



ЛАБОРАТОРНАЯ РАБОТА №4

Vuforia Studio: Создание приложения дополненной реальности (часть II 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 по курсу ДР для ПИТ
03.05.2026	Пирогова М.А.	1	Редактирование Описания Лабораторной работы №4 для курса ТДР/2026 г. (ЛРН№4)

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 промышленного 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 особых 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 (например, изменения их видимости/прозрачности, масштабирования и вращения и т.д.)
 - **Кроме того, рядом** с объектом должны располагаться плоские «экраны» (как минимум – один), 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 **первой части** данной ЛР **не публикуется на сервере**, а лишь сохраняется в созданном Проекте. Его работоспособность оценивается преподавателем в режиме предварительного просмотра (просмотра присланных сканов экрана) средствами редактора **Vuforia Studio**.

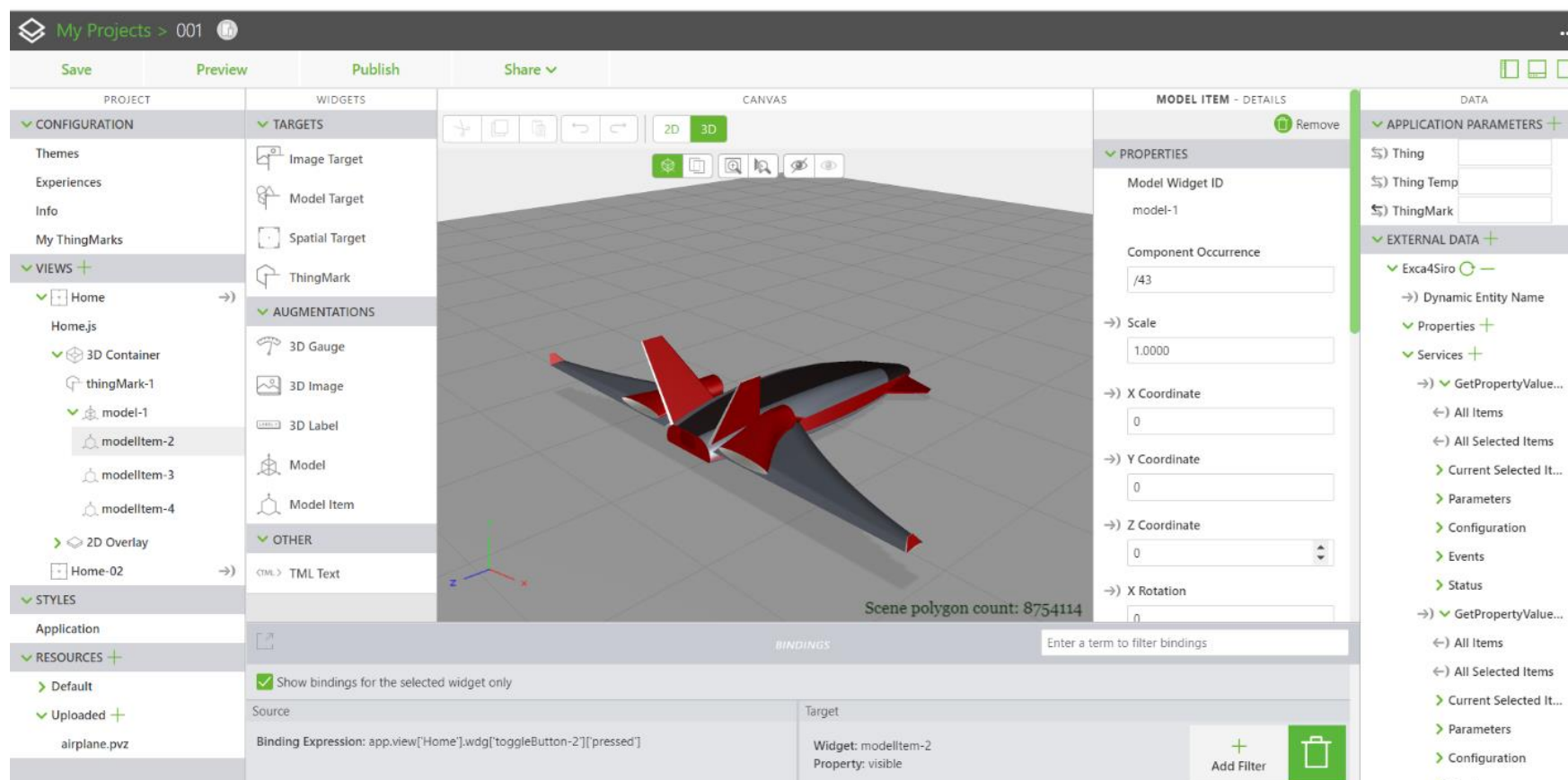
2. Интерфейс Vuforia Studio

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

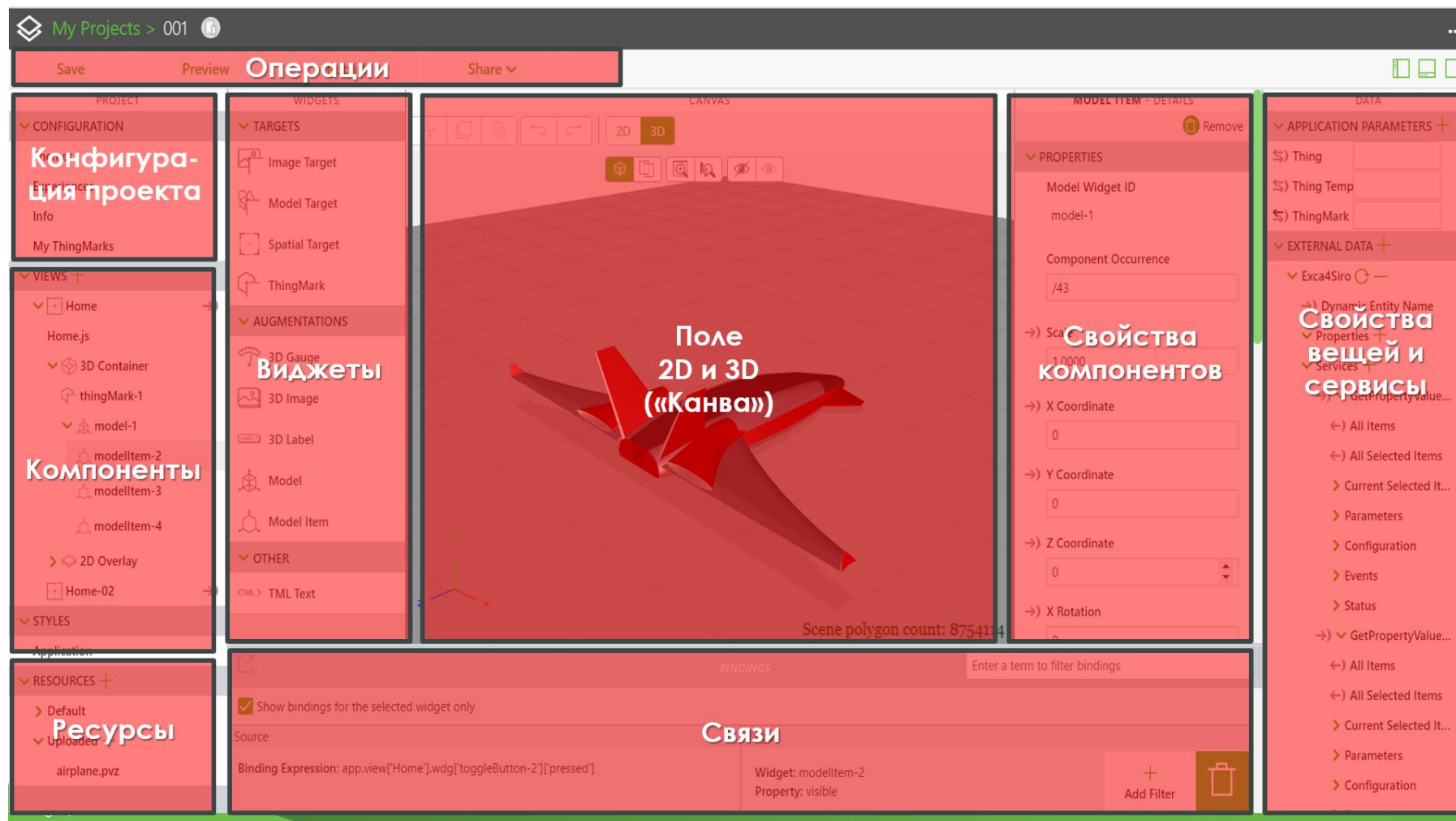
Для получения доступа к интерфейсу **Vuforia Studio** необходимо в окне **URL (URI)** стандартного **Web-просмотрщика** (**PTC** рекомендует преимущественное использование **Google Chrome**) ввести адрес и порт сервера **Vuforia Studio**. Если **Vuforia Studio** установлена на локальную машину разработчика,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



Более 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 **tapret - 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 локальном компьютере под управлением **MS Windows 10**.

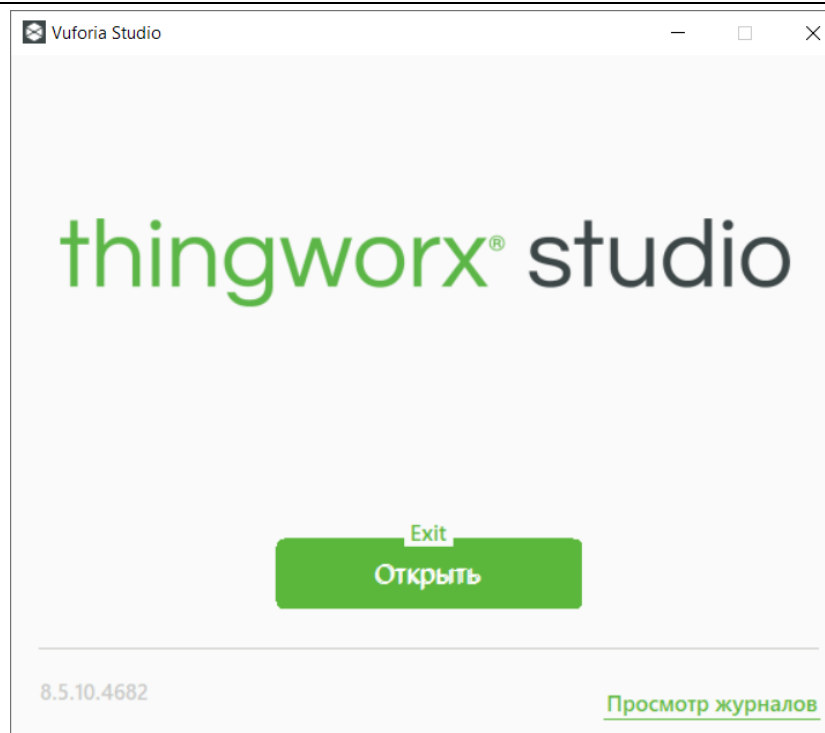
Дожидаемся появления окна входа.

При нажатии на кнопку

Открыть

Автоматически запускается **Web-просмотрщик** «по умолчанию» на этом же компьютере (желательно, по рекомендации **PTC**, 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.

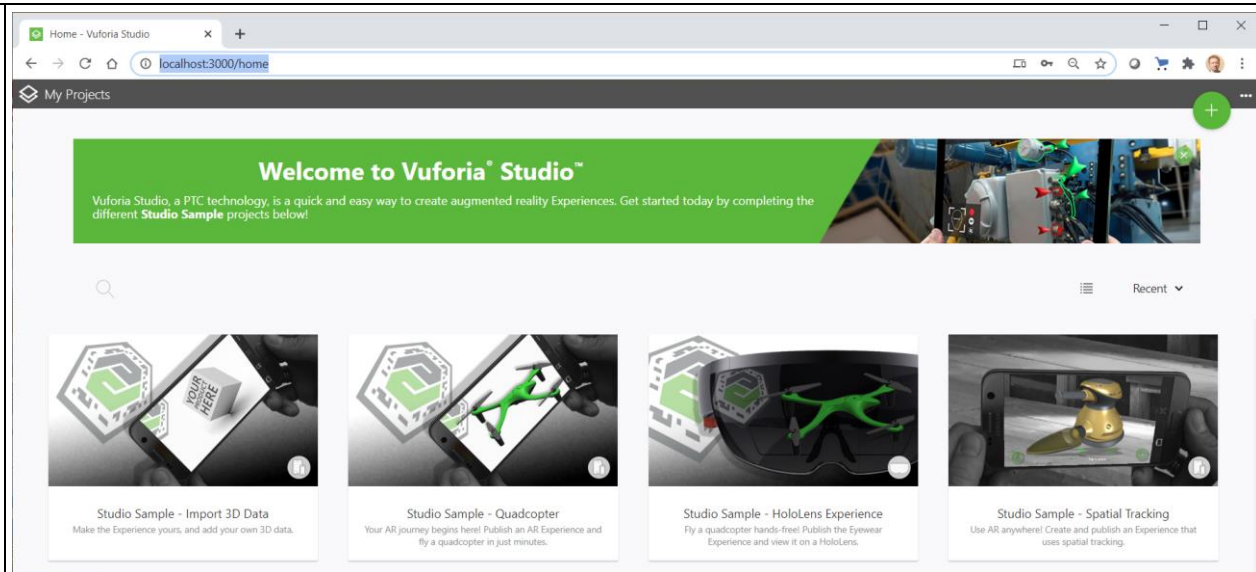
Даже при первом старте **Vuforia Studio** у разработчика имеется возможность работать с четырьмя примерами проектов – **Studio Sample**. Это заранее подготовленные примеры для различных типов конечных устройств и различных методов таргетинга (топологической привязки виртуальных информационных объектов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

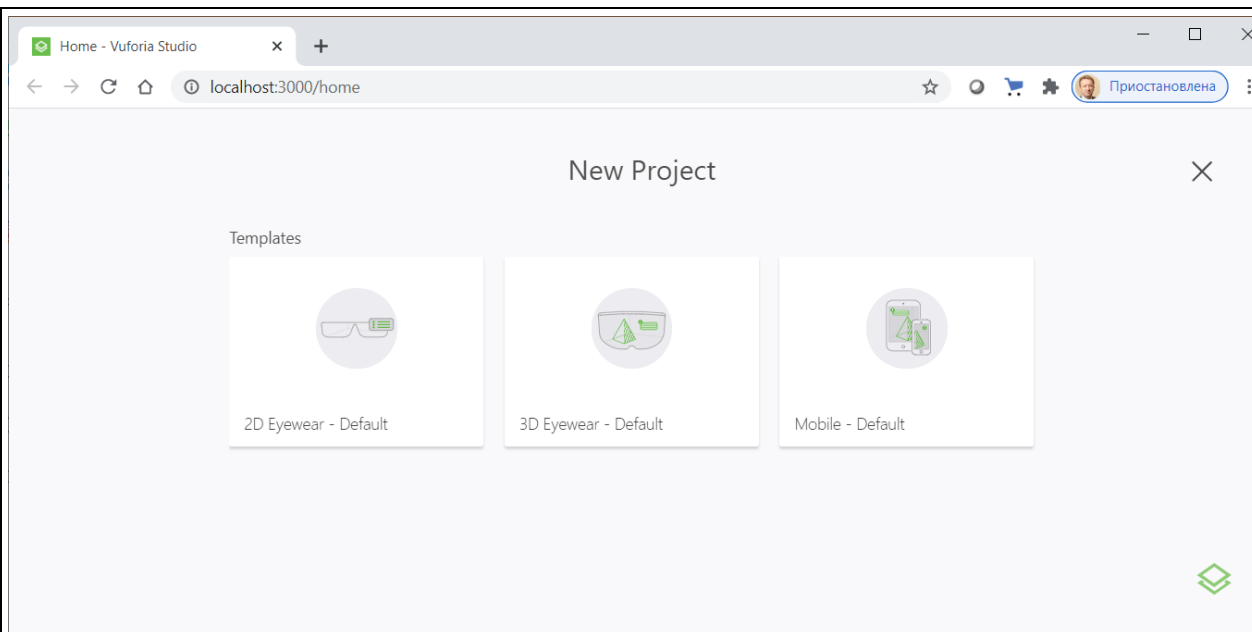


Шаг 3.

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**»

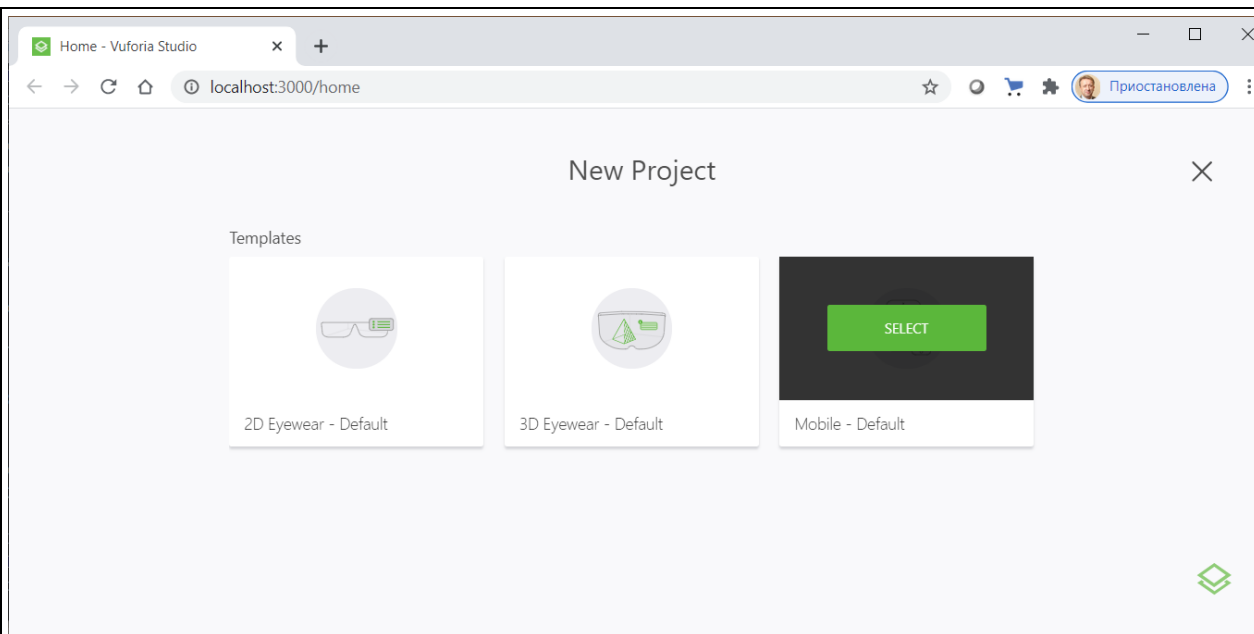


Шаг 4.

Нажимаем кнопку



И выбираем тип проекта «Mobile»



Шаг 5.

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

- **Project Name:**

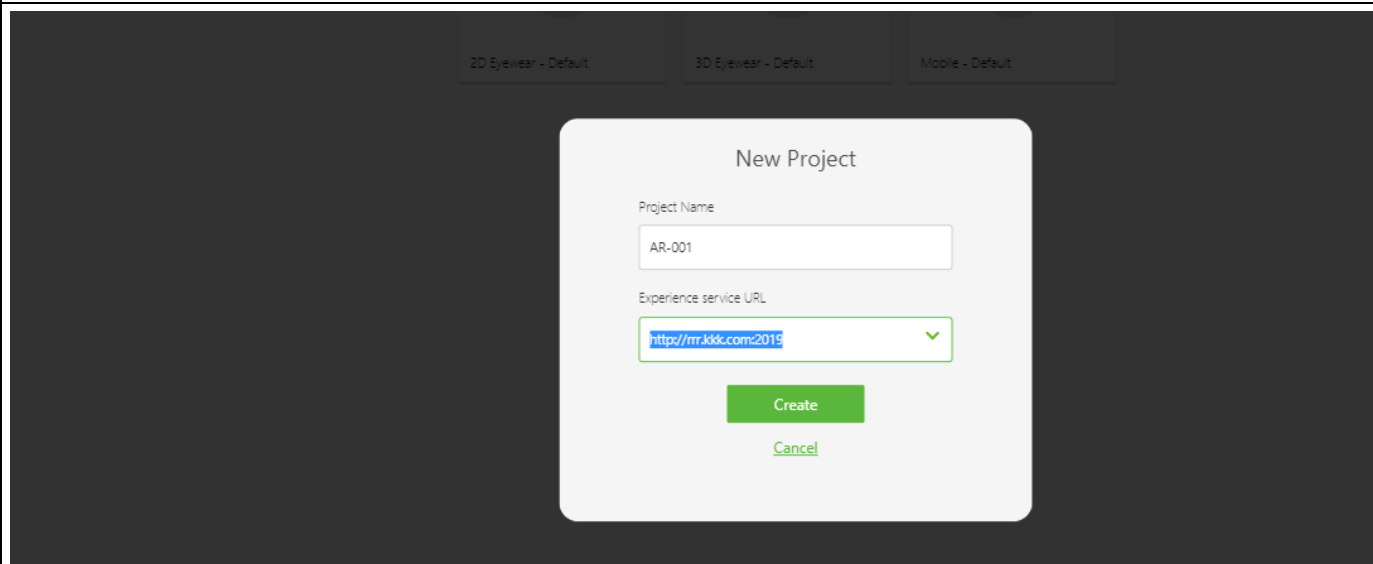
Вводим имя нашего нового проекта.

Например, **AR-001**

- **Experience service URL:**

В этом поле выбираем нужный нам сервер из списка доступных.

На данном этапе можно выбрать [несуществующий] сервер, например <http://rrr.kkk.com:2019>

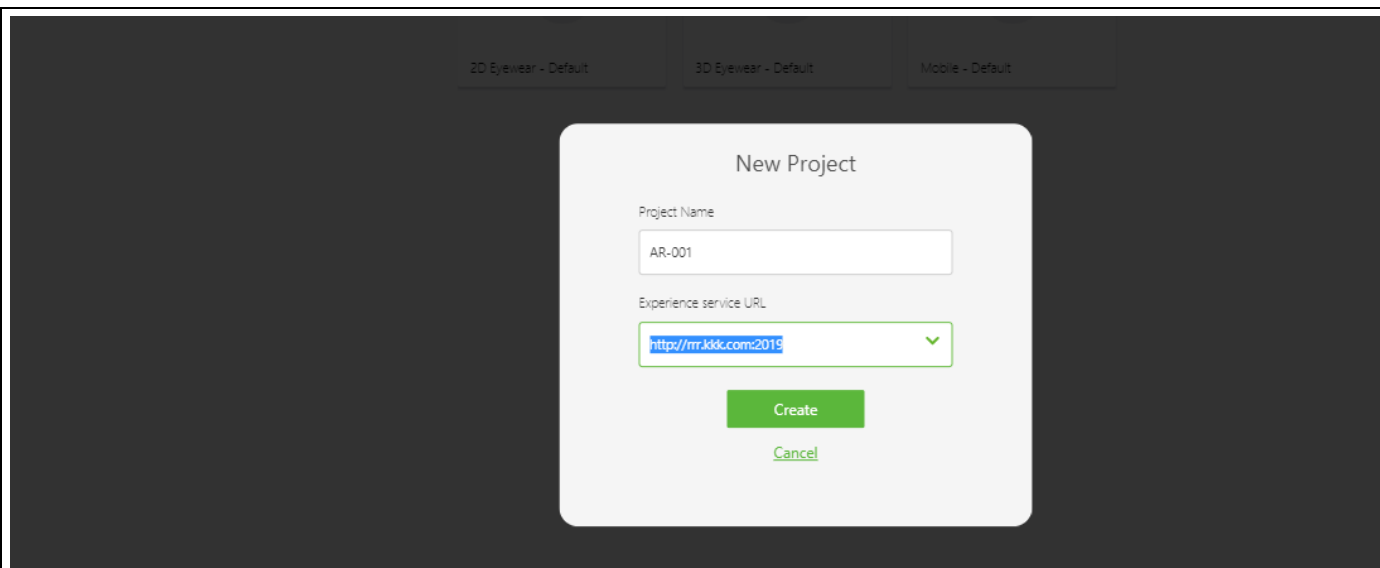


Шаг 6.

Нажимаем кнопку

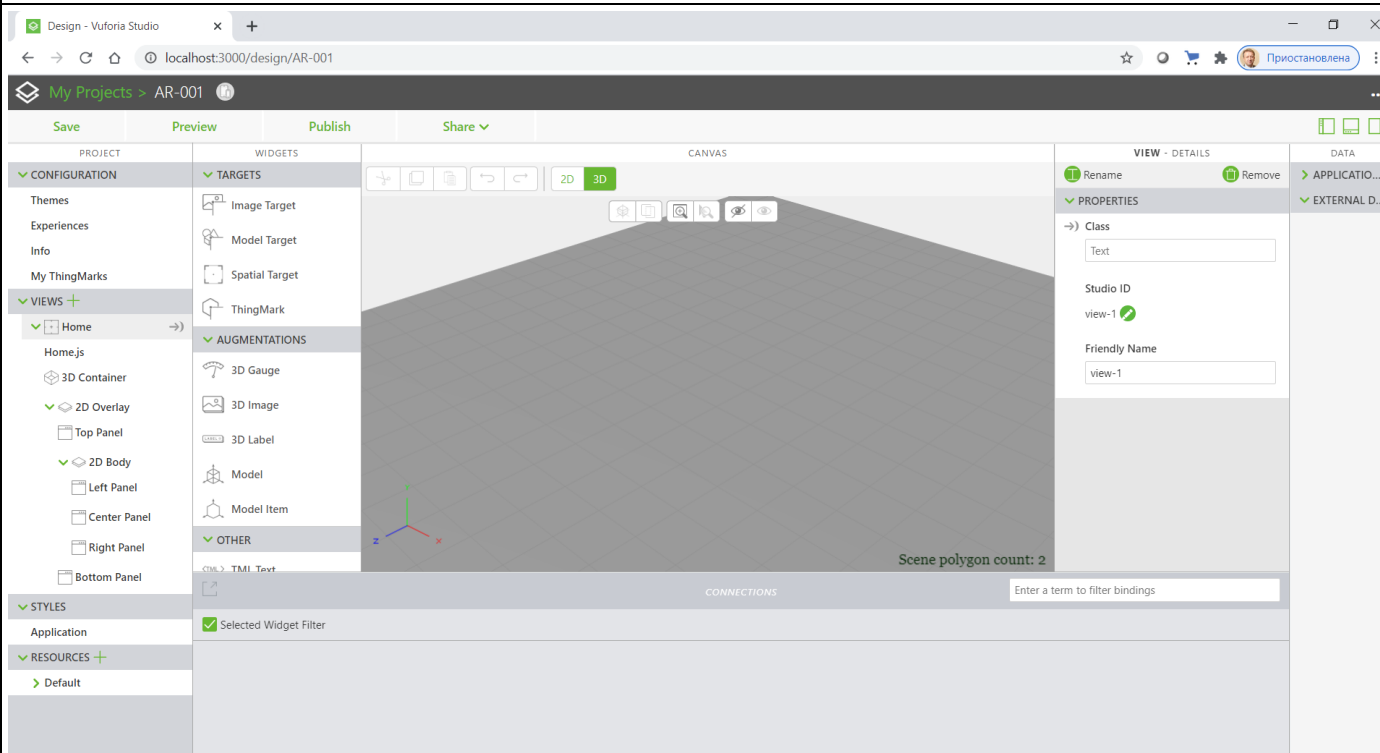
Create

И переходим к новому проекту



Шаг 7.

