Studio Sample**. Это заранее подготовленные примеры для различных типов конечных устройств и различных методов таргетинга (топологической привязки виртуальных информационных объектов к реальному окружению).

**После полноценной регистрации** вы можете самостоятельно выполнить разработку любого из этих подготовленных проектов, поскольку каждый из них сопровождается пошаговым комментарием. Наша задача – создать свой собственный проект и выполнить разработку в рамках этого проекта своего примера **AR-приложения – Experience**.

**Для старта нового проекта нажимаем на кнопку**



в правом верхнем углу

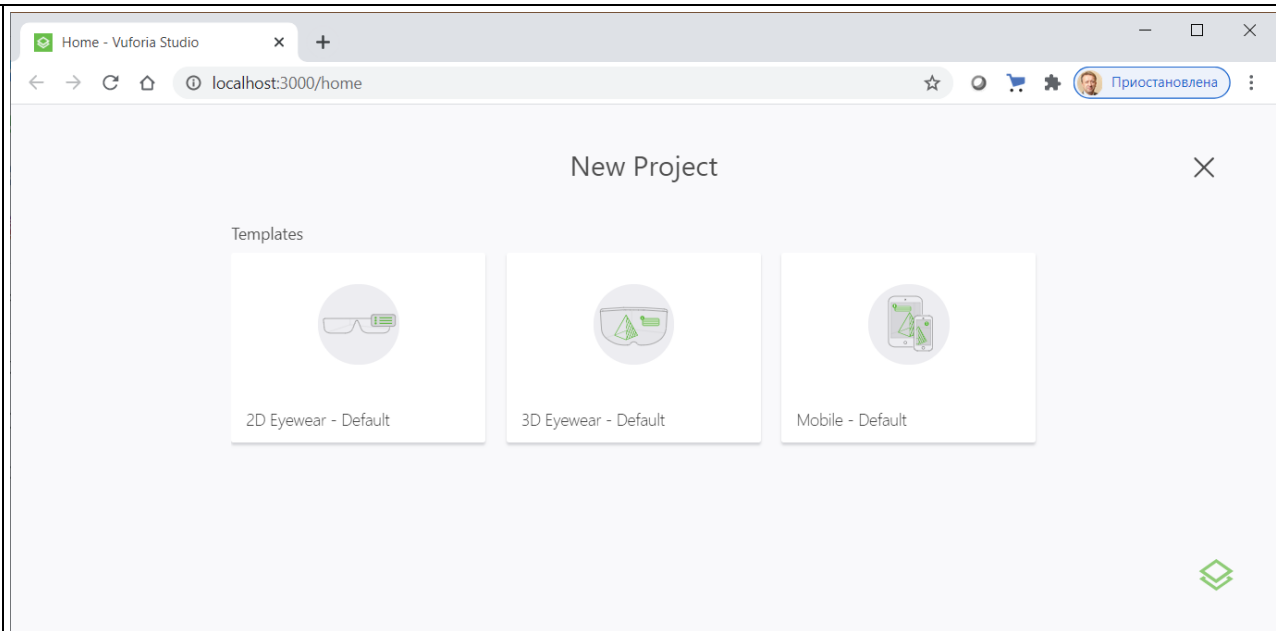


### Шаг 3.

Окно создания нового проекта предлагает выбор типа проекта:

- **AR**-приложение для монокуляров дополненной реальности (**realwear HMT-1 и HMT-z1z**) – проекты типа «**2D Eyewear**» или ещё «**assisted reality**»-проекты;
- **AR**-приложение для любых смартфонов, планшетов а также ноутбуков-трансформеров Yoga – проекты типа «**Mobile**» или ещё «**augmented reality**»;
- **AR**-приложение для очков дополненной реальности **Microsoft Hololens и Hololens2** – проекты типа «**3D Eyewear**» или ещё «**mixed reality**».

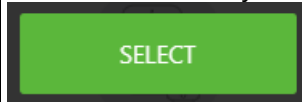
**Воспользуемся наиболее** востребованным и массовым типом проекта – «**augmented reality**» для самого широкого круга устройств просмотра дополненной реальности – «**Mobile**»



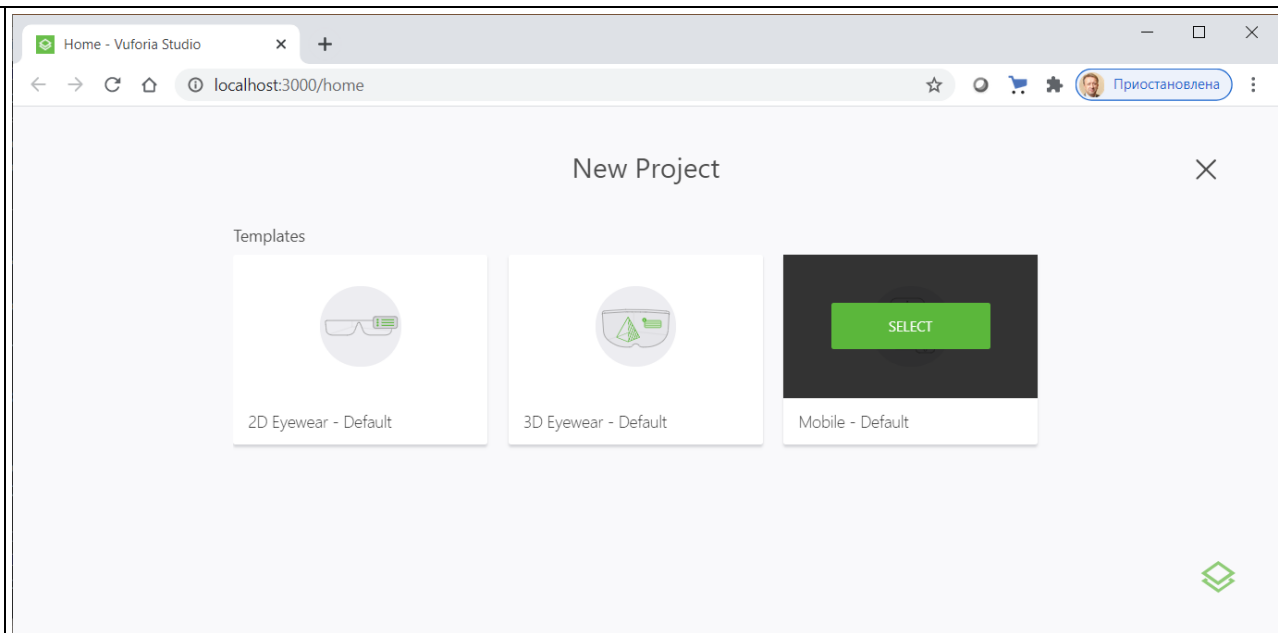


## Шаг 4.

Нажимаем кнопку



И выбираем тип проекта «Mobile»



## Шаг 5.

В появившемся окне **New Project** заполняем поля:

- **Project Name:**

Вводим имя нашего нового проекта.

Например, **AR-001**

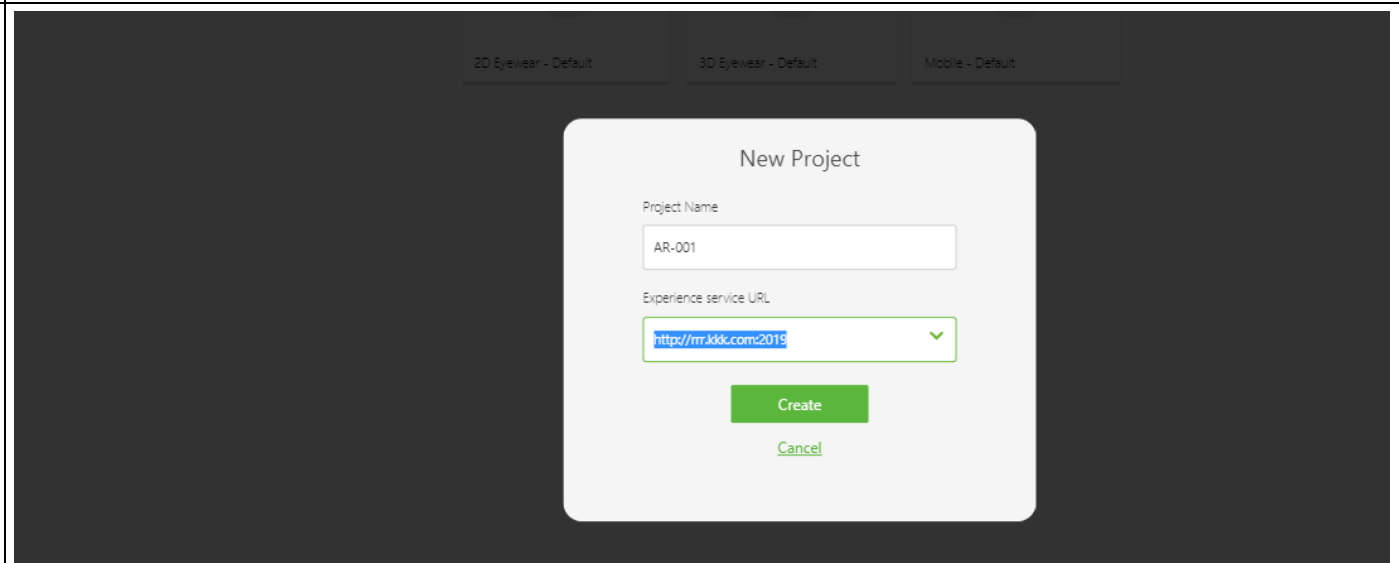
- **Experience service URL:**

В этом поле выбираем нужный нам сервер из списка доступных.

В нашем случае выбираем [несуществующий] сервер

<http://rrr.kkk.com:2019>

**Вспомните, это ли имя вы использовали в ЛР №1 или придумали какое-то свое?**

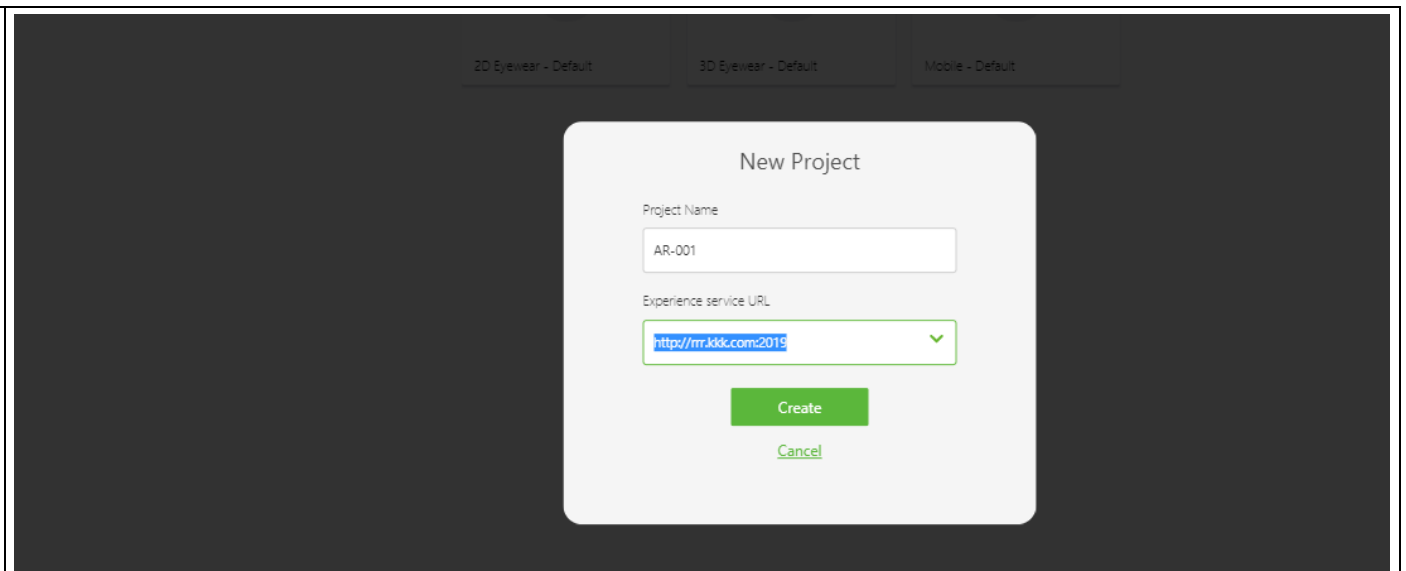


## Шаг 6.

Нажимаем кнопку

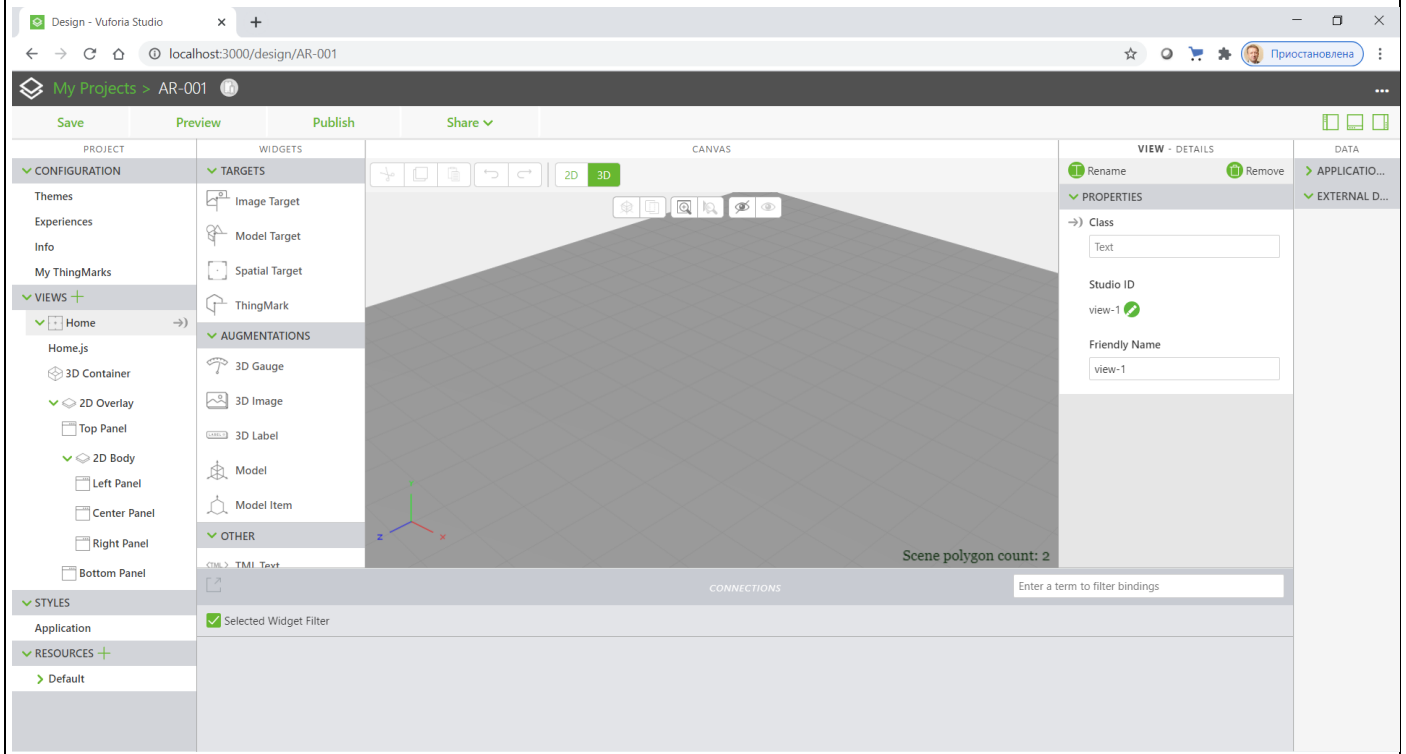
Create

И переходим к новому проекту



## Шаг 7.

В **Web-просмотрщике** открывается стандартный интерфейс **Vuforia Studio**.



## Шаг 11.

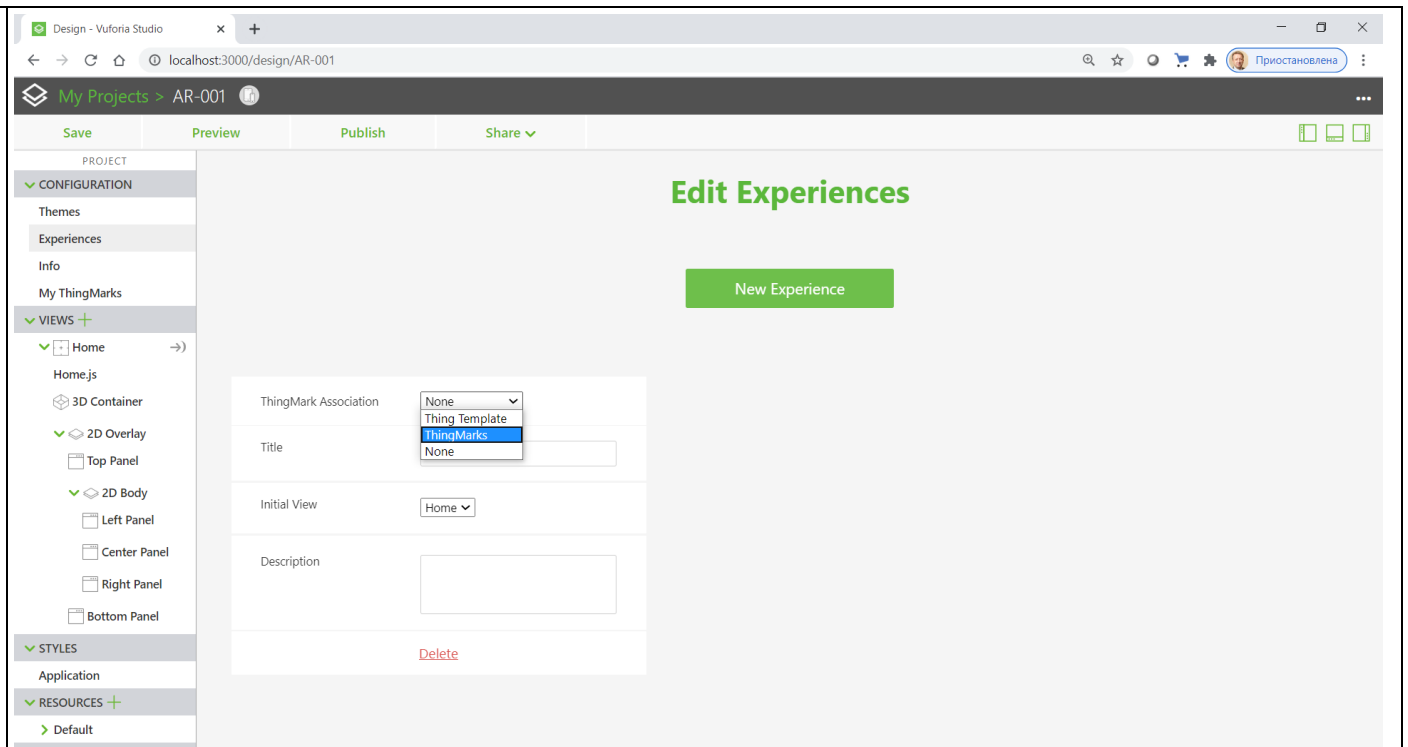
В зоне интерфейса «Конфигурация проекта» выберем пункт «Experience» и заполним необходимые поля.

Для нашего AR-приложения мы будем использовать тип таргетинга «ThingMark» (по специальной метке, содержащей специальный QR-код нашего устройства)

Для этого в разделе

«ThingMark Association»

выберем из выпадающего списка позицию **ThingMark** →



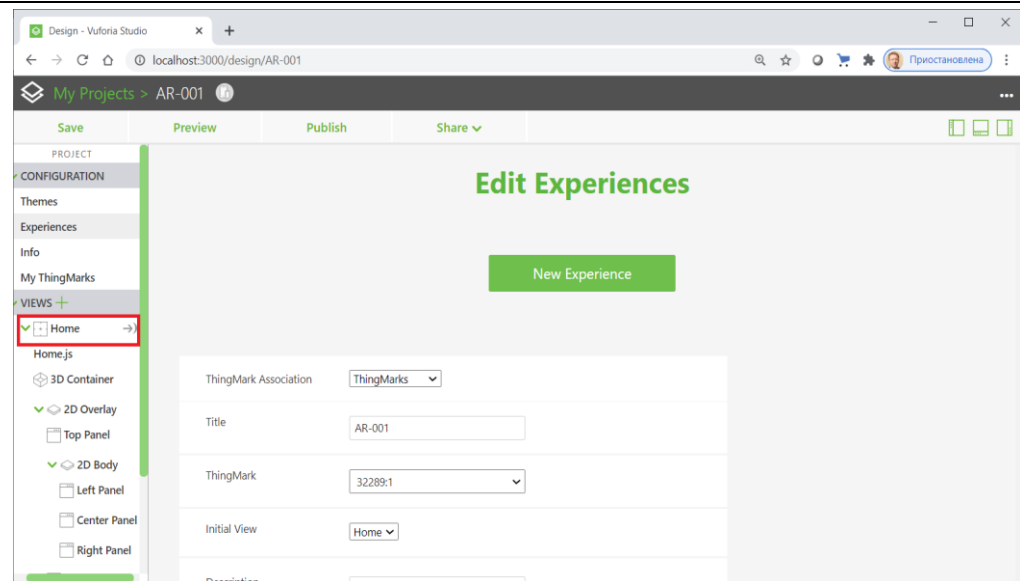
## Шаг 12.

Определение нашего проекта завершено.

Переходим к разработке AR-приложения.

Для этого в зоне интерфейса

«Компоненты» нажимаем позицию **Home**



## Шаг 13.

Мы находимся в режиме работы с **3D-объектами – 3D-канва**.

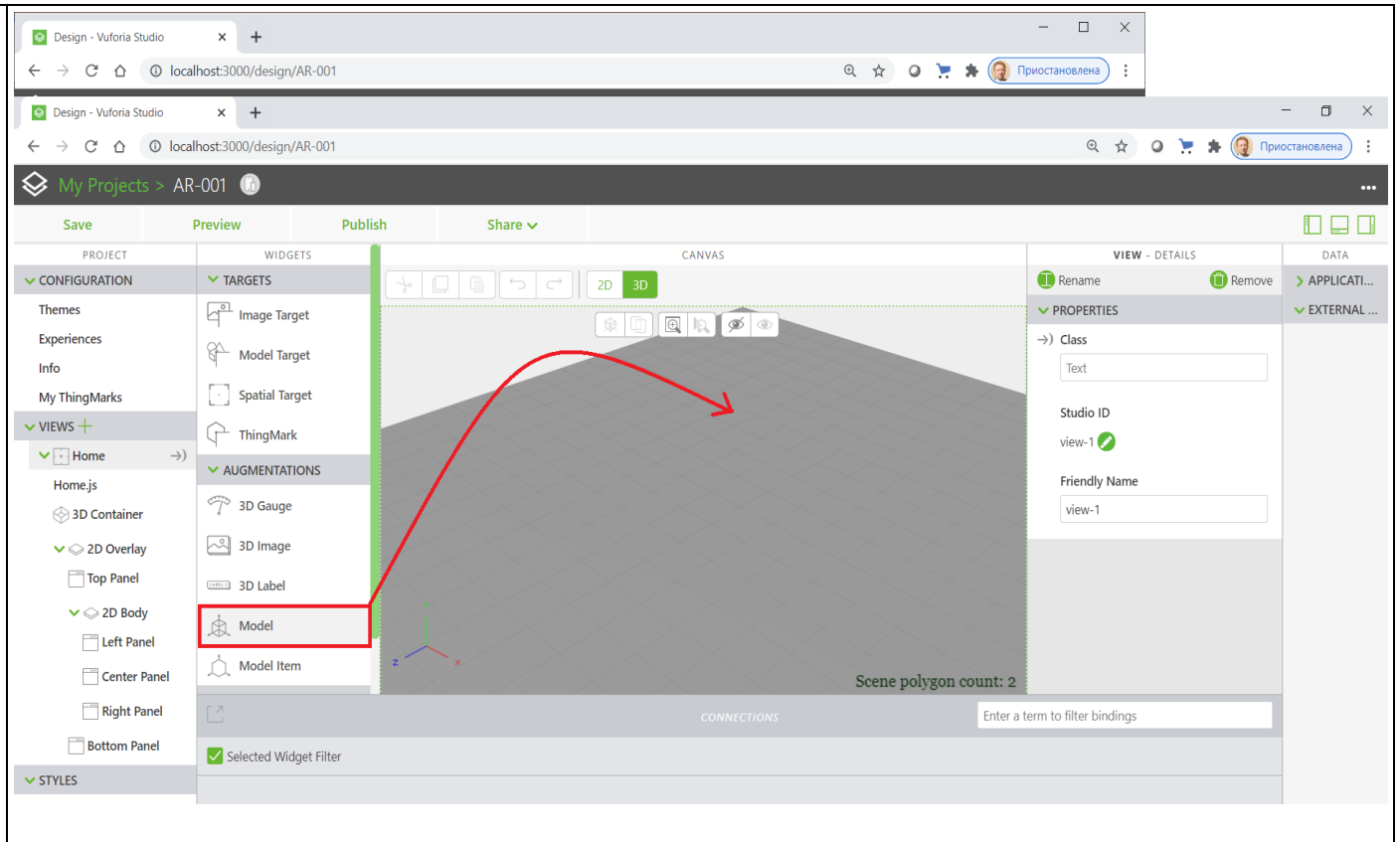
Эта канва используется как инструмент создания/редактирования **3D-сцены**.

Прежде всего на эту сцену надо поместить наш виртуальный объект, который будет в дальнейшем визуализироваться в **AR-Приложении**.

Для этого сначала разместим контейнер нашего **3D-объекта на канве 3D**. В качестве такого контейнера из доступных будем использовать виджет **Model** → Выполняем следующее:

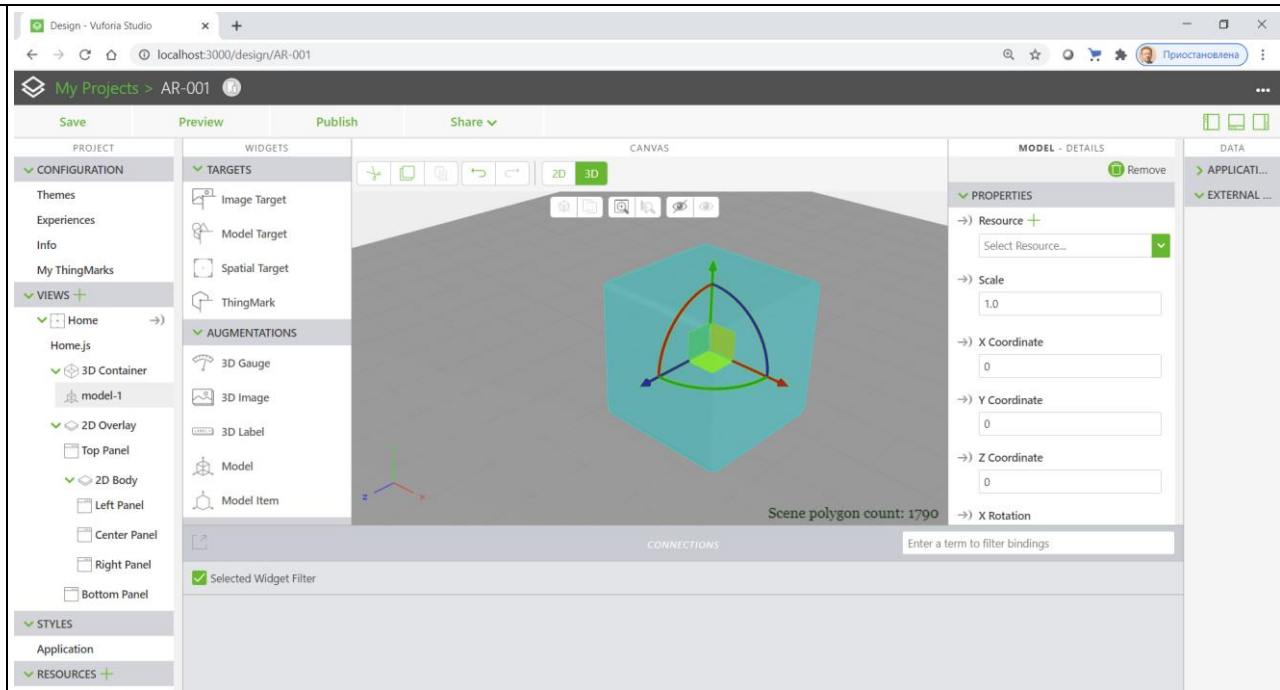
В зоне интерфейса «**виджеты**» находим виджет **Model** и перетаскиваем его «**drag-n-drop**» в зону интерфейса «**канва**».