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



Шаг 11.

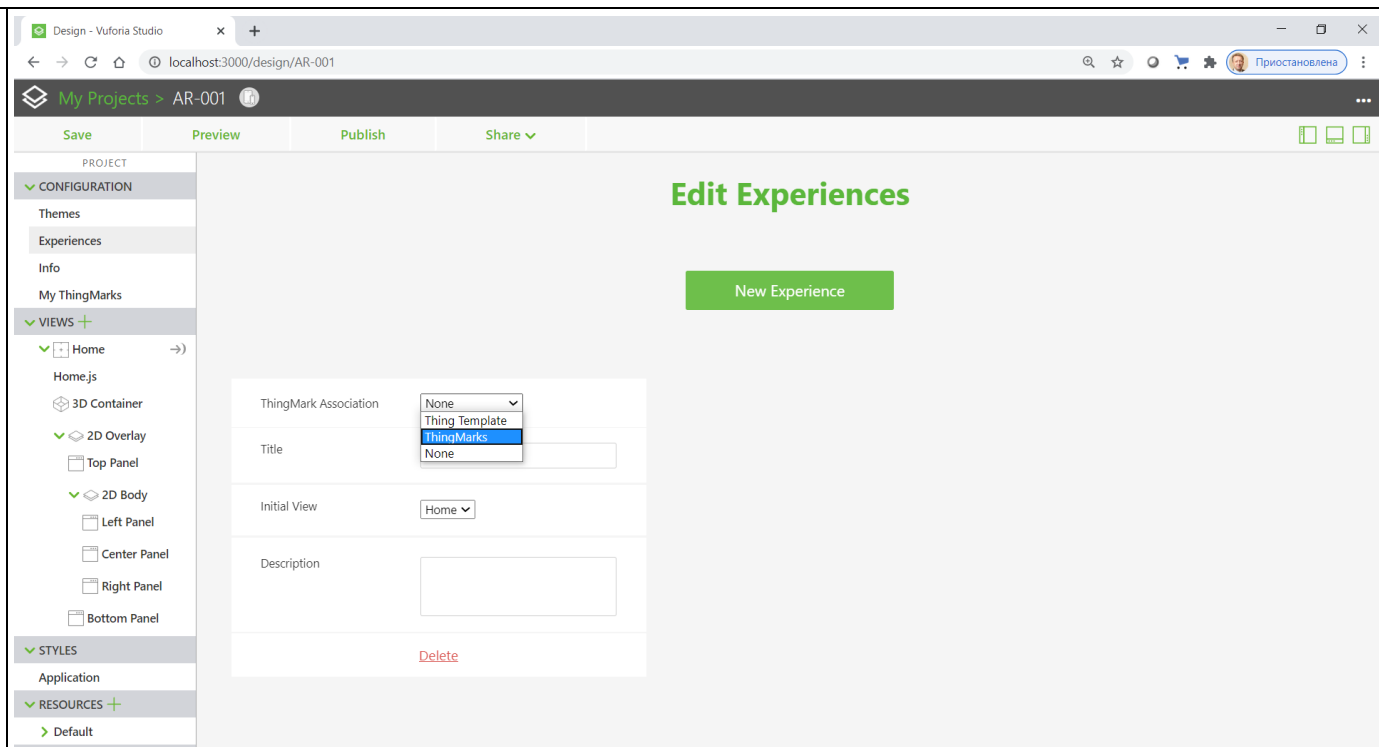
В зоне интерфейса «Конфигурация проекта» выберем пункт «Experience» и заполним необходимые поля.

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** →



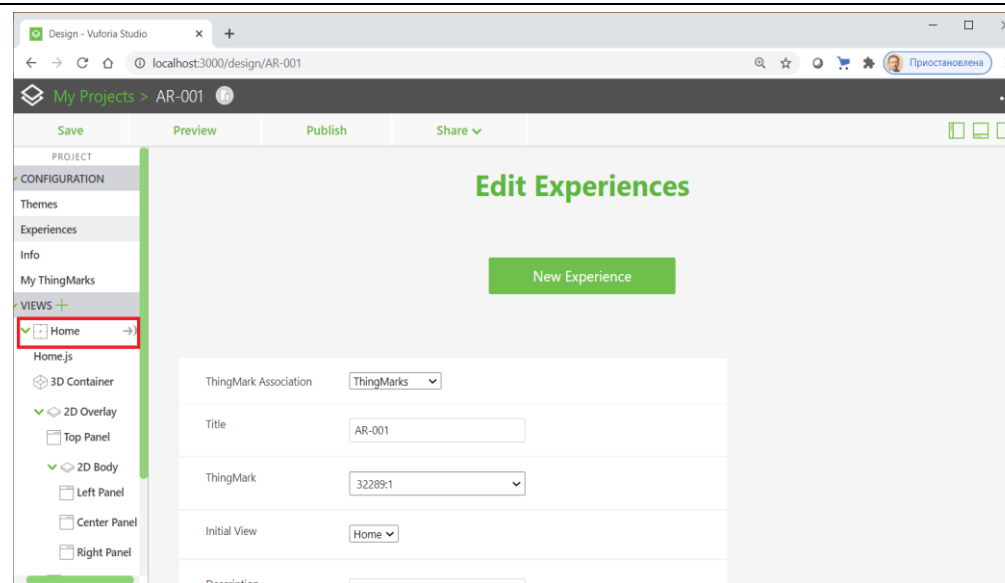
Шаг 12.

Определение нашего проекта завершено.

Переходим к разработке AR-приложения.

Для этого в зоне интерфейса

«Компоненты» нажимаем позицию **Home**



Шаг 13.

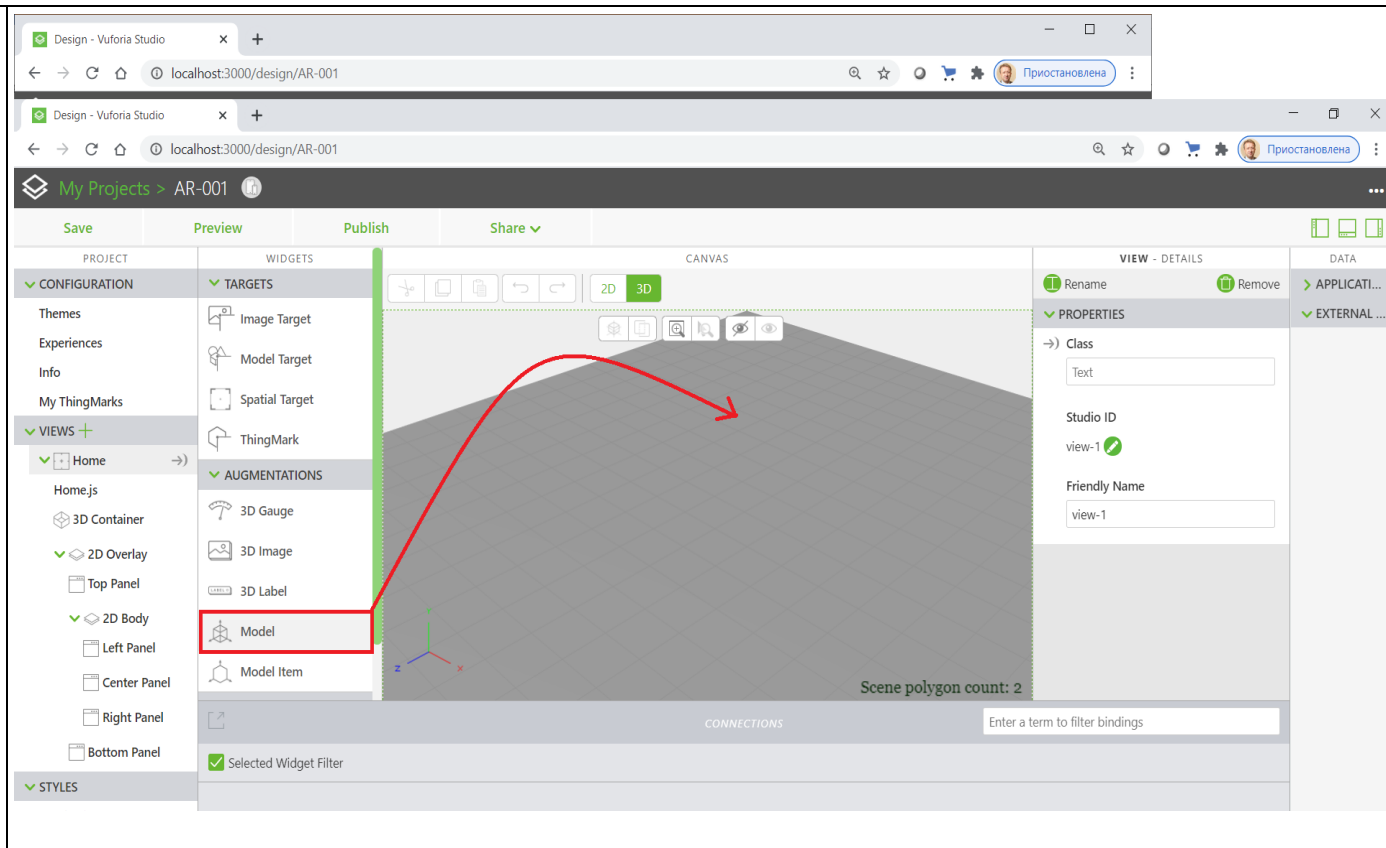
Мы находимся в режиме работы с **3D-объектами – 3D-канва**.

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**:

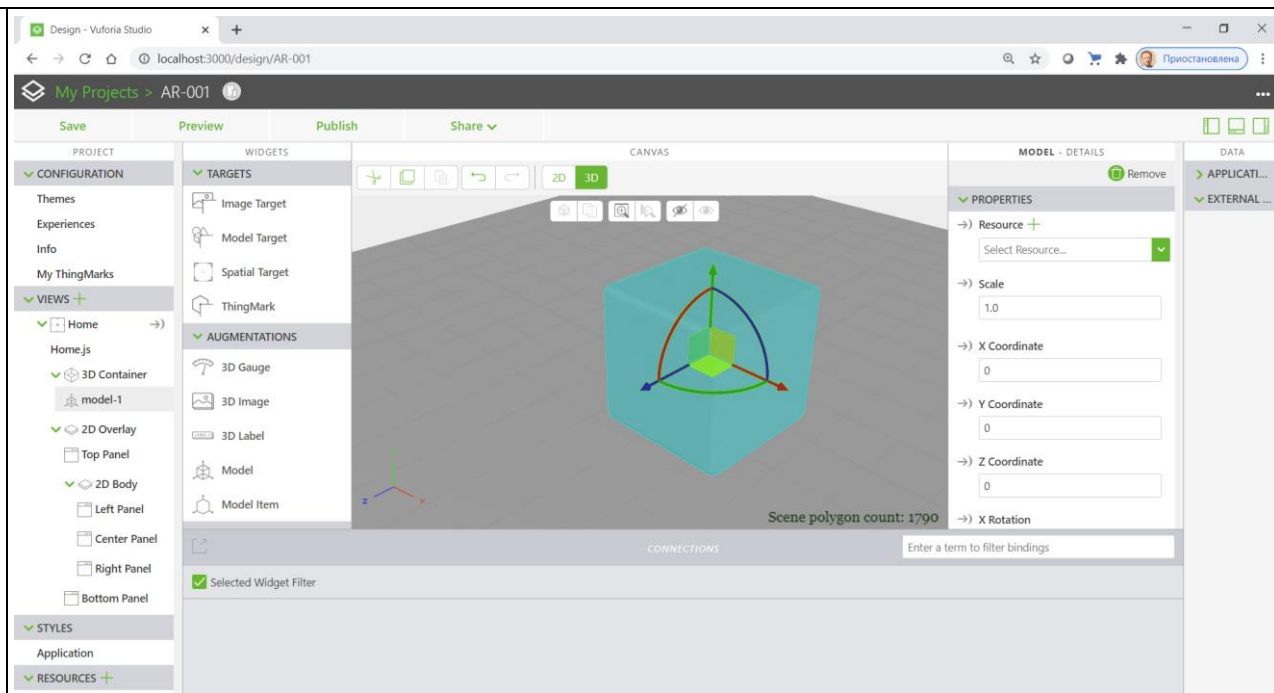
В зоне интерфейса «**виджеты**» находим виджет **Model** и перетаскиваем его «**drag-n-drop**» в зону интерфейса «**канва**».

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



Шаг 14.

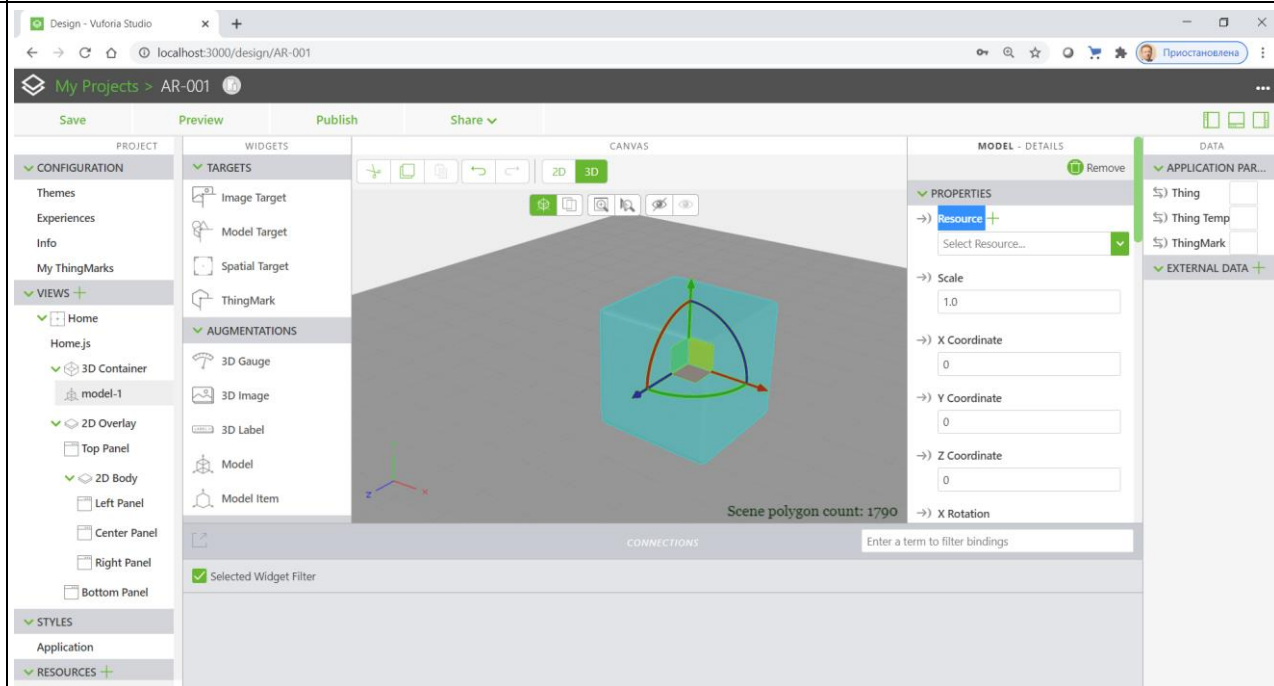
На **3D-канве** разрабатываемого приложения помещён контейнер для виртуального объекта нашего проекта



Шаг 15.

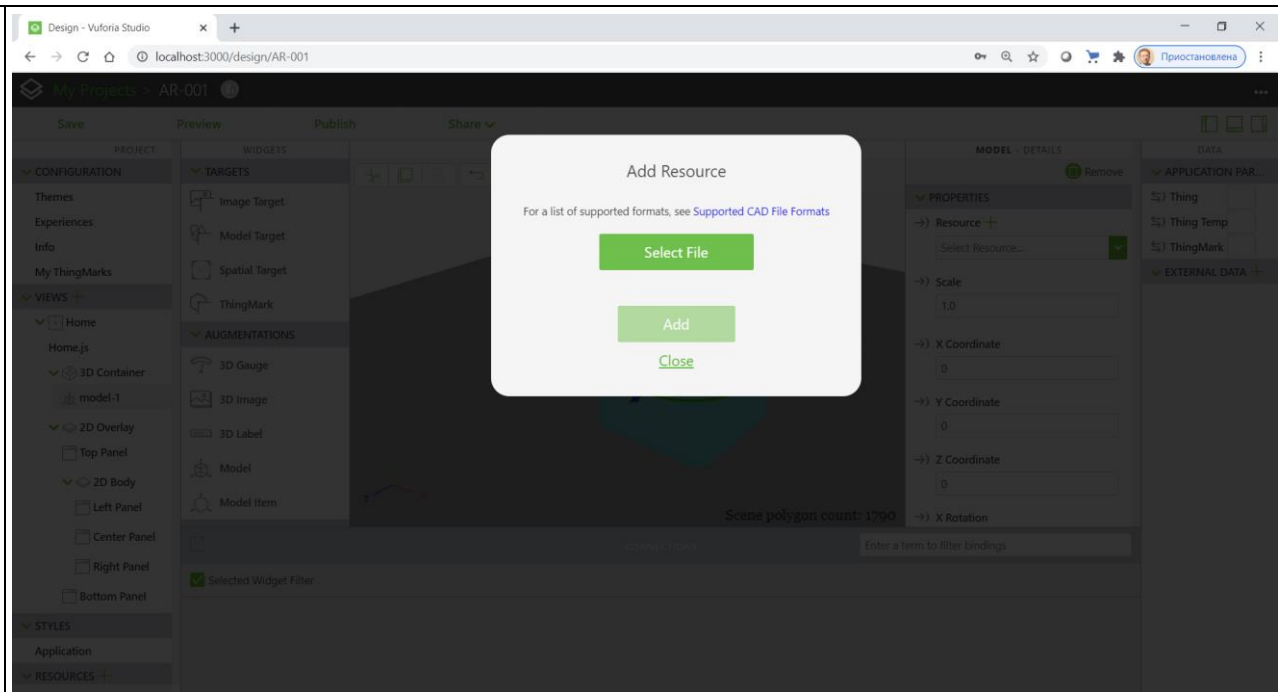
В зоне интерфейса **«Свойства компонента»** выбираем доступную **3D модель** виртуального объекта, а если такового нет, то ищем его в доступной файловой системе.

Для этого нажимаем на символ **«+»** справа от **Resource** в зоне **«Свойства компонента»**.



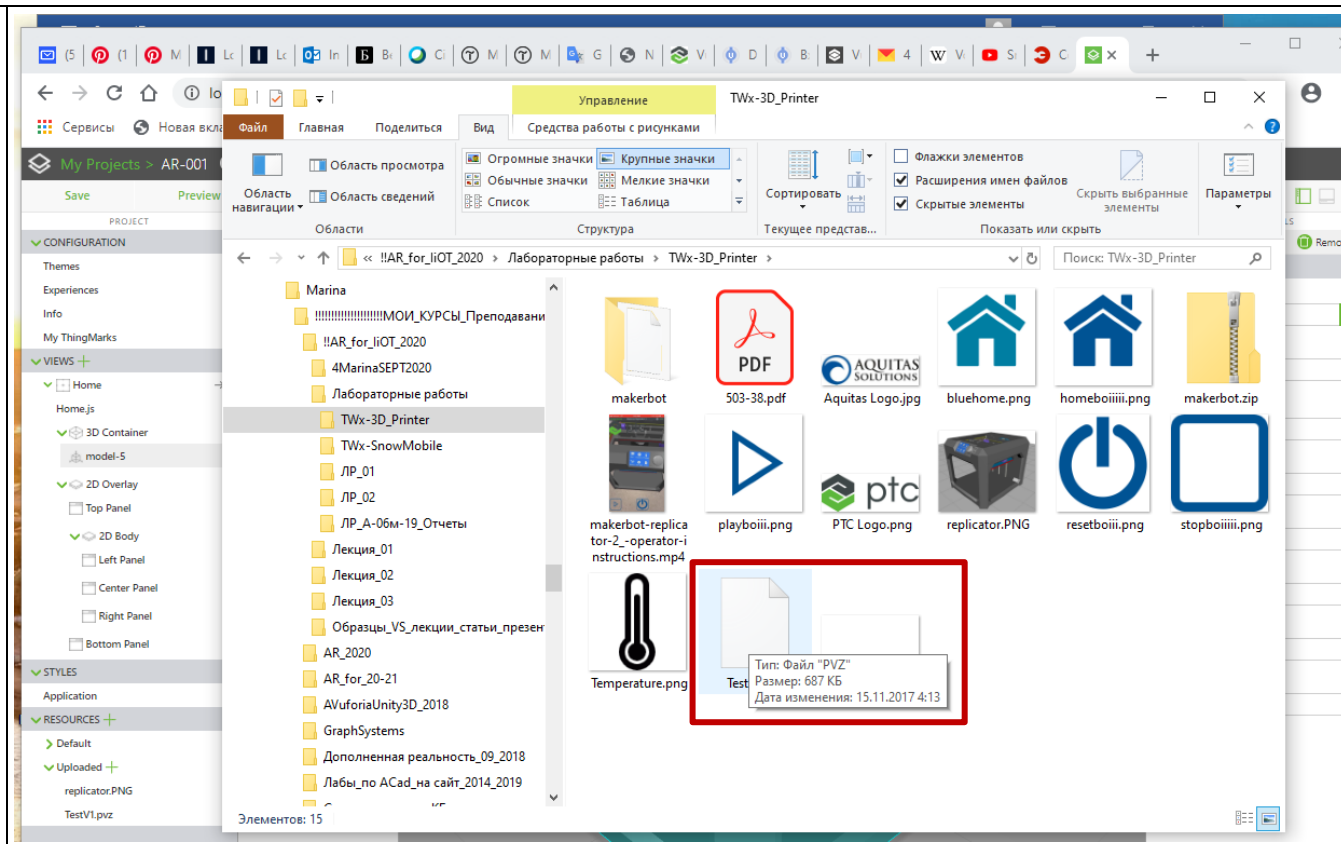
Шаг 16.

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**



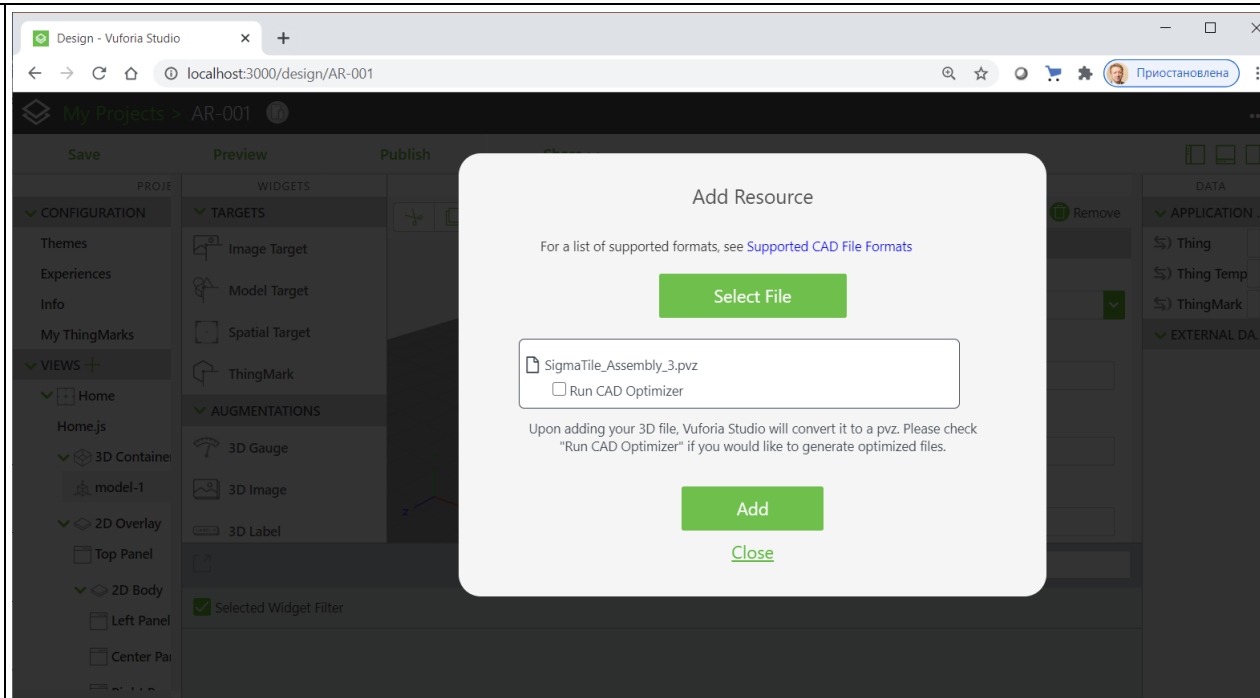
Шаг 17.

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»



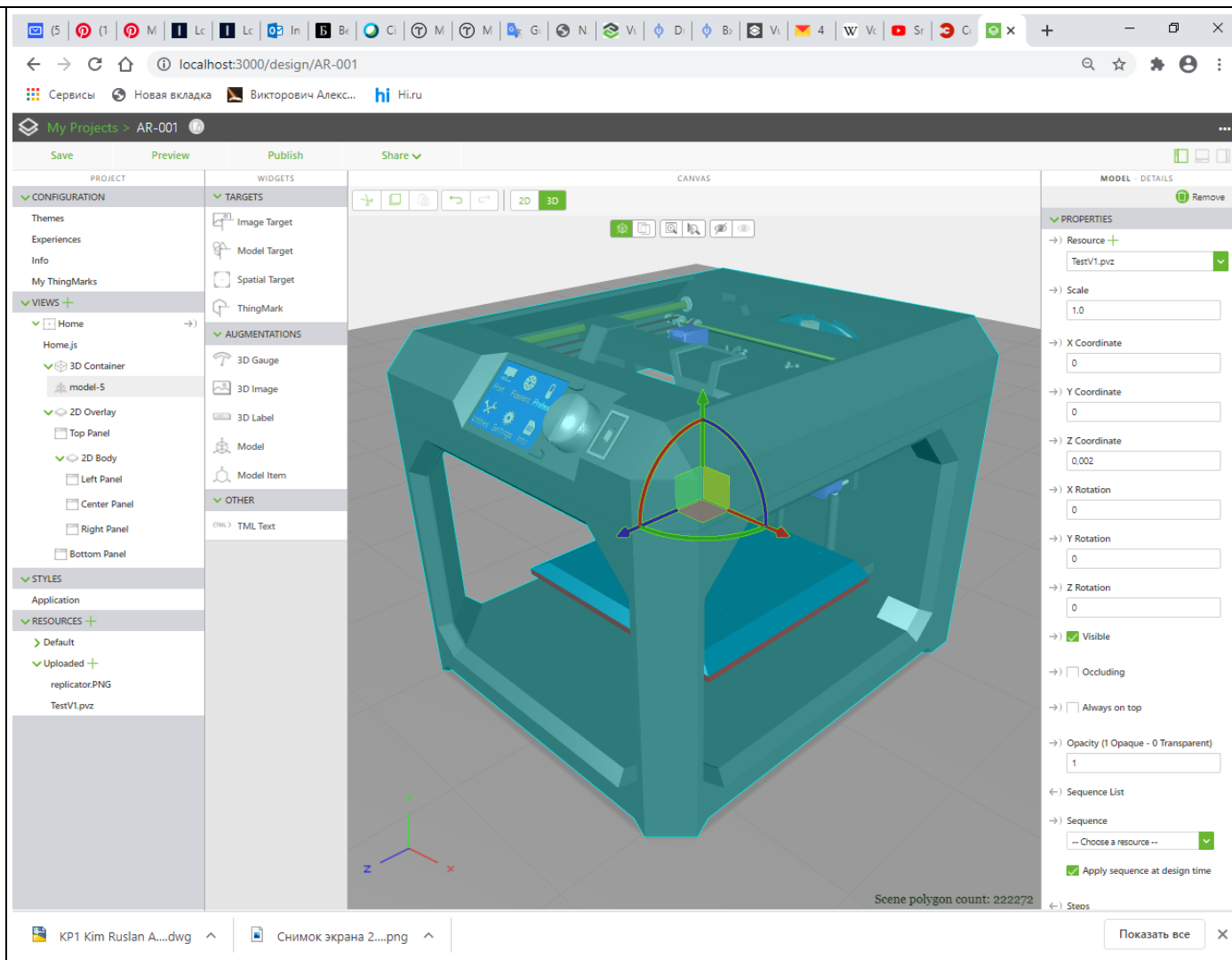
Шаг 19.

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.

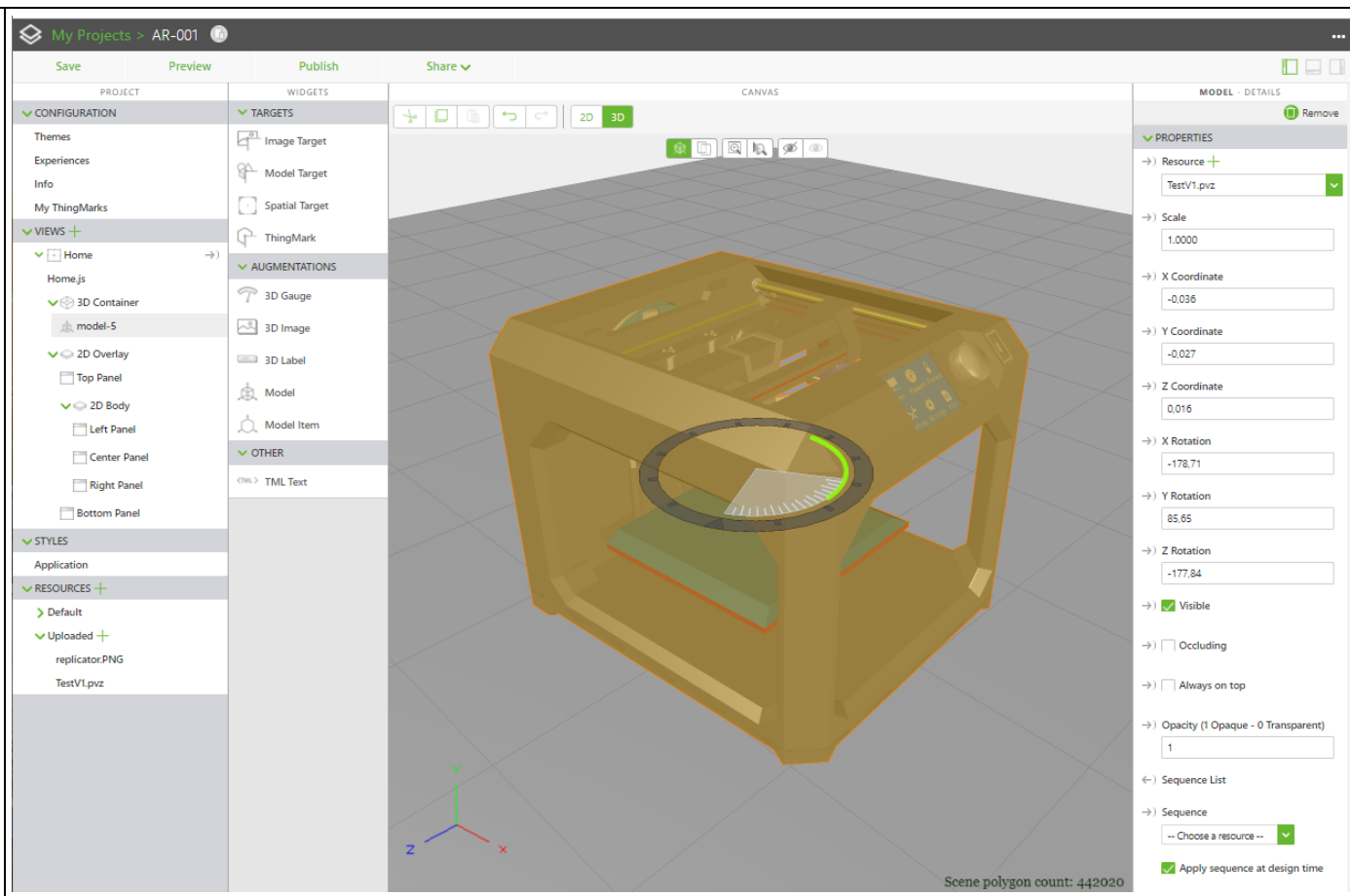
Обратите внимание и зафиксируйте (на бумажке) исходные координаты и углы разворотов (6 степеней свободы) – зона **«Свойства компонентов»**



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.



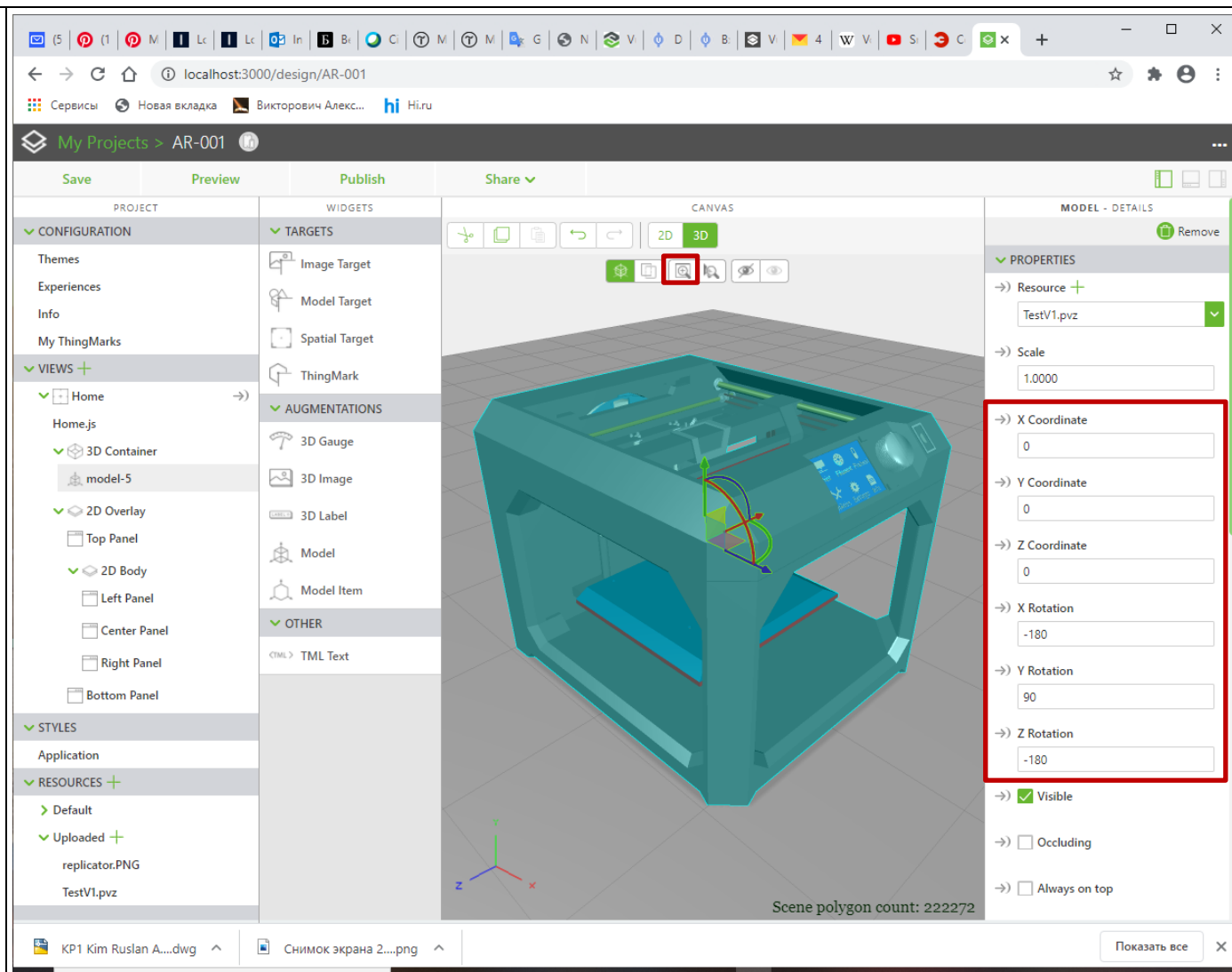
Шаг 21.

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

Как показано на рисунке, выбран 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.

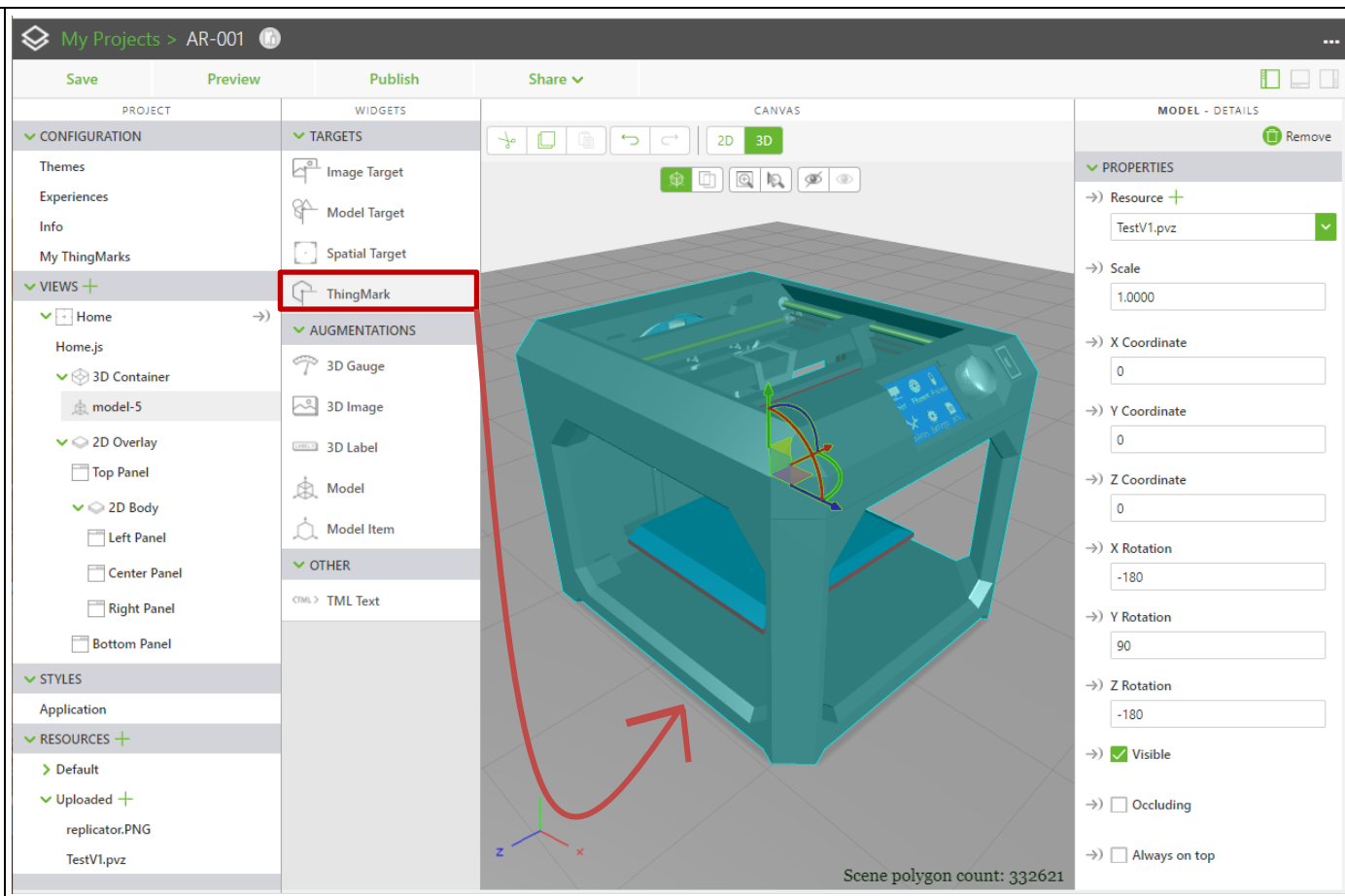
Результат – наша модель располагается горизонтально относительно канвы, Панель управления смотрит «направо» - в положительном направлении оси **X** пространства (глобальной системы координат).



Шаг 22.

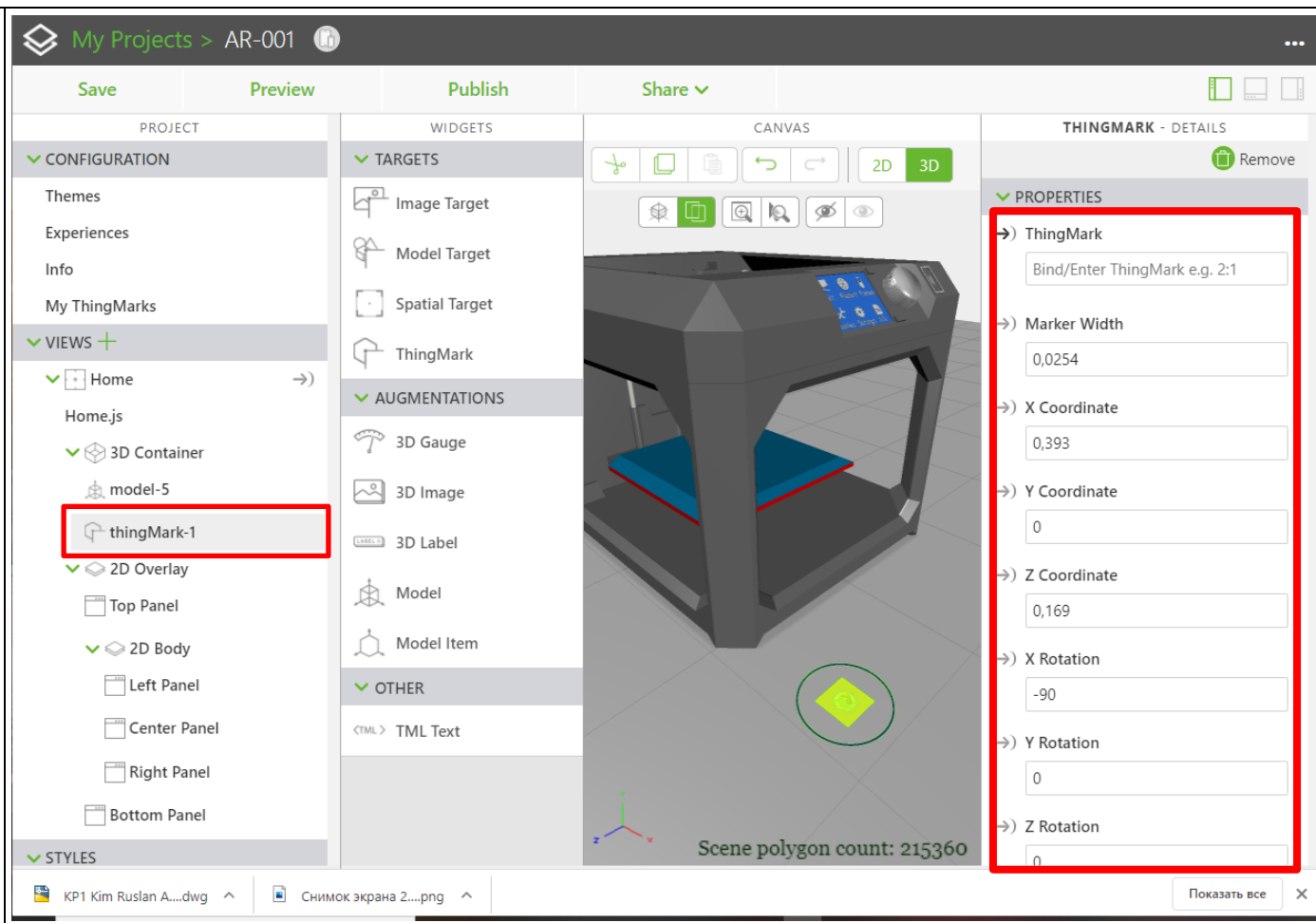
Теперь необходимо сделать топологическую привязку для правильной визуализации нашей виртуальной модели в реальном окружении. Это действие будем выполнять в соответствии с выбранным типом таргетинга – с помощью **ThingMark'a**.

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



Шаг 23.

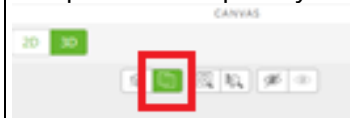
Результат – на канве появляется компонент приложения **ThingMark**, свойства которого – размер, координаты размещения – вы видите в зоне «**Свойства компонента**».



Шаг 24.

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

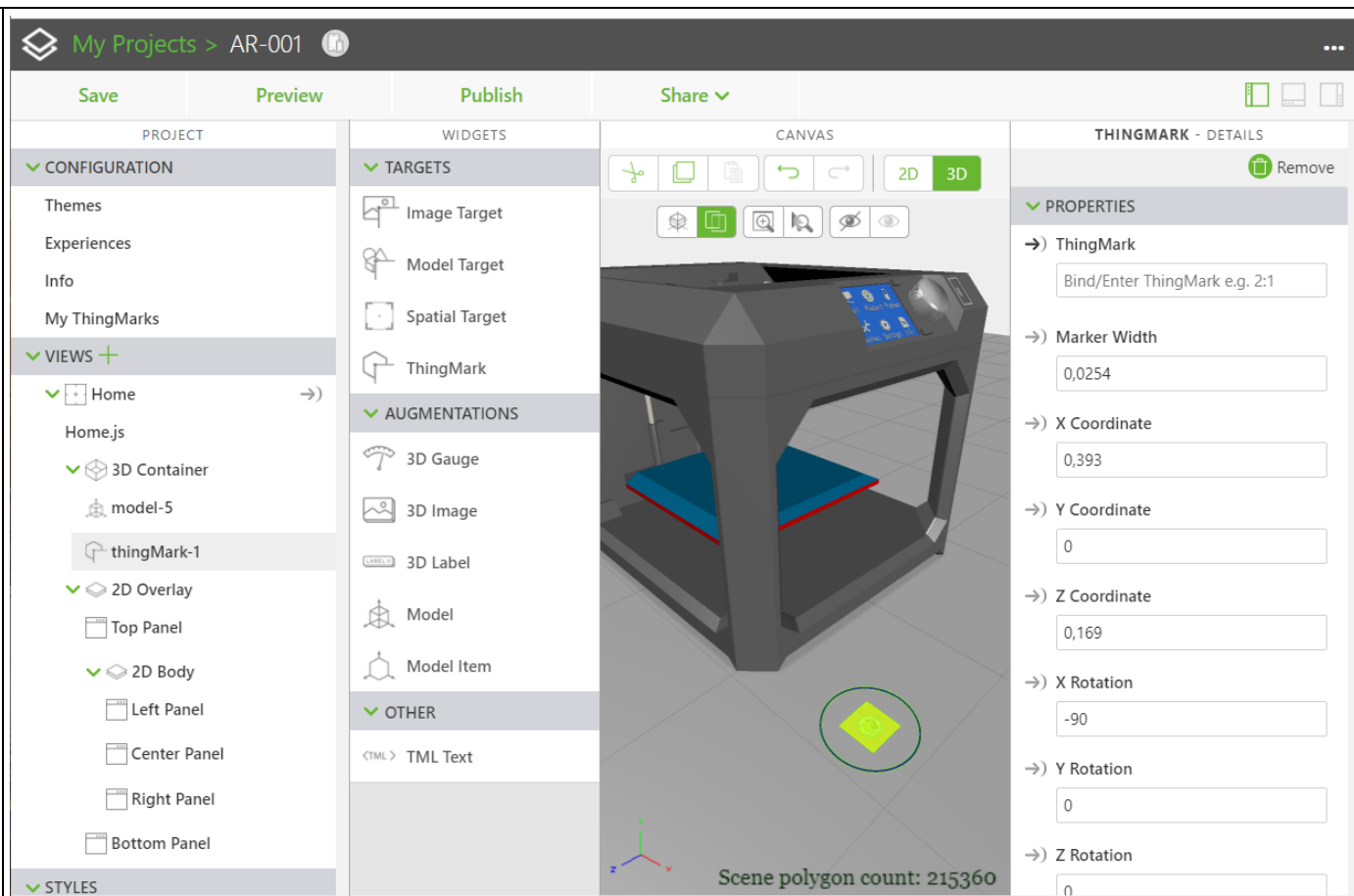
Можно это делать с помощью перемещения-проецирования по поверхностям модели – для этого надо выбрать пиктограмму «**Mate**» на канве



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**.

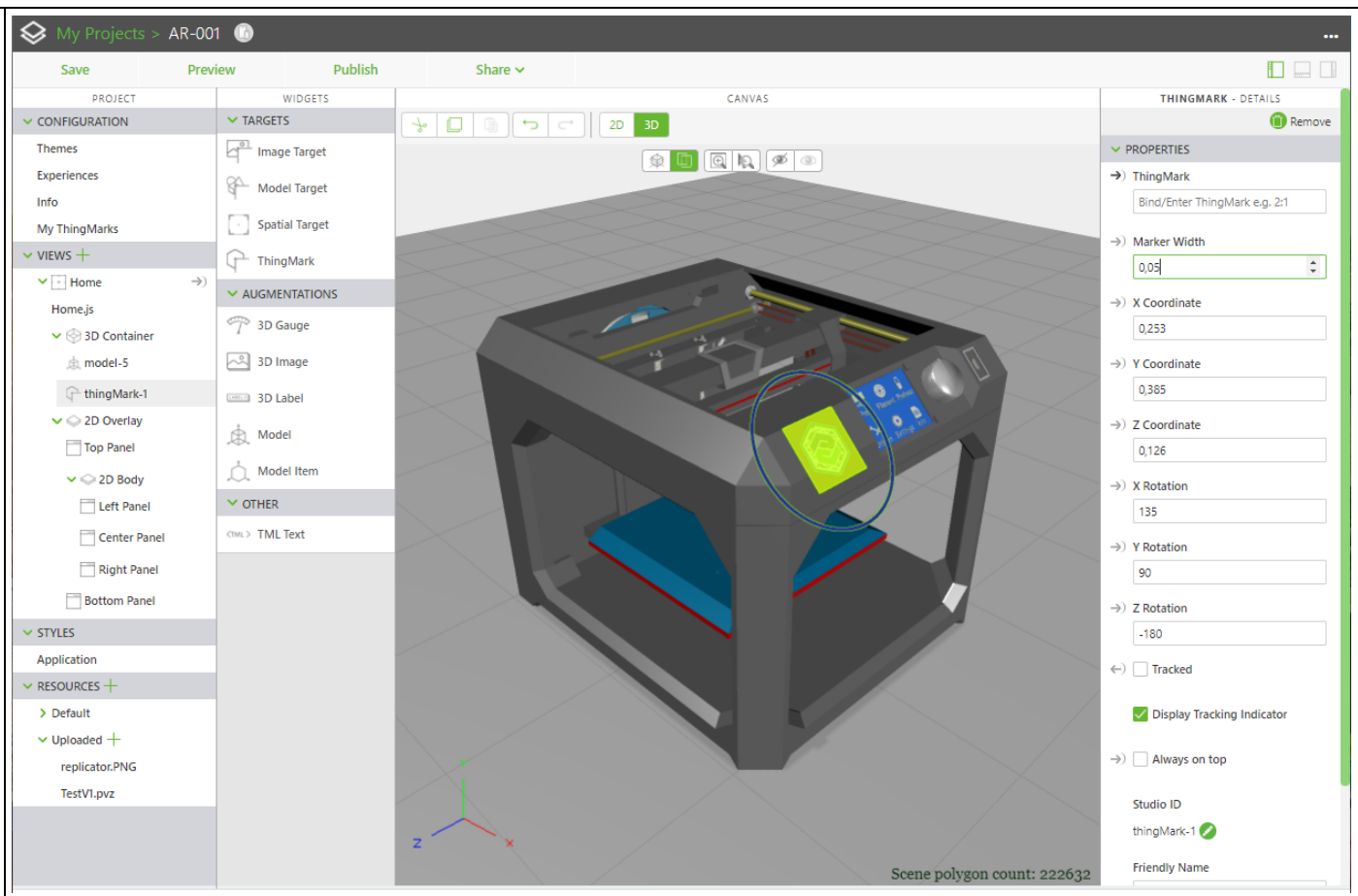
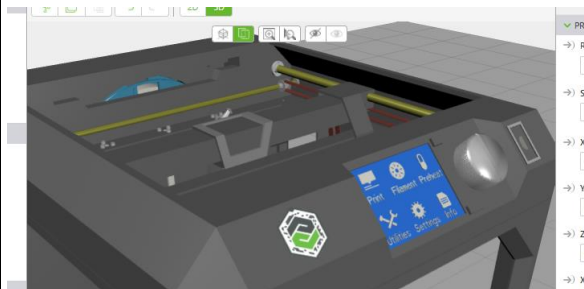
Или можно воспользоваться полями **X, Y, Z, Xrotation, Yrotation, Zrotation** для **ThingMark-1** в зоне интерфейса «Свойства компонента».

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



Шаг 25.

В результате для нашего приложения получаем такое расположение виртуальной марки/таргета **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.