**Обращайте внимание на изменение наполнения зон «Компоненты» и «Свойства компонентов» после выполнения этих действий.**



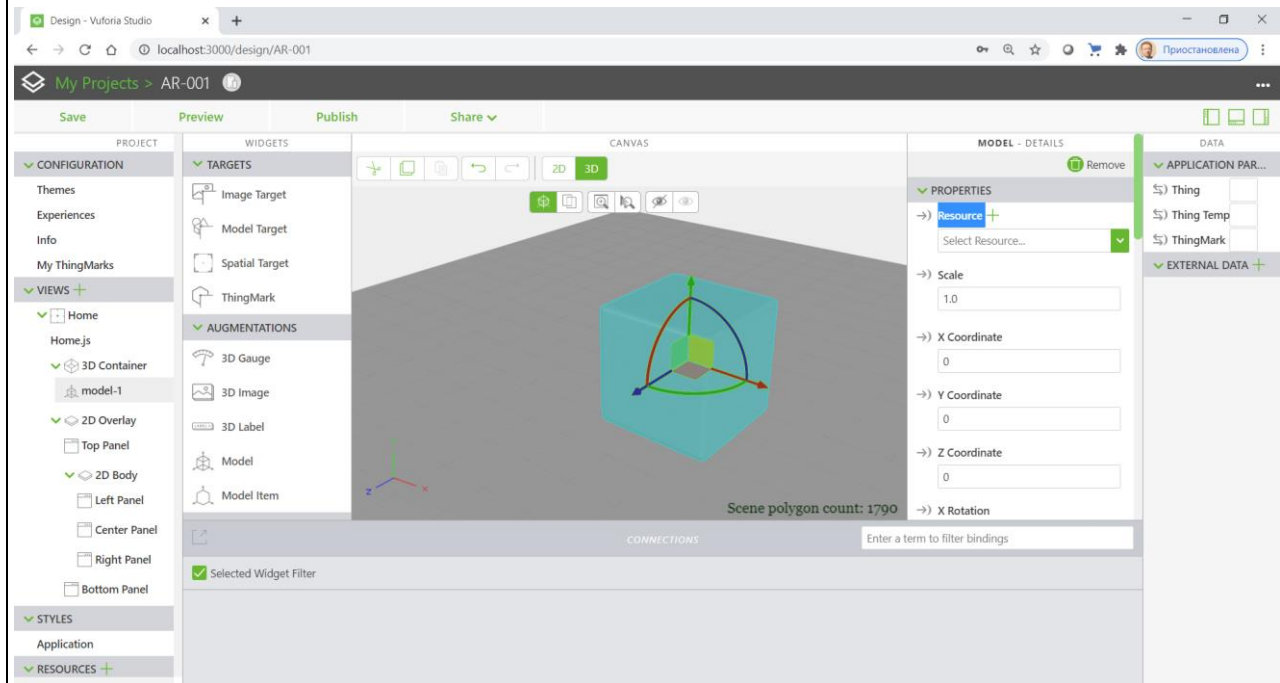
## Шаг 14.

На **3D-канве** разрабатываемого приложения помещён контейнер для виртуального объекта нашего проекта



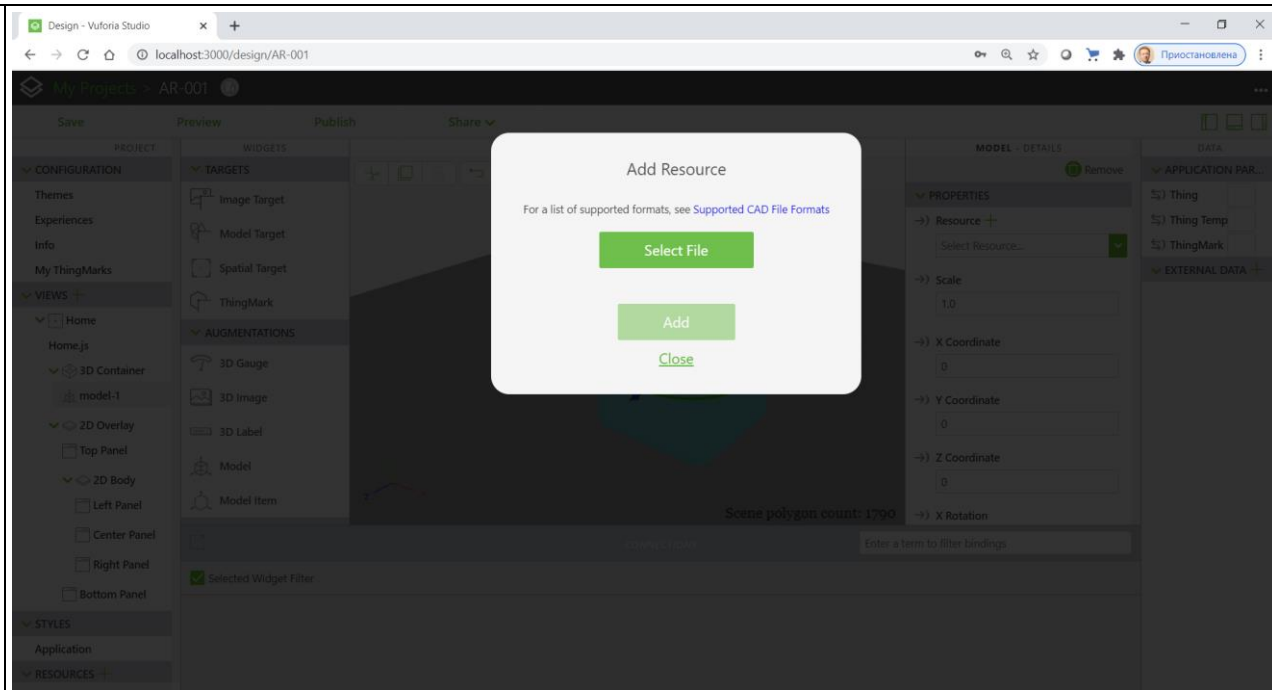
## Шаг 15.

В зоне интерфейса «**Свойства компонента**» выбираем доступную **3D модель** виртуального объекта, а если такового нет, то ищем его в доступной файловой системе. Для этого нажимаем на символ «+» справа от **Resource** в зоне «**Свойства компонента**».



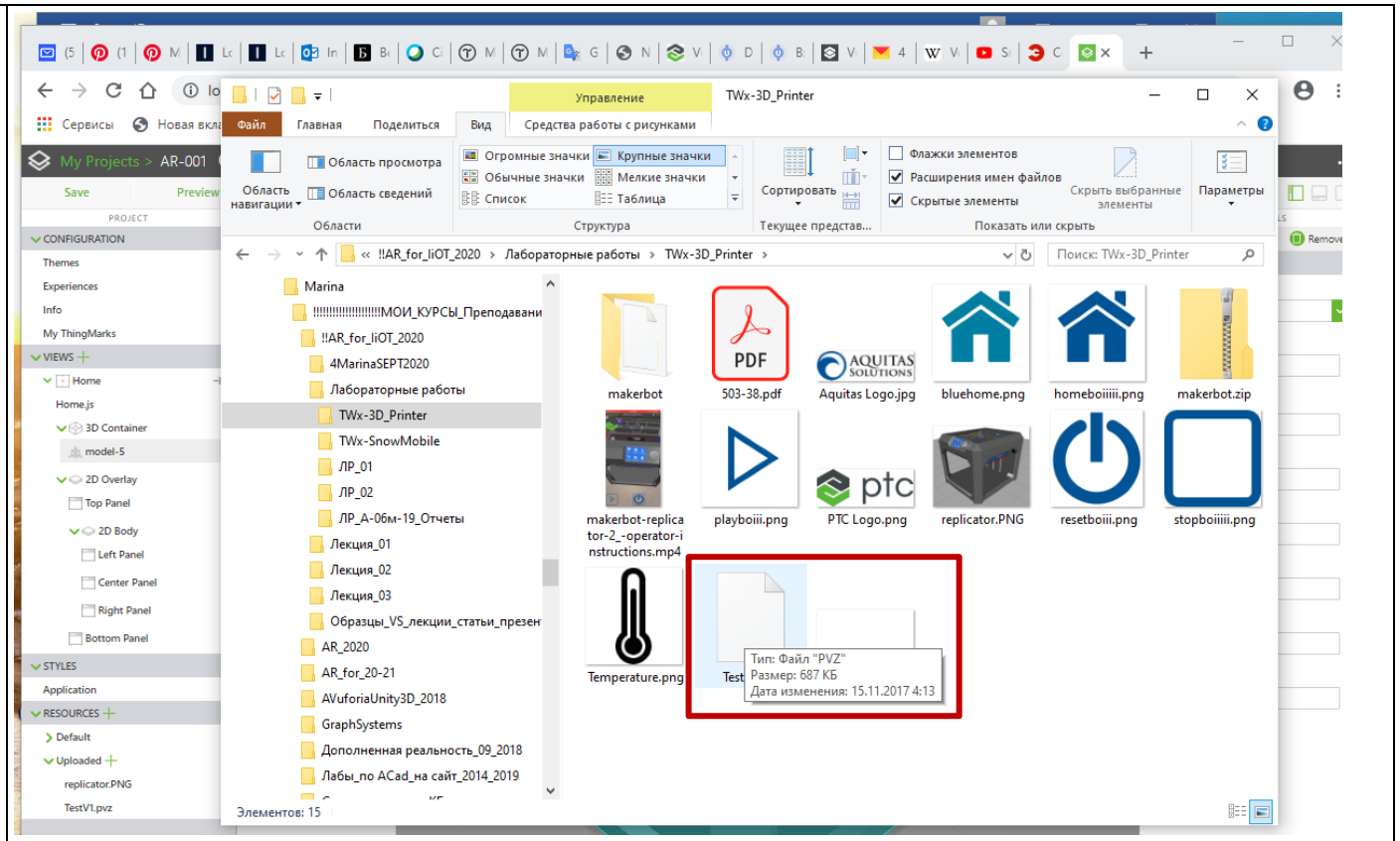
## Шаг 16.

В появляющемся окне «**Add Resource**» нажимаем «**Select File**» для выхода в локальную файловую систему, где должен быть сохранен объект для размещения. В нашем случае – **3D Модель 3D-Принтер MakerBot**



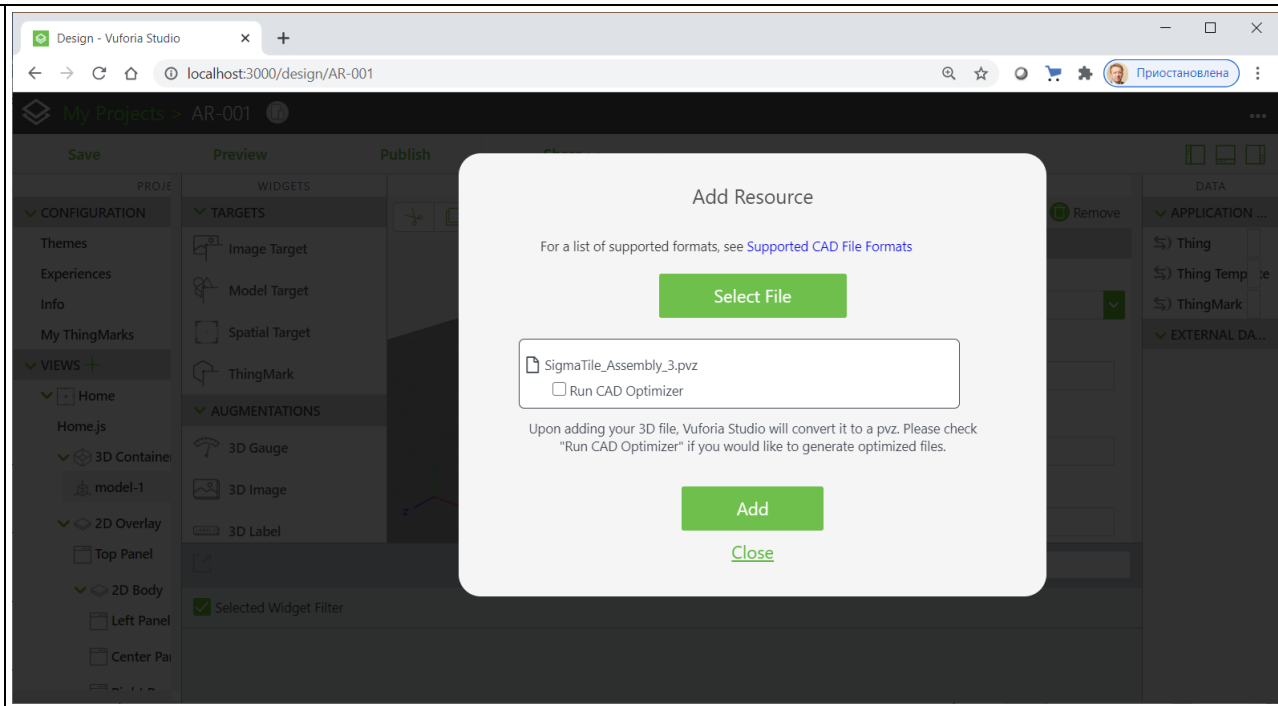
## Шаг 17.

Используя навигатор по файловой системе находим необходимый нам архив моделей (в нашем случае - **TestV1.pvz**). Это формат представления является «родным» для **Vuforia Studio** и может интегрироваться в систему без преобразования.



## Шаг 18.

Загружаем выбранный файл - нажимаем кнопку «Add»





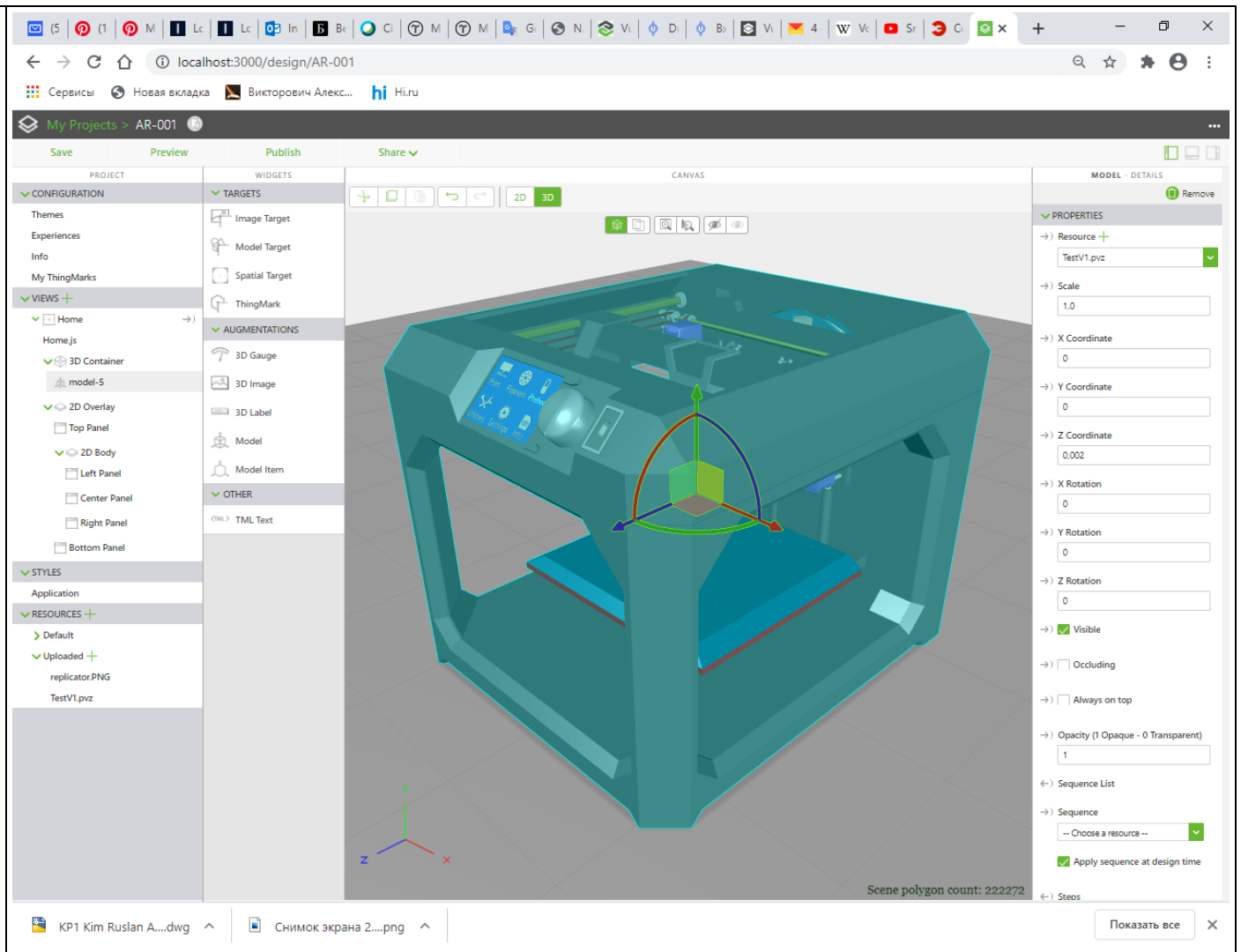
## Шаг 19.

В результате на **3D-канву** в зарезервированный 3D-контейнер загружается выбранная из файла **TestV1.pvz** **3D-модель**.

Разработчик получает возможность изменять масштаб модели, ориентацию ее расположения по осям и изменять углы наклона по осям.

Выполнять это можно как с помощью выведенной на **3D-канву «гизмо»** или с помощью соответствующих полей ввода значений в зоне интерфейса «Свойства компонентов». Гизмо представляет собой 3 взаимно перпендикулярные разноцветные стрелки для выполнения перемещения по соответствующим осям и три расположенные в ортогональных плоскостях дуги, «касание» к которым вызывает вращение в соответствующей плоскости.

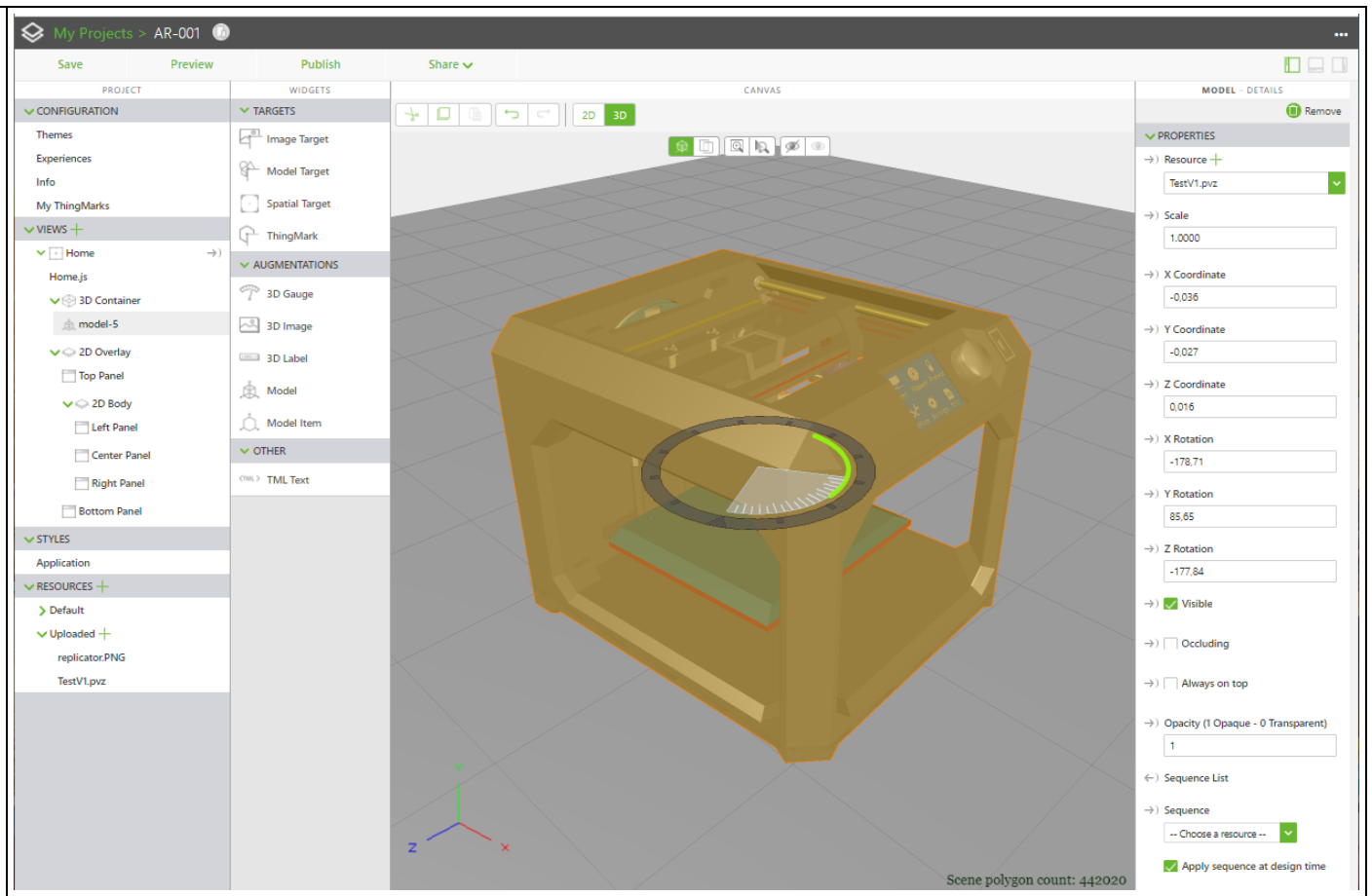
Обратите внимание и зафиксируйте (на бумажке) исходные координаты и углы разворотов (6 степеней свободы) – зона **«Свойства компонентов»**



## Шаг 20.

Попробуем изменить положение нашей модели так, чтобы панель управления смотрела направо, с помощью гизмо. (Если соответствующий символ отсутствует – кликайте на **«Model»** в зоне **«Компоненты»** до его появления).

Выбор объекта для редактирования возможен как непосредственно на **3D-канве**, так и в структуре в зоне **«Компоненты»**. При этом после выбора компонента в зоне **«Свойства»** будут отображаться, в том числе и для редактирования, его свойства.



## Шаг 21.

Выполнить эту же операцию можно точнее путем заполнения соответствующих полей в зоне интерфейса «Свойства компонентов».

Как показано на рисунке, выбрано такое положение **3D-модели**, которое характеризуется координатами размещения **X=0, Y=0, Z=0**.

Вращение модели по оси **X** выполнено на угол -180 градусов, вращение по оси **Z** - на угол -180 градусов, по оси **Y** – на угол 90 градусов.

Результат – наша модель располагается горизонтально относительно канвы, Панель управления смотрит «направо» - в положительном направлении оси **X** пространства (глобальной системы координат).

The screenshot displays the Vuforia Studio interface for a project named 'AR-001'. The central 'CANVAS' area shows a 3D model of a device with a red box highlighting the '3D' mode. The 'MODEL - DETAILS' panel on the right is also highlighted with a red box, showing the following properties:

- X Coordinate: 0
- Y Coordinate: 0
- Z Coordinate: 0
- X Rotation: -180
- Y Rotation: 90
- Z Rotation: -180
- Visible:
- Occluding:
- Always on top:

The 'WIDGETS' panel on the left shows 'Model Target' selected. The bottom status bar indicates 'Scene polygon count: 222272'.

## Шаг 22.

Теперь необходимо сделать топологическую привязку для правильной визуализации нашей виртуальной модели в реальном окружении. Это будем выполнять в соответствии с выбранным типом таргетинга – с помощью ThingMark'a.

В зоне интерфейса «**Виджеты**» находим виджет **ThingMark** и с помощью «**drag-n-drop**» переносим его на **3D-канву** (пока в любое место)

The screenshot displays the Vuforia Studio interface for a project named 'AR-001'. The interface is divided into several panels:

- PROJECT:** Includes 'Save', 'Preview', 'Publish', and 'Share' buttons.
- WIDGETS:** A central panel with categories: 'CONFIGURATION' (Themes, Experiences, Info, My ThingMarks), 'VIEWS' (Home, Home.js, 3D Container, 2D Overlay, Top Panel, 2D Body, Left Panel, Center Panel, Right Panel, Bottom Panel), 'STYLES' (Application), 'RESOURCES' (Default, Uploaded), and 'OTHER' (TML Text). The 'ThingMark' widget is highlighted with a red box.
- CANVAS:** A 3D view of a teal industrial machine model. A red arrow points from the 'ThingMark' widget to the machine. A 3D coordinate system (x, y, z) is visible at the bottom left.
- MODEL - DETAILS:** A right-hand panel showing properties for the selected model, including 'Resource' (TestV1.pvz), 'Scale' (1.0000), 'X Coordinate' (0), 'Y Coordinate' (0), 'Z Coordinate' (0), 'X Rotation' (-180), 'Y Rotation' (90), 'Z Rotation' (-180), and checkboxes for 'Visible' (checked), 'Occluding', and 'Always on top'.

At the bottom right of the canvas, it says 'Scene polygon count: 332621'.

## Шаг 23.

Результат – на канве появляется компонент приложения **ThingMark**, свойства которого – размер, координаты размещения – вы видите в зоне «Свойства компонента».

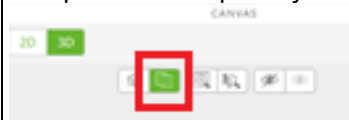
The screenshot displays the Vuforia Studio interface for a project named 'My Projects > AR-001'. The interface is divided into several panels:

- PROJECT:** Contains 'CONFIGURATION' (Themes, Experiences, Info, My ThingMarks), 'VIEWS' (Home, Home.js, 3D Container, model-5, **thingMark-1**, 2D Overlay, Top Panel, 2D Body, Left Panel, Center Panel, Right Panel, Bottom Panel), and 'STYLES'.
- WIDGETS:** Contains 'TARGETS' (Image Target, Model Target, Spatial Target, ThingMark) and 'AUGMENTATIONS' (3D Gauge, 3D Image, 3D Label, Model, Model Item), along with an 'OTHER' section for 'TML Text'.
- CANVAS:** Shows a 3D scene with a machine and a ThingMark component on the floor. The scene polygon count is 215360.
- THINGMARK - DETAILS:** A red-bordered panel showing the properties of the selected 'thingMark-1' component:
  - ThingMark: Bind/Enter ThingMark e.g. 2:1
  - Marker Width: 0,0254
  - X Coordinate: 0,393
  - Y Coordinate: 0
  - Z Coordinate: 0,169
  - X Rotation: -90
  - Y Rotation: 0
  - Z Rotation: 0

## Шаг 24.

Выделяем на **3D-канве** компонент **thingMark-1** и размещаем его с привязкой к модели.

Можно это делать с помощью перемещения-проецирования по поверхностям модели – для этого надо выбрать пиктограмму «**Mate**» на канве



И после этого просто передвигать **ThingMark-1** по модели в поисках нужного расположения. Обратите внимание, как в этом режиме плоский объект ложится по плоскостям. И представьте себе, как бы вы это программировали в **Vuforia Engine+Unity**.

Или можно воспользоваться полями **X, Y, Z, Xrotation, Yrotation, Zrotation** для **ThingMark-1** в зоне интерфейса «Свойства компонента».

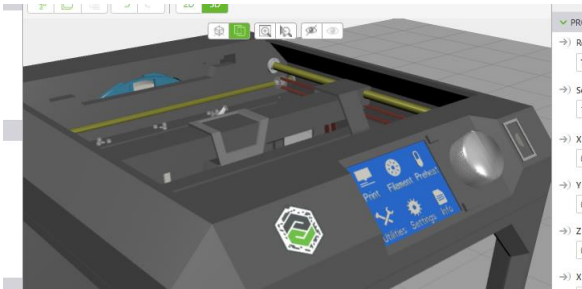
Для изменения масштаба **ThingMark-1** воспользуемся полем **Marker Width** в зоне интерфейса «Свойства компонента».

The screenshot displays the Vuforia Studio interface for project AR-001. The interface is organized into four main sections:

- PROJECT:** Contains a tree view under 'CONFIGURATION' (Themes, Experiences, Info, My ThingMarks) and 'VIEWS' (Home, Home.js, 3D Container, model-5, thingMark-1, 2D Overlay, Top Panel, 2D Body, Left Panel, Center Panel, Right Panel, Bottom Panel).
- WIDGETS:** Lists various components under 'TARGETS' (Image Target, Model Target, Spatial Target, ThingMark) and 'AUGMENTATIONS' (3D Gauge, 3D Image, 3D Label, Model, Model Item), plus an 'OTHER' section with 'TML Text'.
- CANVAS:** Shows a 3D view of a machine model with a ThingMark-1 component placed on its surface. A green circle highlights the ThingMark on the floor. The scene polygon count is 215360.
- THINGMARK - DETAILS:** Shows the properties for the selected ThingMark:
  - ThingMark: Bind/Enter ThingMark e.g. 2:1
  - Marker Width: 0,0254
  - X Coordinate: 0,393
  - Y Coordinate: 0
  - Z Coordinate: 0,169
  - X Rotation: -90
  - Y Rotation: 0
  - Z Rotation: 0

## Шаг 25.

В результате для нашего приложения получаем такое расположение виртуальной марки **thingMark-1** на виртуальной модели нашего устройства, которое соответствовало бы реальному расположению реальной метки на реальном изделии (или рядом с ним в реальной обстановке) – в соответствии с поставленной задачей.



A screenshot of the Vuforia Studio interface. The top bar shows "My Projects &gt; AR-001". Below it are tabs for "Save", "Preview", "Publish", and "Share". The main area is divided into three panels: "PROJECT", "WIDGETS", and "CANVAS". The "PROJECT" panel shows a tree view with "CONFIGURATION", "VIEWS", "STYLES", and "RESOURCES". The "WIDGETS" panel shows a list of widgets including "Image Target", "Model Target", "Spatial Target", "ThingMark", "3D Gauge", "3D Image", "3D Label", "Model", and "Model Item". The "CANVAS" panel shows a 3D model of the device with a virtual marker placed on its surface. The "THINGMARK - DETAILS" panel on the right shows the properties of the selected marker, including "ThingMark", "Marker Width", "X Coordinate", "Y Coordinate", "Z Coordinate", "X Rotation", "Y Rotation", "Z Rotation", "Tracked", "Display Tracking Indicator", "Always on top", "Studio ID", and "Friendly Name". The "Display Tracking Indicator" checkbox is checked. The "Studio ID" is "thingMark-1". The "Scene polygon count" is 222632.

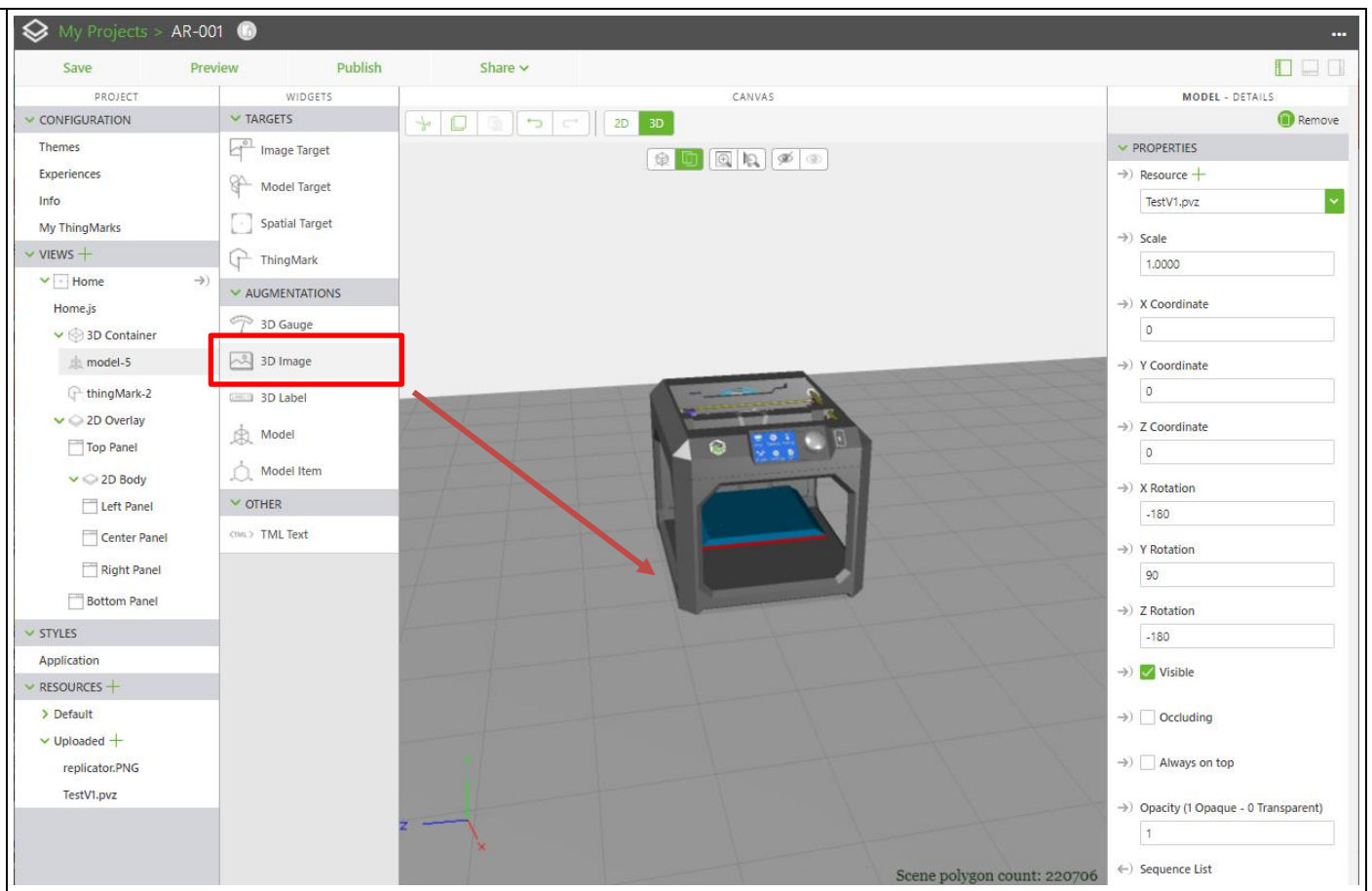
## Шаг 26.

Разместим ещё **3D виджеты**, которые будут воспроизводиться в нашем приложении дополненной реальности.

Разместим на **3D-канве** «картинку» в **3D виртуальном пространстве**. Исходя из задания (см. выше) – разместим два изображения, в каждом из которых будет размещен контент (дополнительная информация) в соответствии со сценарием разрабатываемого **Experience**.

Изображения создаются внешними приложениями и сохраняются в виде файлов форматов **.png, .jpg, .bmp**.

Для этого выберем в зоне интерфейса «**Виджеты**» виджет **3D Image** и перенесём его на **3D-канву** ( пока в любое место).



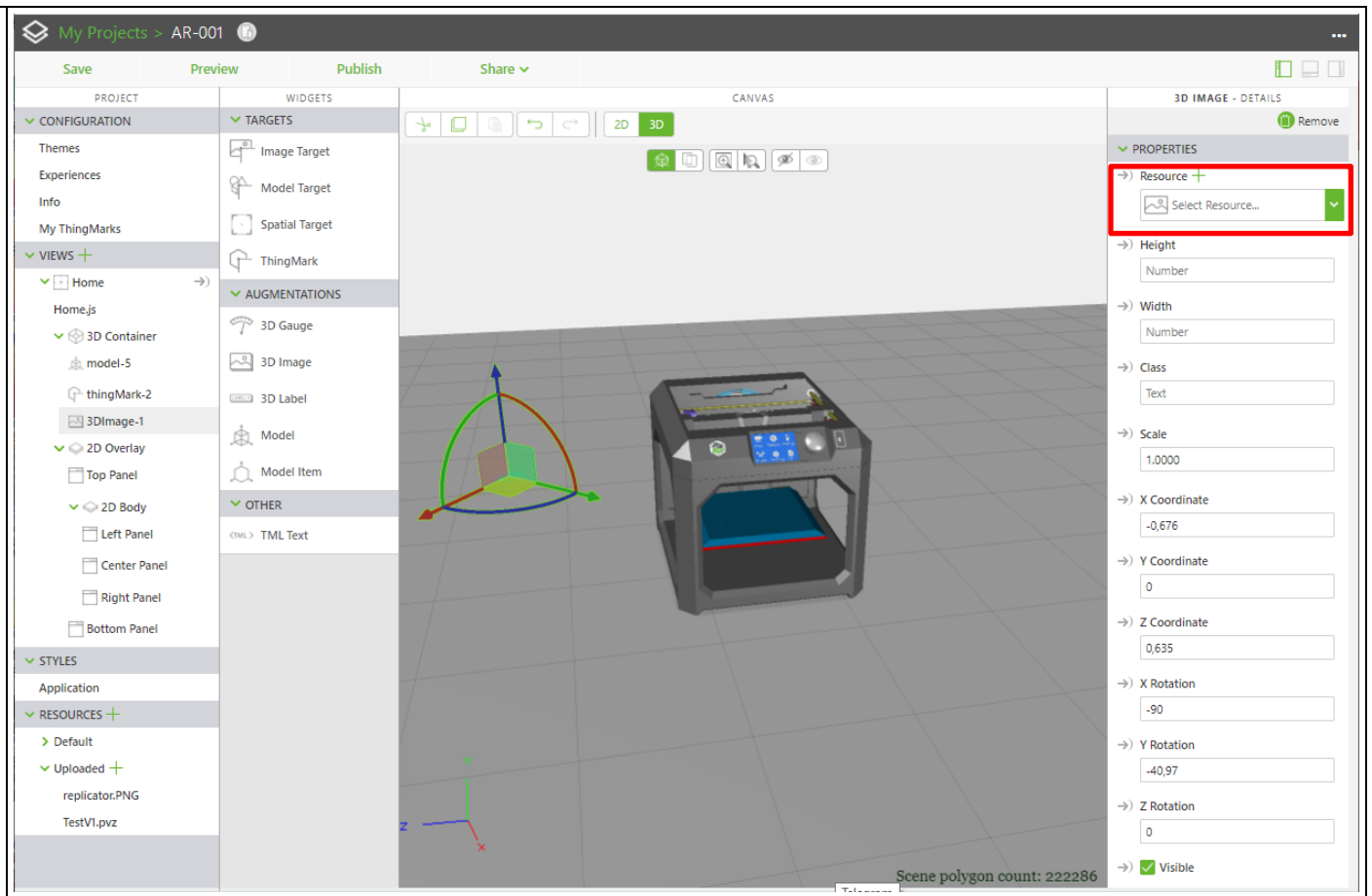


## Шаг 27.

Для того, чтобы выбрать «картинку» - изображение, которое мы хотим разместить в приложении дополненной реальности, воспользуемся либо имеющейся галереей изображений в **Vuforia Studio**, либо загрузим изображение из локальной файловой системы (**ЛФС**).

В любом случае для выполнения выбора необходимо воспользоваться меню «**Resource**» в зоне интерфейса «**Свойства компонента**»

Если нужного изображения в составе галереи **Vuforia Studio** нет, то надо искать требуемое в ЛФС. Для этого нажимаем символ «+» справа от поля **Resource**

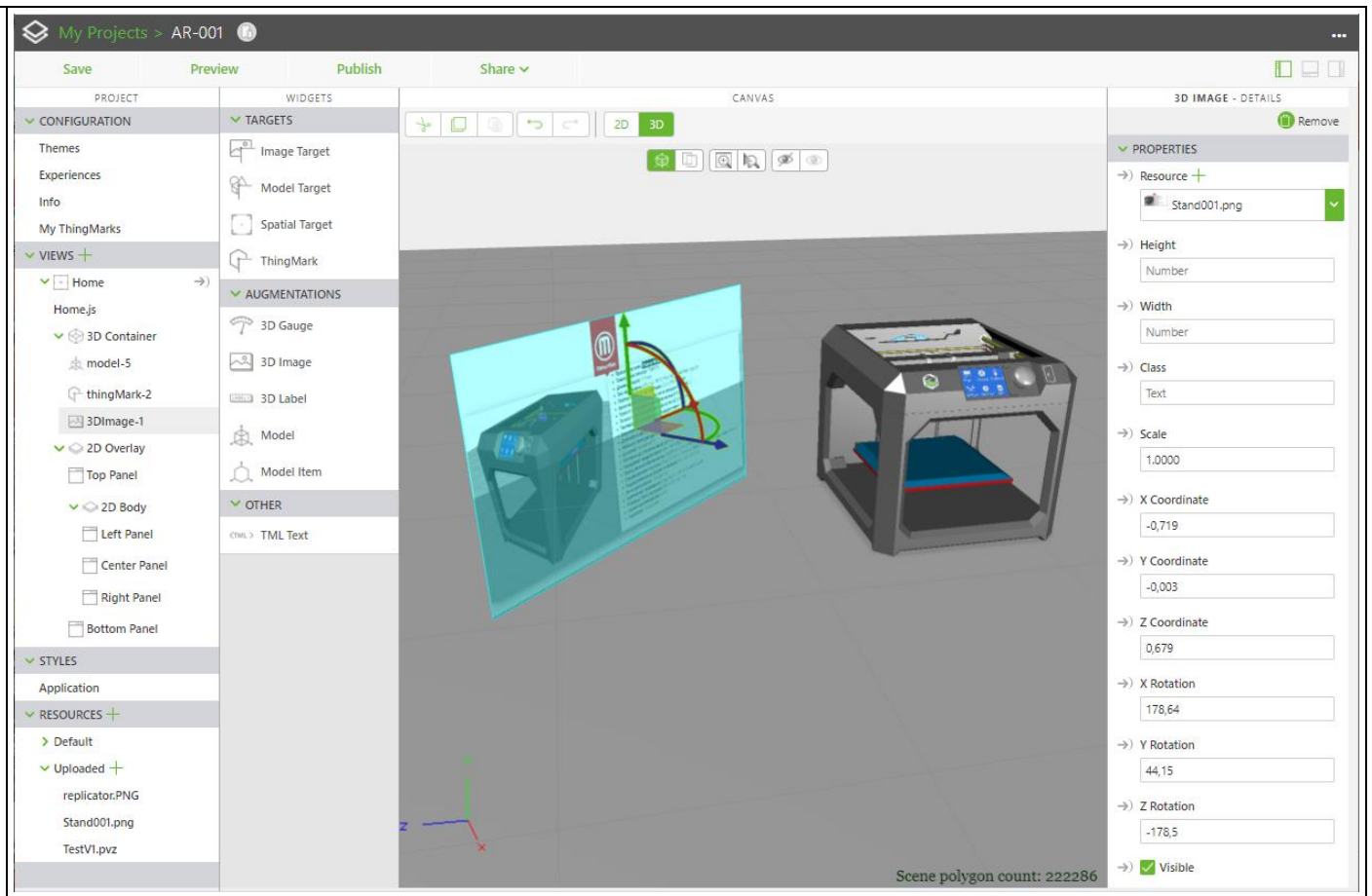
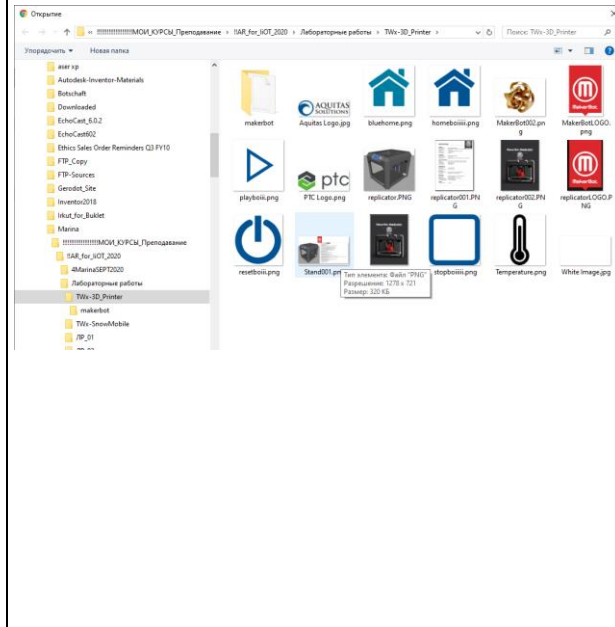


The screenshot displays the Vuforia Studio interface for a project named 'AR-001'. The central canvas shows a 3D scene with a printer model and a coordinate system. On the right, the '3D IMAGE - DETAILS' panel is open, showing the 'PROPERTIES' section. The 'Resource' field is highlighted with a red box, and the 'Select Resource...' dropdown menu is open, indicating the process of choosing a resource from the local file system. The 'Visible' checkbox is checked.

## Шаг 28.

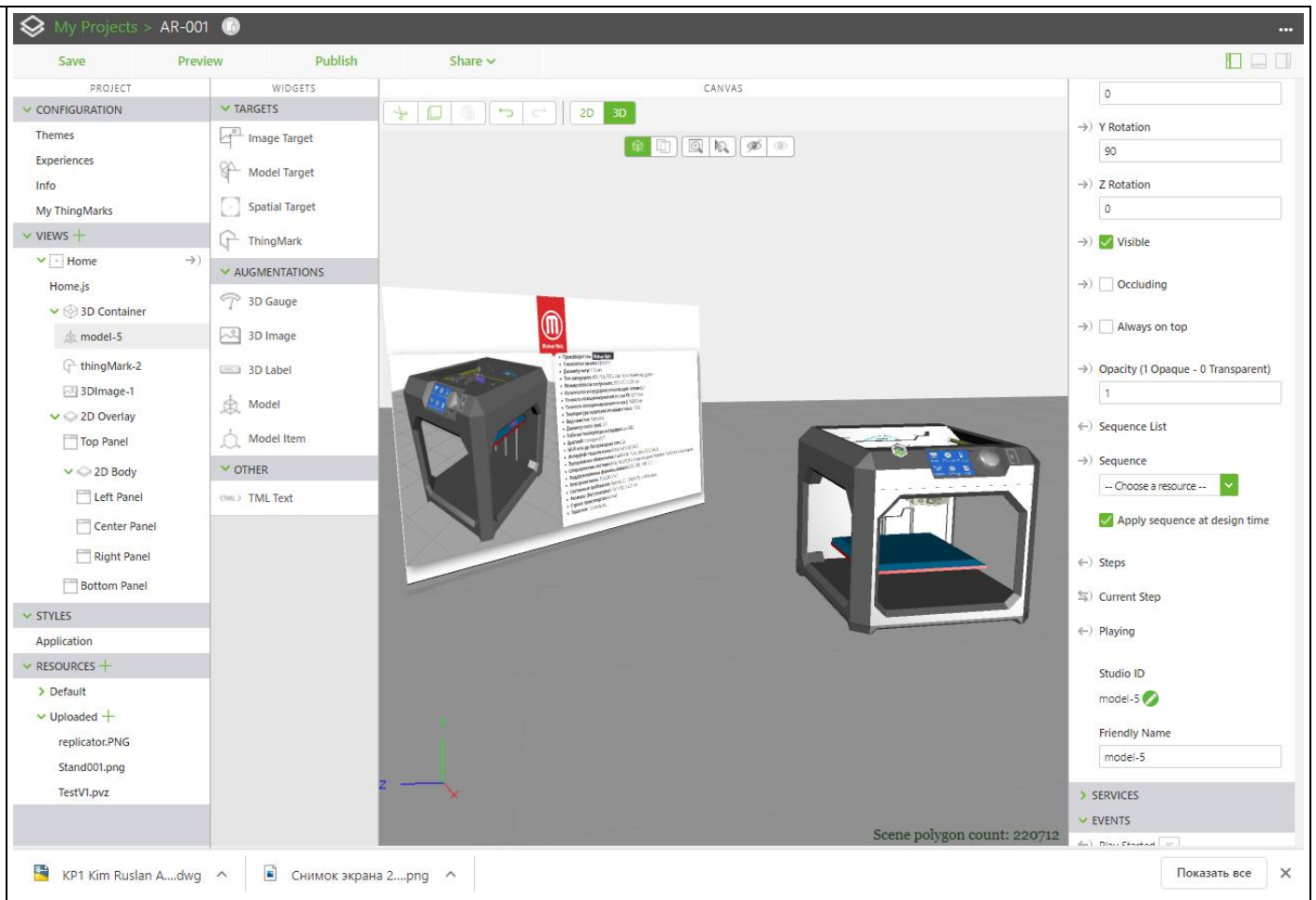
Средствами вызванного навигатора файловой системы находим необходимое изображение.

Например, в нашем случае - это файл **Stand001.png**.



## Шаг 29.

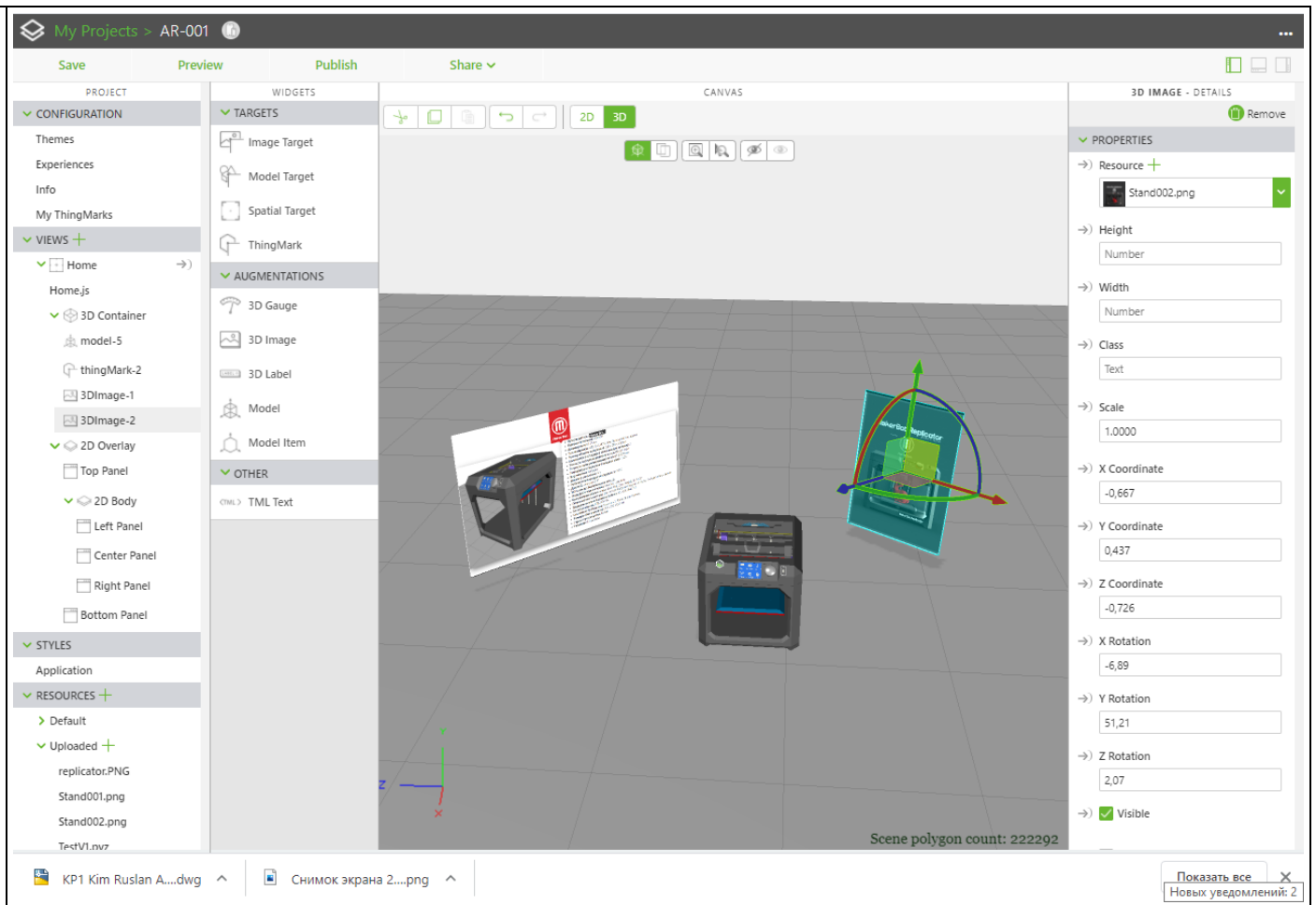
Размещаем изображение для **3D Image** на **3D-канве** проекта, используя те же методы, которые применялись при размещении **3D-модели** (шаги 20, 21).



## Шаг 30.

Размещаем второе изображение для **3D Image** на **3D-канве** проекта, используя те же методы, которые применялись при размещении **3D-модели** (шаг 20, шаг 21). Например, в нашем случае - это файл **Stand002.png**.

При размещении этих виджетов (в сцене это стенды, позиционируемые вокруг объекта - модели принтера), пользуйтесь инструментами разворота сцены (**левая, правая, средняя клавиша мыши → RMB, LMB, «колесико»**), для уточнения позиции каждого элемента активизируйте в зоне **«Компоненты»** в поле **VIEW** соответствующие компоненты. Здесь – это **3D Image-1** и **3D Image-1** соответственно. При этом символ гизмо появляется у выбранного компонента для манипулирования им.



## Шаг 31.

Разместим ещё один **3D-виджет** - виртуальную **3D-надпись**, текст. Например, мы хотим, чтобы при нахождении нужного нам принтера высвечивалось некоторое текстовое обозначение.

Разместим на **3D-канве** виджет **3D Label**.

The screenshot displays the Vuforia Studio interface for a project named 'AR-001'. The interface is divided into several panels:

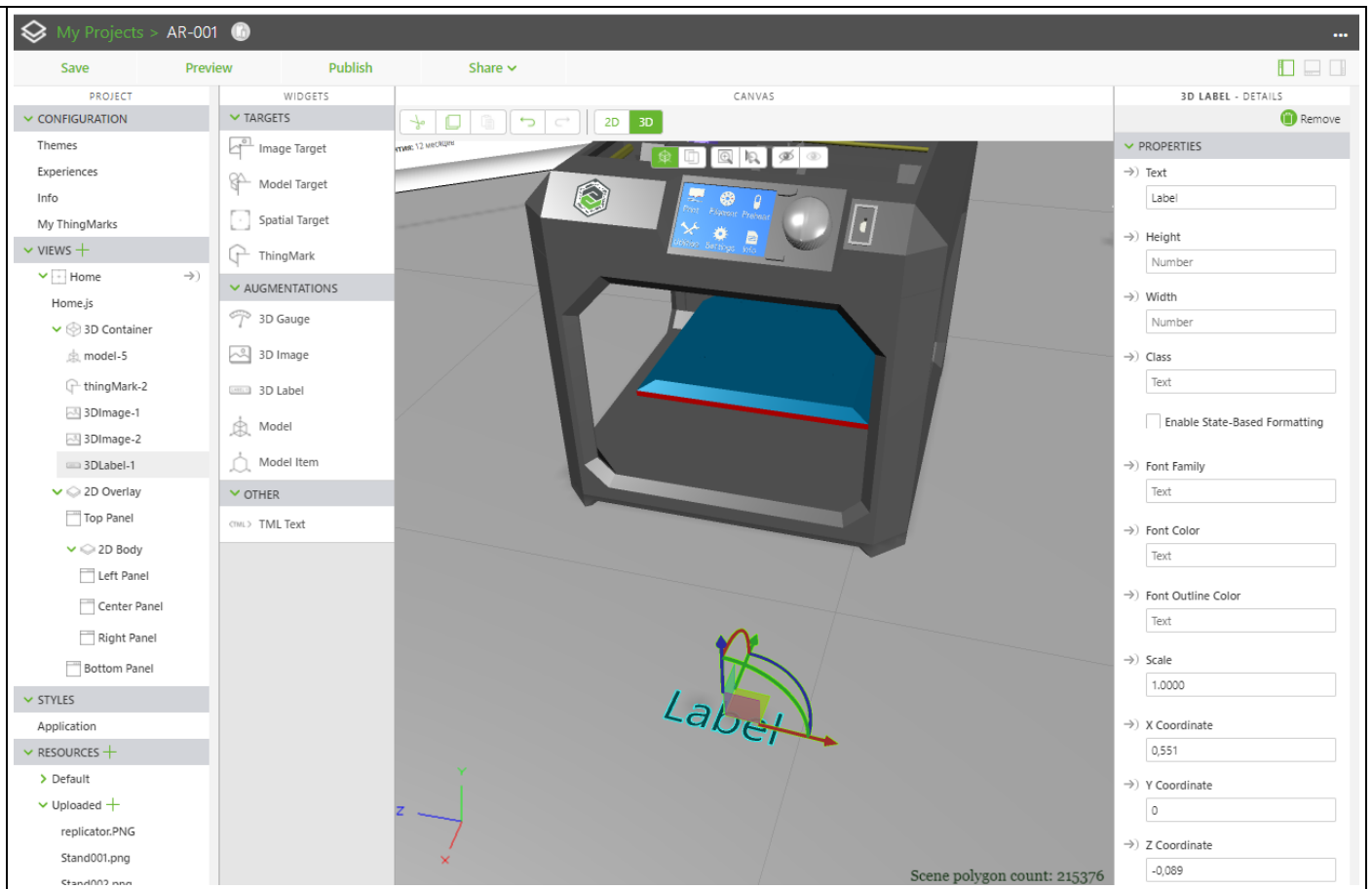
- PROJECT:** Contains sections for CONFIGURATION, VIEWS, STYLES, and RESOURCES.
- WIDGETS:** Lists various widget types under TARGETS, AUGMENTATIONS, and OTHER. The '3D Label' widget is highlighted with a red box.
- CANVAS:** The central 3D workspace showing a scene with a printer model and a 3D label widget being placed. A red arrow points from the '3D Label' widget in the WIDGETS panel to the label in the scene.
- THINGMARK - DETAILS:** A panel on the right showing the properties of the selected ThingMark, including fields for ThingMark, Marker Width, X, Y, Z Coordinates, X, Y, Z Rotations, and checkboxes for Tracked, Display Tracking Indicator, and Always on top.

At the bottom right of the canvas, it indicates 'Scene polygon count: 220718'.