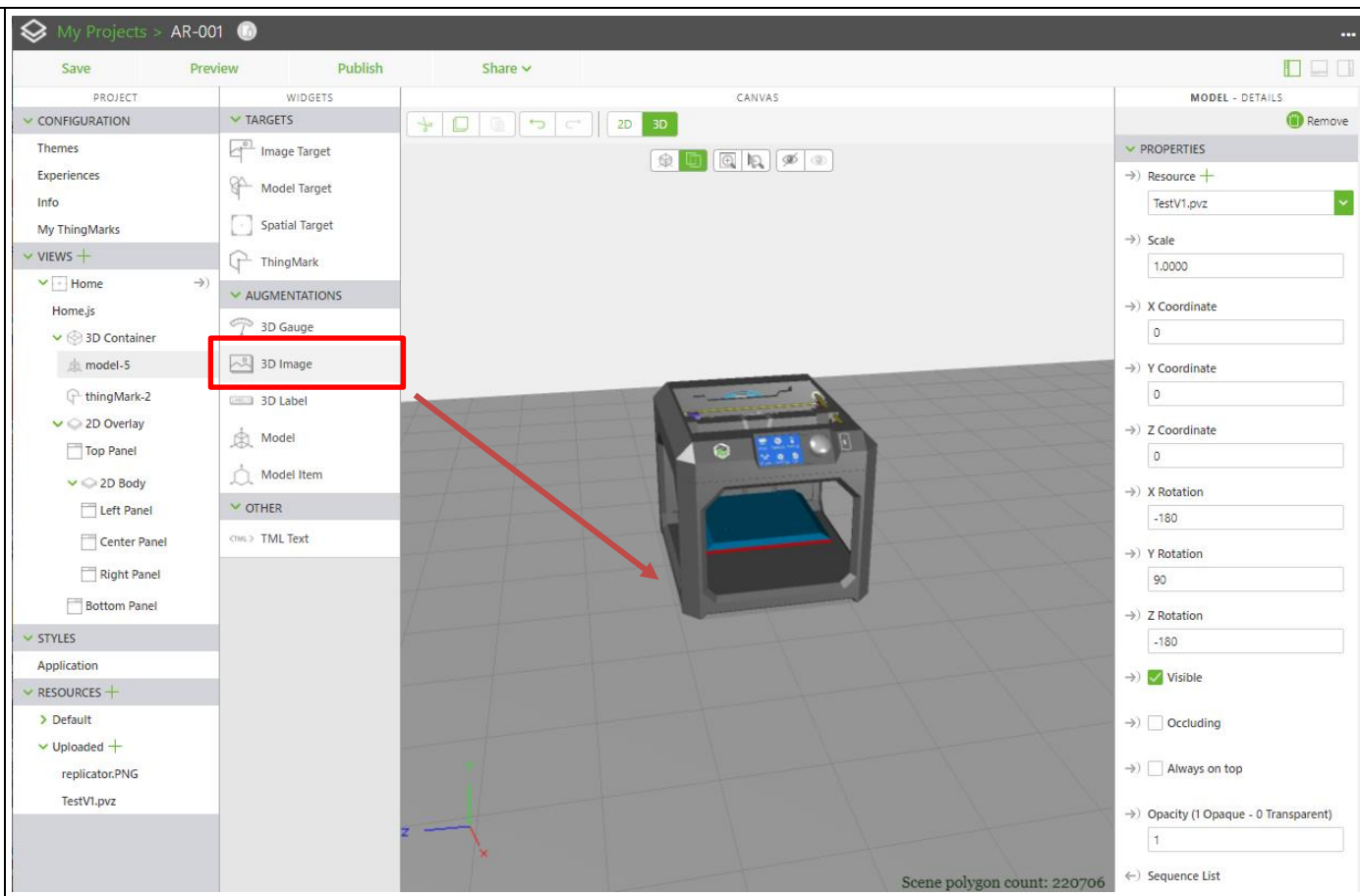


Шаг 26.

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).

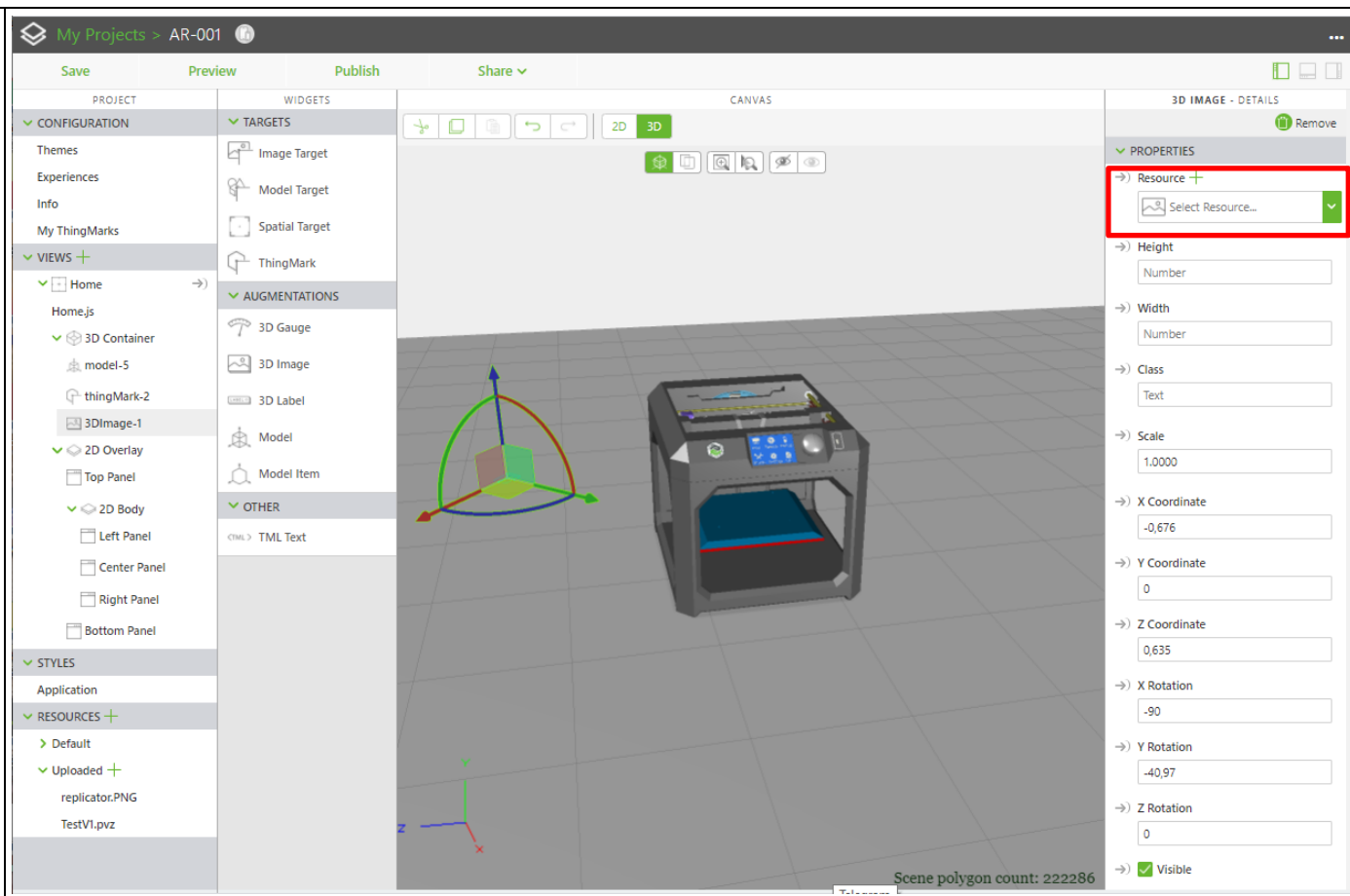


Шаг 27.

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**»

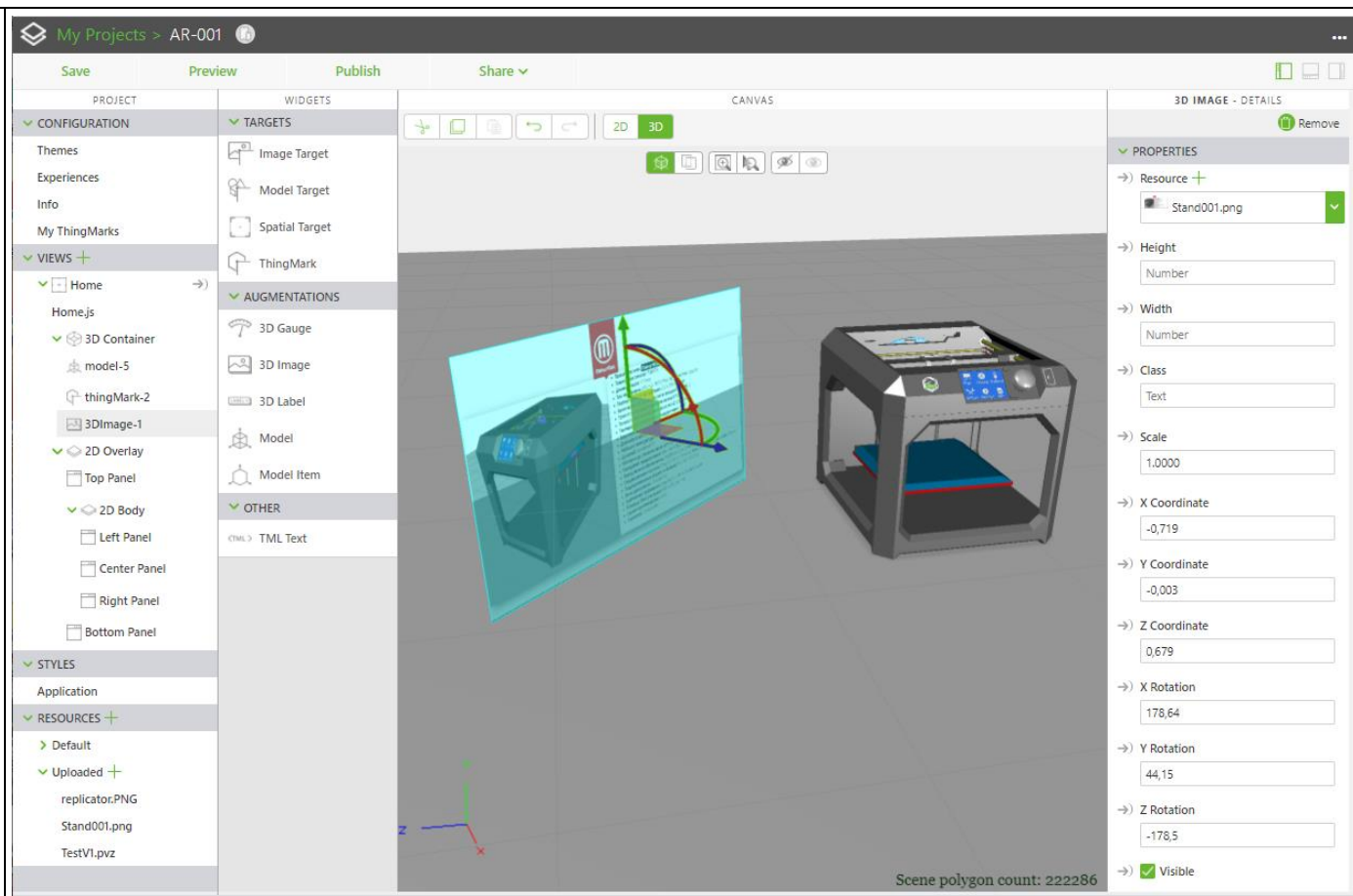
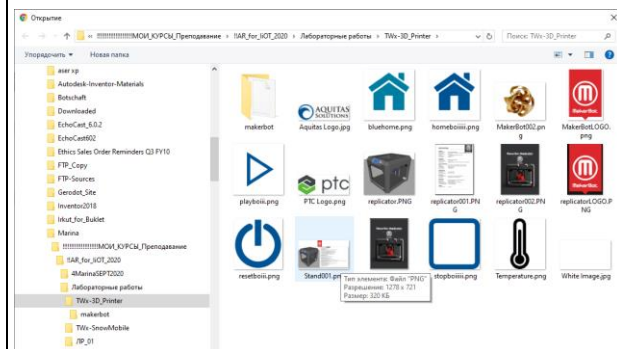
Если нужного изображения в составе галереи **Vuforia Studio** нет, то надо искать требуемое в ЛФС. Для этого нажимаем символ «+» справа от поля **Resource**



Шаг 28.

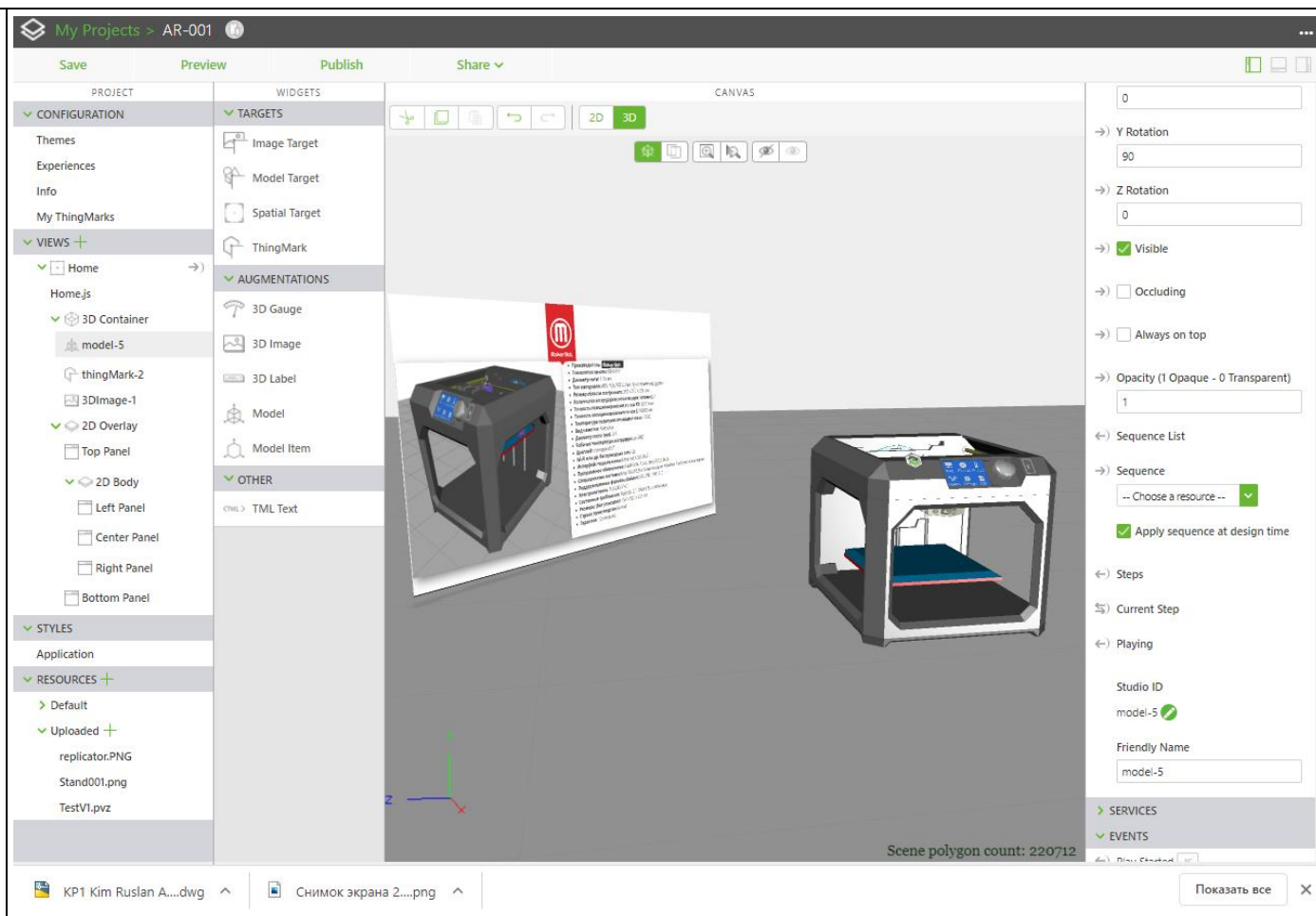
Средствами вызванного навигатора файловой системы находим необходимое изображение.

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).

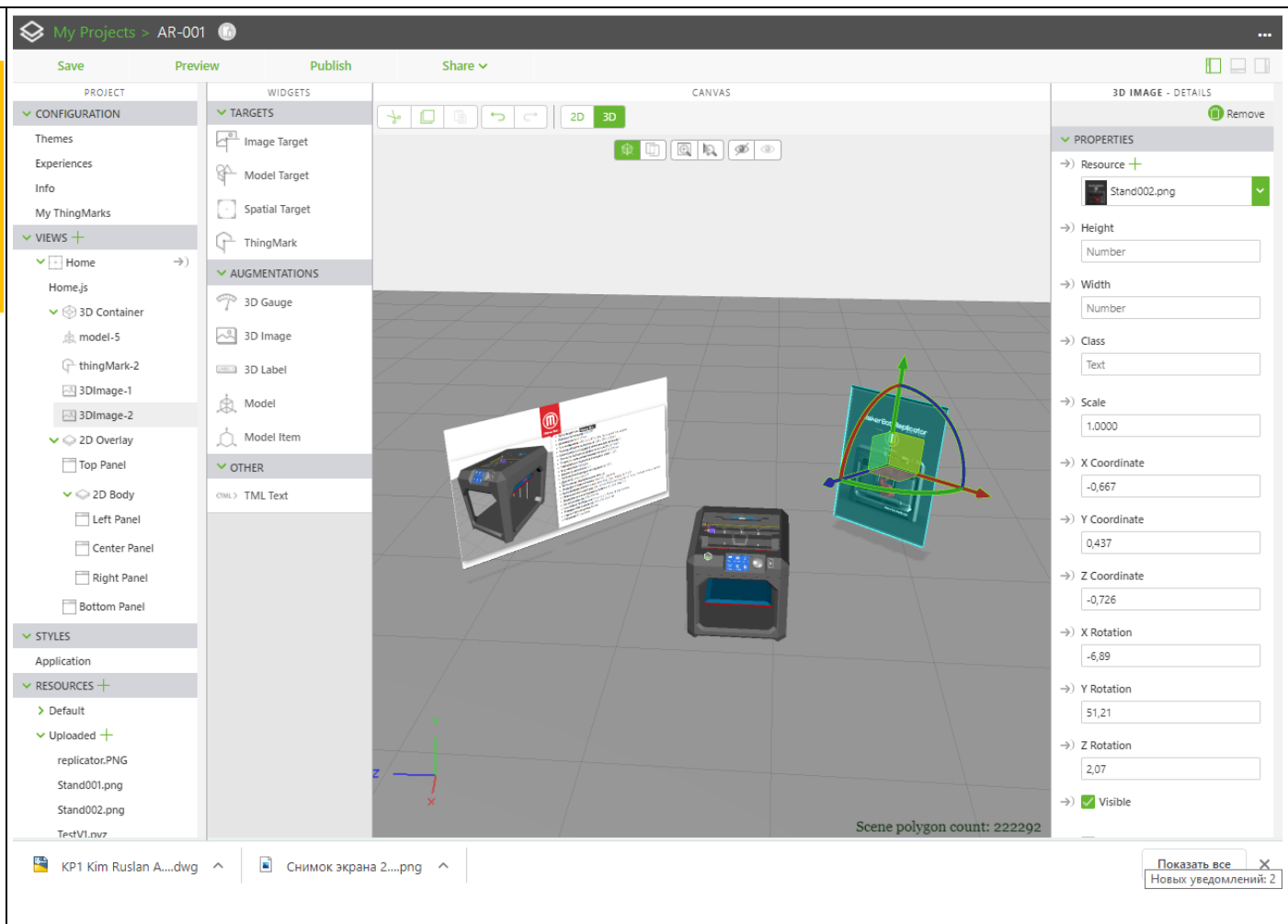


Шаг 30.

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**.

МОЖЕТЕ ОБОЙТИСЬ ОДНИМ ЭКРАНОМ!

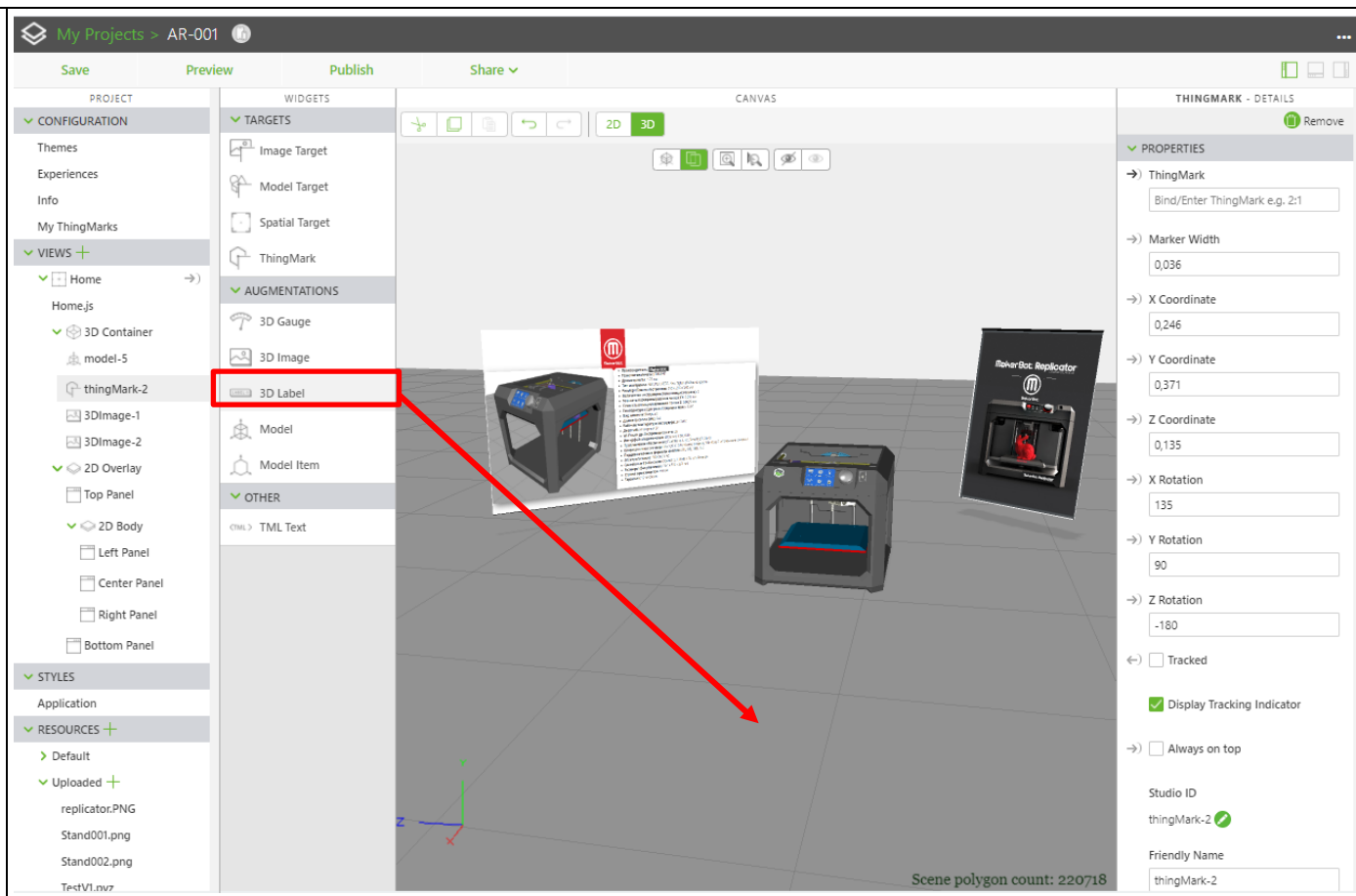
При размещении этих виджетов (в сцене это стенды, позиционируемые вокруг объекта - модели принтера), пользуйтесь инструментами разворота сцены (**левая, правая, средняя клавиша мыши → RMB, LMB, «колесико»**), для уточнения позиции каждого элемента активизируйте в зоне **«Компоненты»** в поле **VIEW** 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.

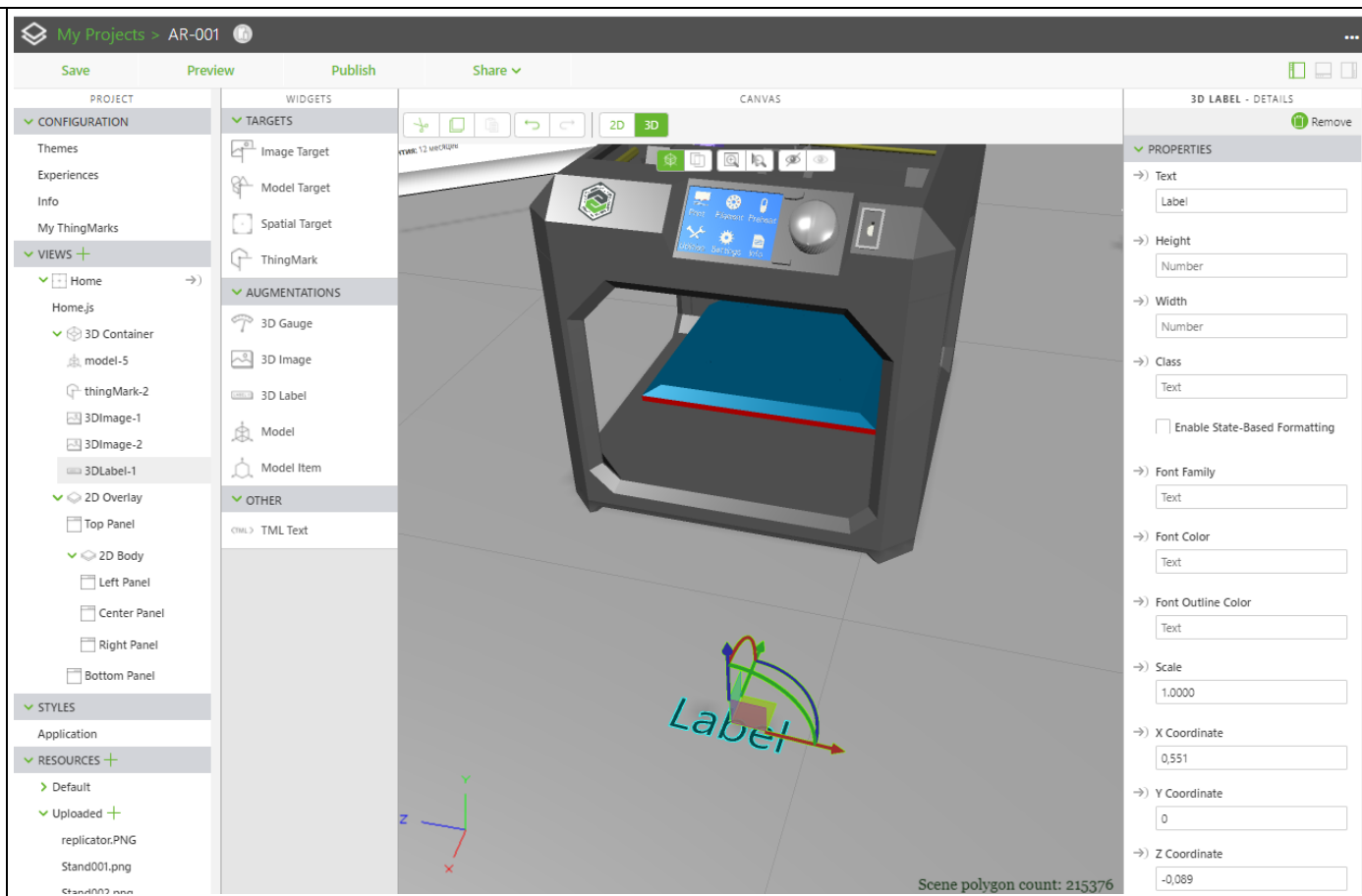
Разместим на **3D-канве** виджет **3D Label**.