## Шаг 32.

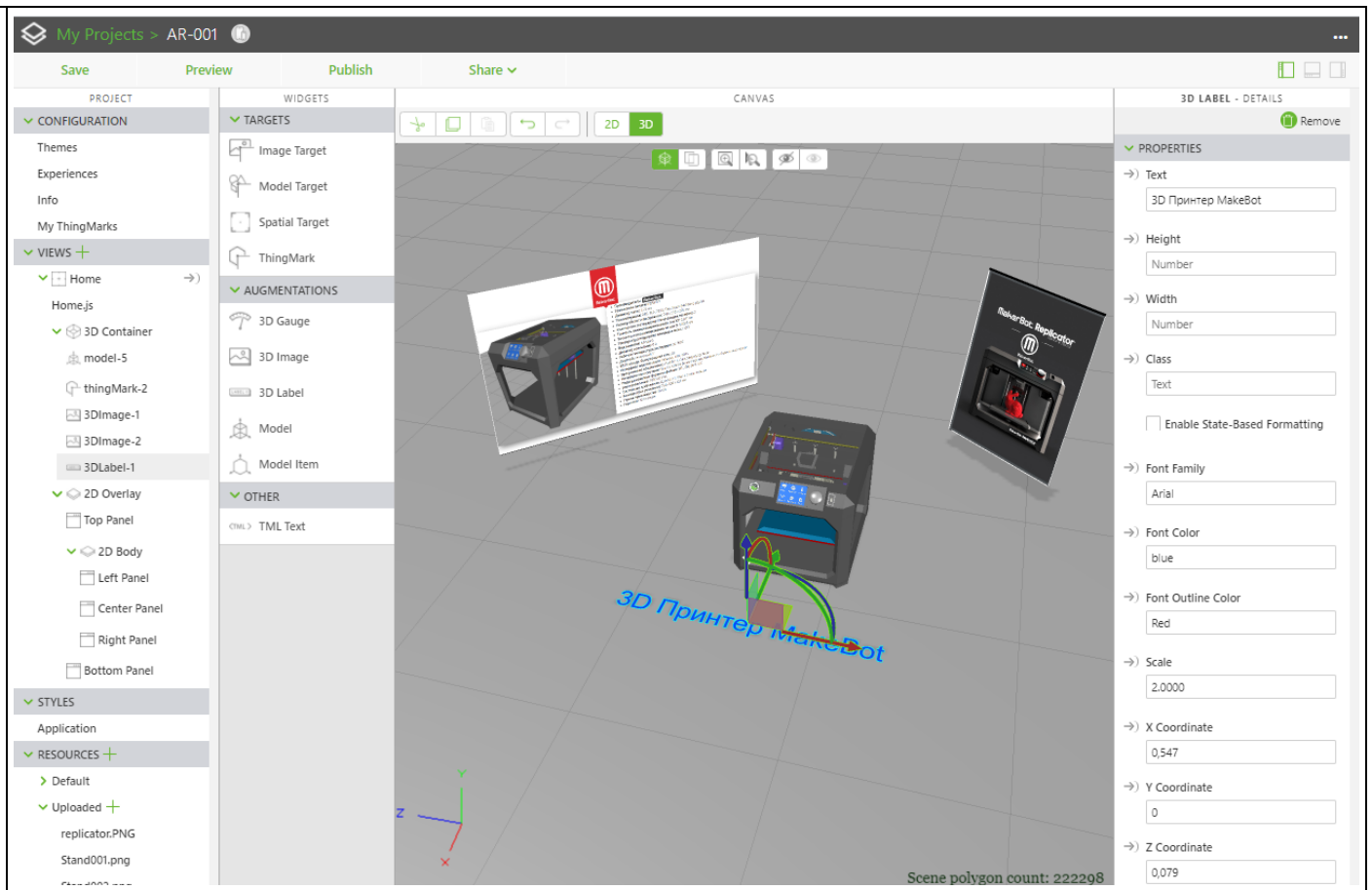
Выполним масштабирование и ориентацию его размещения аналогично тому, как это выполнялось при размещении модели и изображения (шаги 20, 21, 30).

Например, положим эту надпись – «ковриком».



## Шаг 33.

Заменяем стандартный текст «Label» на тот, который мы хотим поместить в приложение дополненной реальности. Например, «3D Принтер MakerBot».



The screenshot displays the Vuforia Studio interface for an AR application. The top navigation bar includes 'Save', 'Preview', 'Publish', and 'Share'. The main workspace is divided into three panels: 'PROJECT', 'WIDGETS', and 'CANVAS'. The 'PROJECT' panel on the left shows a tree view with categories like 'CONFIGURATION', 'VIEWS', '2D Overlay', 'STYLES', and 'RESOURCES'. The 'WIDGETS' panel in the middle lists various AR widgets such as 'Image Target', 'Model Target', '3D Gauge', and '3D Label'. The 'CANVAS' panel on the right shows a 3D scene with a printer model, a 3D label, and a 2D overlay. The '3D LABEL - DETAILS' panel on the far right shows the properties for the selected label, including text, height, width, class, font family, font color, font outline color, scale, and coordinates. The scene polygon count is 222298.

## Шаг 34.

Настройки **3D-Label** могут выполняться путем изменения параметров в зоне «Свойства компонент». Можно менять шрифт, цвета (символов и контуров), размер (**Scale**), видимое расположение текста, например: в режимах «**Billboard**» (надпись будет всегда повернута лицом к «зрителю»), «**Always on top**» (надпись будет всегда визуально перекрывать любое другое изображение, воспроизводится всегда «поверх» всего остального), «**Visible**» (видимость)

0.5000

→ X Coordinate  
0.027

→ Y Coordinate  
0.1

→ Z Coordinate  
-0.042

→ X Rotation  
-90

→ Y Rotation  
0

→ Z Rotation  
0

→  Visible

→  Billboard

→  Occluding

→  Always on top

→ Opacity (1 Opaque - 0 Transparent)

Unit: 35710

Enter a term to filter bindings

My Projects > AR-001

Save Preview Publish Share

PROJECT WIDGETS

CONFIGURATION TARGETS

Themes Experiences Info My ThingMarks

VIEWS +

Home Home.js

3D Container model-5

thingMark-2

3DImage-1

3DImage-2

3DLabel-1

2D Overlay

Top Panel

2D Body

Left Panel

Center Panel

Right Panel

Bottom Panel

STYLES

Application

RESOURCES +

Default

Uploaded +

replicator.PNG

Stand001.png

Stand002.png

Image Target

Model Target

Spatial Target

ThingMark

3D Gauge

3D Image

3D Label

Model

Model Item

OTHER

HTML TML Text

CANVAS

2D 3D

Scene polygon count: 222298

Font Family  
Times

Font Color  
Red

Font Outline Color  
Yellow

Scale  
2.0000

X Coordinate  
0.344

Y Coordinate  
-0.02

Z Coordinate  
0.074

X Rotation  
-76.03

Y Rotation  
88.75

Z Rotation  
0

Visible

Billboard

Occluding

Always on top



## Шаг 35.

А для инвентарного номера принтера из БД предприятия заготовим другую компоненту типа **3D Label**.

Предлагаемое размещение – на сцене.

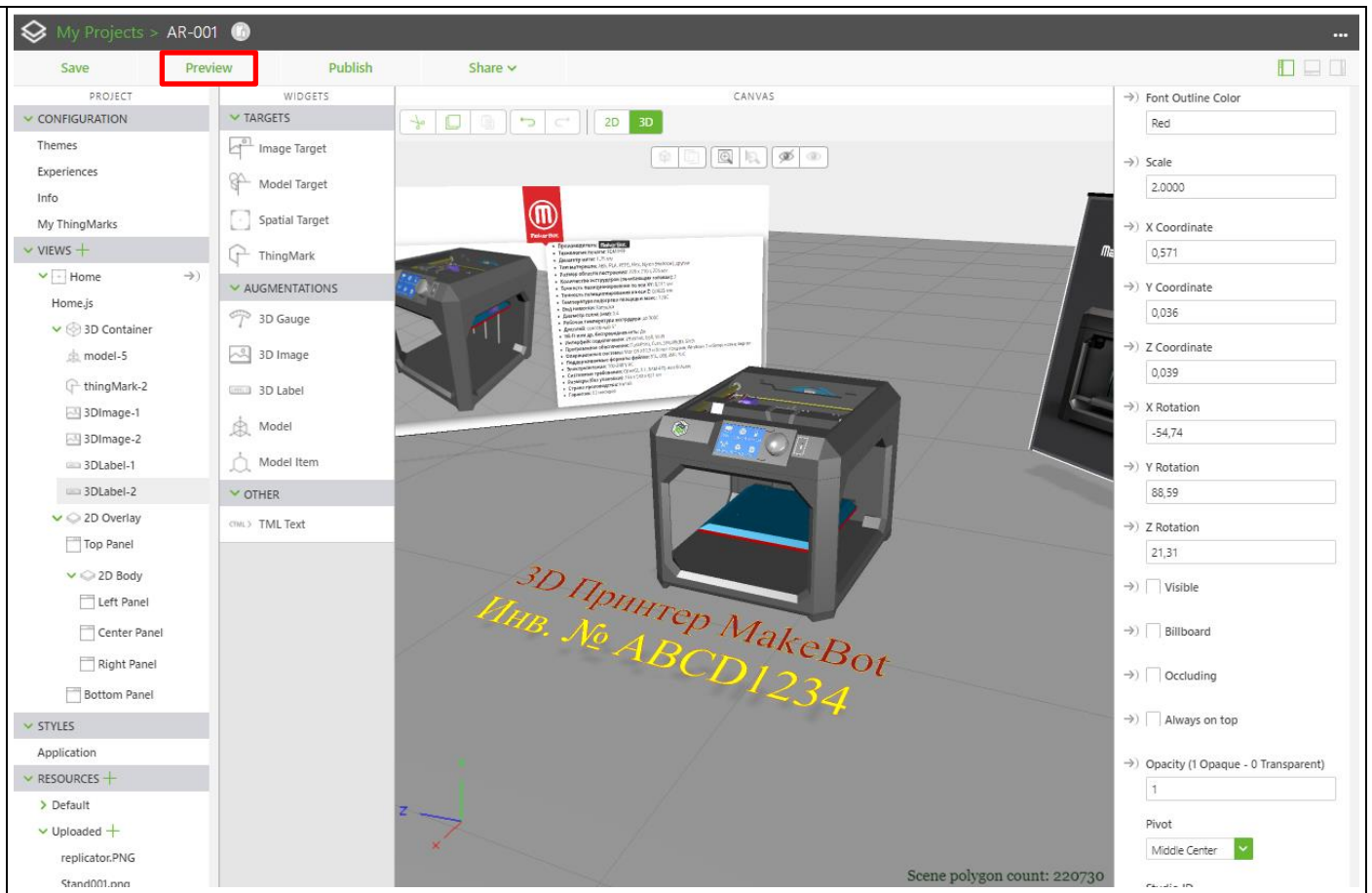
**ВАЖНО!!** Для будущего применения сделаем эту надпись невидимой → снимем флажок видимости в свойствах. В дальнейшем, в соответствии со сценарием, будем управлять видимостью с помощью **2D-Интерфейса**.

The screenshot displays the Vuforia Studio interface for a project named 'AR-001'. The interface is divided into several panels:

- PROJECT:** Includes sections for CONFIGURATION (Themes, Experiences, Info, My ThingMarks), VIEWS (Home, Home.js, 3D Container, model-5, thingMark-2, 3DImage-1, 3DImage-2, 3DLabel-1, 3DLabel-2, 2D Overlay, Top Panel, 2D Body, Left Panel, Center Panel, Right Panel, Bottom Panel), STYLES (Application), and RESOURCES (Default, Uploaded, replicator.PNG, Stand001.png).
- WIDGETS:** Includes TARGETS (Image Target, Model Target, Spatial Target, ThingMark) and AUGMENTATIONS (3D Gauge, 3D Image, 3D Label, Model, Model Item). An 'OTHER' section contains 'TML Text'.
- CANVAS:** A 3D scene showing a printer model on a grey floor. A yellow 3D label is placed on the floor with the text '3D Принтер MakeBot Инв. № ABCD1234'. A red 'M' logo is visible on a wall in the background.
- Properties Panel:** Located on the right, it shows various settings for the selected widget. The 'Visible' checkbox is highlighted with a red box. Other settings include Font Outline Color (Red), Scale (2,0000), X Coordinate (0,571), Y Coordinate (0,036), Z Coordinate (0,039), X Rotation (-54,74), Y Rotation (88,59), Z Rotation (21,31), Billboard, Occluding, Always on top, Opacity (1 Opaque - 0 Transparent) (1), and Pivot (Middle Center).

## Шаг 36.

Проверим, что у нас получается в проекте.  
Для этого откроем ещё одну страницу в Web-просмотрщике, нажав кнопку «Preview» в зоне интерфейса «Операции»



## Шаг 37.

На странице предварительного просмотра **Preview** выберем модель **мобильного устройства**, для которого разрабатывается данный проект и ориентацию экрана выбранного устройства.

В центральной части страницы будет воспроизведено поведение будущего **AR-приложения**.

В этом режиме **Перемещения, повороты и приближения/удаления** мобильного устройства относительно таргета имитируются нажатием и удержанием клавишей мыши (**LMB** – панорамирование, **RMB** – вращение, колесо или средняя **клавиша** – масштабирование)



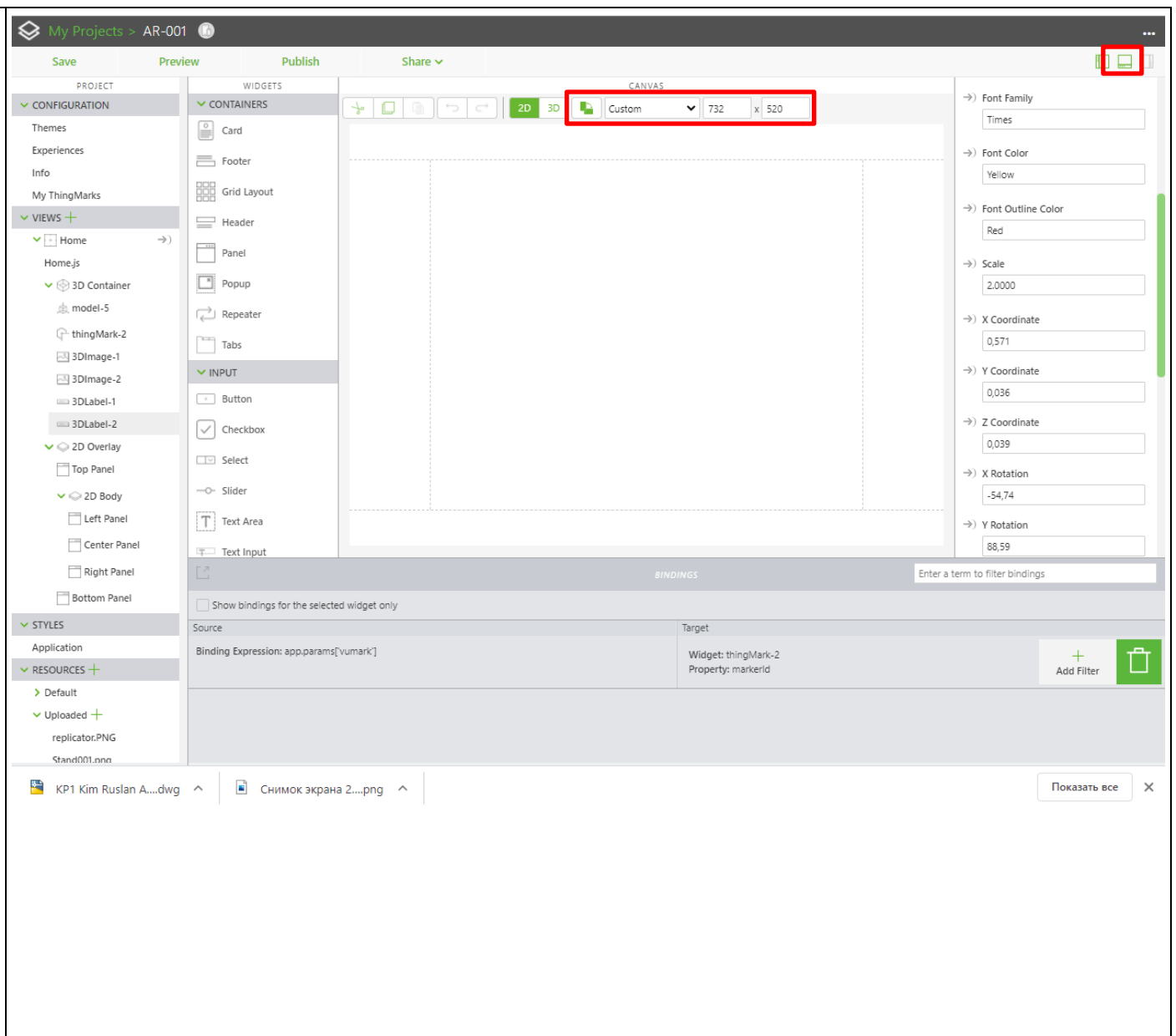
## Шаг 38.

Переходим к использованию виджетов **2D-канвы**. **2D- канва** отвечает за постоянное, неизменяемое при пространственных перемещениях, визуальное воспроизведение информации, связанной с **AR-приложением**.

Большинство **2D-виджетов** используют **multitouch**-функции экранов мобильных устройств и поэтому при размещении **2D-виджетов** на канве необходимо учитывать эргономику работы с **multitouch**- экранами. Стандартно **2D-канва** разделена на верхнюю часть – **Top Panel**, нижнюю часть, – **Bottom Panel**, левую часть – **Left Panel**, правую часть – **Right Panel** и центральную часть – **Central Panel**. Все части **2D-канвы**, кроме центральной, рекомендованы для размещения в них **2D-виджетов**.

Чтобы учесть особенности области просмотра МУ (соотношение сторон, кол-во точек на дюйм, ориентацию экрана и т.д.) в меню 2D канвы появляются дополнительные элементы меню → **Flip Orientation** и тип устройства (выбор из списка).

Для контроля связей между управляющими элементами **2D-Интерфейса** и поведением **3D-компонентов** предлагается открыть зону «Связи» → выбор иконки в верхнем правом углу.

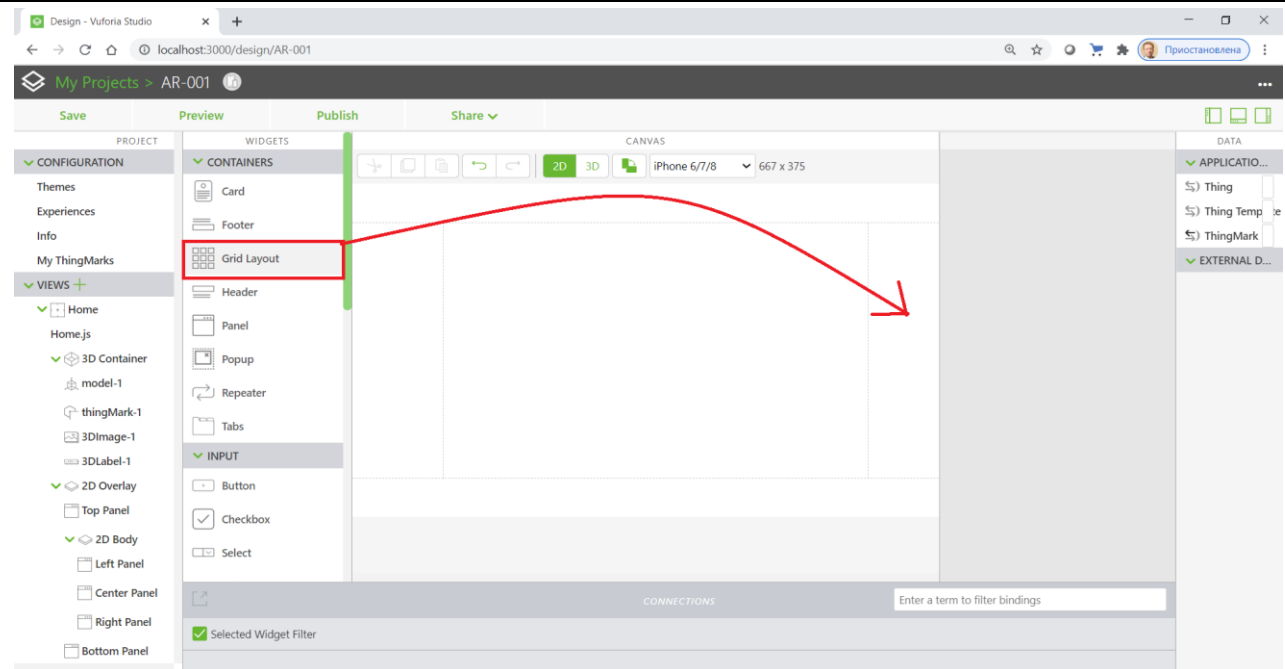


## Шаг 39.

Начнем с самого распространенного виджета – виджета разметки зон – **Grid Layout**.

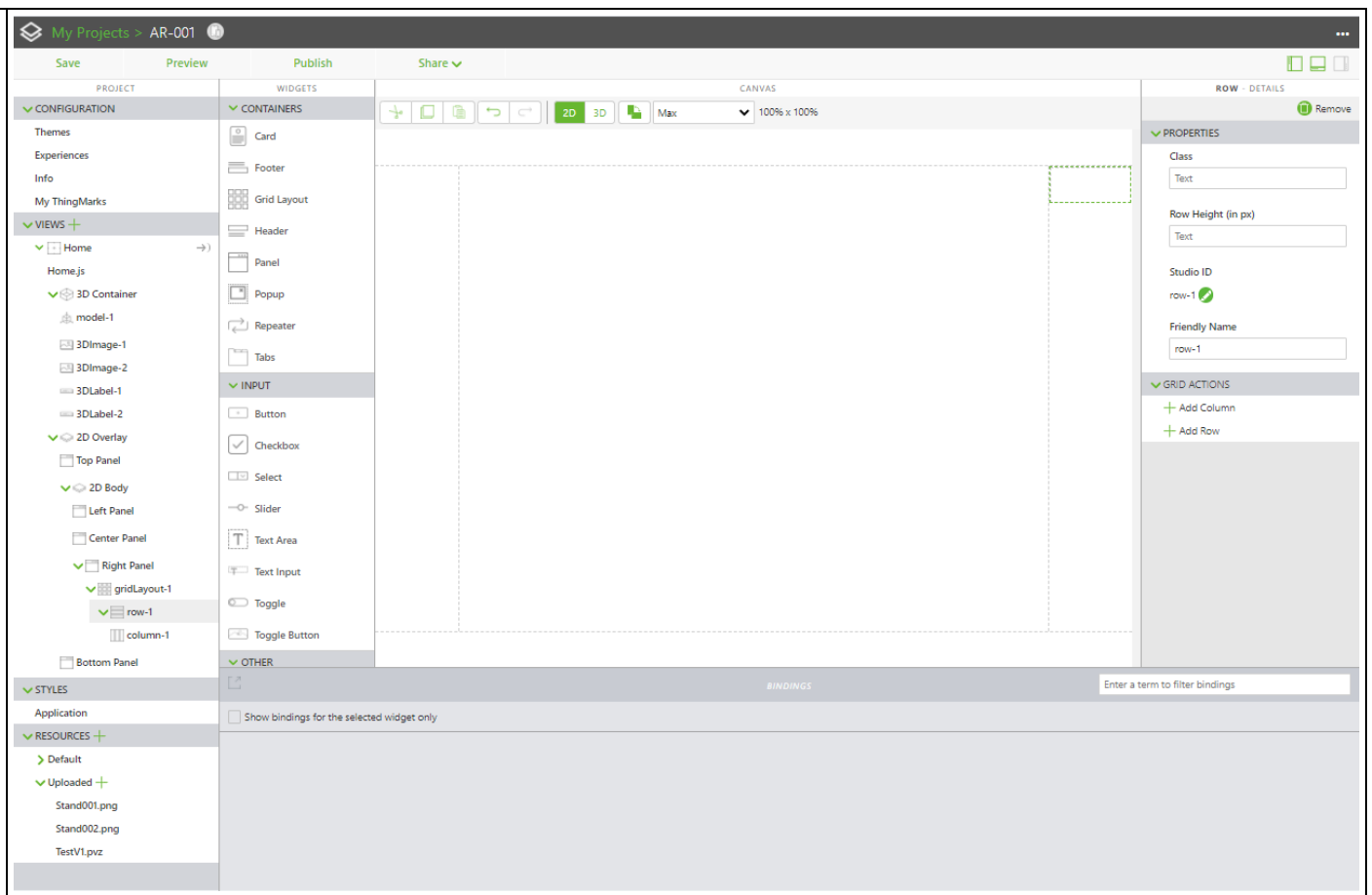
Он позволяет распределять/размечать отдельные элементы интерфейса для их размещения в панелях. Сами по себе панели не имеют характеристик, позволяющих это делать. См. «**Свойства Компонент**». Т.е. элементы интерфейса нельзя разместить в середине, справа или слева от уже имеющихся за счет их личных свойств. Для этого нужно использовать сетку (**Grid Layout**), в ячейки которых будут размещаться элементы интерфейса. Мы хотим разметить под размещение различных виджетов правую часть **2D-канвы**. Для этого поместим туда **2D-виджет Grid Layout** и создадим в нем несколько ячеек для размещения в них отдельных **2D-виджетов**.

Выбираем в зоне интерфейса «**Виджеты**» виджет **Grid Layout** и переносим его на правую часть **2D-канвы**.



## Шаг 40.

В результате в правой части канвы сразу (по умолчанию) получаем разметку в один ряд (**Row**), т.е. сетку с одним рядом и одной колонкой.



## Шаг 41.

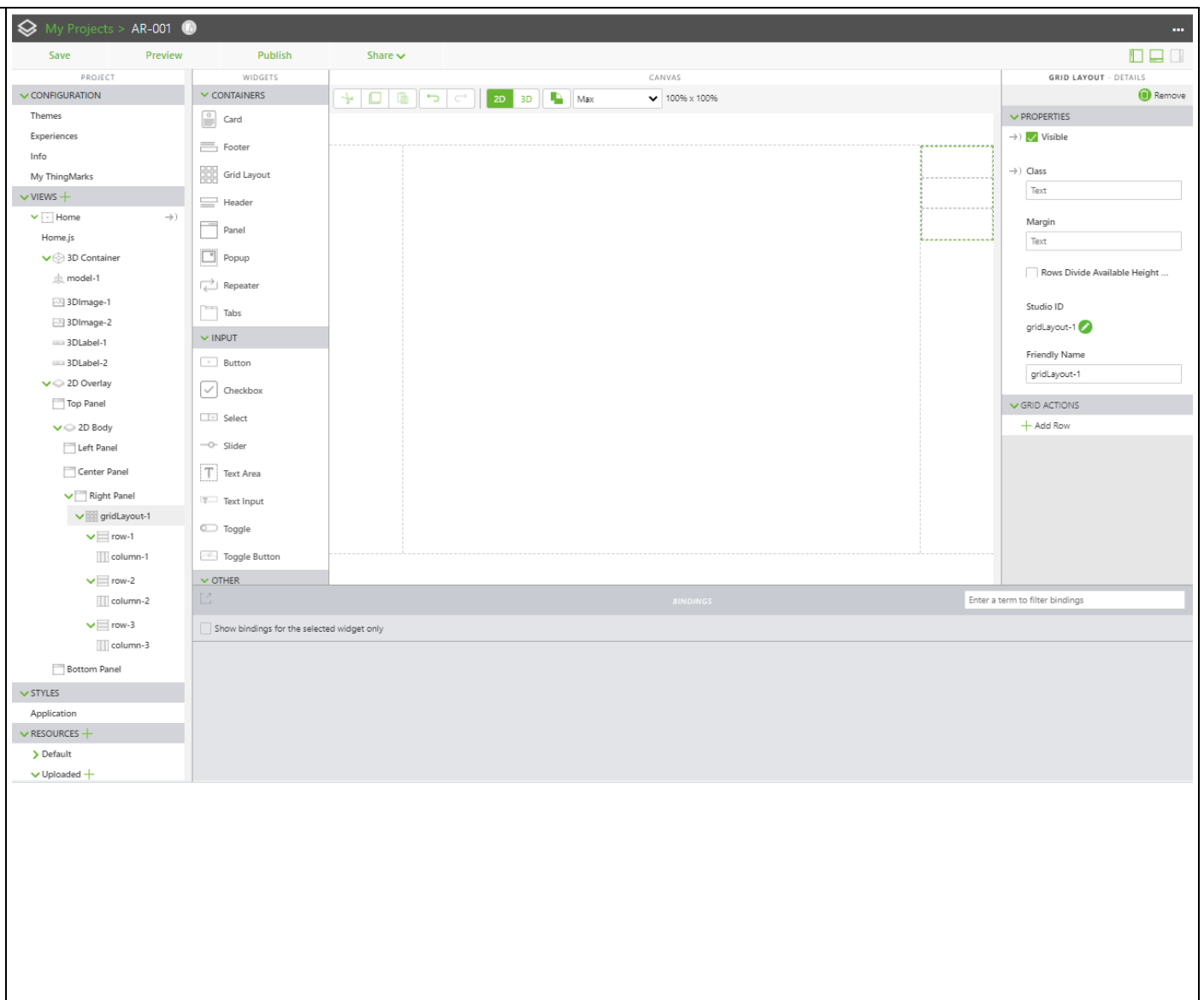
Добавляем в правой части канвы новые ячейки виджета **Grid Layout** (ряды в нашем случае) путем выполнения действия **Add Row** в зоне интерфейса «Свойства компонентов» - в нашем случае **Grid Layout**

GRID ACTIONS

+ Add Row

Каждое нажатие **Add Row** добавляет ещё один ряд в виджете **Grid Layout**.

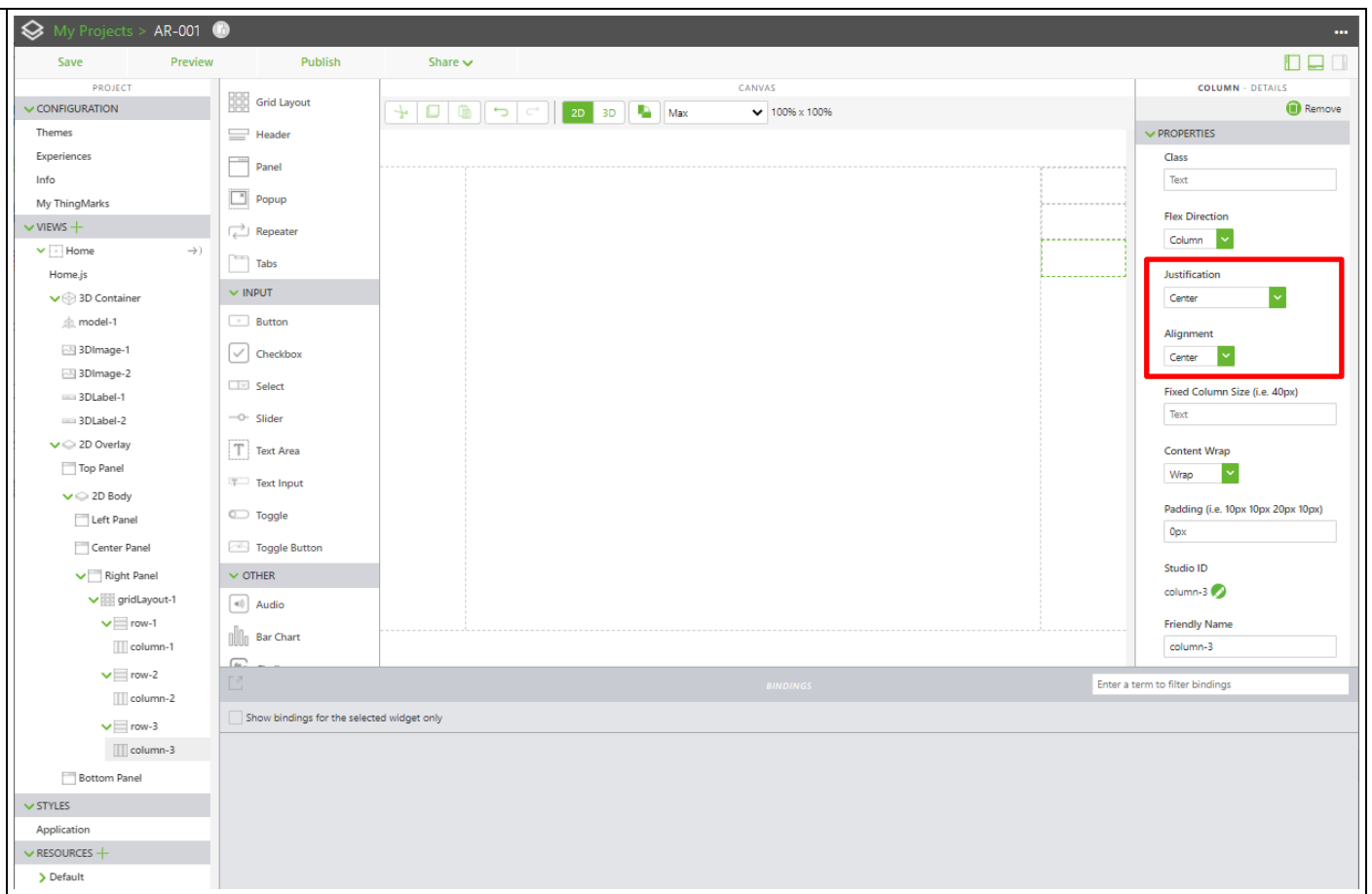
Т.о. мы резервируем на **2D** канве места для размещения элементов будущего плоского интерфейса. В качестве элементов **2D Интерфейса** будем использовать **2D Виджеты**. В нашем случае предлагается использовать **Toggle Button** (кнопка переключения), **Slider** (слайдер), которые будут размещаться в отдельный **Row** нашего **Grid Layout**. Элементы управления будем использовать для **вкл./откл.** визуализации компонентов **Проекта AR-001** и для изменения параметров обзора модели (**вращение, масштабирование, прозрачность отдельных деталей, цвет, перемещение и т.д.**).



## Шаг 42.

Результат троекратного вызова создания колонки в виджете **Grid** → **GRID ACTION** → **+...**

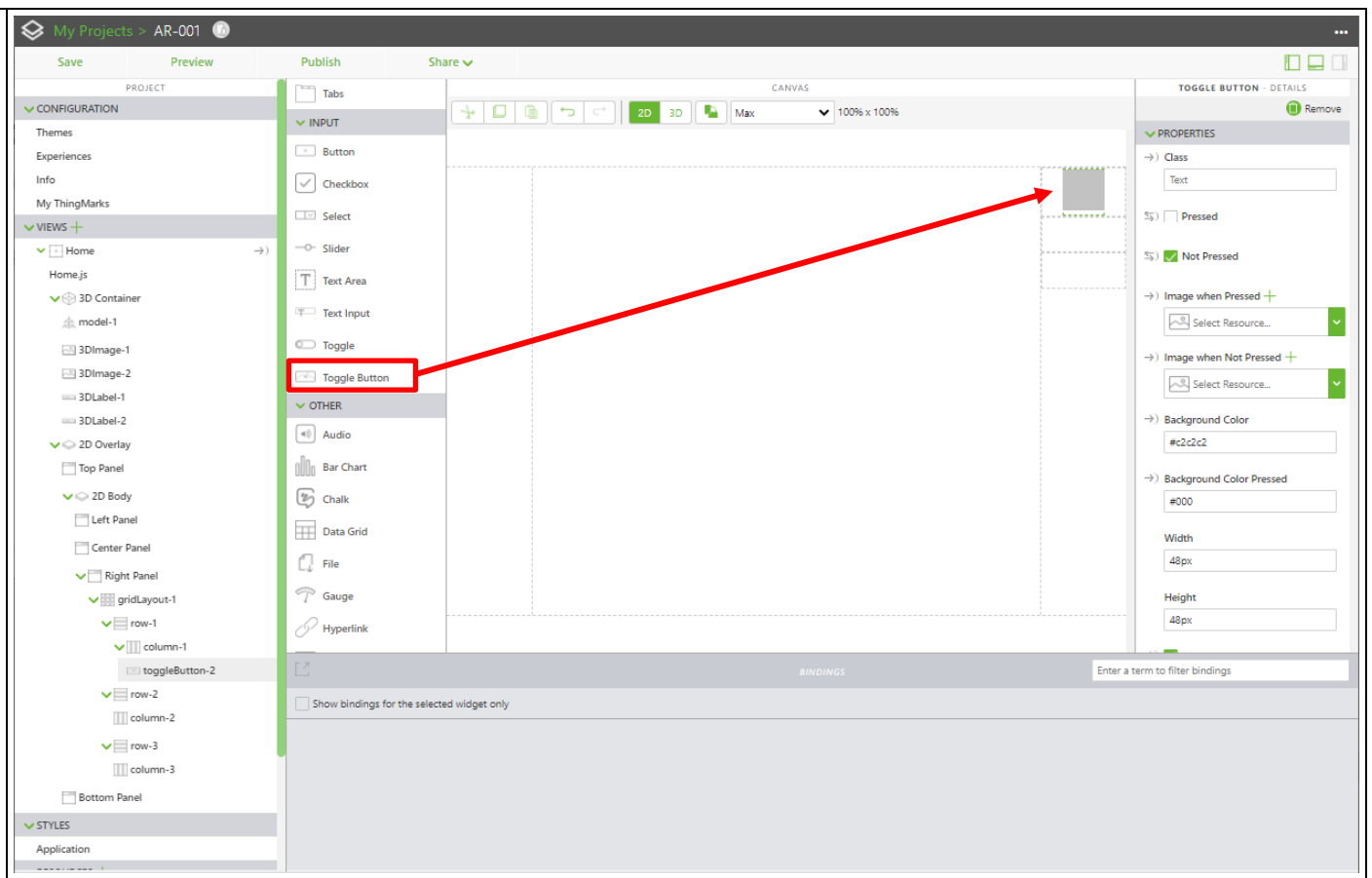
В свойствах строк (рядов) можно задать выравнивание помещаемых в них виджетов. В нашем случае зададим параметры выравнивания по горизонтали и вертикали как «**Center**».





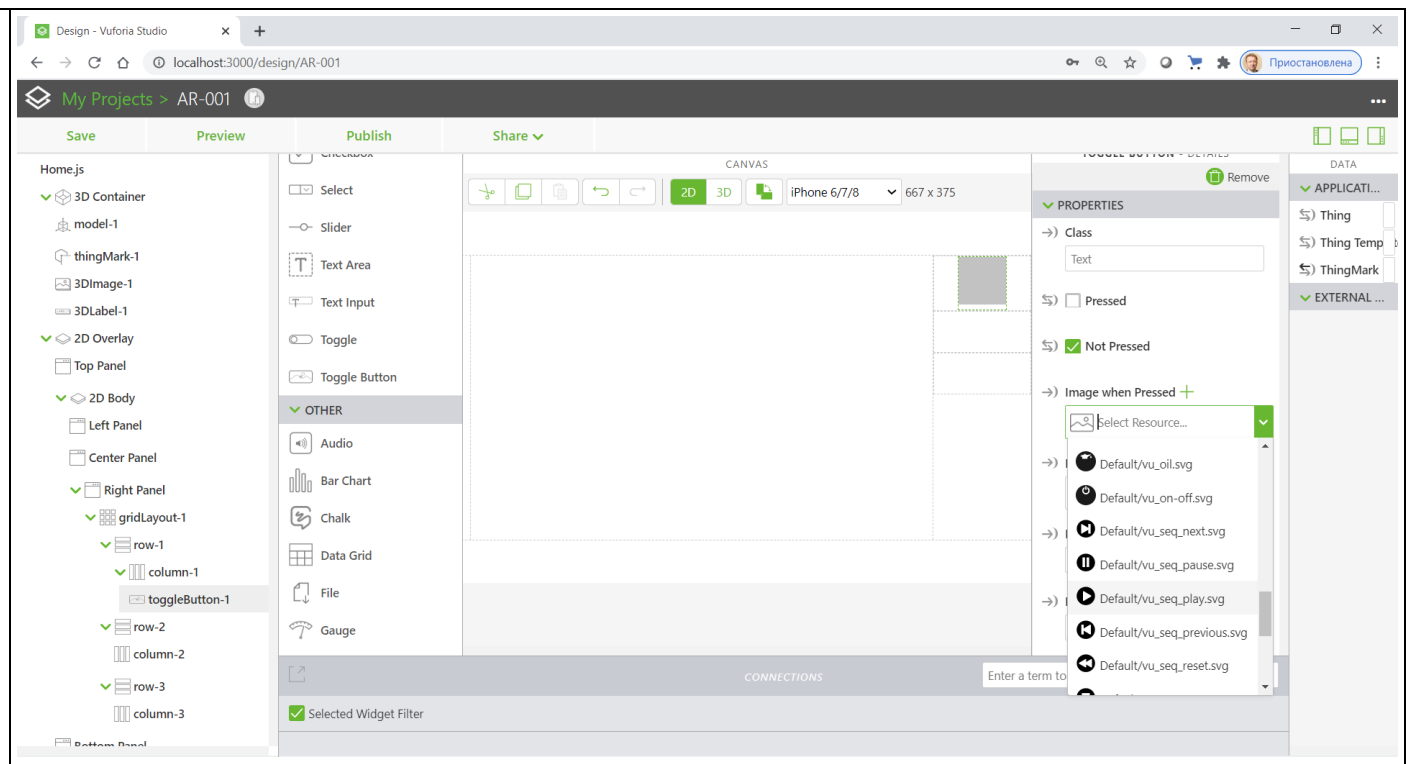
### Шаг 43.

Разместим в первом ряду виджета **Grid Layout 2D-виджет** – «переключатель» **Toggle Button**. Сделаем это с помощью процедуры «**Drag-n-Drop**».



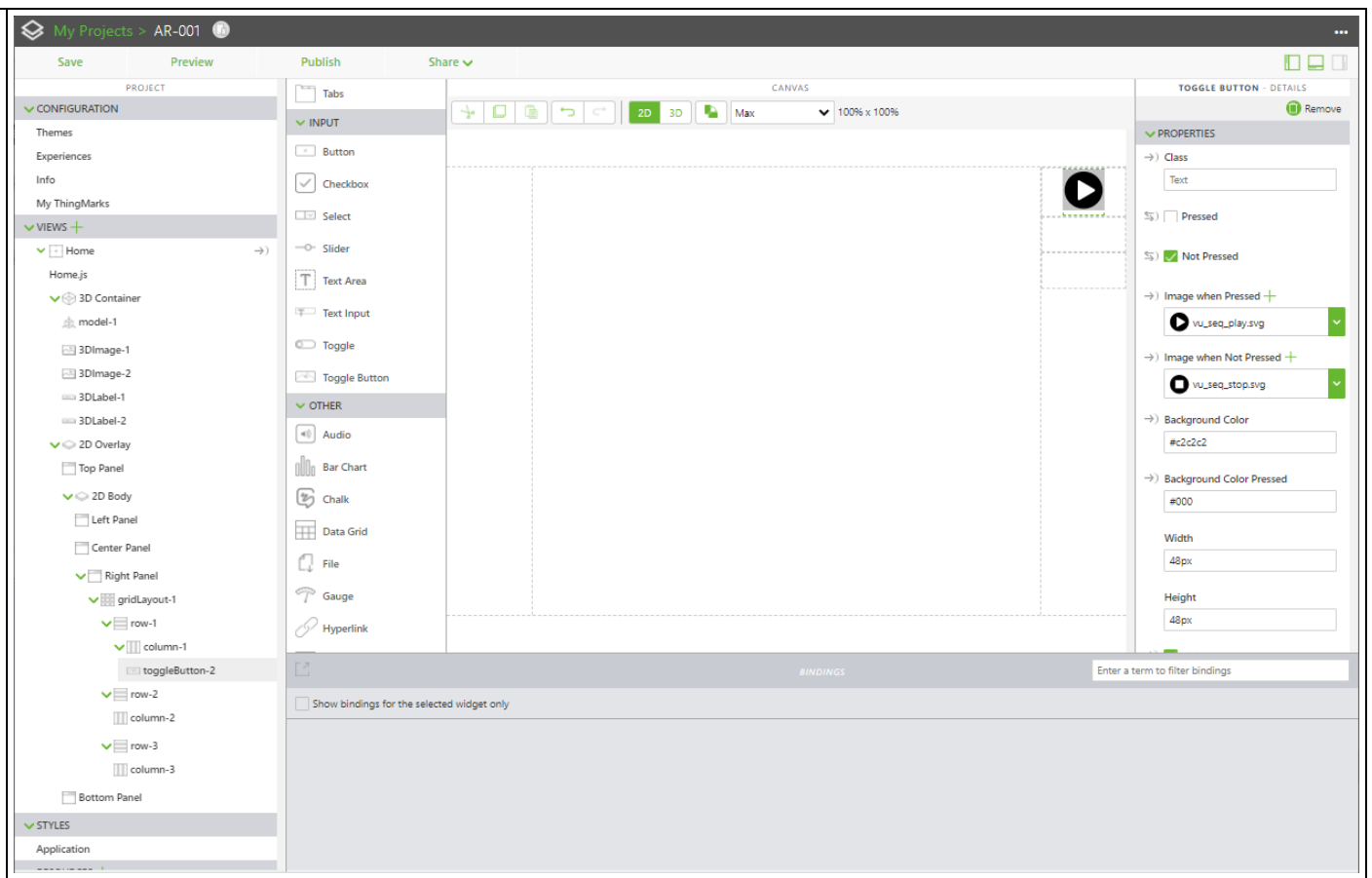
## Шаг 44.

Из галереи пиктограмм (**Images**) выберем изображения для «нажатого» (**Pressed**) переключателя и для «ненажатого» (**Not Pressed**) переключателя.



## Шаг 45.

2D-виджет **Toggle Button** размещен в первом ряду 2D-виджета **Grid Layout**.

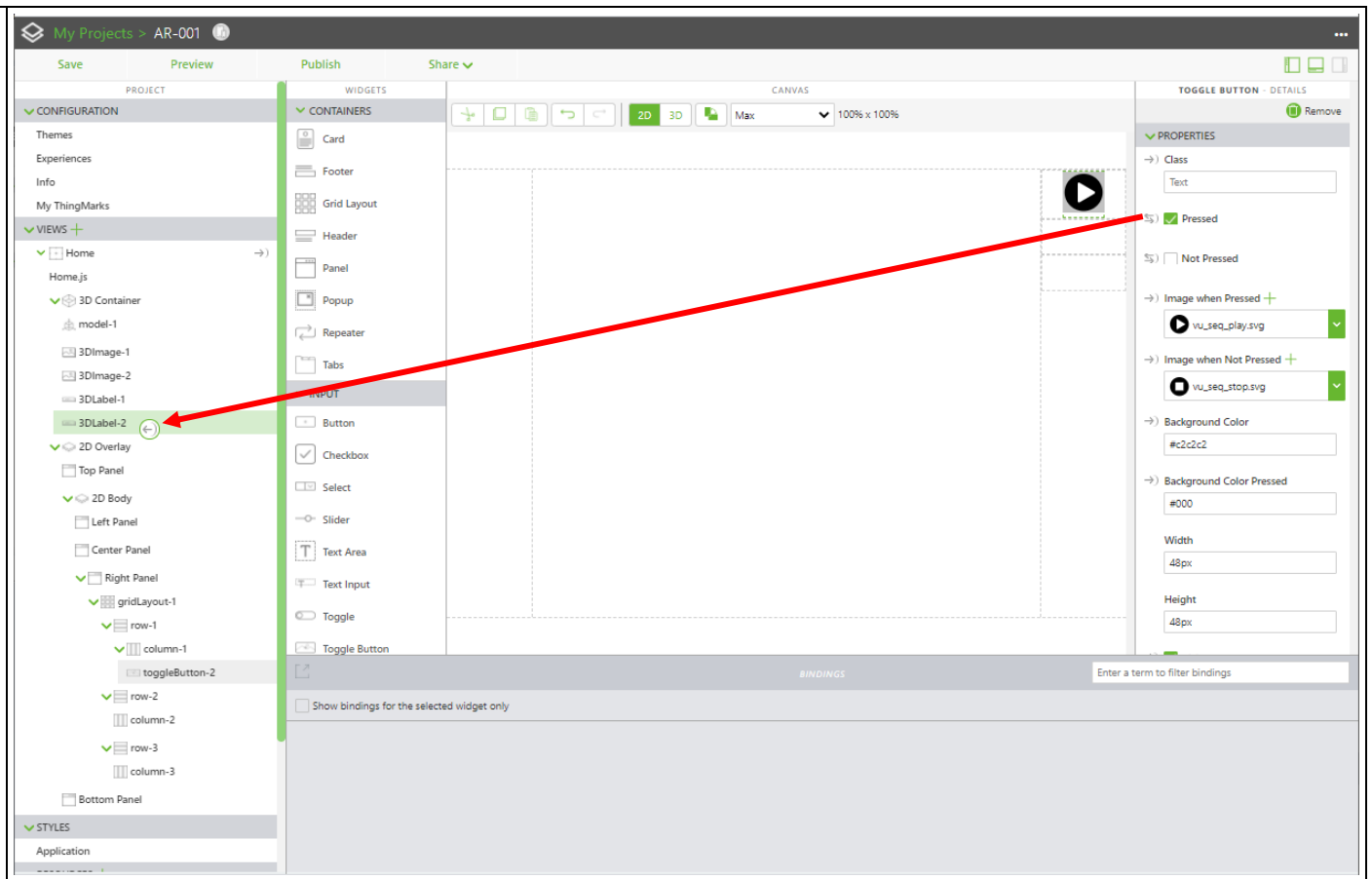


## Шаг 46.

Закрепим за размещенным 2D-виджетом **Toggle Button** управление визуализацией **3D Label-2**, размещенного ранее на **3D-канве**.

Будем считать что **3D Label-2** изначально скрыт и появляется в приложении только при нажатом (**Pressed**) **2D-виджете Toggle Button**.

Для выполнения этой функции переносим «**drag-n-drop**» маркер поля **Pressed** на строку «**3D Label-2**». Иными словами – связываем свойство «нажата» виджета **Toggle Button** с виджетом «**3D Label-2**».

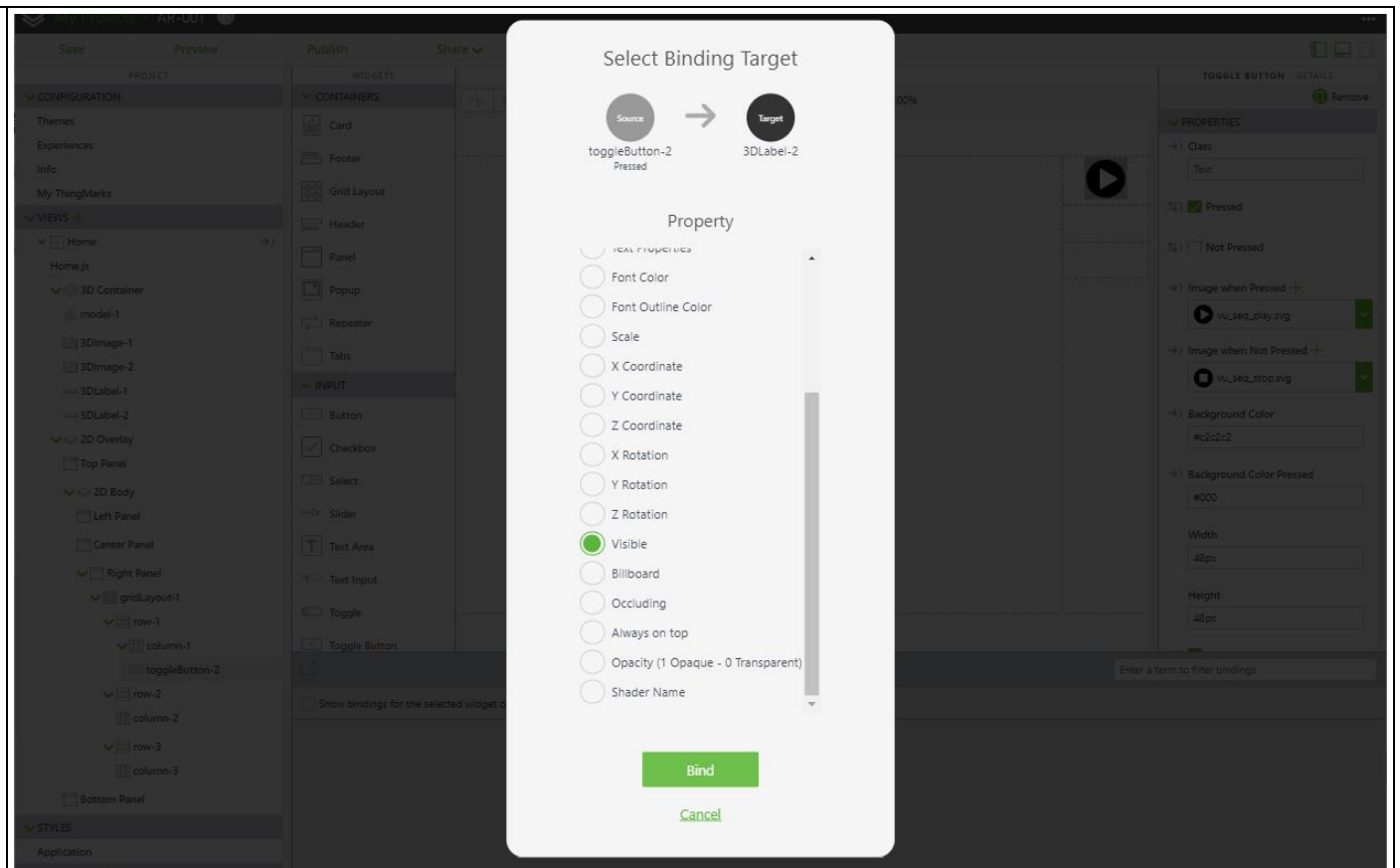


## Шаг 47.

В результате появляется меню связывания, в котором надо теперь только выбрать то свойство **3D-виджета 3D Label-2**, на которое будет влиять состояние **Pressed** 2D-виджета **Toggle Button**.

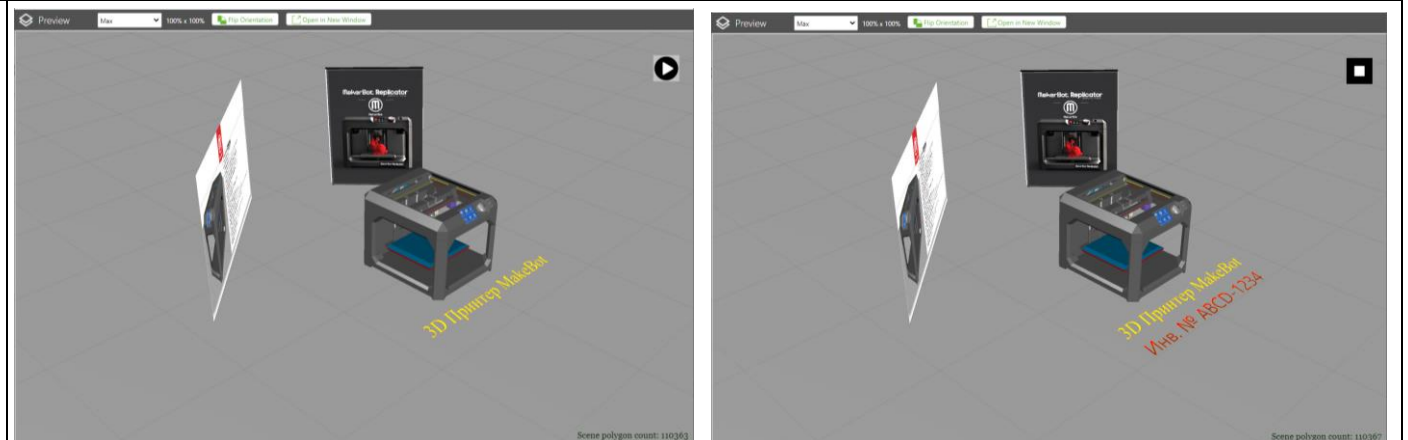
В нашем случае это должно быть свойство **Visible** (Видимость).

Выбираем это свойство и выполняем связывание нажатием кнопки **Bind** в нижней части меню.



## Шаг 48.

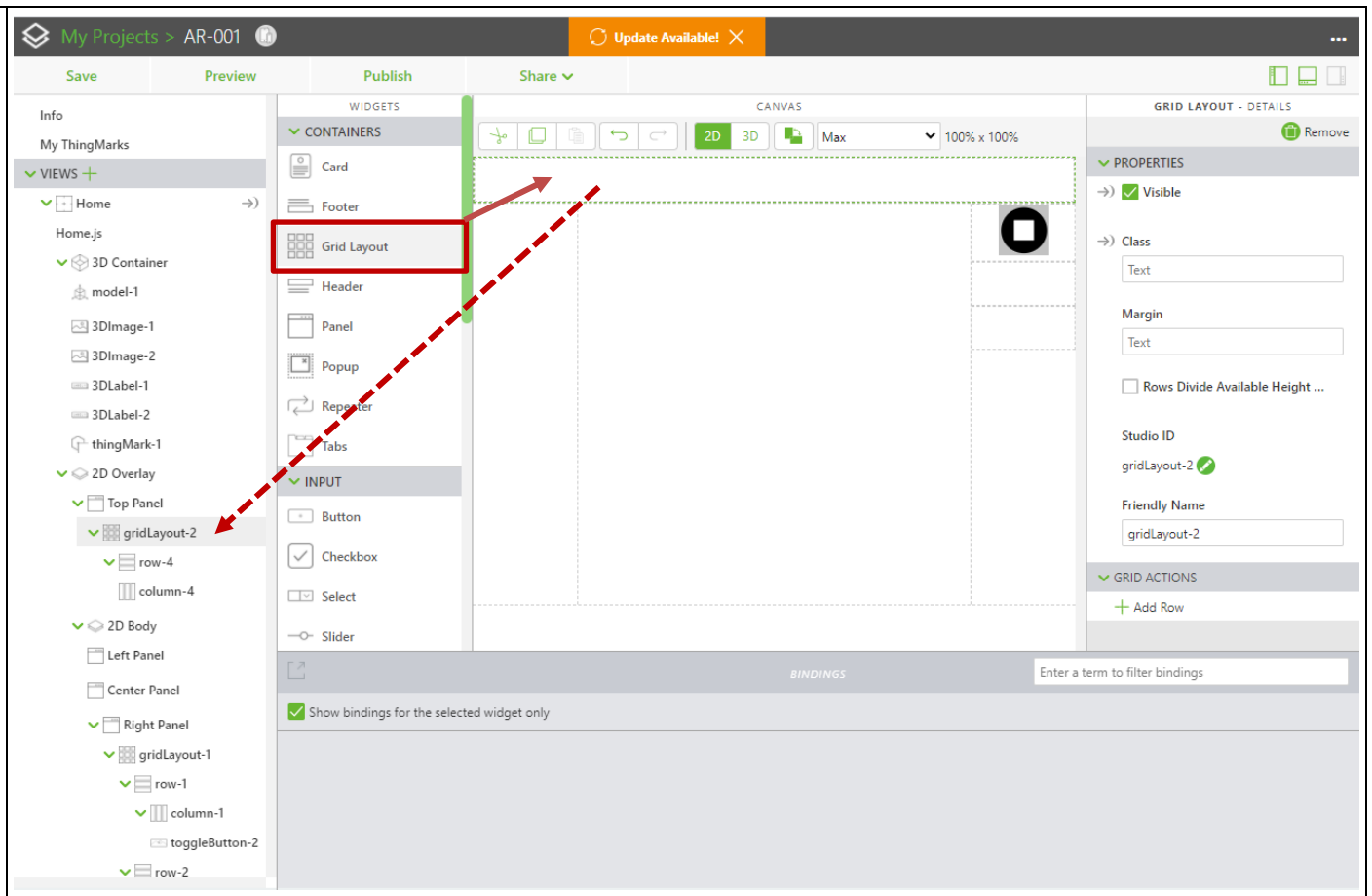
Можно выполнить **Preview** работы переключателя **Toggle Button**.