Шаг 32.

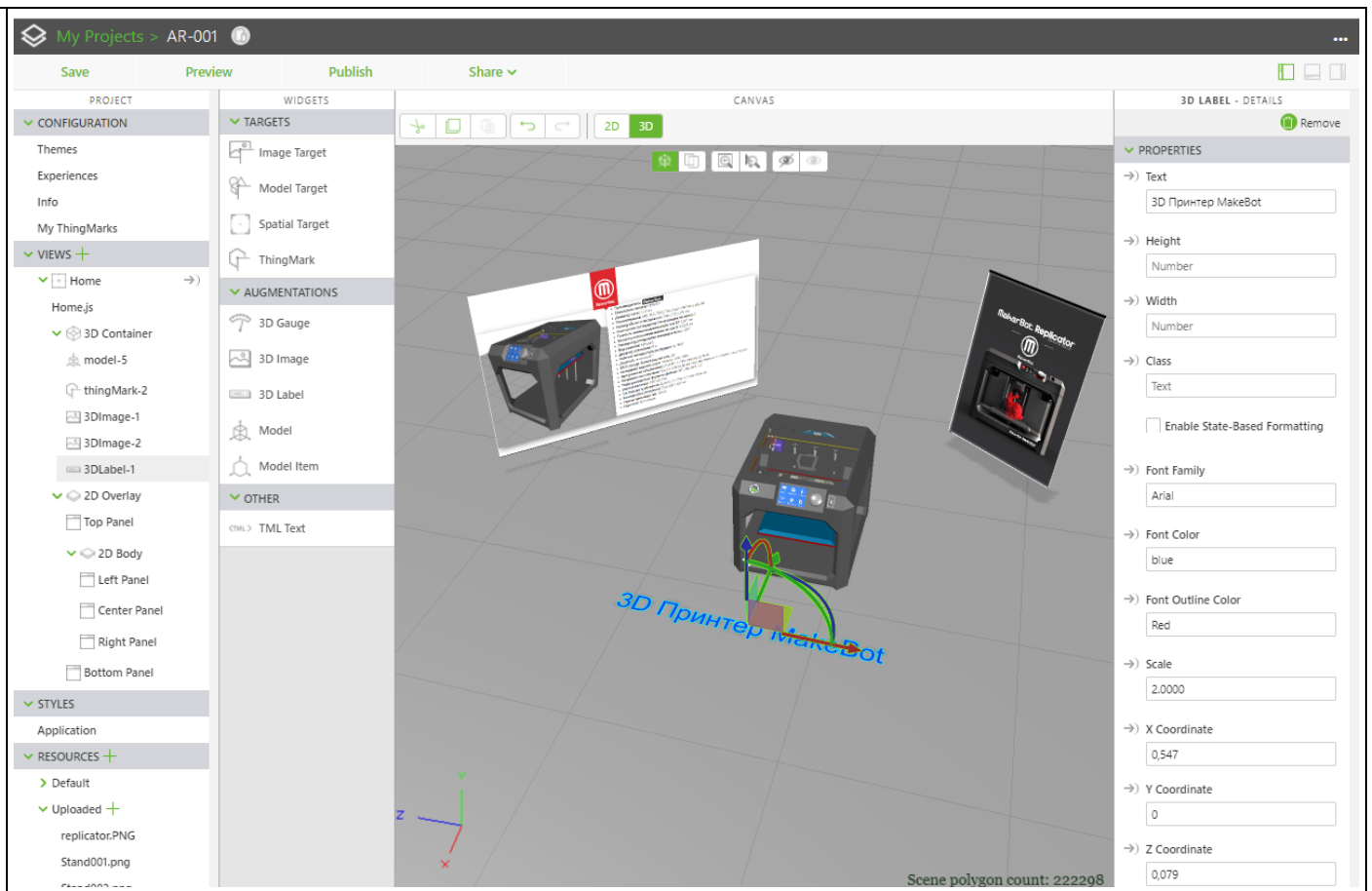
Выполним масштабирование и ориентацию его размещения аналогично тому, как это выполнялось при размещении модели и изображения (**шаги 20, 21, 30**).

Например, положим эту надпись на пол – «ковриком».



Шаг 33.

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



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, воспроизводиться 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

- CONFIGURATION
 - 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

WIDGETS

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

STYLES

- Application
- RESOURCES
 - Default
 - Uploaded
 - replicator.PNG
 - Stand001.png
 - Stand002.png

CANVAS

3D Image Target Bot

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

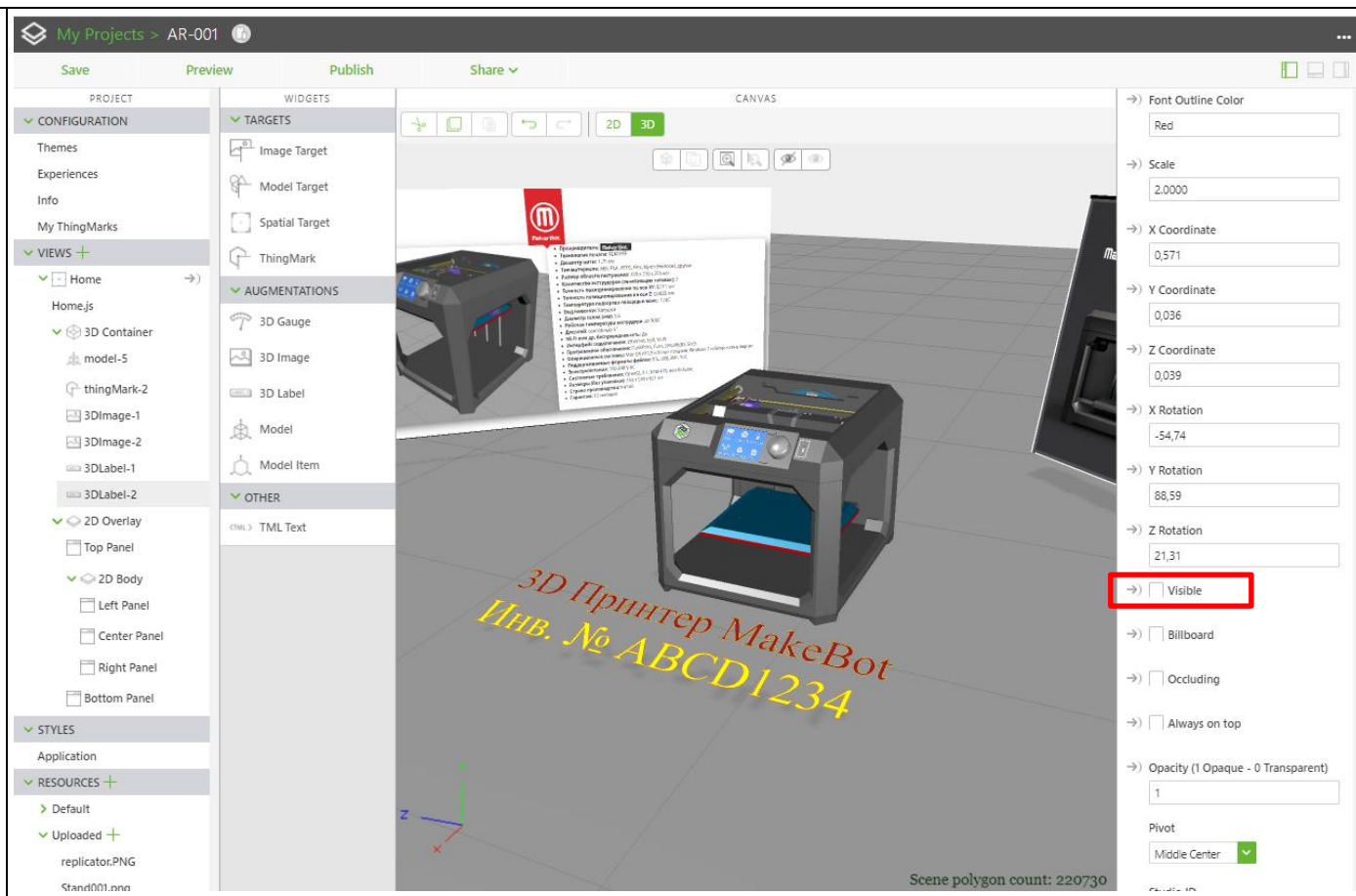
☐ Always on top

Шаг 35.

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

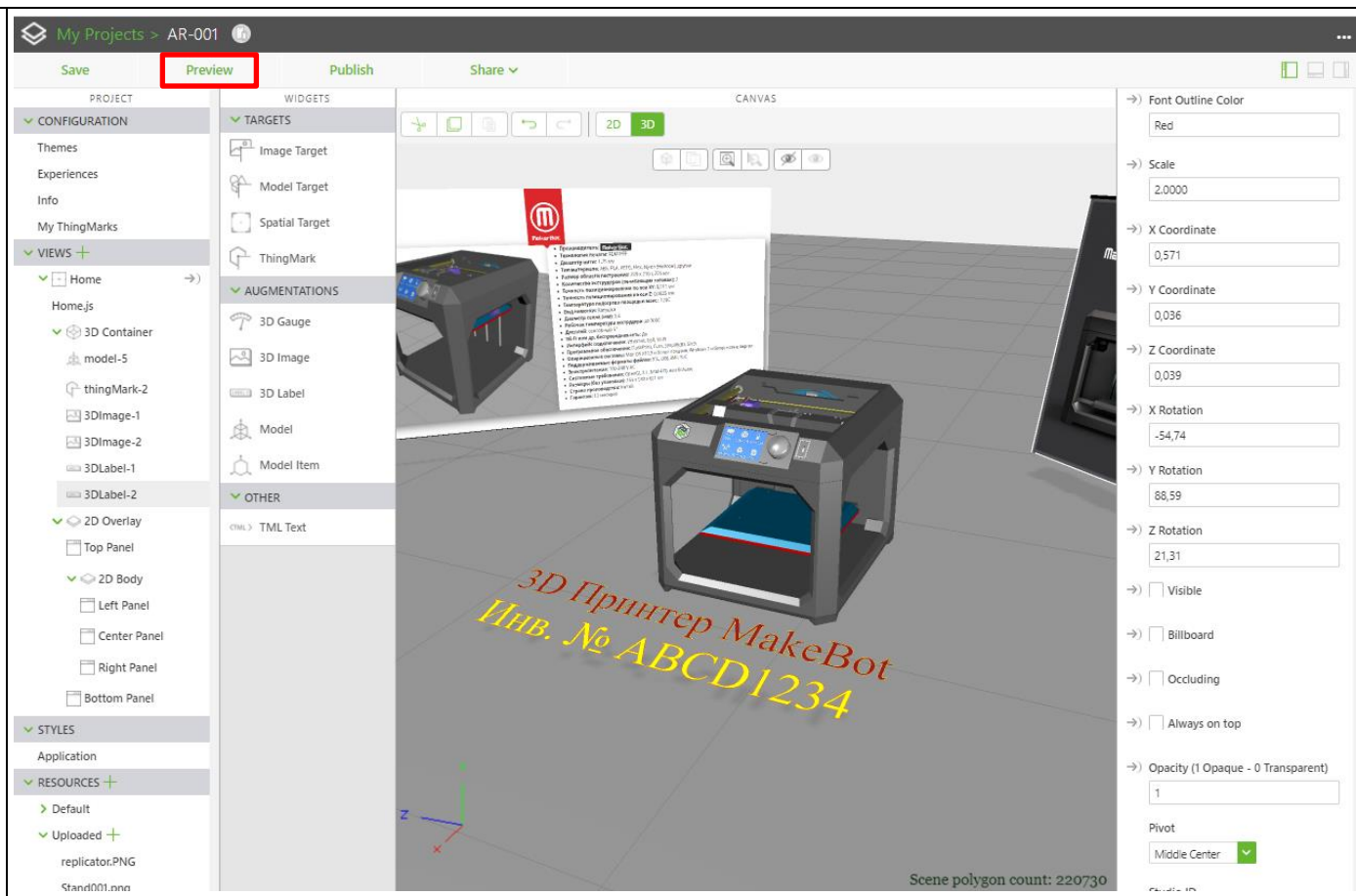
Предлагаемое размещение – на сцене.

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**.



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»

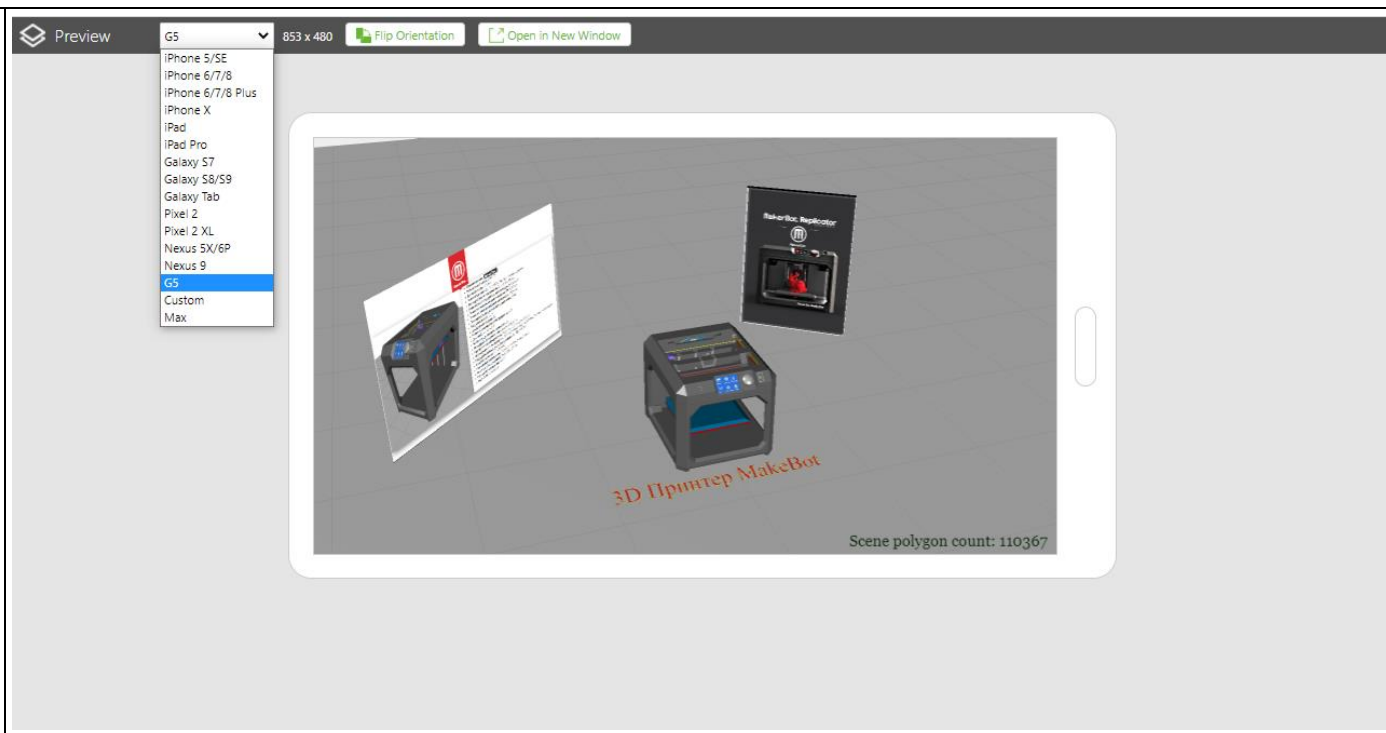


Шаг 37.

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)



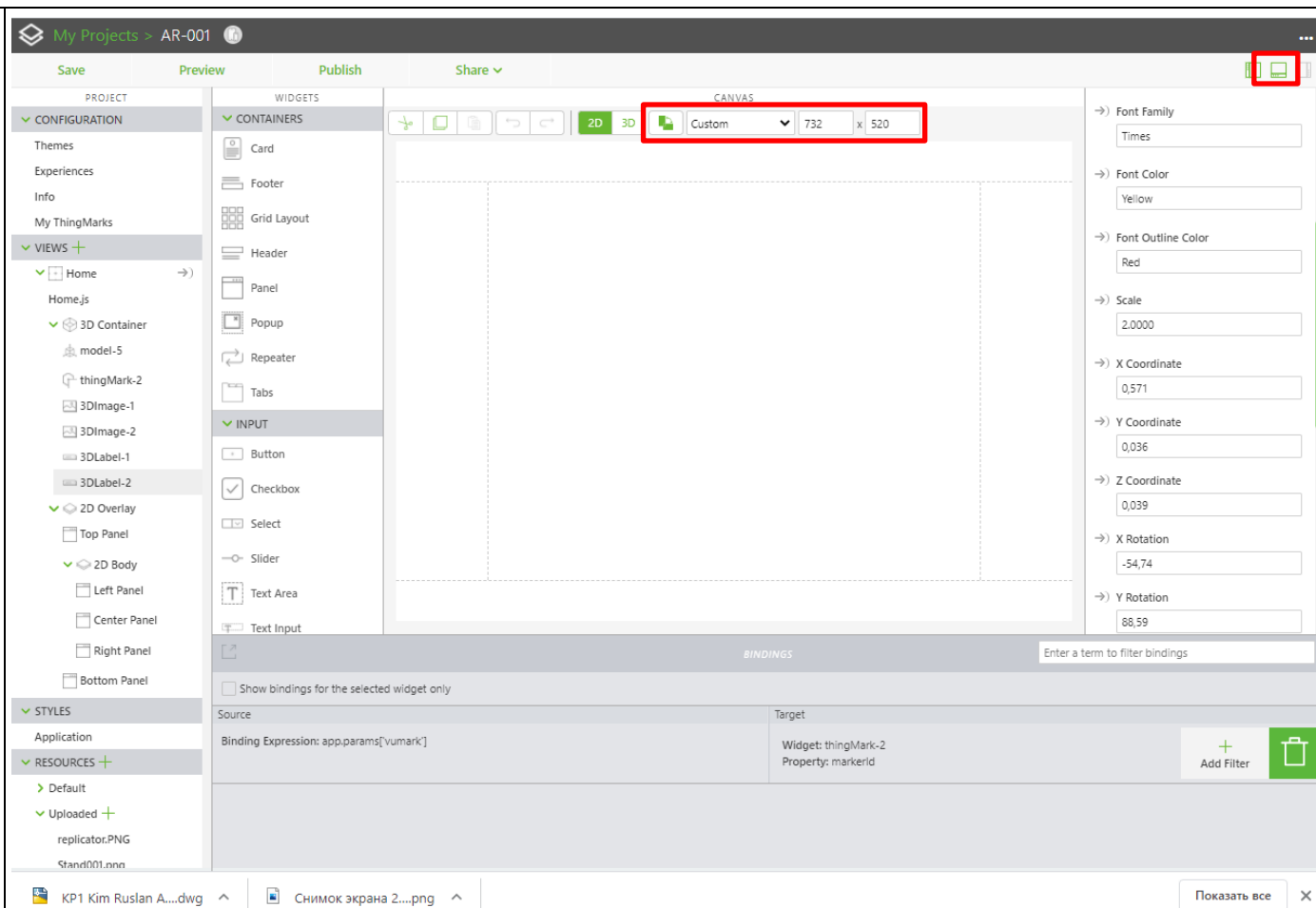
Шаг 38.

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.

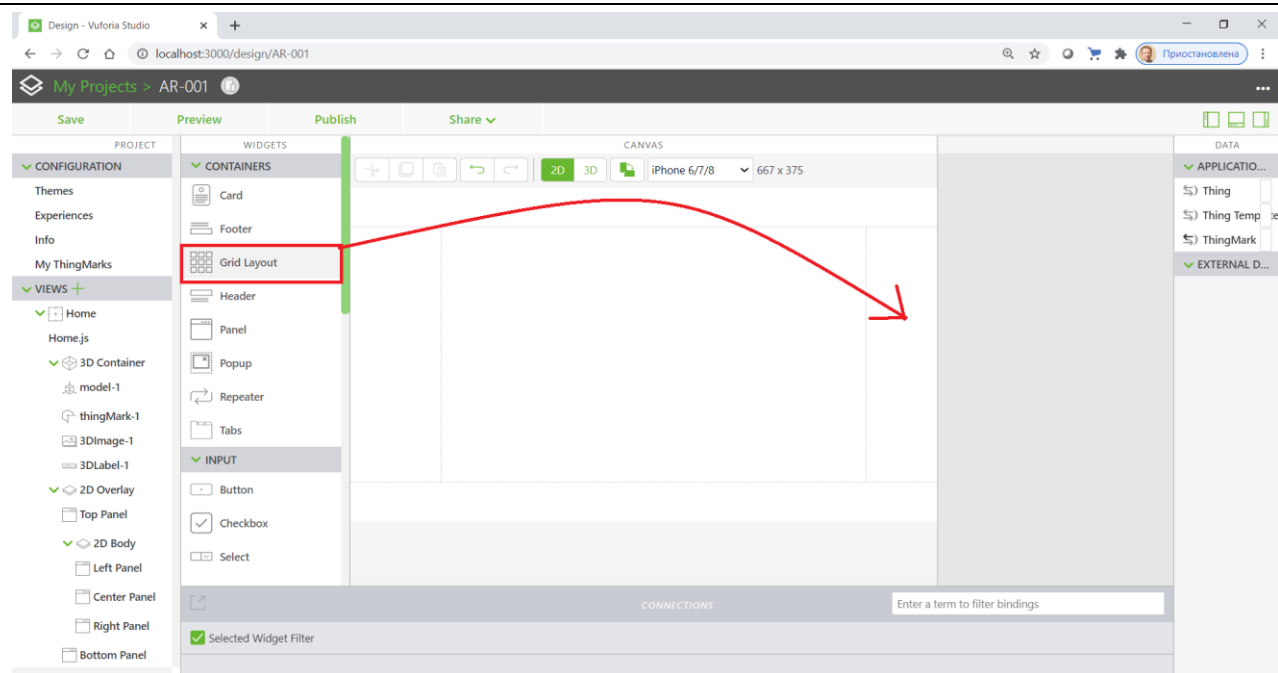


Шаг 39.

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

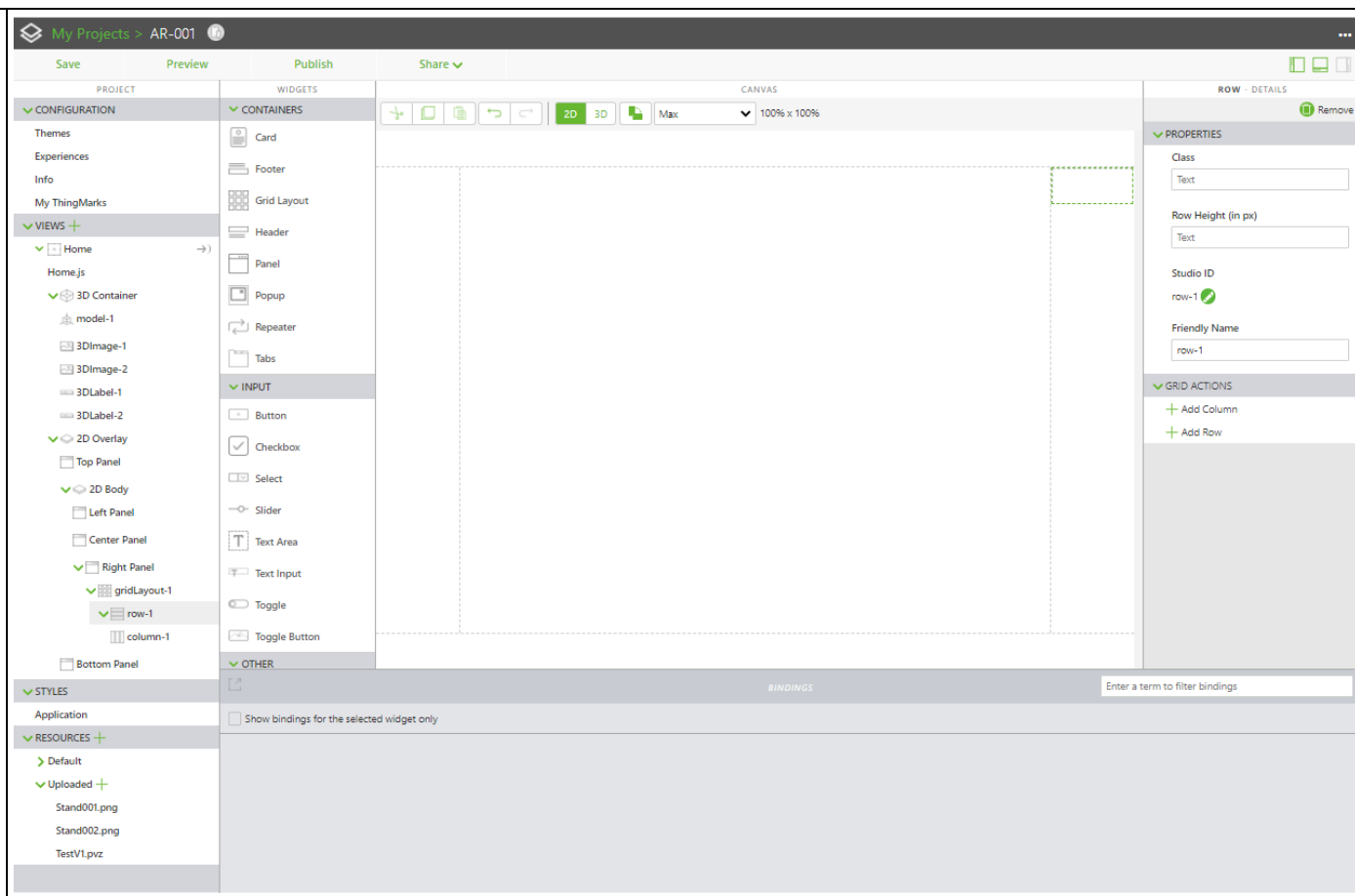
Он позволяет распределять/размечать отдельные элементы интерфейса для их размещения в панелях. Сами по себе панели не имеют характеристик, позволяющих это делать. См. «**Свойства Компонент**». Т.е. элементы интерфейса нельзя разместить в середине, справа или слева от уже имеющихся за счет их личных свойств. 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.