## Шаг 49.

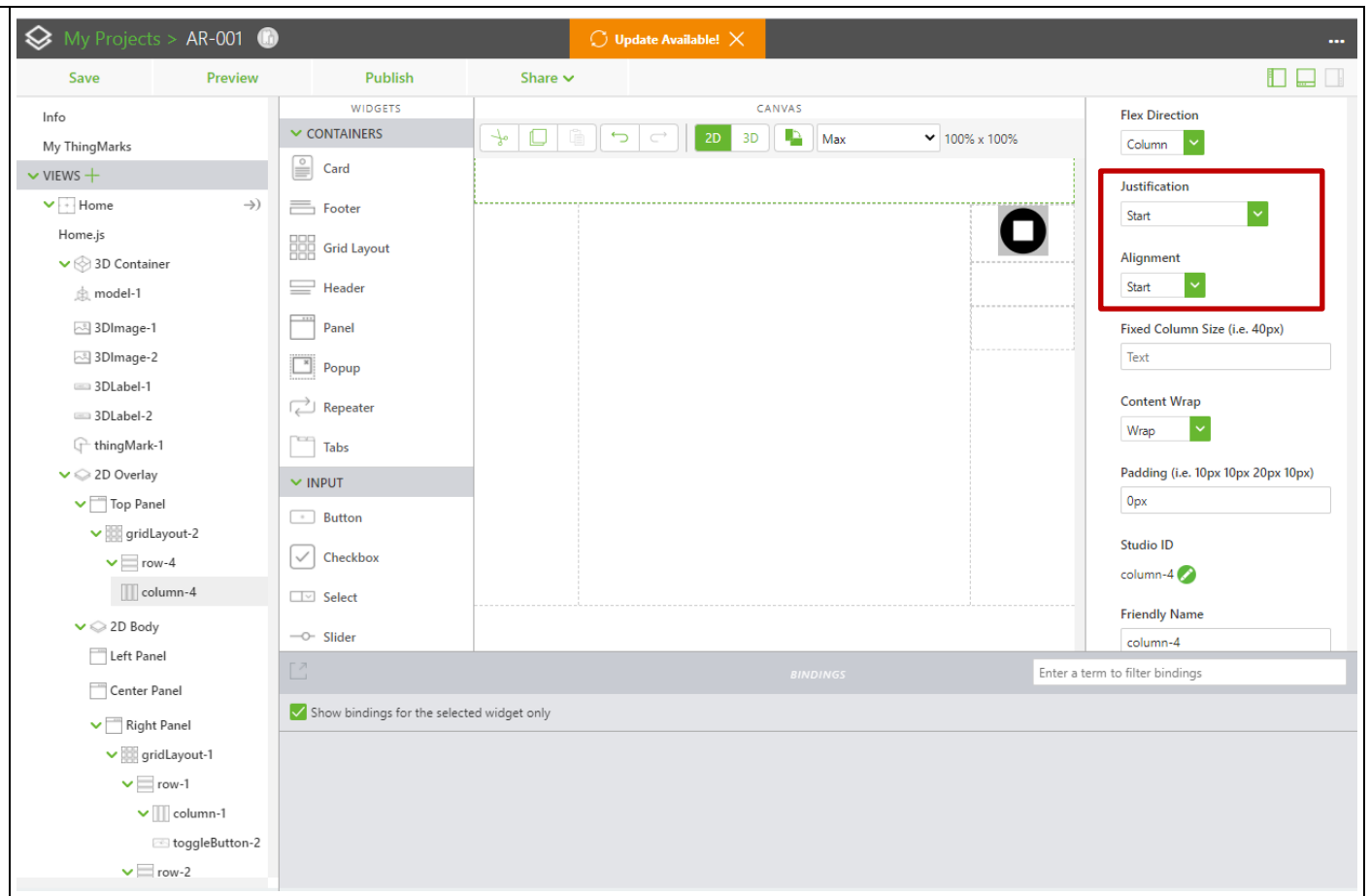
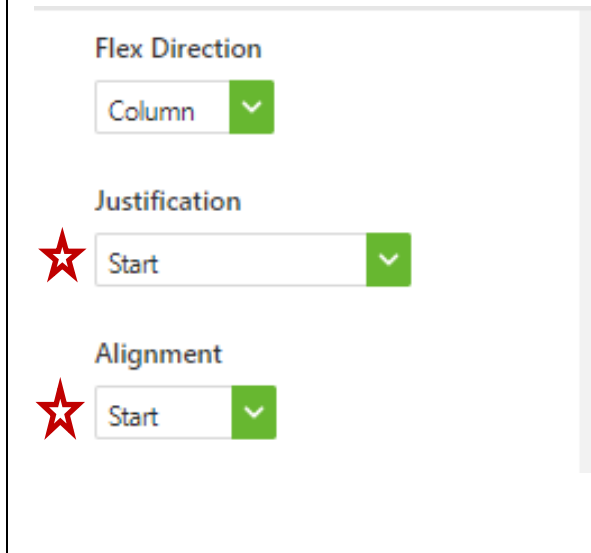
Добавляем **2D Виджет**, в котором будет размещаться, например логотип МЭИ или подобное изображение. Т.е. – добавляем в **2D Канву** ( иногда можно услышать жаргонизм на «стекло») **2D-Image** (см. зону «Виджеты» редактора ). Логотип хотим разместить в верхнем левом углу. Контейнер, отвечающий за это, - «разметчик» позиций – **Grid Layout** - помещаем на **Top Panel**.

**NB!!** – Не забывайте следить за изменением состава компонент в областях интерфейса редактора **Vuforia Studio** – левая часть экрана.



## Шаг 50.

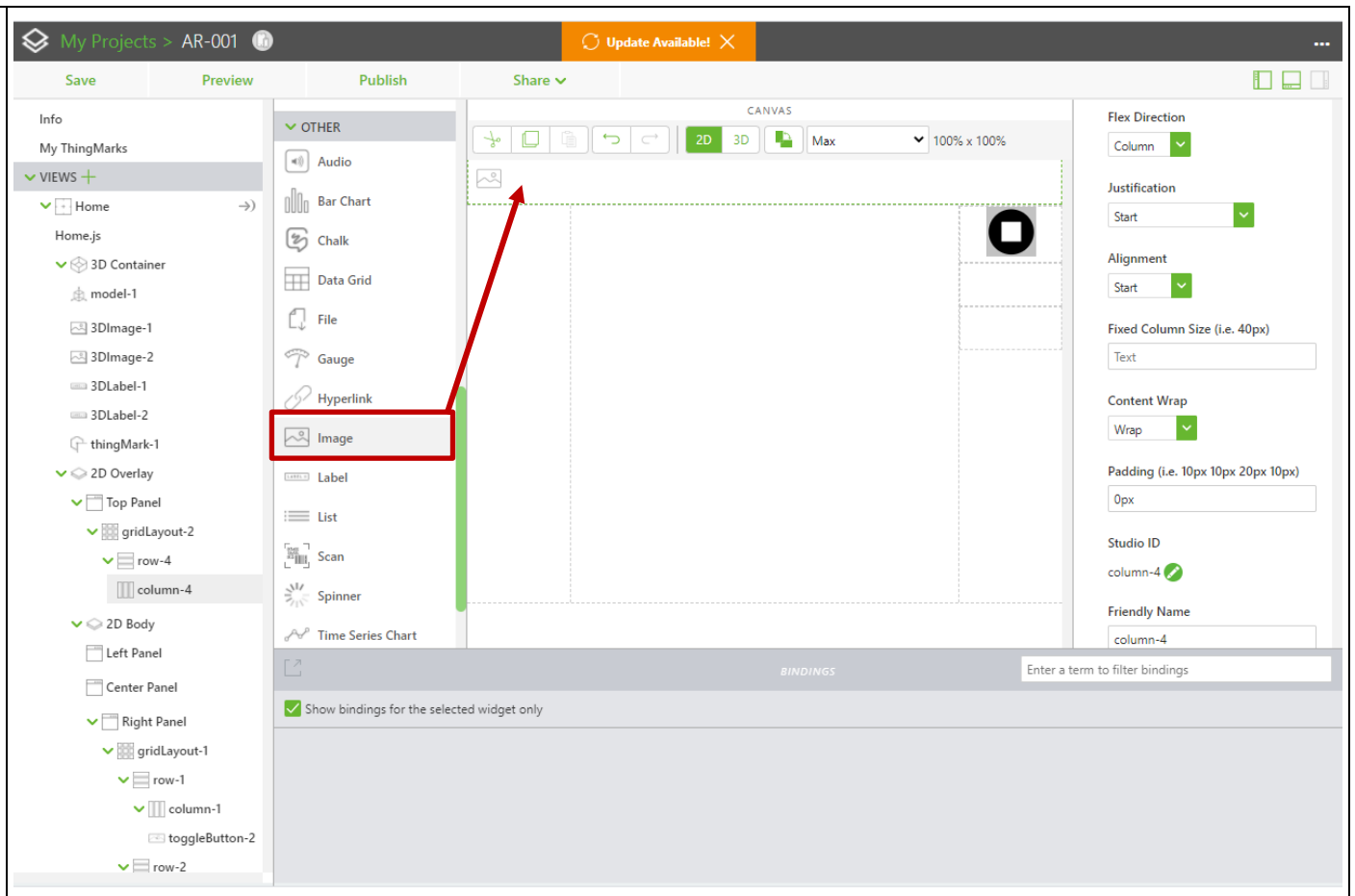
**Grid Layout** уже содержит одну колонку и один ряд. Нам достаточно этого для разметки левого верхнего угла в колонке. Настраиваем колонку для позиционирования того виджета, который будем здесь размещать, т.е. – **логотипа МЭИ. Выравнивание – по**



## Шаг 51.

Добавляем **2D Виджет Image** из **WIDGET→OTHER**.

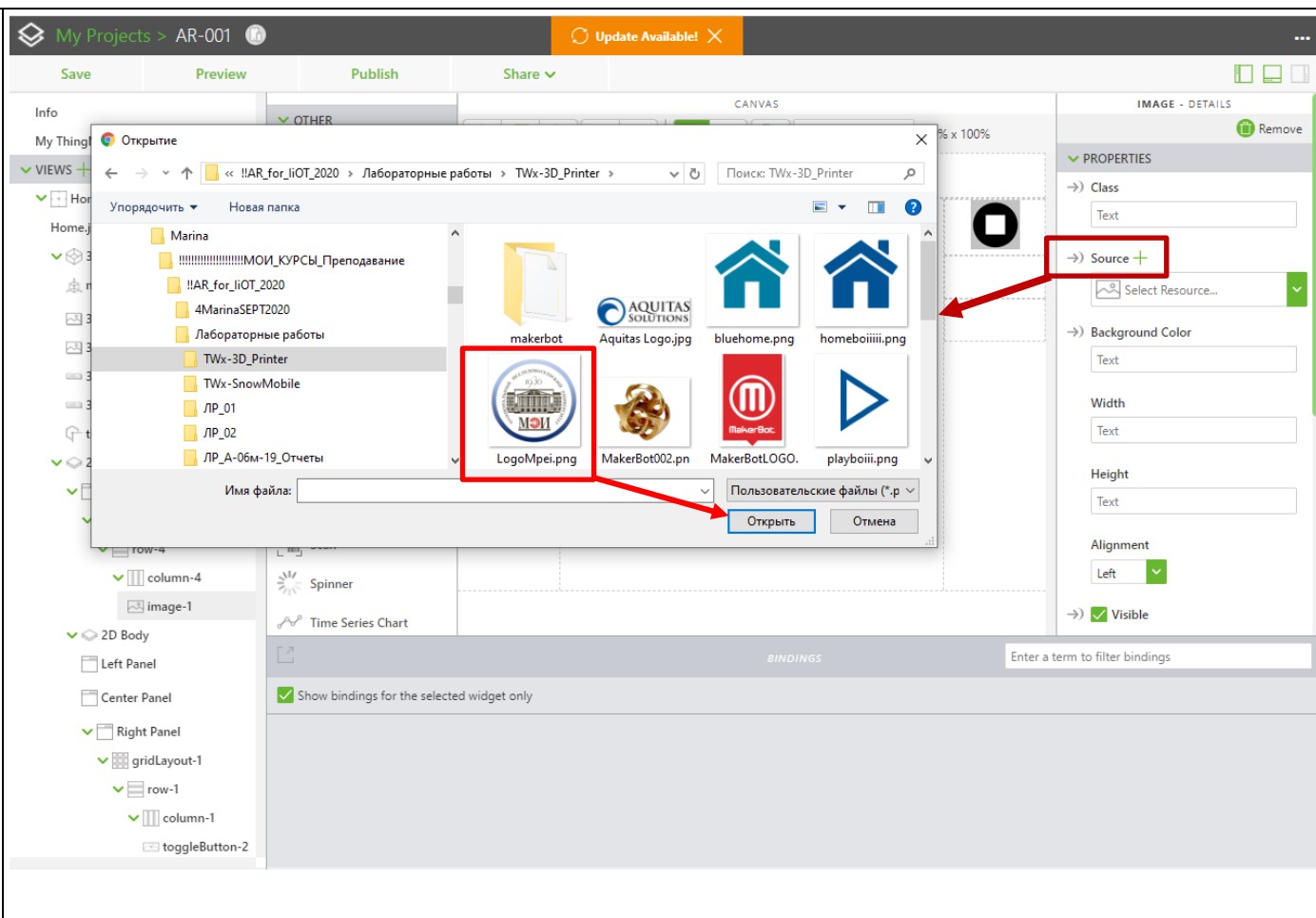
При этом видно, что размещаемый виджет размещается сразу в левый верхний угол, как мы и хотели.





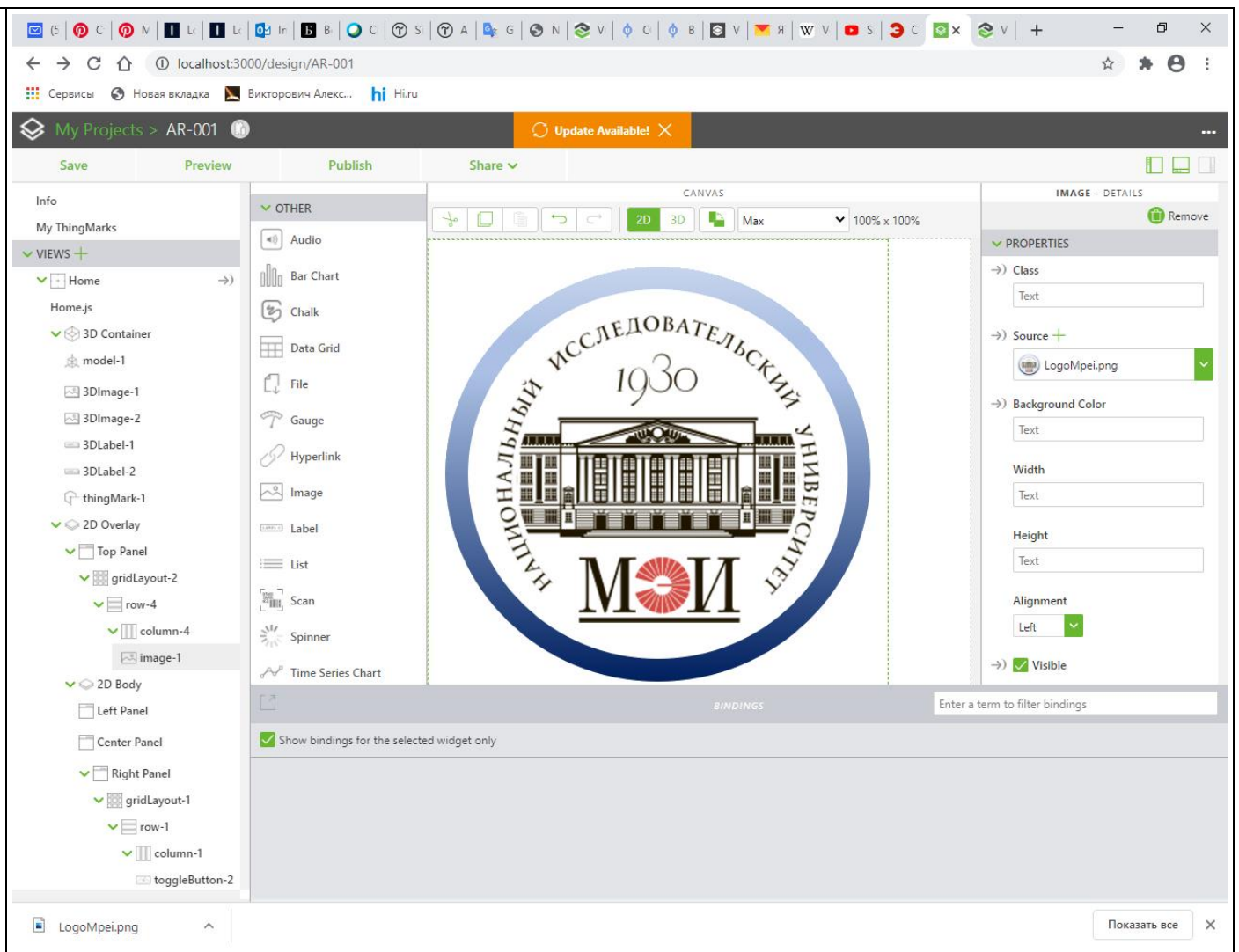
## Шаг 52.

В размещенный **Image** надо поместить самую картинку – т.е. **логотип МЭИ**, заранее подготовленный и находящийся в рабочей директории Проекта локальной ФС. Для этого в зоне редактора «**Свойства компонентов**» для компонента **Image** выбираем в позиции **Source** иконку «+». В открывшемся навигаторе локальной ФС находим нужный нам рисунок.



## Шаг 53.

Полученный результат нас не может удовлетворить – все это нужно промасштабировать, чтобы вписать в **Top Panel**. Берем за аксиому, что вертикальный и горизонтальные размеры картинки (логотипа) должны не превышать **60x60** пикселей.



## Шаг 54.

В зоне «Свойства компонента» для компонента **Image** вводим описанные выше значения размера логотипа:

→ Source +  
LogoMpei.png

→ Background Color  
Text

Width  
60px

Height  
60px

My Projects > AR-001 Update Available! X

Save Preview Publish Share

Info  
My ThingMarks  
VIEWS +  
Home →  
Home.js  
3D Container  
model-1  
3DImage-1  
3DImage-2  
3DLabel-1  
3DLabel-2  
thingMark-1  
2D Overlay  
Top Panel  
gridLayout-2  
row-4  
column-4  
image-1  
2D Body  
Left Panel  
Center Panel  
Right Panel  
gridLayout-1  
row-1  
column-1  
toggleButton-2

OTHER  
Audio  
Bar Chart  
Chalk  
Data Grid  
File  
Gauge  
Hyperlink  
Image  
Label  
List  
Scan  
Spinner  
Time Series Chart

CANVAS  
2D 3D Max 100% x 100%

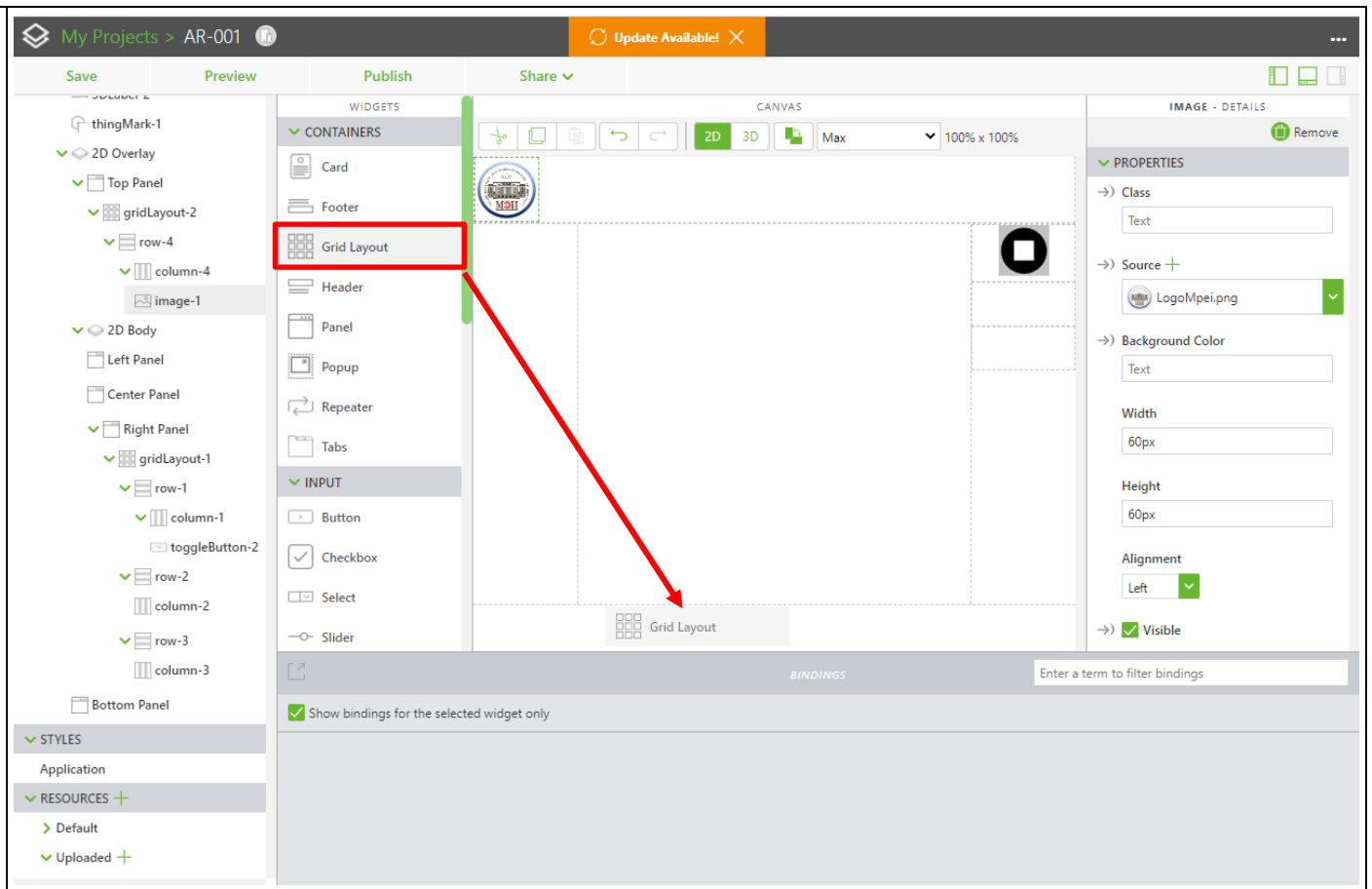
IMAGE - DETAILS  
Remove

PROPERTIES  
Class  
Text  
Source +  
LogoMpei.png  
Background Color  
Text  
Width  
60px  
Height  
60px  
Alignment  
Left  
Visible

BINDINGS  
Enter a term to filter bindings  
Show bindings for the selected widget only

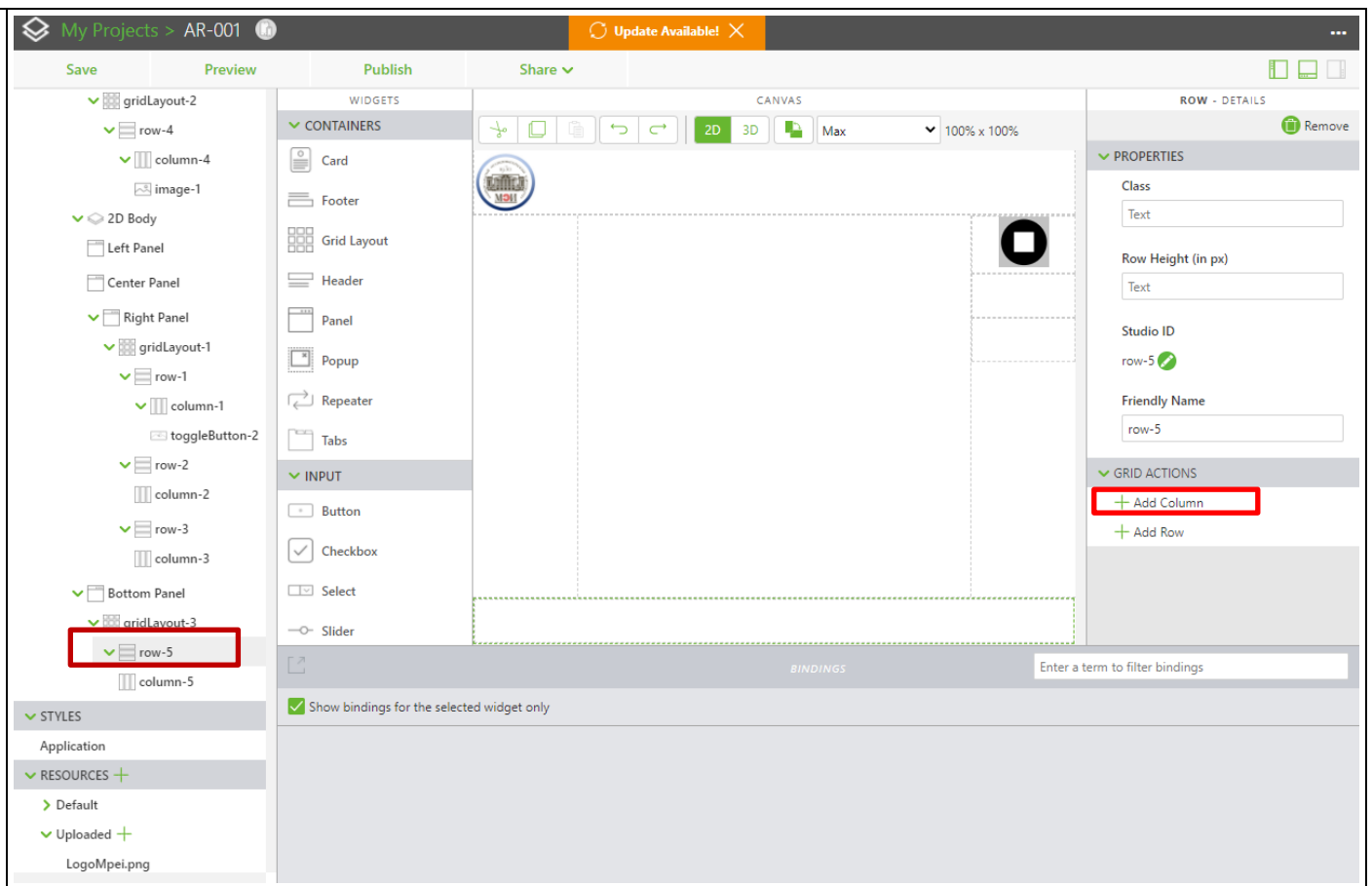
## Шаг 55.

Для реализации возможных пунктов сценария разрабатываемого **Experience** – например, увеличение/уменьшение трехмерного объекта (модели принтера), изменение прозрачности объекта, вращение вокруг оси и пр. – необходимо использовать **2D – Виджет Slider**. Будем размещать эти слайдеры в **Bottom Panel** плоского интерфейса. В нашем примере будем управлять **zoom-ированием** и **масштабированием** модели принтера. Вновь с помощью **Grid Layout** размечаем **Bottom Panel**.



## Шаг 56.

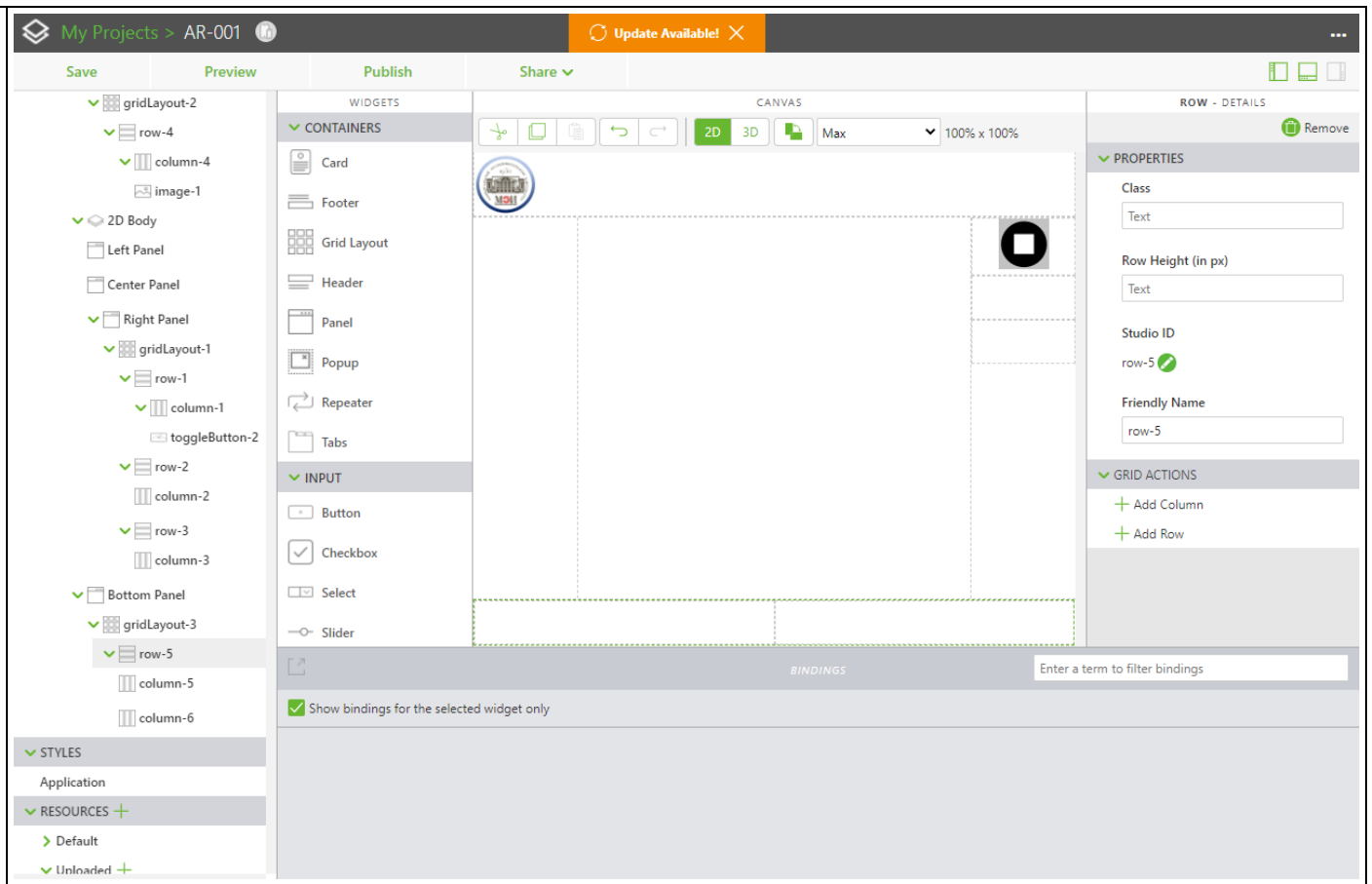
С помощью **Grid Layout** размечаем **Bottom Panel** на 2 столбца → в «Свойствах компонент» В ряду (Row) добавляем **Column** → **Add Column**.