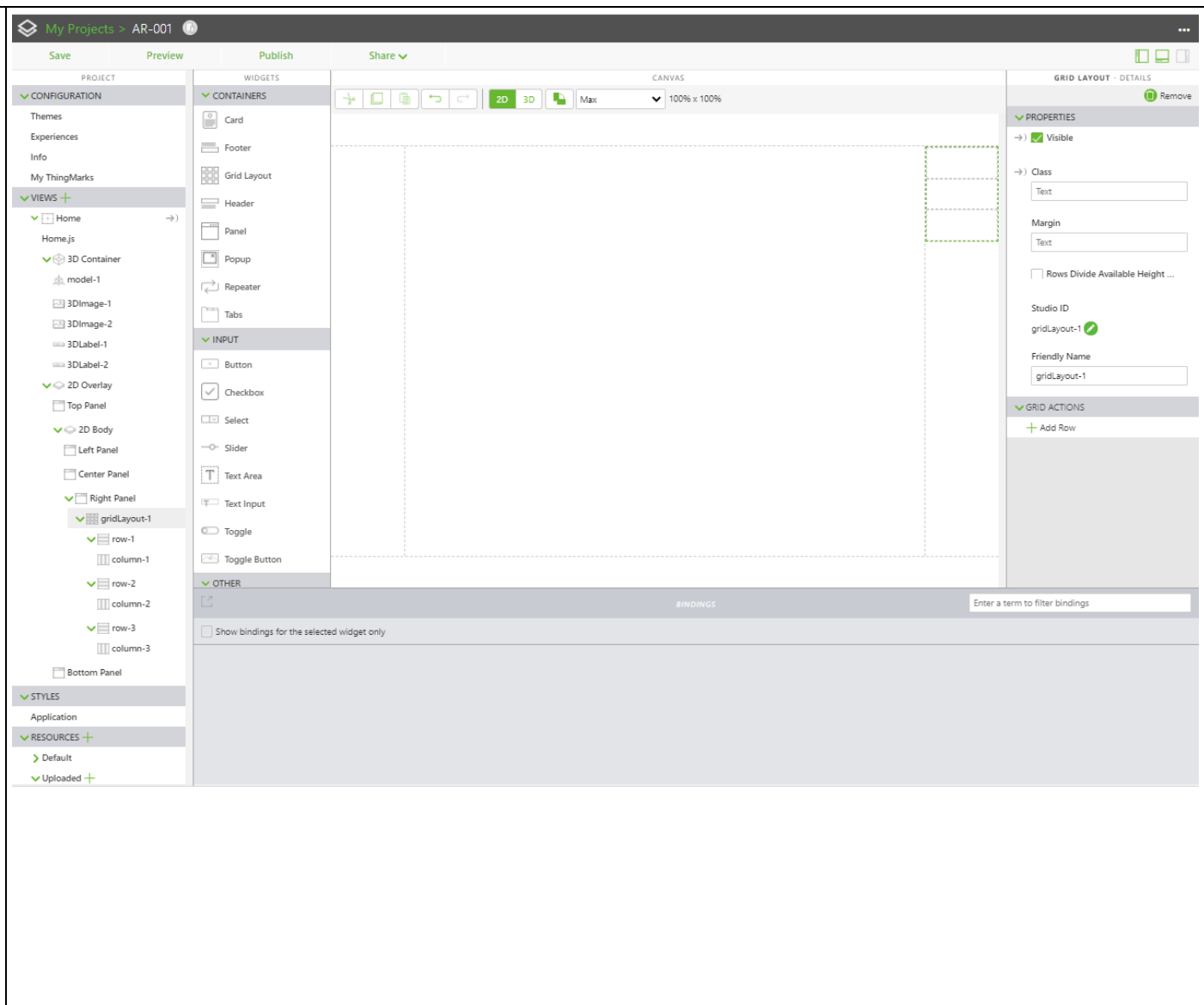
Добавляем в правой части канвы новые ячейки виджета **Grid Layout** (ряды в нашем случае) путем выполнения действия **Add Row** в зоне интерфейса «Свойства компонентов» - в нашем случае **Grid Layout**

GRID ACTIONS

+ Add Row

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

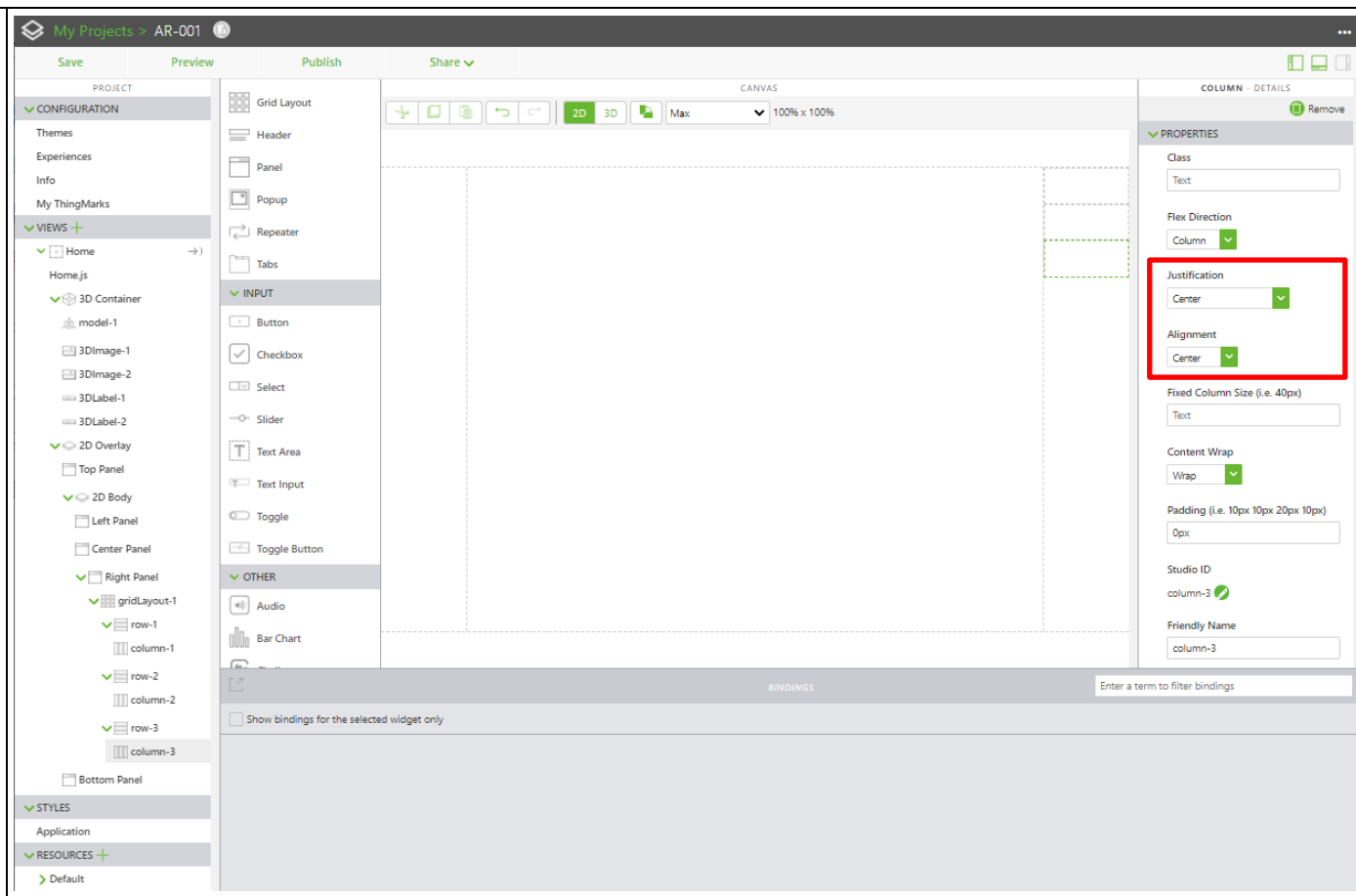
Т.о. мы резервируем на **2D канве** места для размещения элементов будущего плоского интерфейса. В качестве элементов **2D Интерфейса будем использовать 2D Виджеты**. В нашем случае предлагается использовать **Toggle Button** (кнопка переключения), **Slider** (слайдер), которые будут размещаться в отдельный **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.

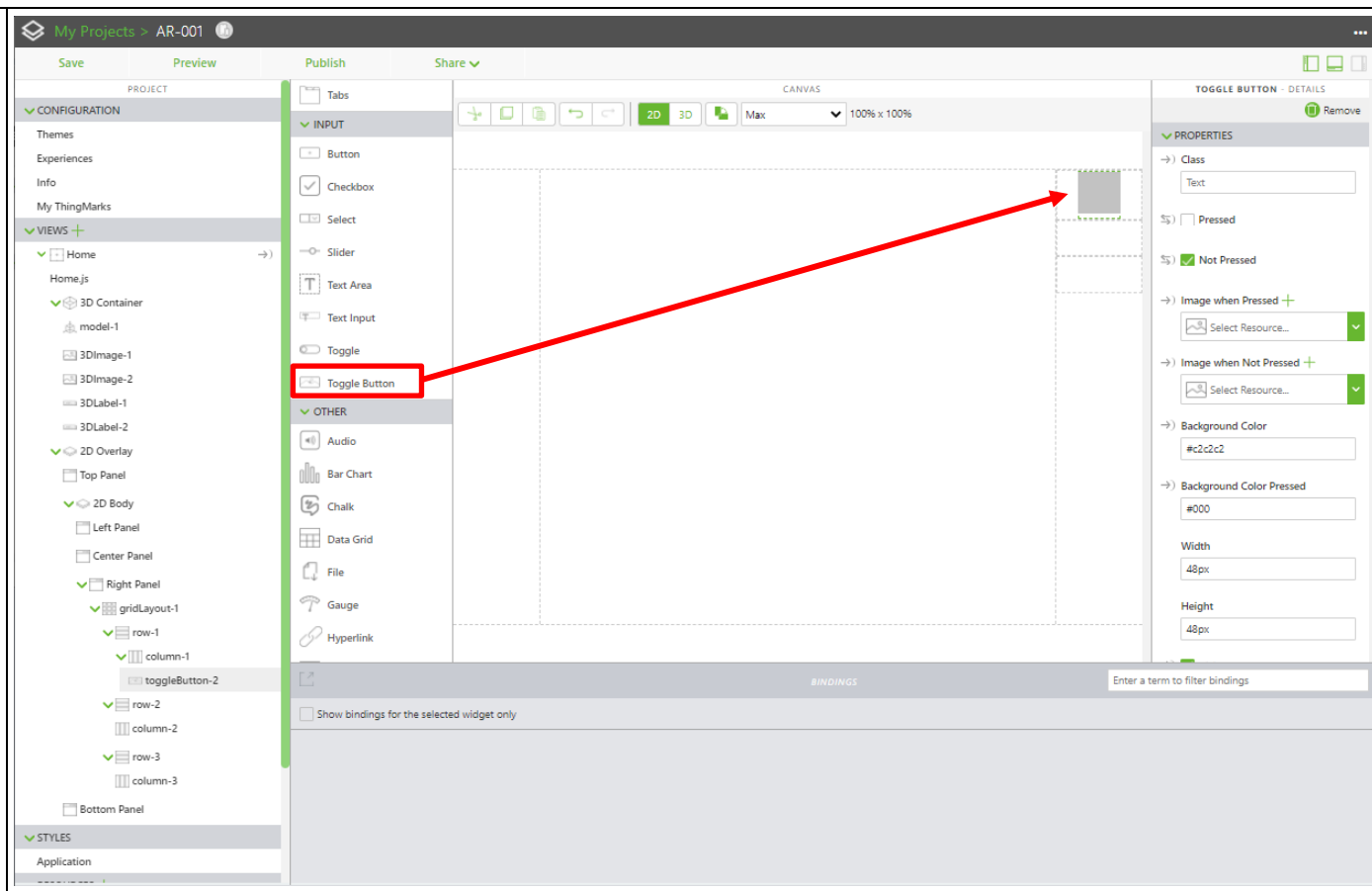
Результат троекратного вызова создания колонки в виджете **Grid** → **GRID ACTION** → **+...**

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



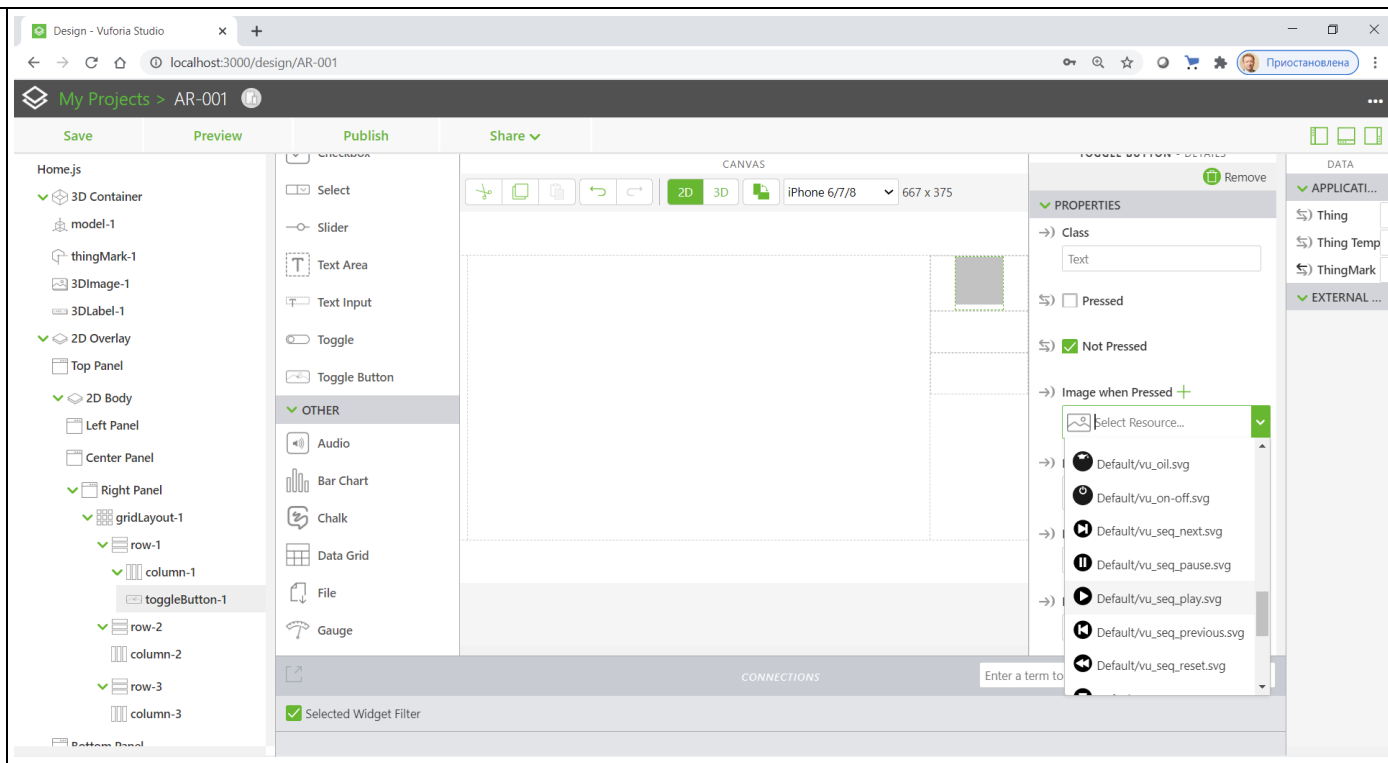
Шаг 43.

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



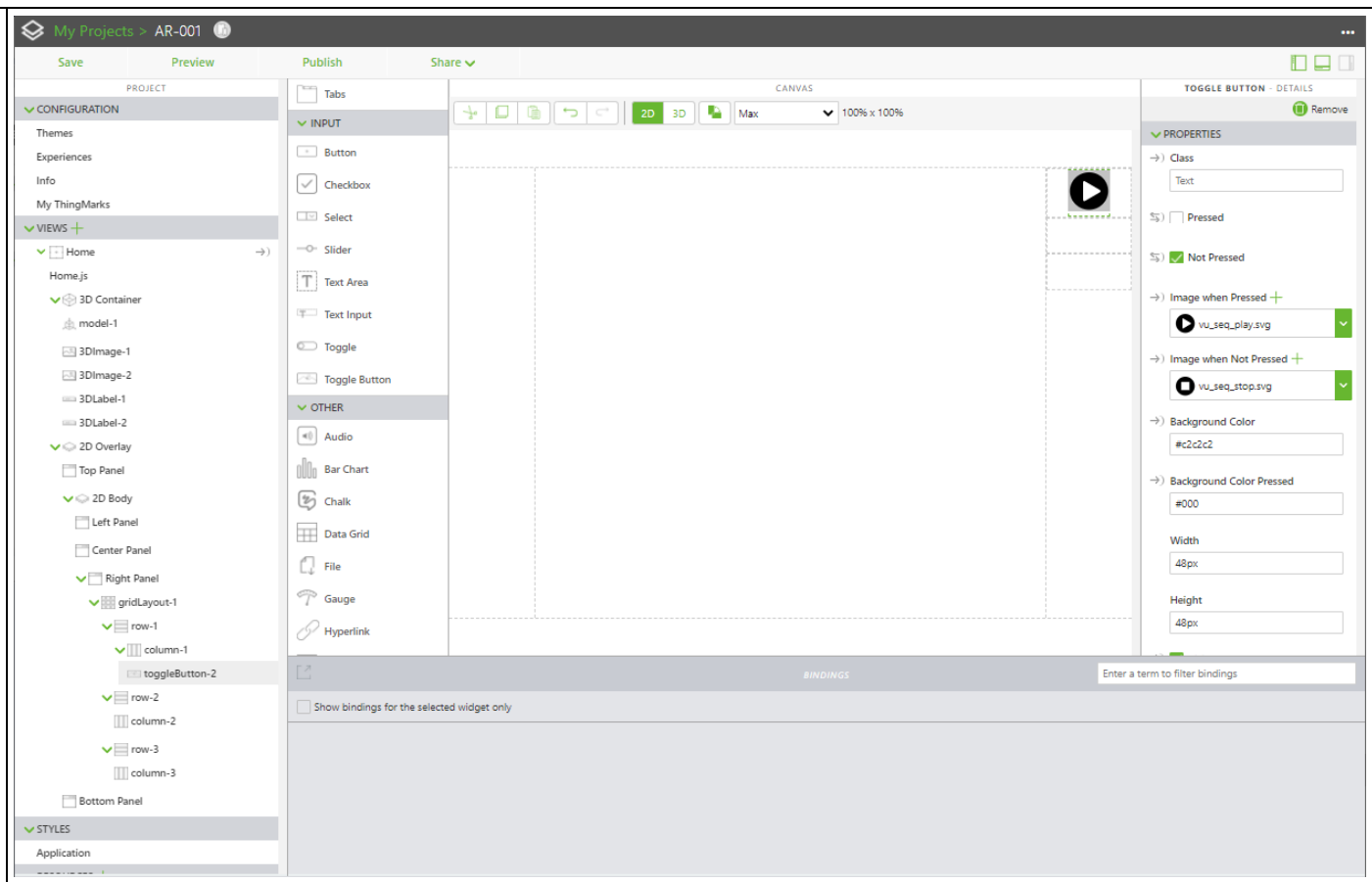
Шаг 44.

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**.

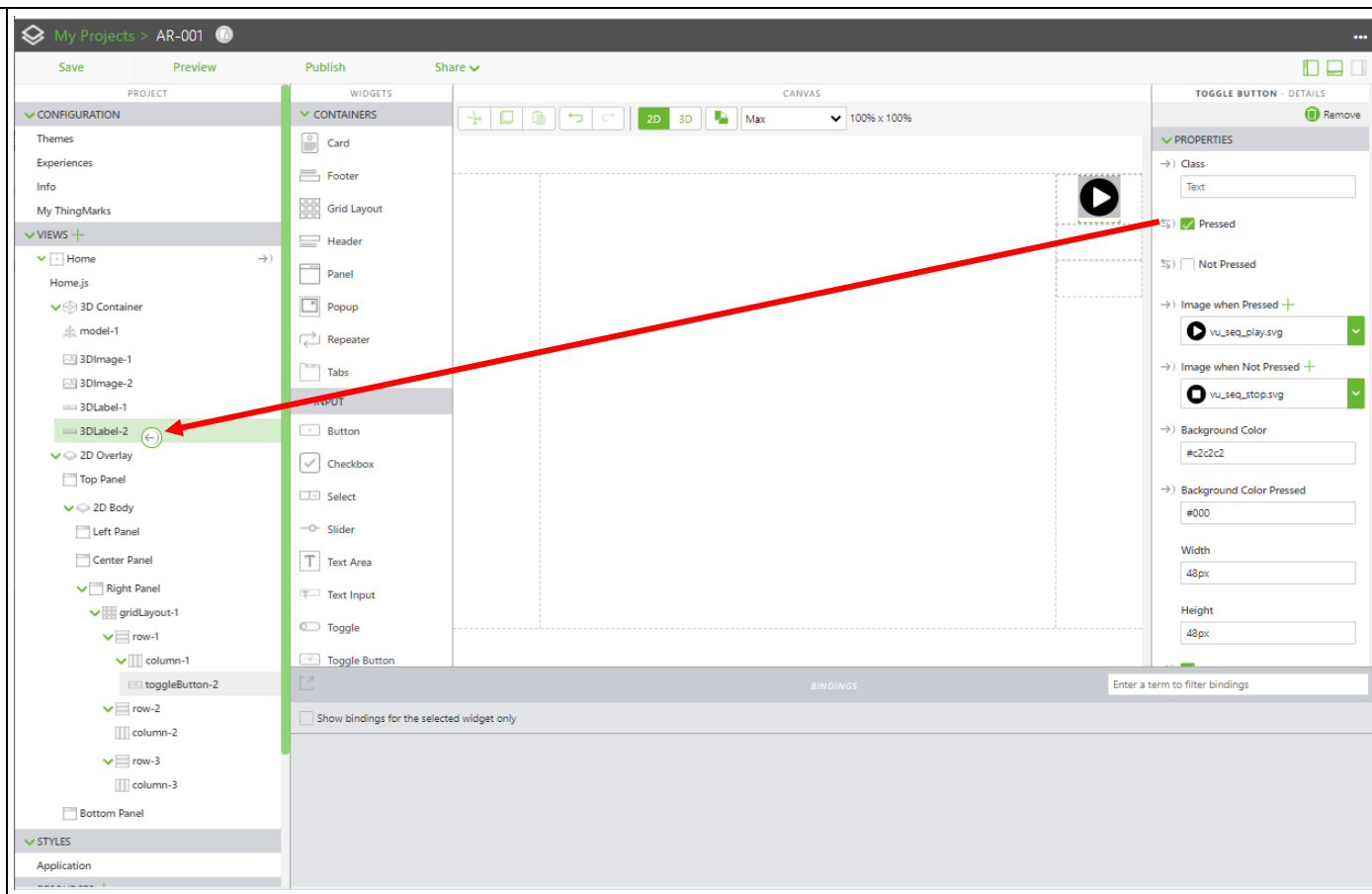


Шаг 46.

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**».

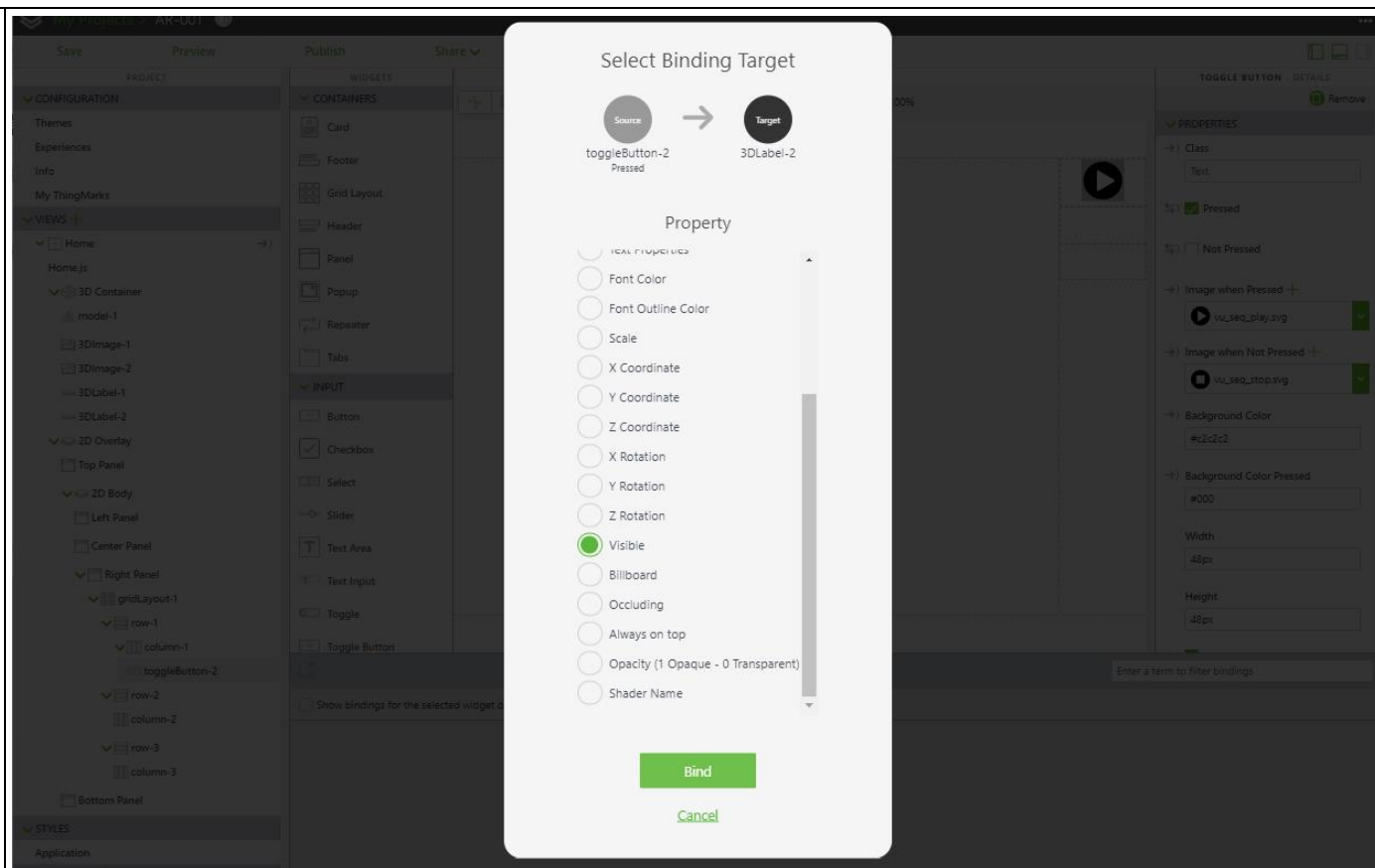


Шаг 47.

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**.

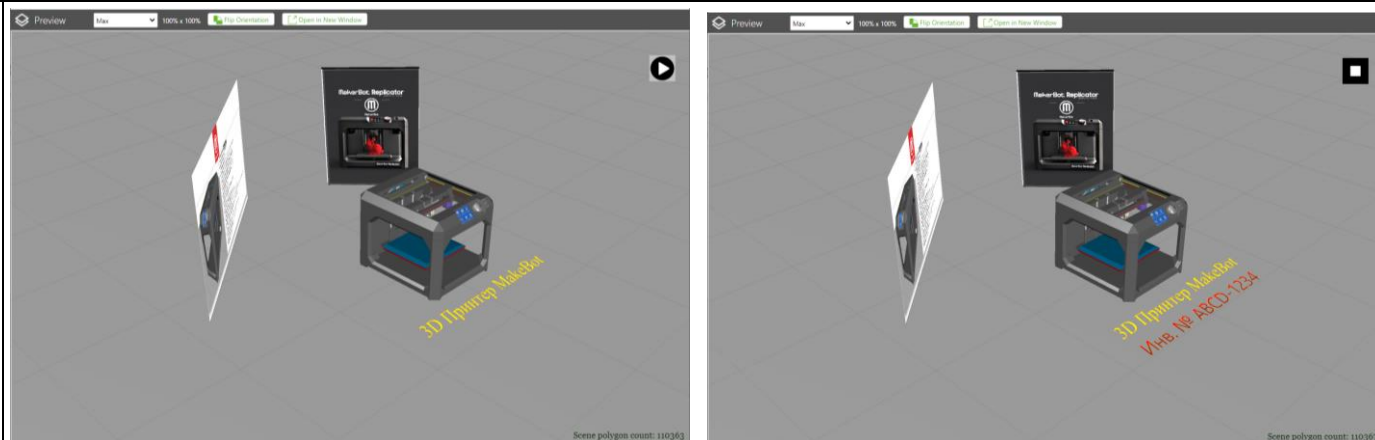
В нашем случае это должно быть свойство **Visible** (Видимость).

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



Шаг 48.

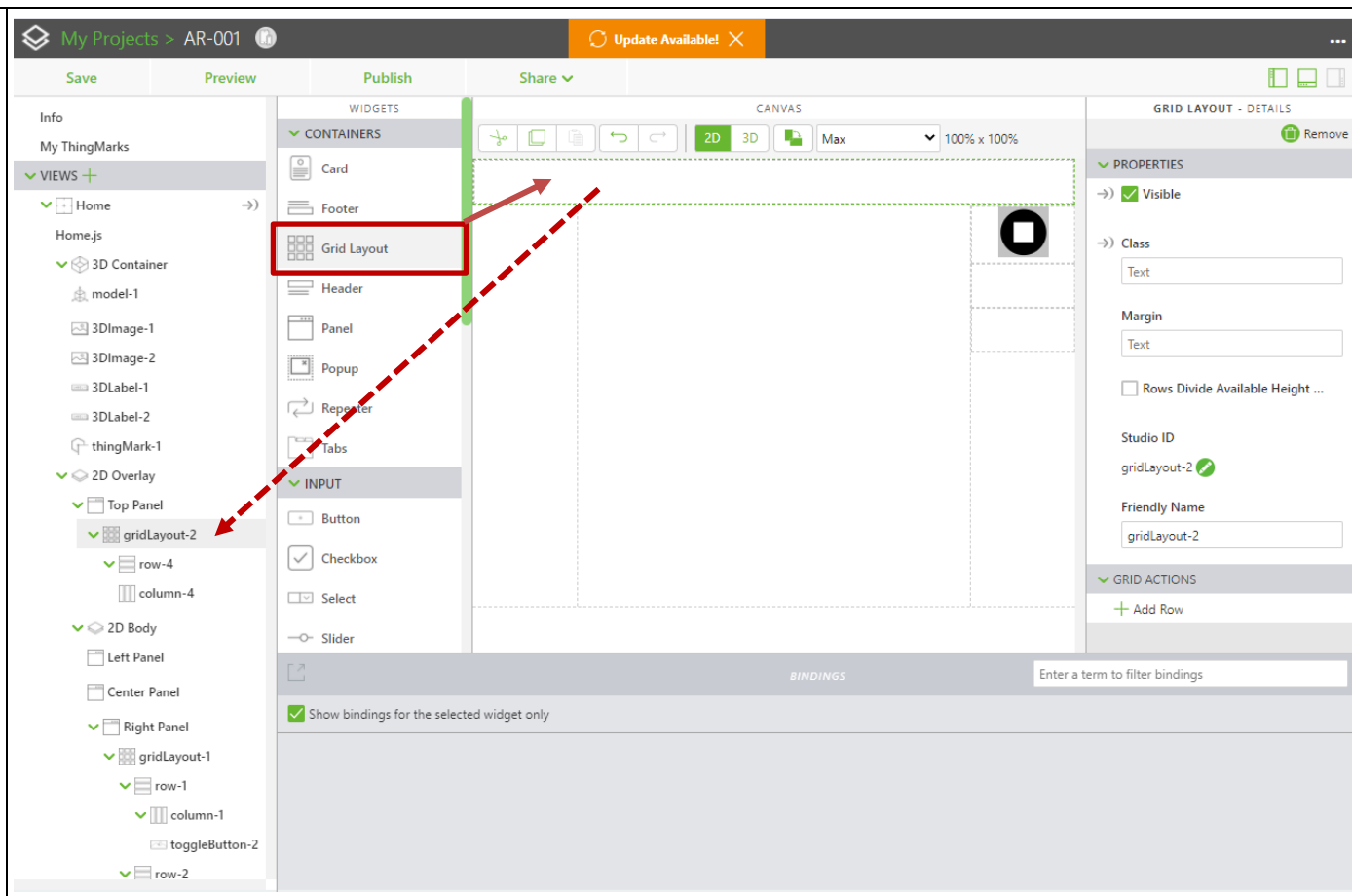
Можно выполнить **Preview** работы переключателя **Toggle Button**.



Шаг 49.

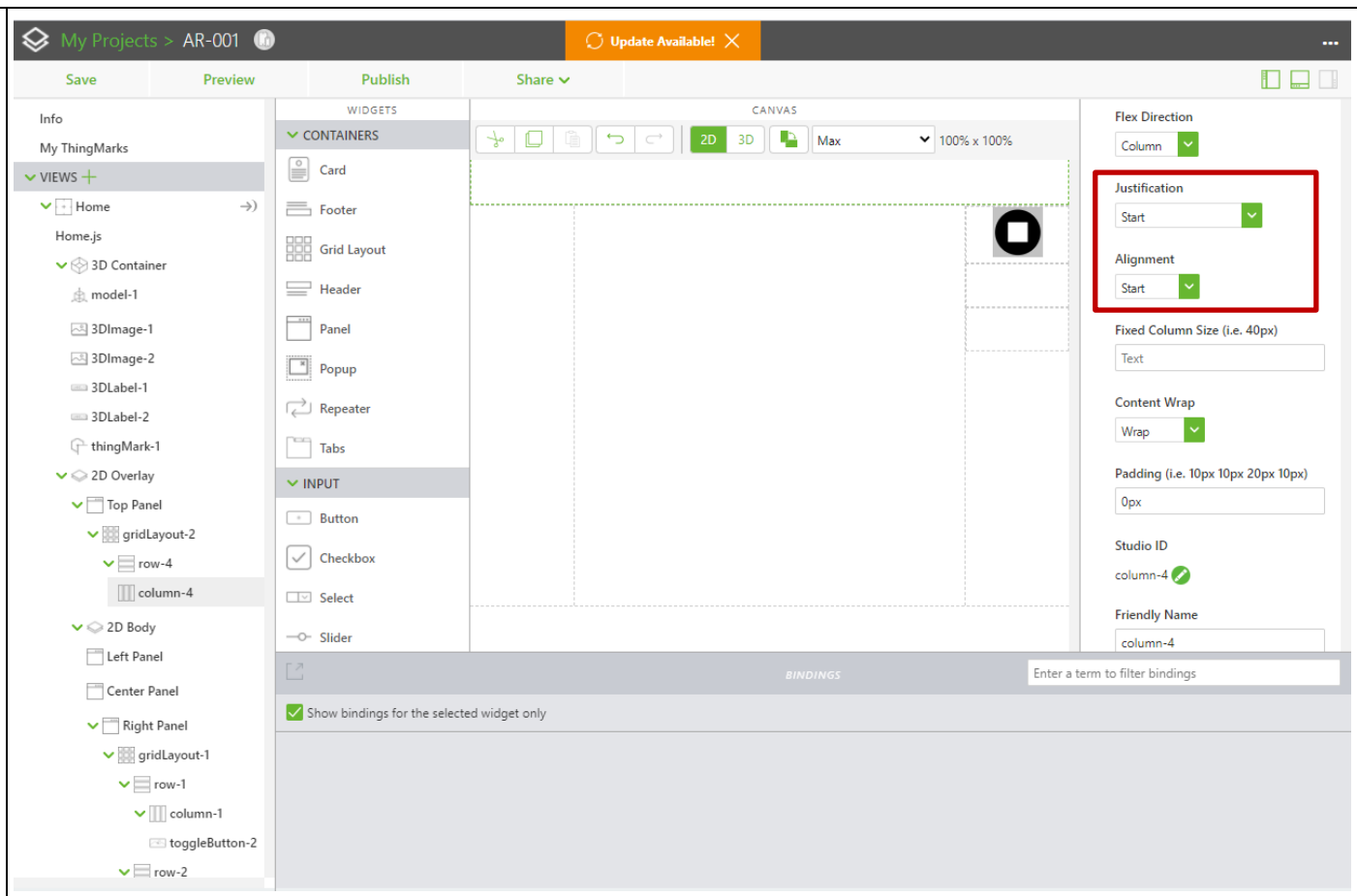
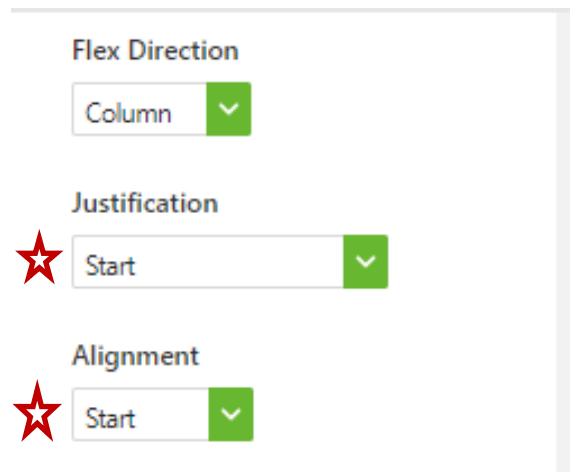
Добавляем **2D Виджет**, в котором будет размещаться, например логотип МЭИ или подобное изображение. Т.е. – добавляем в **2D Канву** (иногда можно услышать жаргонизм на «стекло») **2D-Image** (см. зону «Виджеты» редактора). Логотип хотим разместить в верхнем левом углу. Контейнер, отвечающий за это,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

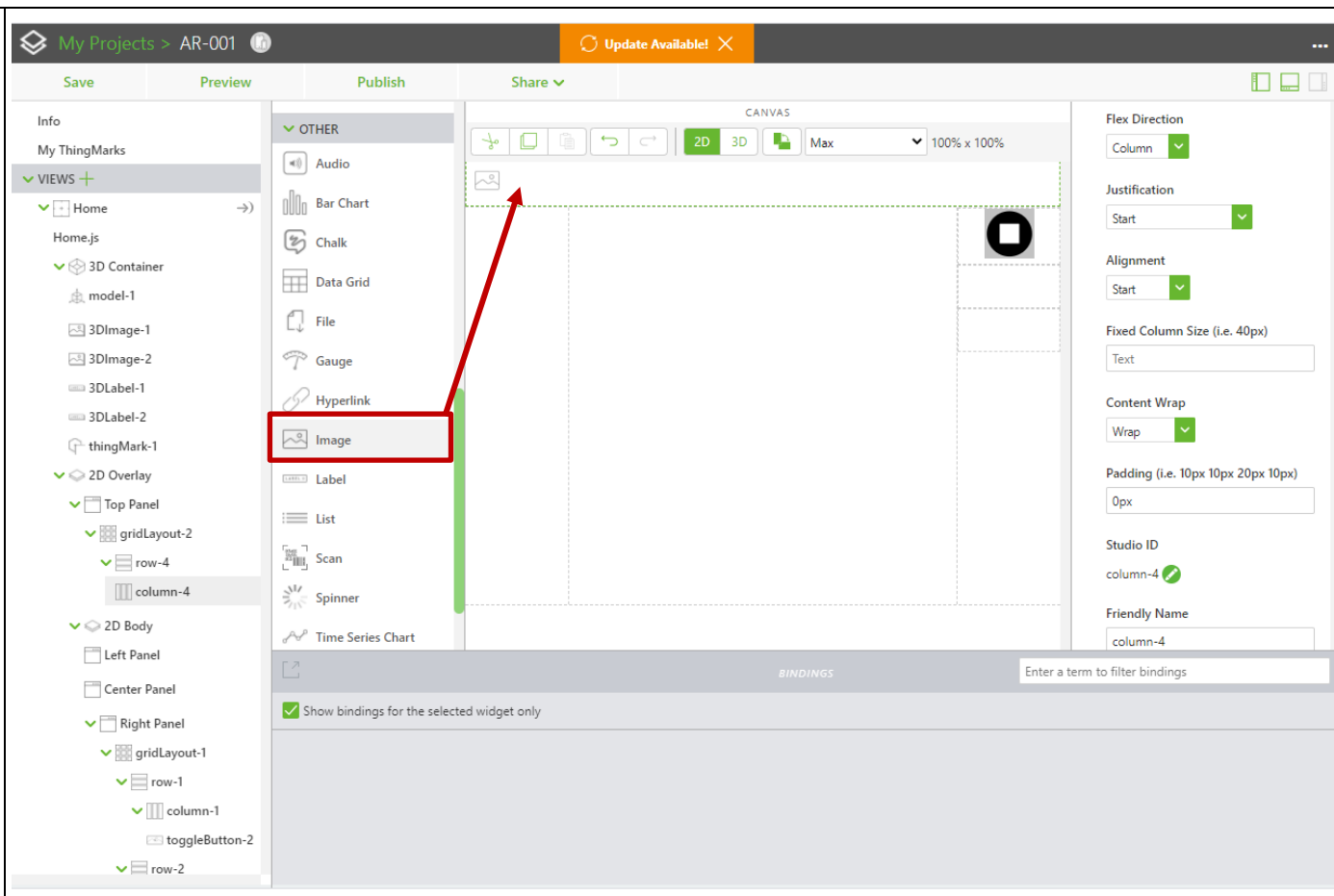
Grid Layout уже содержит одну колонку и один ряд. Нам достаточно этого для разметки левого верхнего угла в колонке. Настраиваем колонку для позиционирования того виджета, который будем здесь размещать, т.е. – **логотипа МЭИ. Выравнивание – по**



Шаг 51.

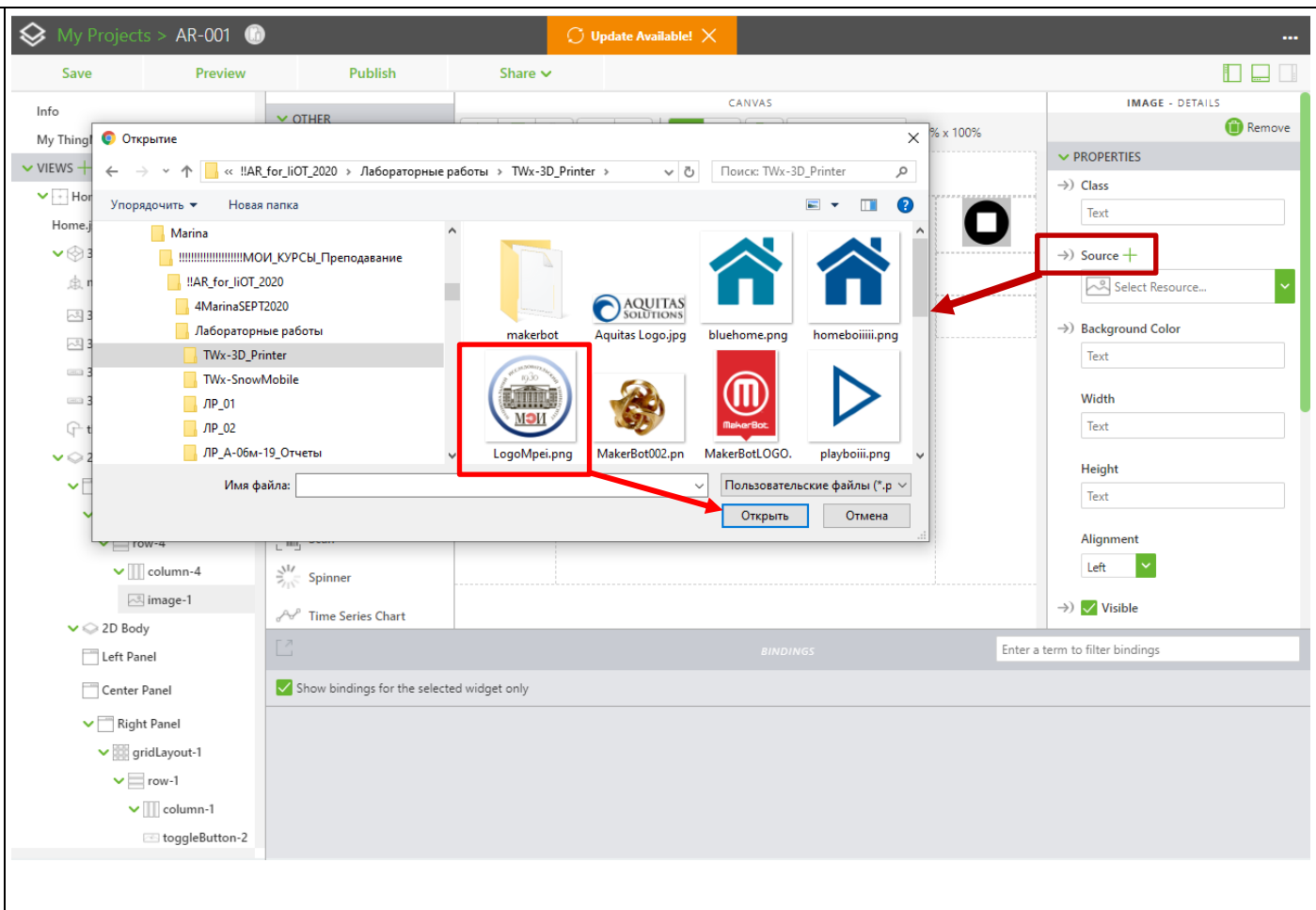
Добавляем **2D Виджет Image** из **WIDGET→OTHER**.

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



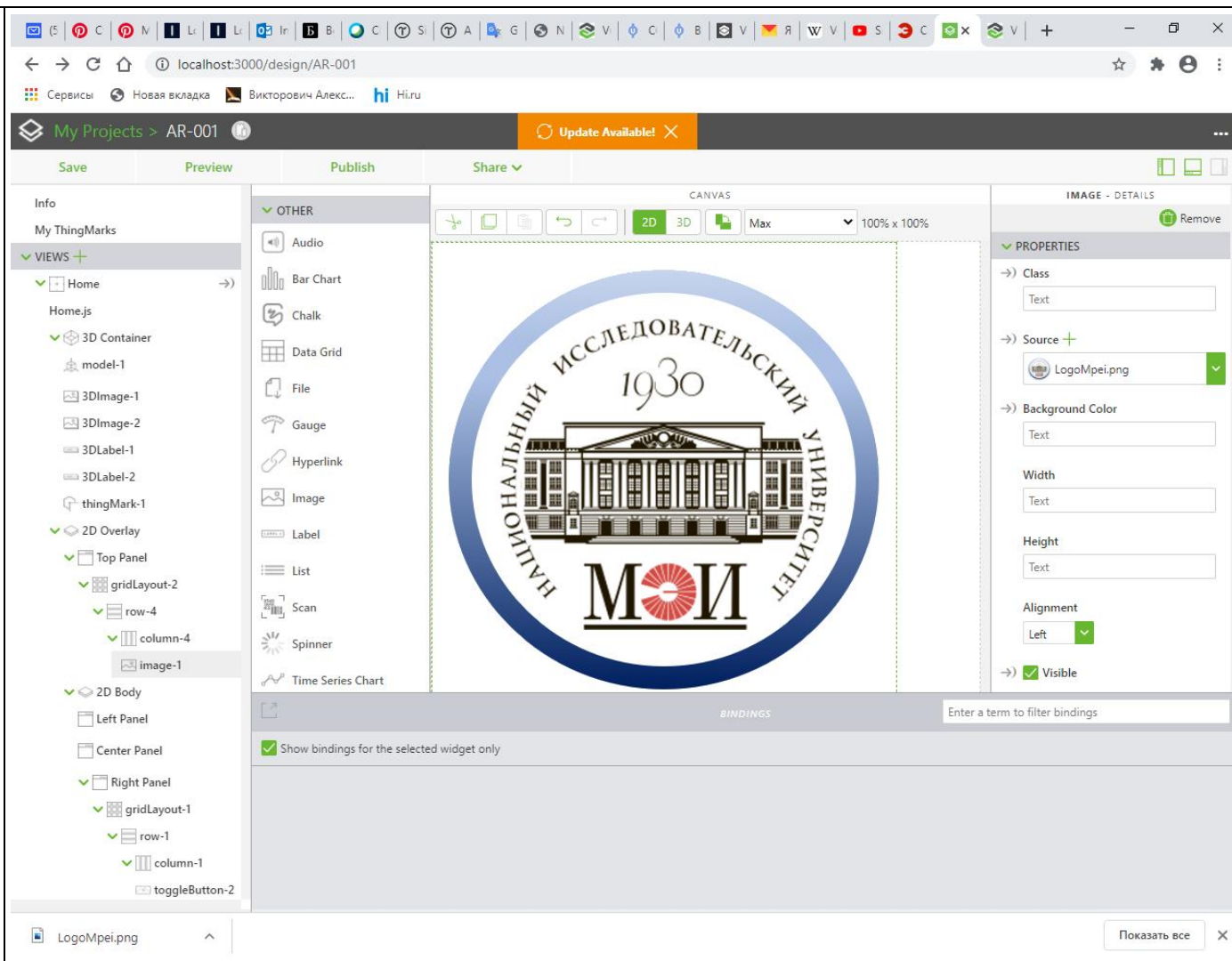
Шаг 52.

В размещенный **Image** надо поместить саму 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) не должны превышать **60x60** пикселей.



Шаг 54.

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

→ Source +

LogoMpei.png

→ Background Color

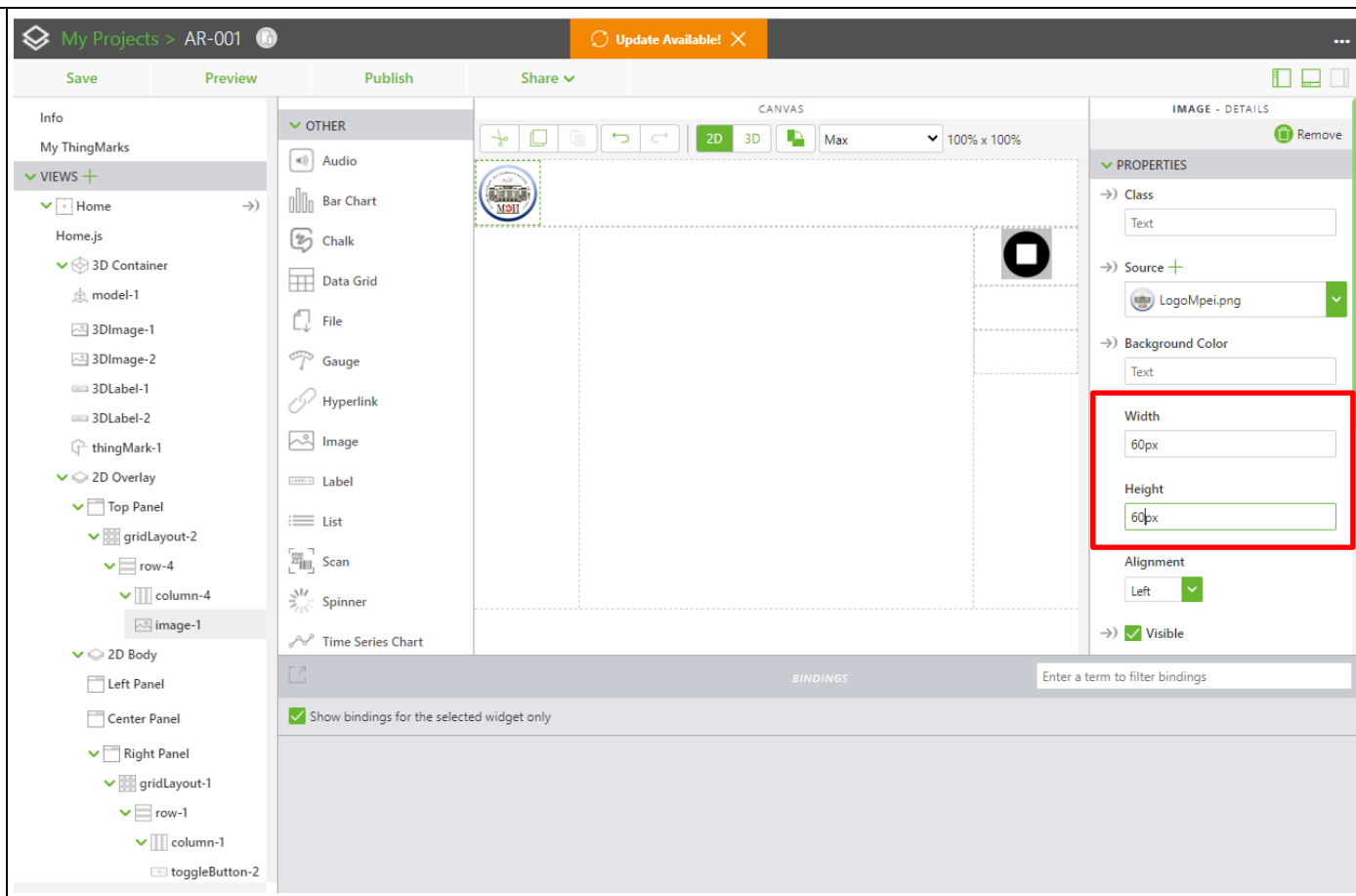
Text

Width

60px

Height

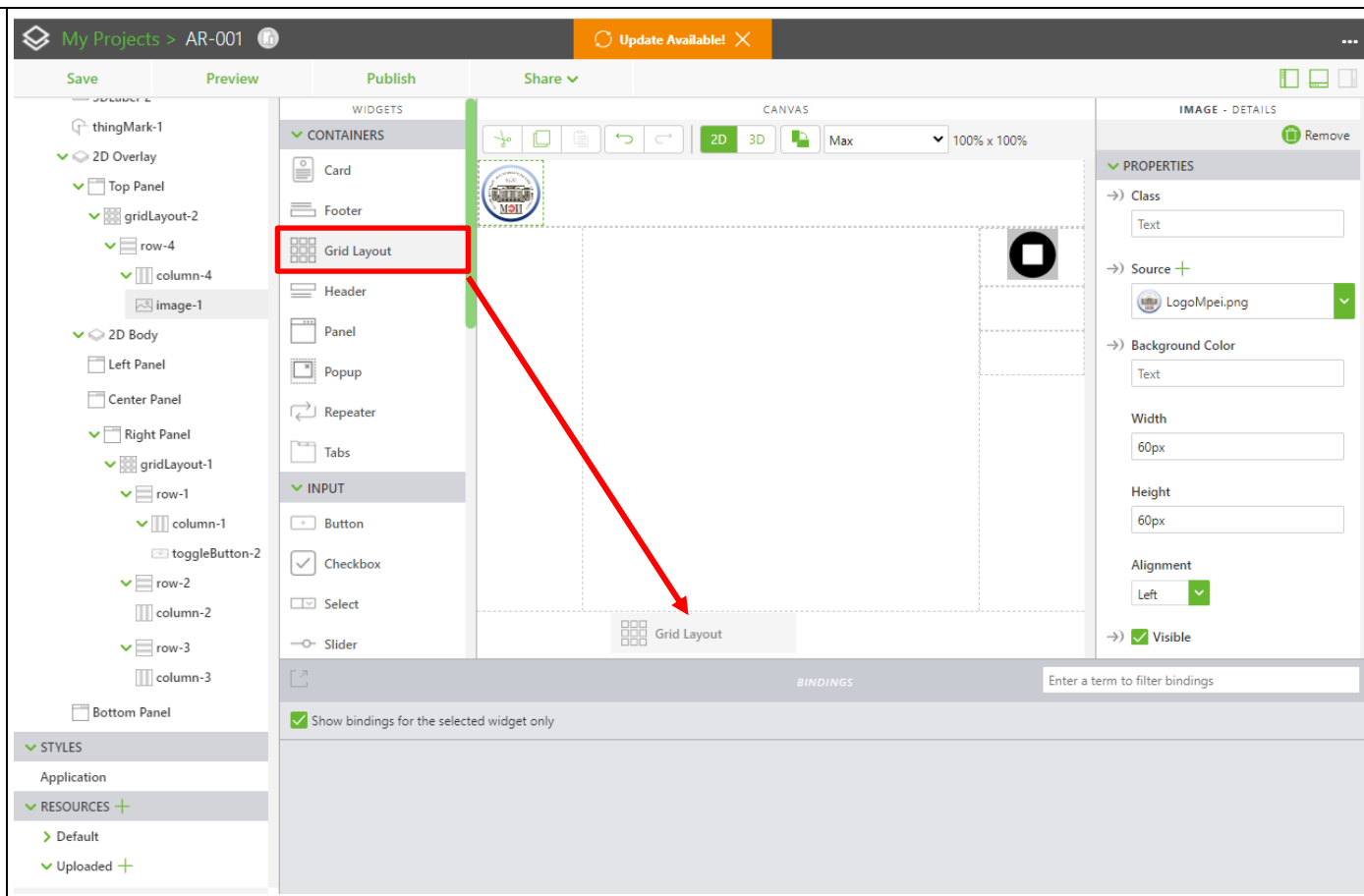
60px



Шаг 55.