## Шаг 57.

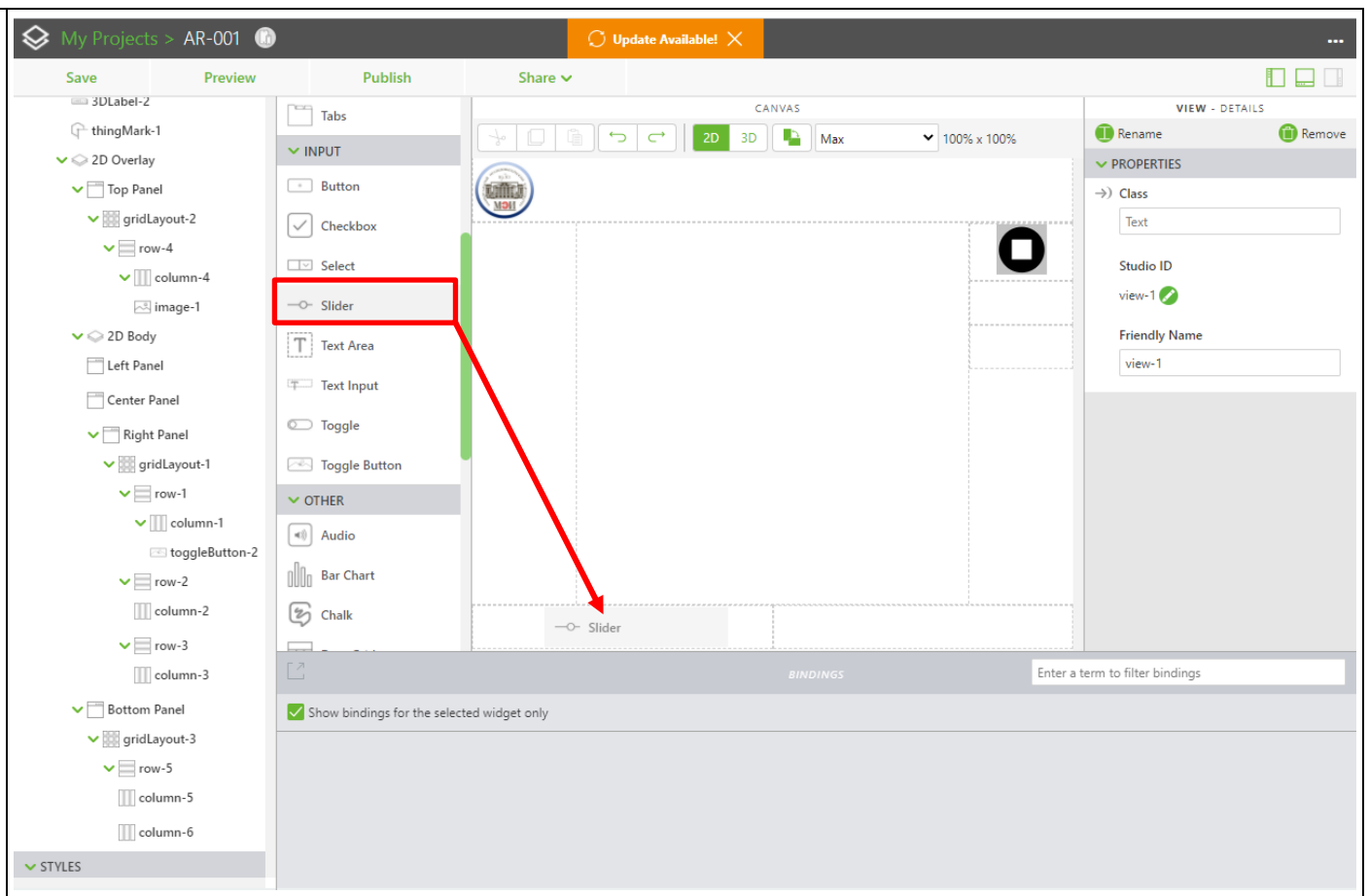
В результате добавления колонки в нижней части плоского меню (**2D CANVAS**) появились 2 колонки, в которые позже будем размещать слайдеры.

**ВНИ!** – по мере выполнения операций следите за наполнением структуры Конфигурации Проекта – зона «Компонентов».



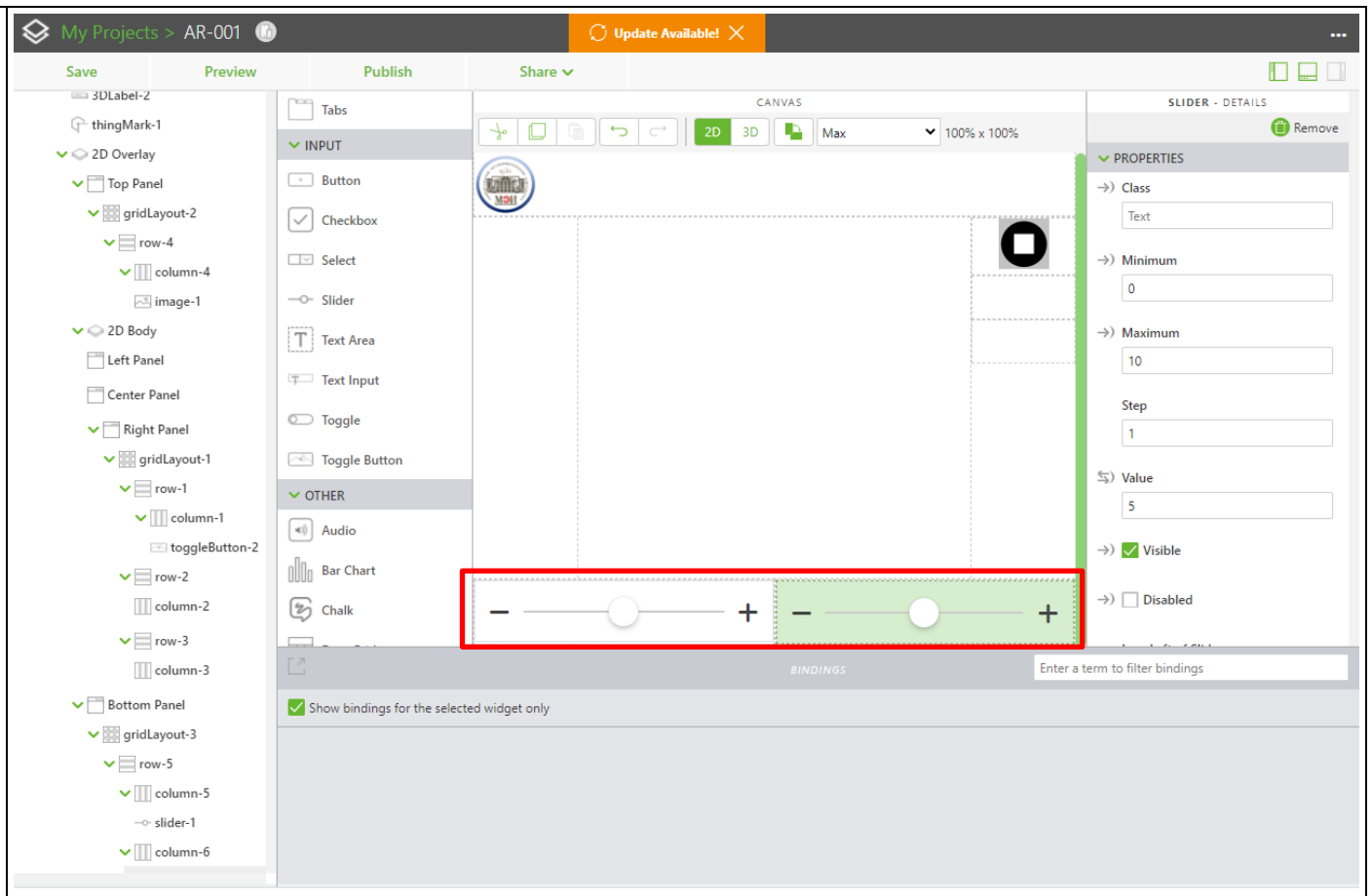
## Шаг 58.

Помещаем слайдер в первую колонку размеченного **Grid'ом Bottom Panel**.



## Шаг 59.

Повторяя описанную на предыдущем шаге процедуру, размещаем на **Bottom Panel**, в разных колонках, **2 слайдера**.





## Шаг 60.

Настроим первый слайдер на управление вращением **3D Model** принтера: настраиваем параметры слайдера – **шаг** (слишком мелкий шаг для мобильного устройства лучше не брать), **начало и конец** вращения (**0 – 360 градусов**), **Value** (начальная позиция самого слайдера).

The screenshot displays the Vuforia Studio interface for configuring a slider widget. The top navigation bar includes 'My Projects > AR-001' and an 'Update Available!' notification. The main workspace is divided into several panels:

- Left Panel:** A tree view showing the project structure, including '2D Overlay', '2D Body', and 'Bottom Panel'. A slider widget is selected under 'column-6'.
- WIDGETS Panel:** A list of available widgets such as Card, Footer, Grid Layout, Header, Panel, Popup, Repeater, Tabs, Button, Checkbox, Select, Slider, Text Area, and Text Input. The 'Slider' widget is highlighted.
- CANVAS:** A central workspace showing a 3D printer model and a slider widget. The slider is currently set to a value of 15.
- SLIDER - DETAILS Panel:** A configuration panel for the selected slider widget. The 'PROPERTIES' section is expanded, and the following values are set:
  - Minimum: 0
  - Maximum: 360
  - Step: 15
  - Value: 180The 'Visible' checkbox is checked, and the 'Disabled' checkbox is unchecked. The 'Icon Left of Slider' is set to 'ion-minus-round'.

## Шаг 61.

Настроим первый слайдер на управление вращением модели принтера вокруг оси Y (вертикальная) с выполнением поворота на 360 градусов, как было описано выше.

Для выполнения этой функции переносим методом «drag-n-drop» маркер поля **Value** на строку **model-1** в зоне «Компоненты».

The screenshot displays the Vuforia Studio interface for a project named 'AR-001'. The interface is divided into several panels:

- Left Panel (Component Tree):** Shows a hierarchical view of the application components. Under '3D Container', 'model-1' is selected and highlighted with a green background. A red arrow points from this component to the 'Value' field in the properties panel.
- WIDGETS Panel:** Lists various UI components like Card, Footer, Grid Layout, Header, Panel, Popup, Repeater, Tabs, Button, Checkbox, Select, Slider, Text Area, and Text Input.
- CANVAS:** The central workspace showing a 3D model of a printer. A slider widget is visible at the bottom of the canvas, with a red dashed box around it.
- SLIDER - DETAILS Panel:** Shows the configuration for the selected slider widget. The 'Value' field is set to 180 and is highlighted with a red box. Other properties include Class (Text), Minimum (0), Maximum (360), Step (15), Visible (checked), and Icon Left of Slider (ion-minus-round).

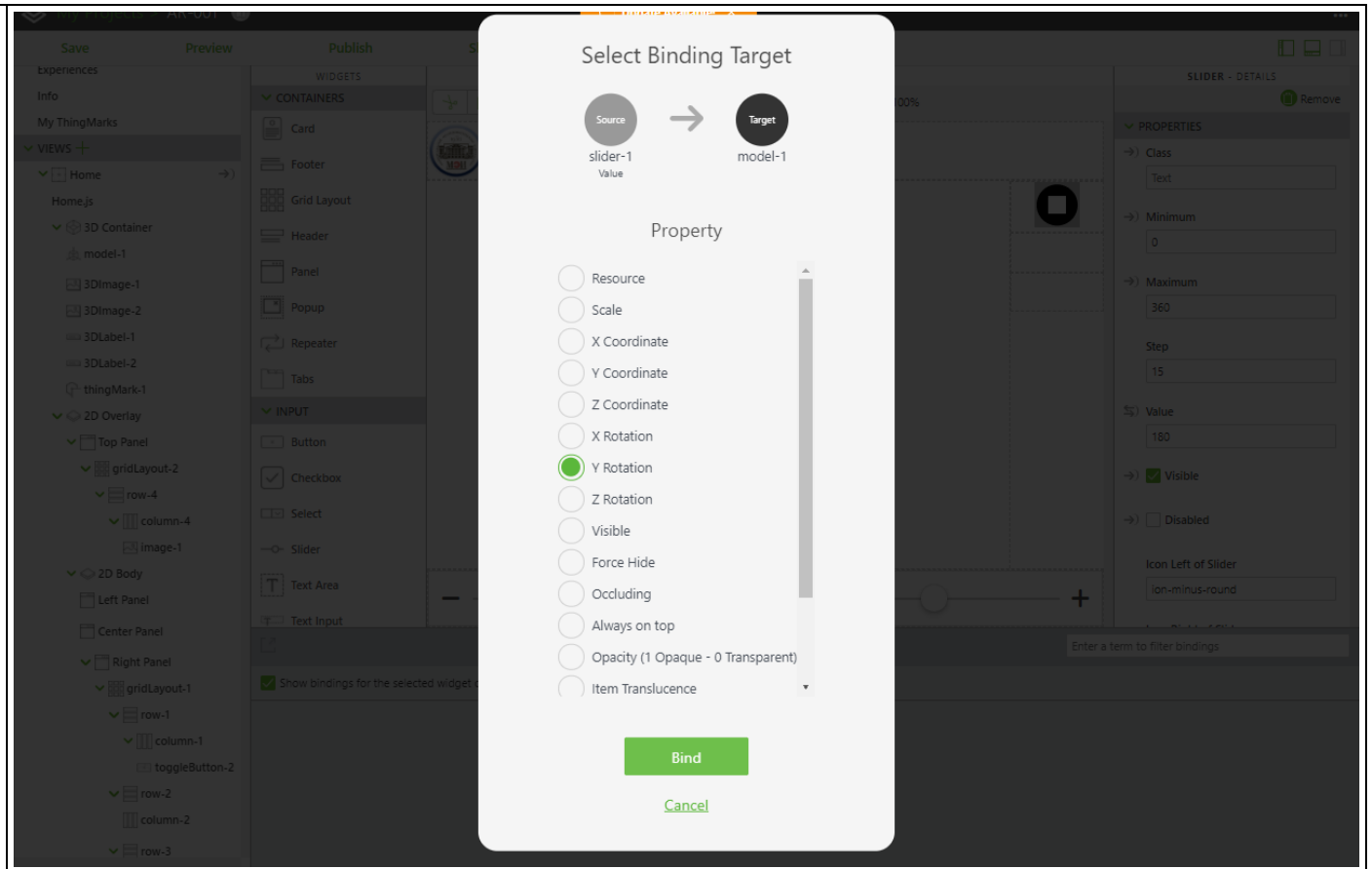
The bottom of the image shows the Windows taskbar with the search bar and various application icons. The system tray shows the time as 13:11 on 13.10.2020.

## Шаг 62.

В результате появляется меню связывания, в котором надо теперь только выбрать то свойство **3D-виджета 3D model-1**, на которое будет влиять значение, генерируемое слайдером **slider-1**.

В нашем случае это должно быть свойство **Y- Rotation** (Вращение вокруг оси **Y**).

Выбираем это свойство и выполняем связывание нажатием кнопки **Bind** в нижней части меню.



## Шаг 63.

Аналогичные действия производим со вторым слайдером **slider-2**. Настраиваем его на масштабирование **3D Модели** от 0.5 до 2.0 с шагом 0.25.

The screenshot displays the Vuforia Studio interface for a project named "AR-001". The interface is divided into several panels:

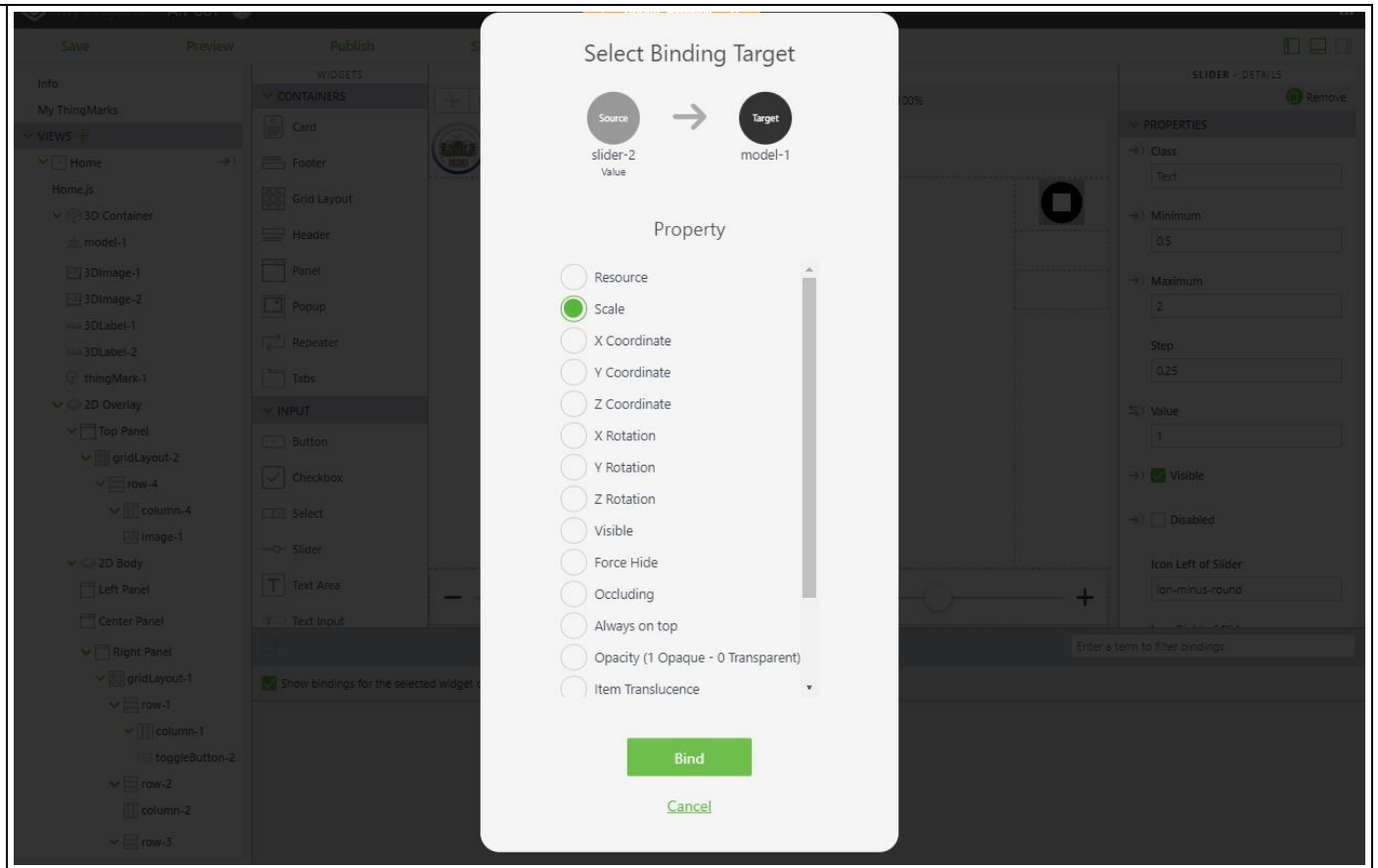
- Info Panel (Left):** Shows a tree view of the application structure. Under "3D Container", "model-1" is selected, indicated by a red arrow.
- WIDGETS Panel (Middle-Left):** Lists various widget types such as Card, Footer, Grid Layout, Header, Panel, Popup, Repeater, Tabs, Button, Checkbox, Select, Slider, Text Area, and Text Input.
- CANVAS (Center):** Shows a 3D model of a slider widget on a grid. The slider has a value of 1, with minimum and maximum values of 0.5 and 2.0 respectively.
- SLIDER - DETAILS Panel (Right):** Shows the configuration options for the selected slider widget. The "Value" field is highlighted with a red box and contains the number 1. Other visible fields include "Class" (Text), "Minimum" (0.5), "Maximum" (2), "Step" (0.25), "Visible" (checked), and "Icon Left of Slider" (ion-minus-round).

## Шаг 64.

В результате появляется меню связывания, в котором надо теперь только выбрать то свойство **3D-виджета 3D model-1**, на которое будет влиять значение, генерируемое слайдером **slider-2**.

В нашем случае это должно быть свойство **Scale** (масштабирование).

Выбираем это свойство и выполняем связывание нажатием кнопки **Bind** в нижней части меню.



## Шаг 65.

В результате в зоне «Связи» можно наблюдать еще две образовавшиеся связи для элементов интерфейса и модели изучаемого устройства.

The screenshot displays the Vuforia Studio interface. On the left, the 'VIEWS' tree shows a hierarchy: Home > Home.js > 3D Container > model-1. The 'WIDGETS' panel lists various UI components, with 'Slider' selected under the 'INPUT' category. The central 'CANVAS' shows a 3D scene with a slider widget. On the right, the 'MODEL - DETAILS' panel shows properties for 'TestV1.pvz', including Scale, X, Y, and Z coordinates and rotations. At the bottom, the 'BINDINGS' panel is highlighted with a red box, showing two binding entries:

Source	Target
Binding Expression: app.view[Home].wdg[slider-1][value]	Widget: model-1 Property: ry
Binding Expression: app.view[Home].wdg[slider-2][value]	Widget: model-1 Property: scale

## Шаг 66.

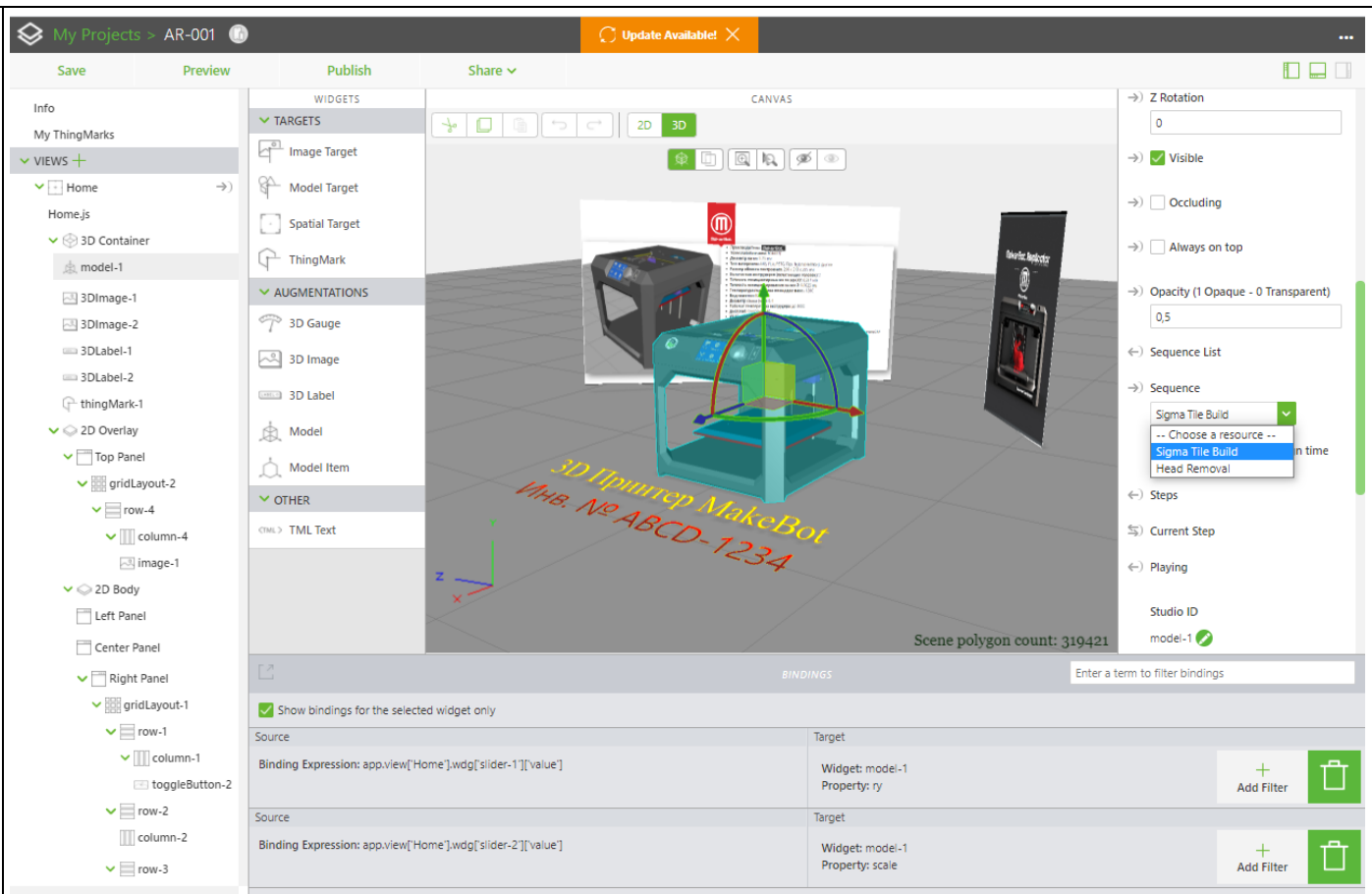
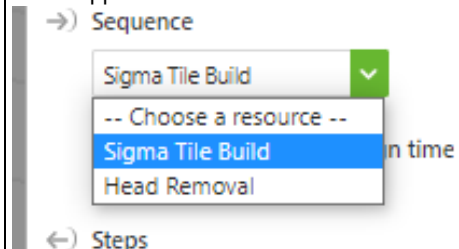
Просмотрите промежуточный результат в режиме **Preview**.



## Шаг 67.

Состав контента **AR-проекта** может содержать визуализацию производственных процессов, пошагового описания ремонтных операций, для которых может потребоваться сбор/разбор устройства и т.д. Эти визуализации – последовательности, **sequences** - определяются на составе базовой модели (в нашем случае – это **3D модель** принтера), сохраняются в формате **.pvi** и конвертируются в общий архив модели **.pvz**. В разрабатываемом **AR Проекте** мы хотим развернуть такую последовательность (если она есть, заранее подготовленная) и в **2D-Интерфейсе** управлять ее пошаговой визуализацией (**просмотр вперед, просмотр назад**) с помощью **2D Виджетов**. Например – с помощью уже известного нам **Toggle Button**.

Вначале определим наличие **sequences** в архиве модели. Для этого в зоне «**Компоненты**» выбираем нашу модель, а в зоне «**Свойства компонентов**» найдем свойство **sequences**. В открывшемся списке, если он не пуст, выберем **нужную нам sequence**. В нашем примере эта последовательность называется →

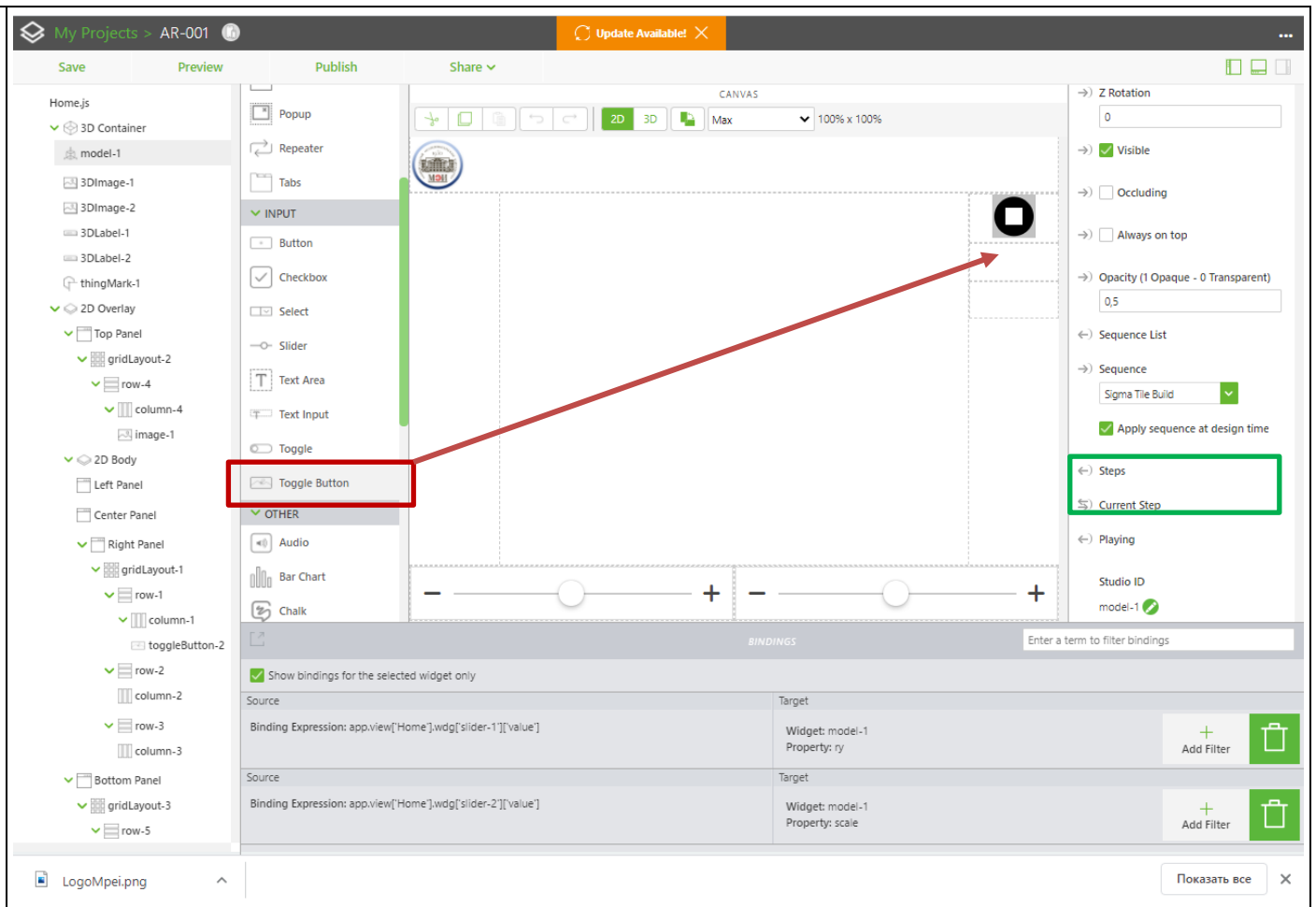


Выбранная последовательность **Sigma Tile Build** содержит пошаговое описание процесса **3D принтинга** (аддитивного производства) кожуха (корпус и крышка) для размещения в нем микрокомпьютера **RPi** и блока датчиков **SenseHat**.



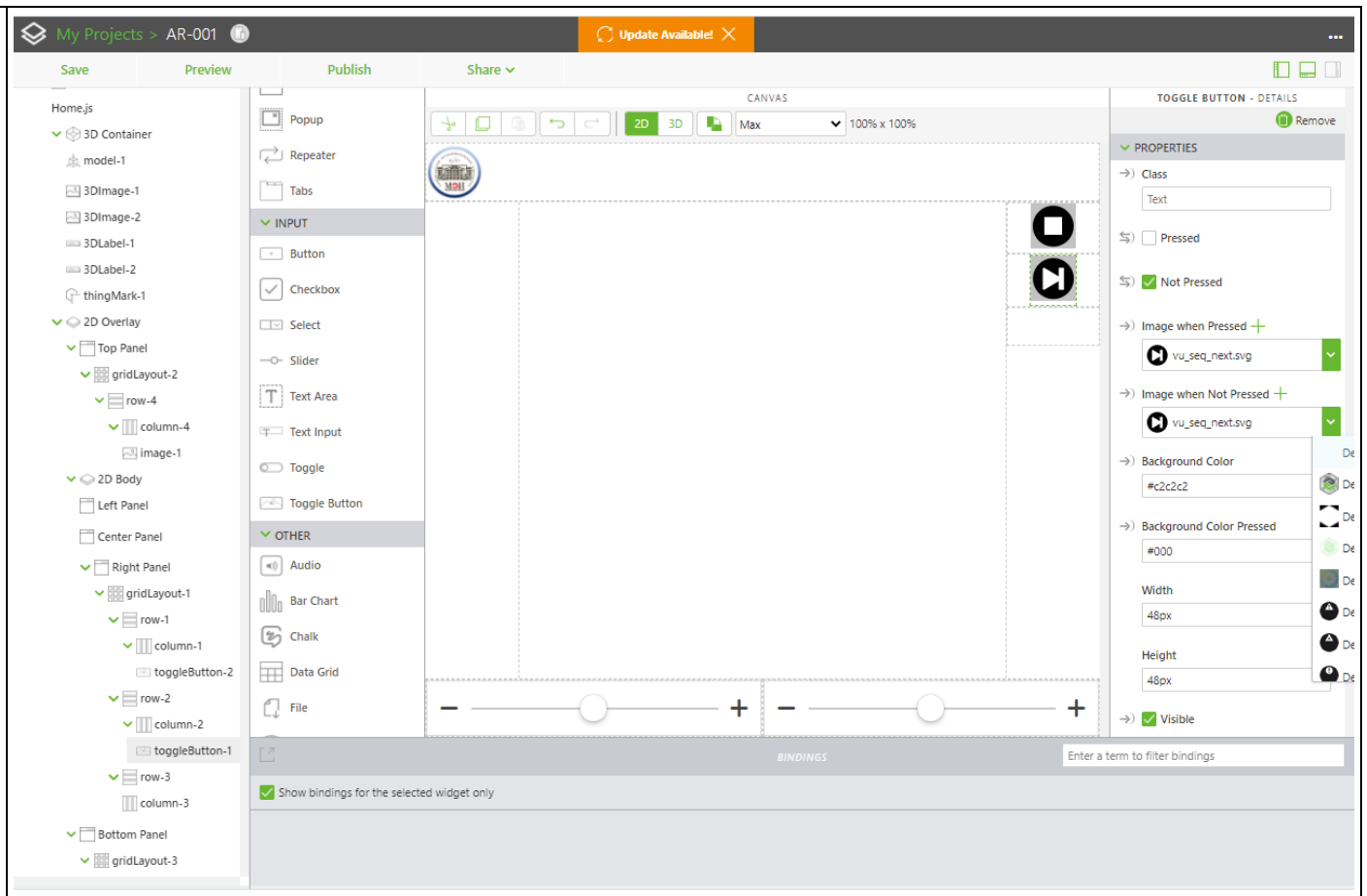
## Шаг 68.

Для размещения в **2D Интерфейсе** элементов управления выбранной последовательностью, переходим в и размещаем **Toggle Button (в 2D – Канве)** в очередном ряду правой панели «Стекла» - **Right Panel**. Этот **Toggle Button** будет управлять просмотром последовательности пошагово «вперед».



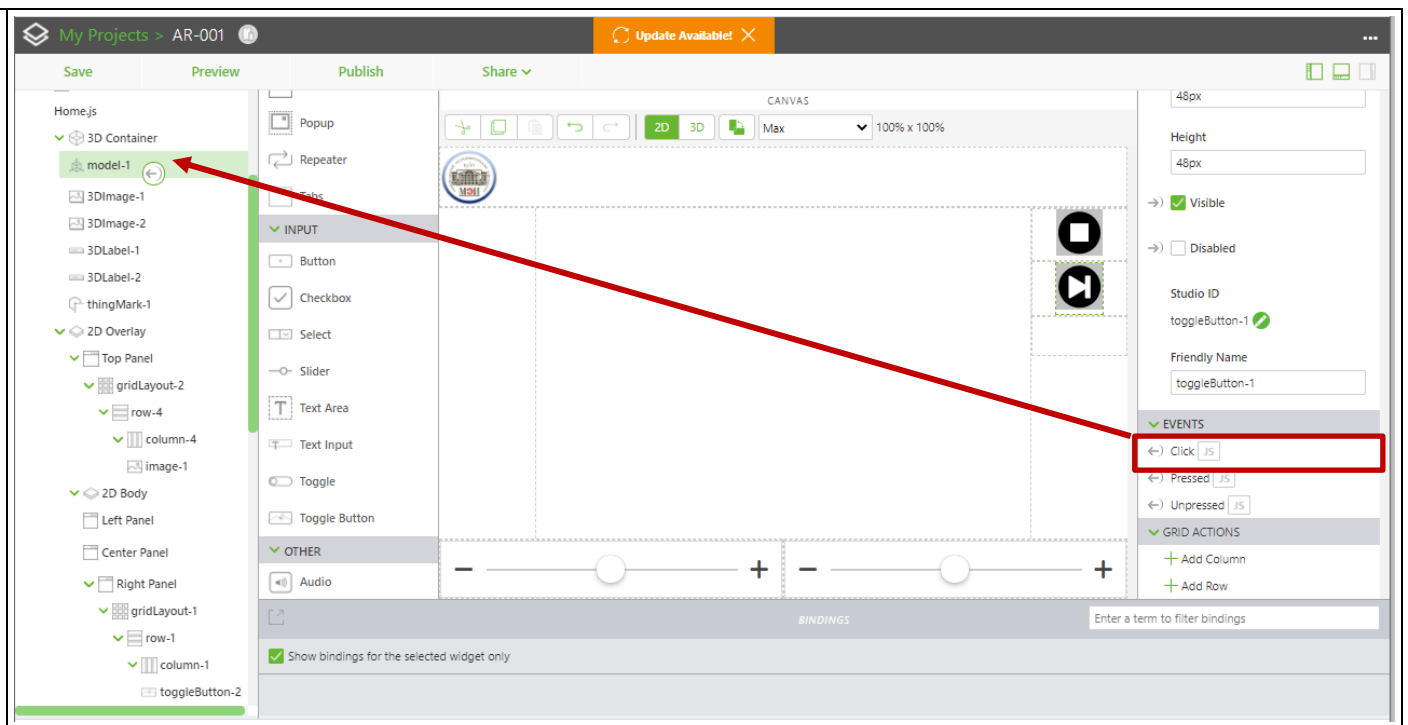
## Шаг 69.

Из галереи пиктограмм (**Images**) выберем изображения для «нажатого» (**Pressed**) переключателя и для «не нажатого» (**Not Pressed**) переключателя «**Просмотр вперед**».



## Шаг 70.

Управление пошаговым просмотром будем осуществлять с использованием свойства **Click** кнопки **Toggle Button**, которое располагается в подмножестве **EVENTS** (события) в зоне «Свойства компонентов». Связываем **Click** с **3D-моделью**.

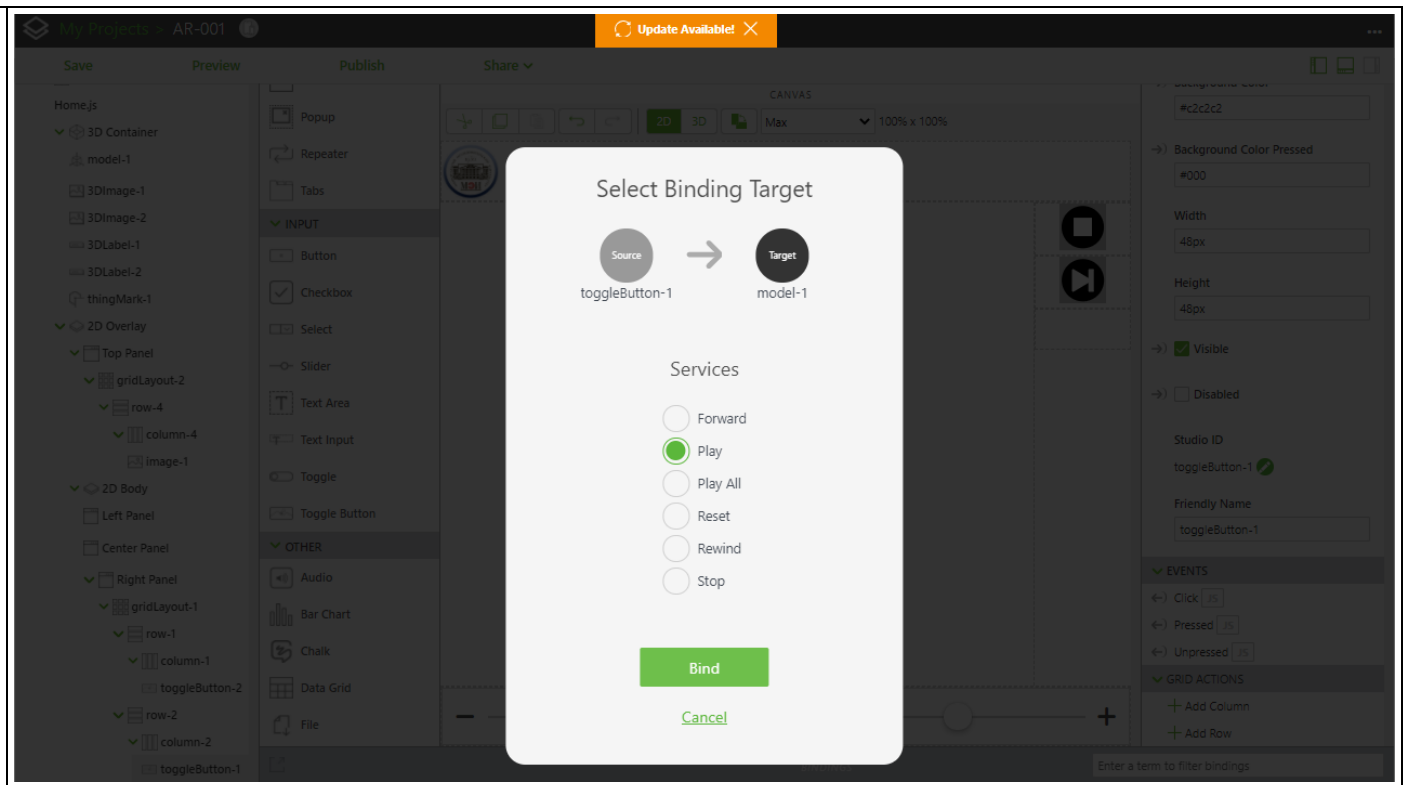


## Шаг 71.

В результате появляется меню связывания, в котором надо теперь только выбрать тот **СЕРВИС 3D model-1**, который будет запускаться при нажатии на связываемый Toggle Button

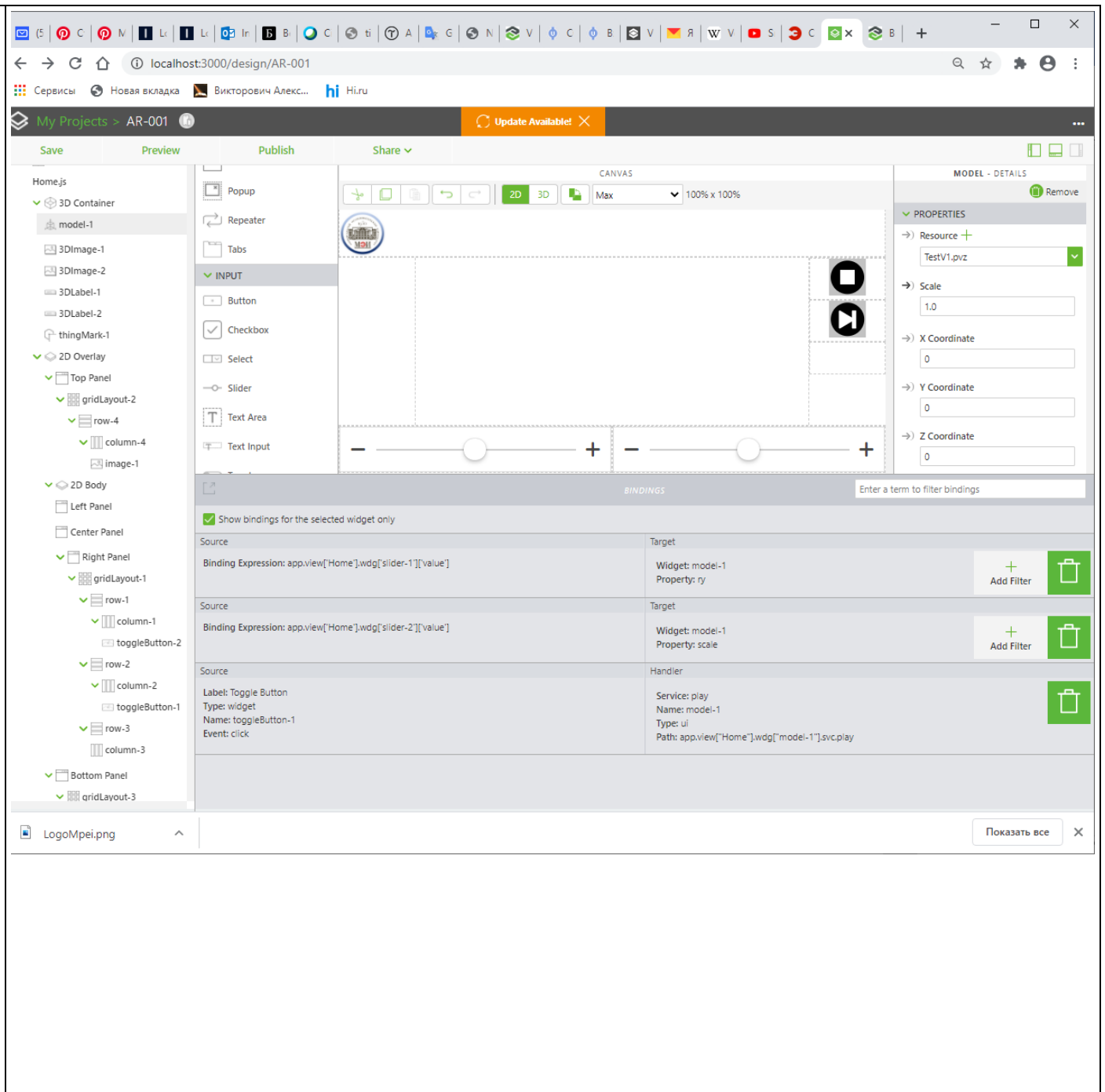
В нашем случае это должен быть **СЕРВИС Play**. Обратите внимание -- **СЕРВИС Play All** мы не выбираем потому, что он реализует полный просмотр последовательности без разделения на шаги.

Выбираем этот **СЕРВИС** и выполняем связывание нажатием кнопки **Bind** в нижней части меню.



## Шаг 72.

Обратите внимание – в результате количество связей у модели постоянно увеличивается. Это можно проконтролировать в зоне «Связи» для выбранной модели в зоне «Компоненты».



The screenshot displays the Vuforia Studio interface for a project named 'AR-001'. The interface is divided into several panels:

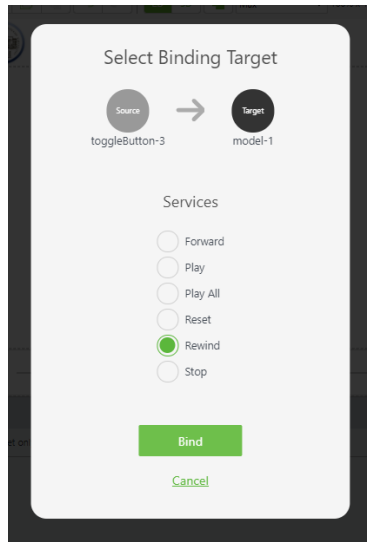
- Left Panel (Component Tree):** Shows a hierarchical structure of components. The 'model-1' component is selected, revealing its sub-components: '3D Container', '3D Image-1', '3D Image-2', '3D Label-1', '3D Label-2', 'thingMark-1', '2D Overlay', 'Top Panel', 'gridLayout-2', 'row-4', 'column-4', 'image-1', '2D Body', 'Left Panel', 'Center Panel', 'Right Panel', 'gridLayout-1', 'row-1', 'column-1', 'toggleButton-2', 'row-2', 'column-2', 'toggleButton-1', 'row-3', 'column-3', 'Bottom Panel', and 'gridLayout-3'.
- Center Panel (Canvas):** Displays a 3D scene with a 'Popup' widget and a 'Slider' widget. The 'Slider' widget is currently selected.
- Right Panel (MODEL - DETAILS):** Shows the properties of the selected widget. The 'PROPERTIES' section includes: 'Resource' (TestV1.pvz), 'Scale' (1.0), 'X Coordinate' (0), 'Y Coordinate' (0), and 'Z Coordinate' (0).
- Bottom Panel (BINDINGS):** Shows a table of bindings for the selected widget. The table has columns for 'Source', 'Target', and 'Handler'. Three bindings are listed:

Source	Target	Handler
Binding Expression: app.view["Home"].wdg["slider-1"]["value"]	Widget: model-1 Property: ry	
Binding Expression: app.view["Home"].wdg["slider-2"]["value"]	Widget: model-1 Property: scale	
Label: Toggle Button Type: widget Name: toggleButton-1 Event: click		Service: play Name: model-1 Type: ui Path: app.view["Home"].wdg["model-1"].svc.play

At the bottom of the interface, there is a file manager showing 'LogoMpei.png' and a button labeled 'Показать все'.

## Шаг 73.

Вспомним, что сценарием предусмотрен просмотр последовательности не только вперед, но и назад. Для создания соответствующей кнопки – **2D Виджета Toggle Button** повторим шаги 68 – 71, выбрав другую пиктограмму для кнопки и другой **СЕРВИС** для связывания (**Rewind**):



Source	Handler
Label: Toggle Button Type: widget Name: toggleButton-3 Event: click	Service: rewind Name: model-1 Type: ui Path: app.view["Home"].wdg["model-1"].svc.rewind

## Шаг 74.

Проверяем полученный результат в режиме Preview.



Откат на шаг назад →

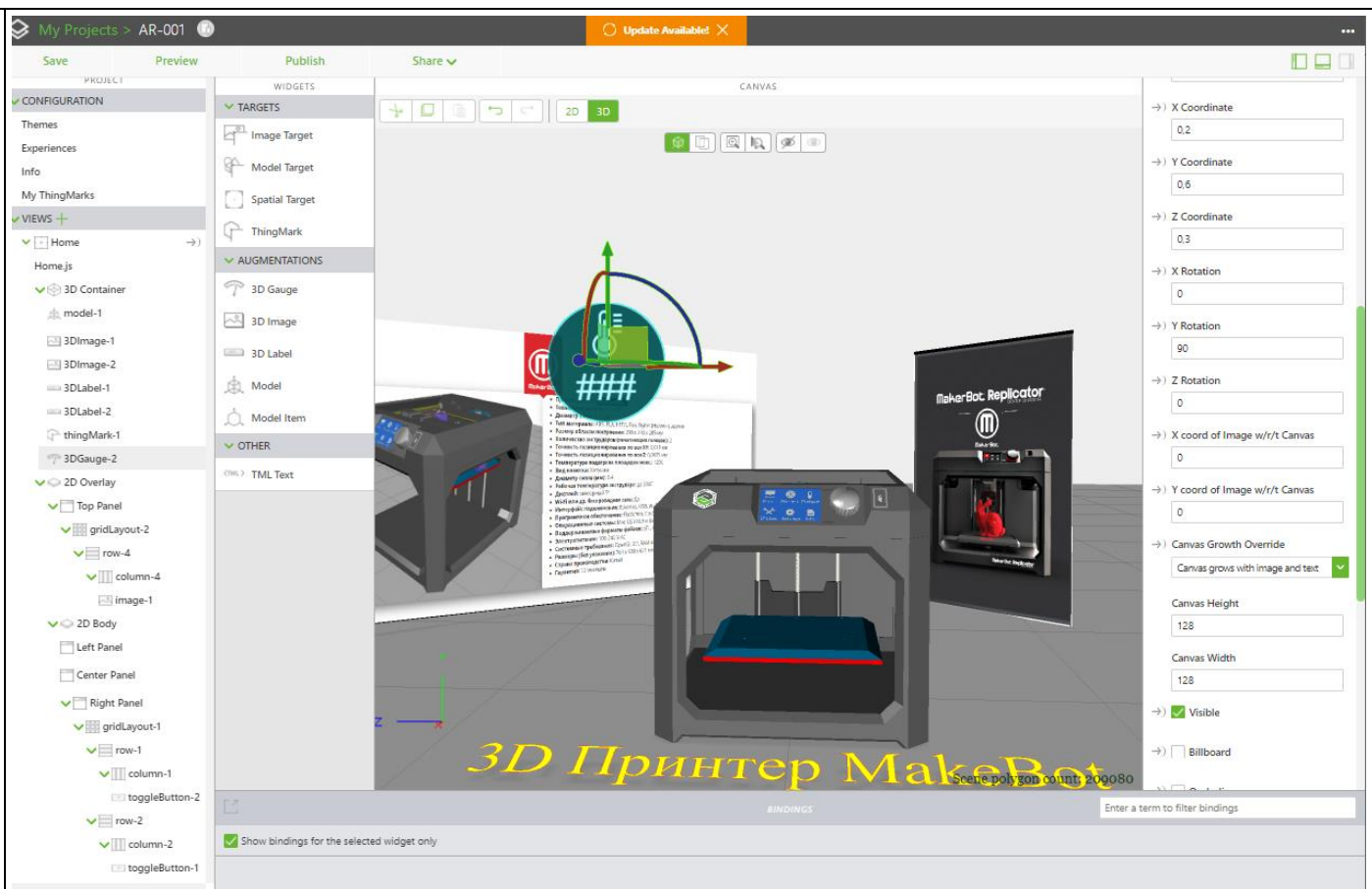


## Шаг 75.

Созданный **AR – проект (Experience)** должен содержать объекты, предусматривающие связывание с ним информационной модели «умная вещь», в качестве информационного наполнения дополненной реальности. Информационная модель умной вещи в рамках данной ЛР нам не доступна. Но мы можем заранее предусмотреть для нашего **Experience** виджеты, в которые в ходе выполнения полноценной работы Приложения ДР в составе развернутой платформы **IoT** должны будут транслироваться актуальные значения характеристик умной вещи от сервера **IoT**. Вопросы связывания с сервером и публикации нашего проекта будут рассмотрены в следующих ЛР. Такие Виджеты (**2D** и **3D**) предусмотрены в системе и называются **Gauge** (прибор, циферблат. Индикатор, индикаторная панель и пр.)

Наиболее удобными для нас являются **3D Gauge**.

Переходим к размещению этих виджетов в сцене. Разместим в нужном нам месте индикаторы нужного размера, например для отображения температуры филамента в сопле и запаса филамента. Актуальные значения предполагается получать с сервера **IoT**.



Обратите внимание на свойства **3D Gauge** – вы можете управлять видимостью, размерами, координатами, положением относительно смотрящего (отключайте **Billboard**, если не хотите, чтобы индикатор все время смотрел на вас), и т.д. Размещение этого виджета происходит аналогично предыдущим **3D Виджетам (3D Model, 3D Image, 3D Label)**.



## Шаг 76.

Размещаем второй 3D Gauge аналогичным образом

The screenshot displays the Vuforia Studio interface for an AR application. The main canvas shows a 3D scene with a printer, a gauge, and a sign. The left sidebar contains a 'WIDGETS' panel with '3D Gauge' selected. The right sidebar shows the 'Resource' and 'Font' settings for the selected widget. The 'Font' settings include 'Font' (Arial), 'Font Size' (40px), 'Text Attributes' (fill:rgb(255, 255, 255, 1);textbaselin), and various coordinate and rotation fields. The 'Scene polygon count' is 207512.

**WIDGETS**

- TARGETS
  - Image Target
  - Model Target
  - Spatial Target
  - ThingMark
- AUGMENTATIONS
  - 3D Gauge
  - 3D Image
  - 3D Label
  - Model
  - Model Item
- OTHER
  - TML Text

**Resource**

- vu\_sinewave.svg

**Font**

- Font: Arial
- Font Size: 40px
- Text Attributes: fill:rgb(255, 255, 255, 1);textbaselin
- Enable State-Based Formatting:

**Text Properties**

- X coord of Text w/r/t Canvas: 64
- Y coord of Text w/r/t Canvas: 94
- Scale: 1.5000
- X Coordinate: 0.2
- Y Coordinate: 0.6
- Z Coordinate: 0.1
- X Rotation: 0
- Y Rotation: 0

Scene polygon count: 207512

## Шаг 77.

Проверяем полученный результат в режиме Preview.



## 4. Практическое задание

---

<p><b>Шаг 1.</b> Создайте <b>Project – AR-Проект (Experience)</b>. Имя «_____» Тип таргетинга - <b>ThingMark</b></p>	
<p><b>Шаг 2.</b> Разместите в проекте <b>3D-модель (архив .pvz)</b></p>	
<p><b>Шаг 3.</b> Используйте в проекте <b>3D-виджеты</b>:</p> <ul style="list-style-type: none"><li>• <b>3D-Image</b></li><li>• <b>3D-Label</b></li></ul>	
<p><b>Шаг 4.</b> Используйте в проекте 2D-виджеты:</p> <ul style="list-style-type: none"><li>• <b>Grid (Grid Layout)</b></li><li>• <b>Toggle Button</b> для управления свойствами 3D-виджетов;</li><li>• <b>Slider</b> для управления свойствами 3D-виджетов;</li><li>• <b>Image</b> для размещения логотипов разработчика.</li></ul>	
<p><b>Шаг 5.</b> Выполните превью проекта, убедитесь в правильном выполнении всех предыдущих шагов.</p>	
<p><b>Шаг 6.</b> Продемонстрируйте результат – просмотр проекта средствами Preview (в рамках последующих ЛР после полноценной регистрации)</p>	

---

**В соответствии с изложенной в пункте 4. структурой самостоятельной работы в рамках Лабораторной работы №2 каждый студент выполняет AR - Проект в соответствии с индивидуальным заданием. Описание задания вместе с исходными данными ( файл .rvz, файл .pdf) разослано по электронной почте.**