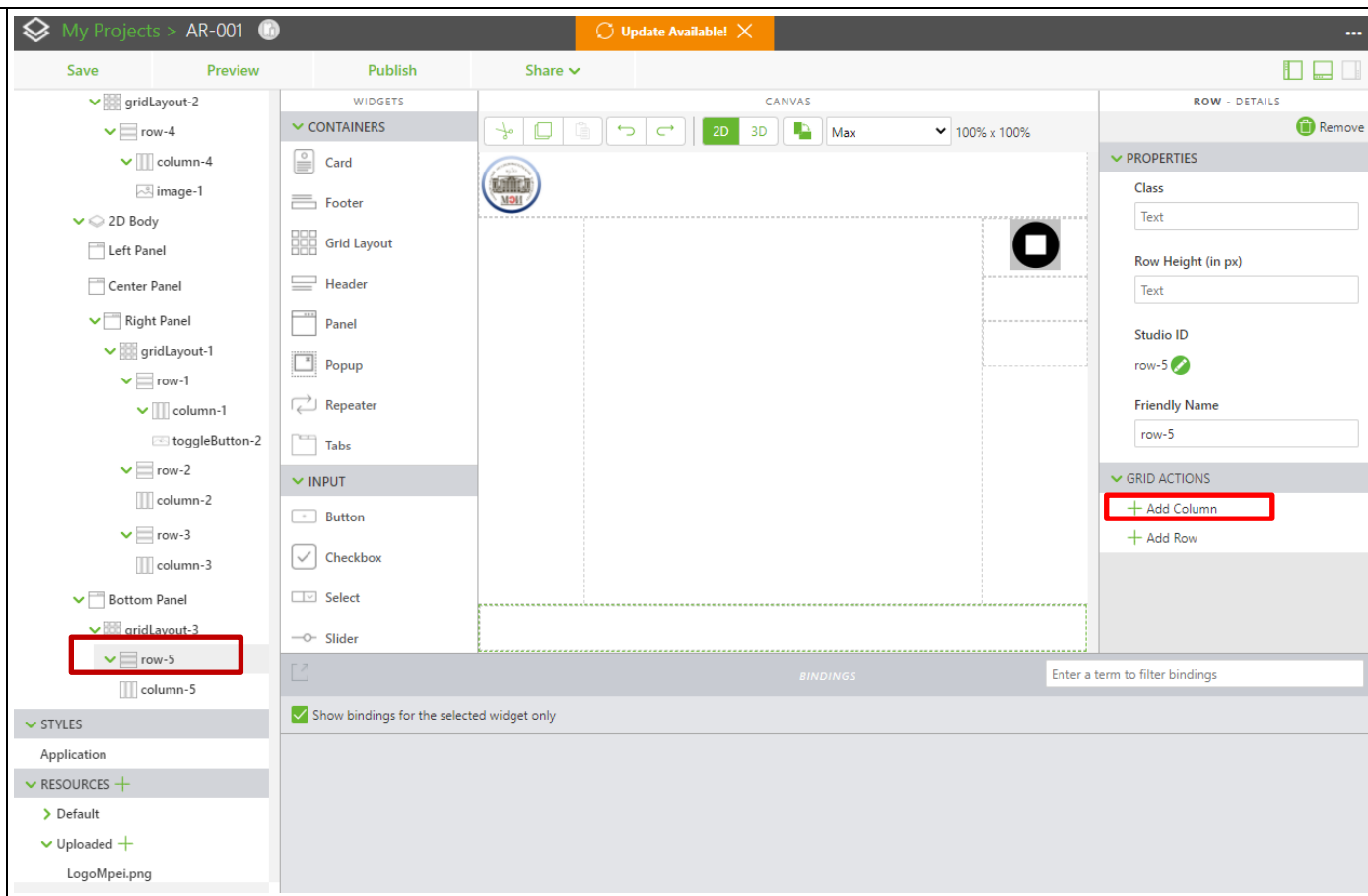
Для реализации возможных пунктов сценария разрабатываемого **Experience** – например, увеличение/уменьшение трехмерного объекта (модели принтера), изменение прозрачности объекта, вращение вокруг оси и пр. – необходимо использовать **2D – Виджет Slider**. Будем размещать эти слайдеры в **Bottom Panel**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

МОЖЕТЕ РАЗМЕСТИТЬ ТОЛЬКО ОДИН СЛАЙДЕР И ОДНО ДЕЙСТВИЕ!



Шаг 56.

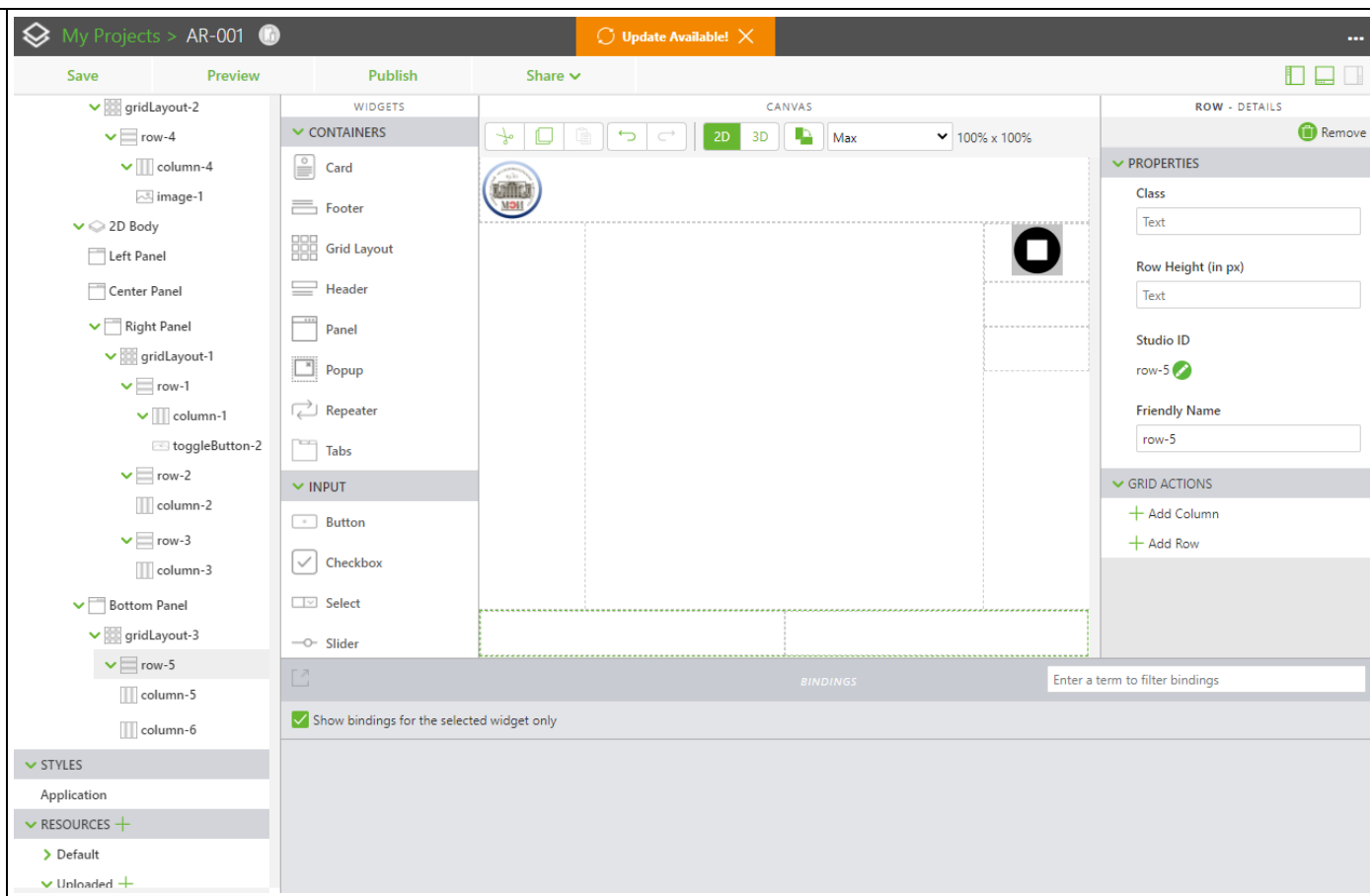
С помощью **Grid Layout** размечаем **Bottom Panel** на 2 столбца → в «Свойствах компонент» В ряду (Row) добавляем **Column** → **Add Column**.



Шаг 57.

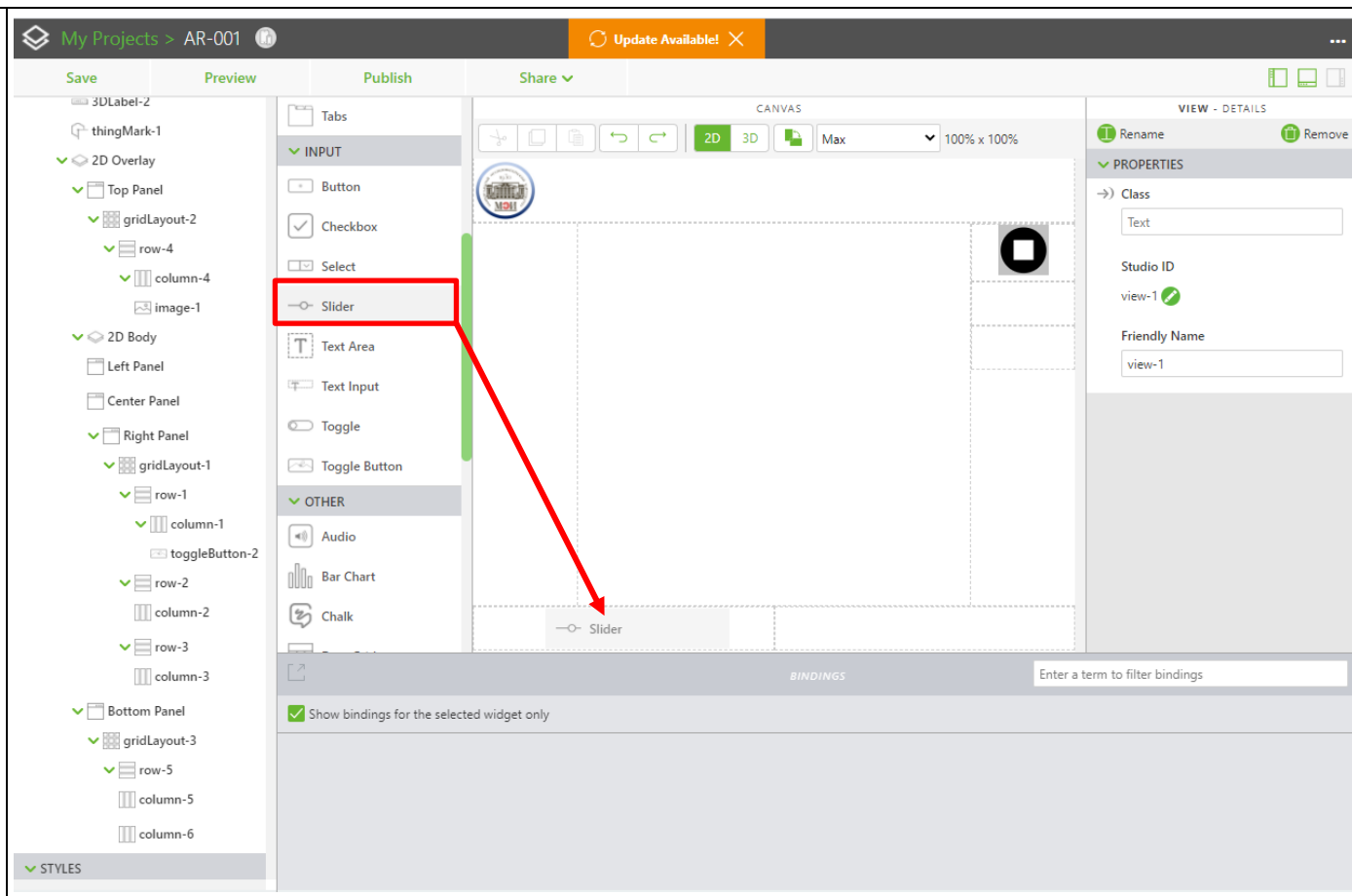
В результате добавления колонки в нижней части плоского меню (**2D CANVAS**) появились 2 колонки, в которые позже будем размещать слайдеры.

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



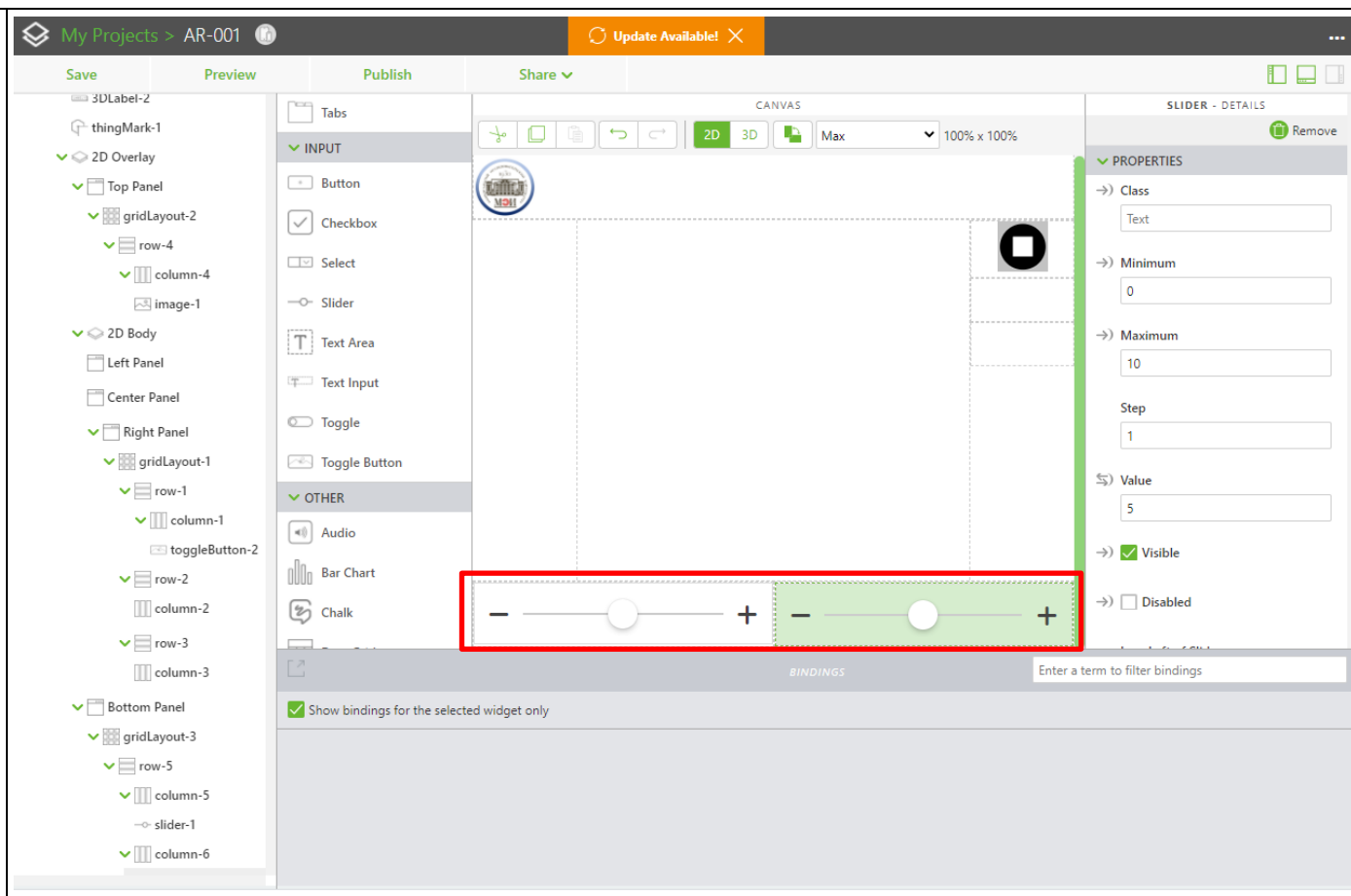
Шаг 58.

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



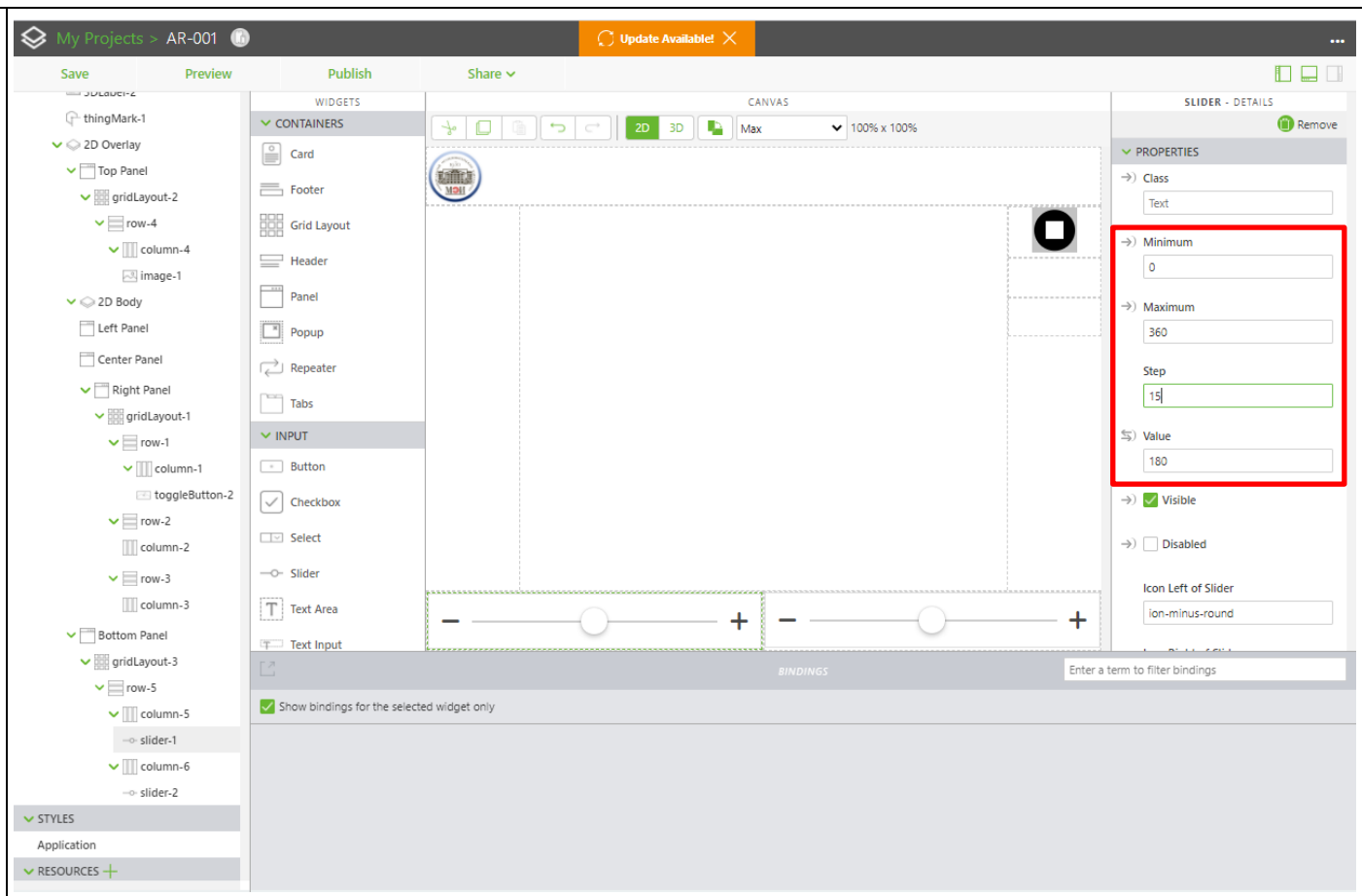
Шаг 59.

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



Шаг 60.

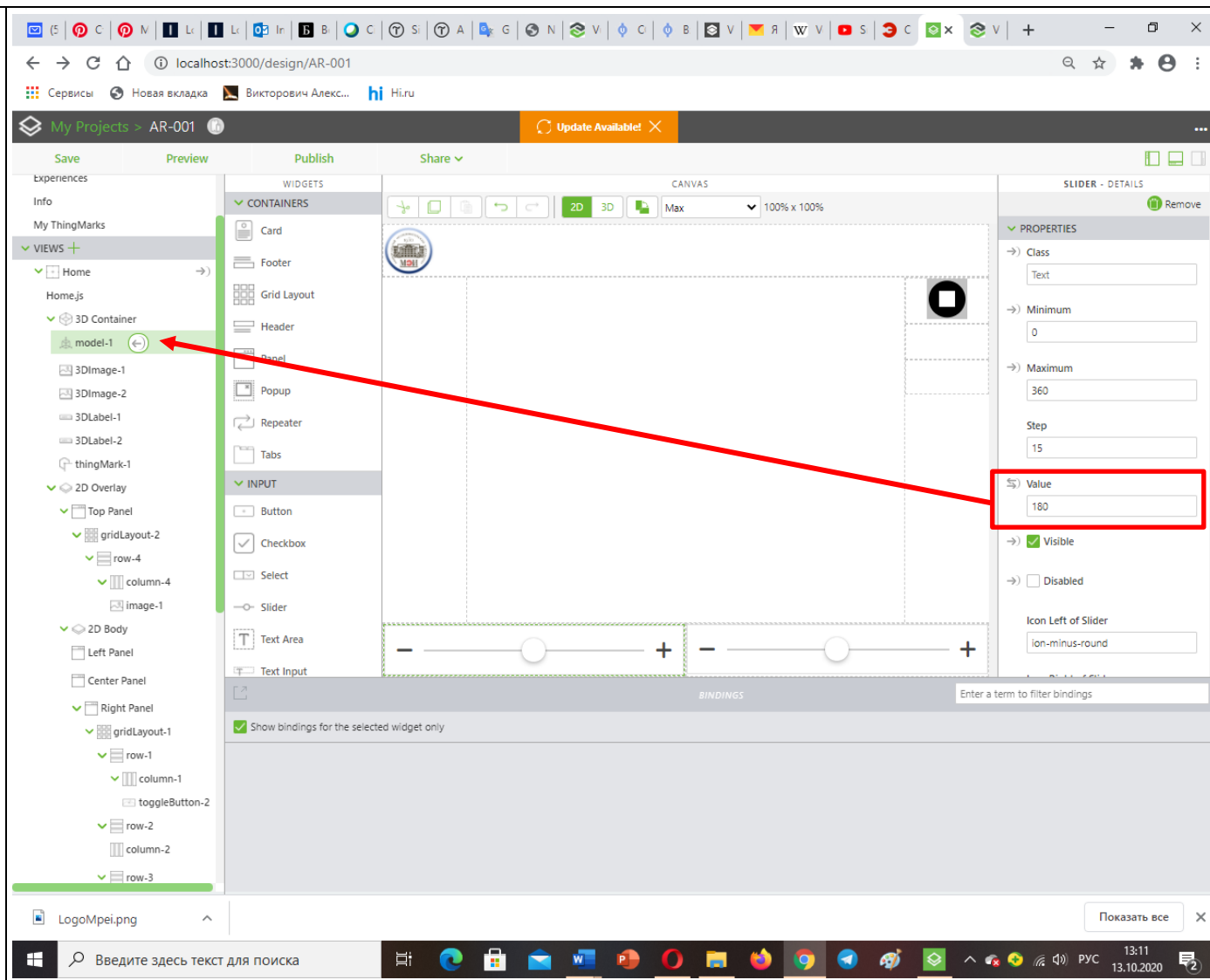
Настроим первый слайдер на управление вращением **3D Model** принтера: настраиваем параметры слайдера – **шаг** (слишком мелкий шаг для мобильного устройства лучше не брать), **начало и конец** вращения (**0 – 360 градусов**), **Value** (начальная позиция самого слайдера).



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».

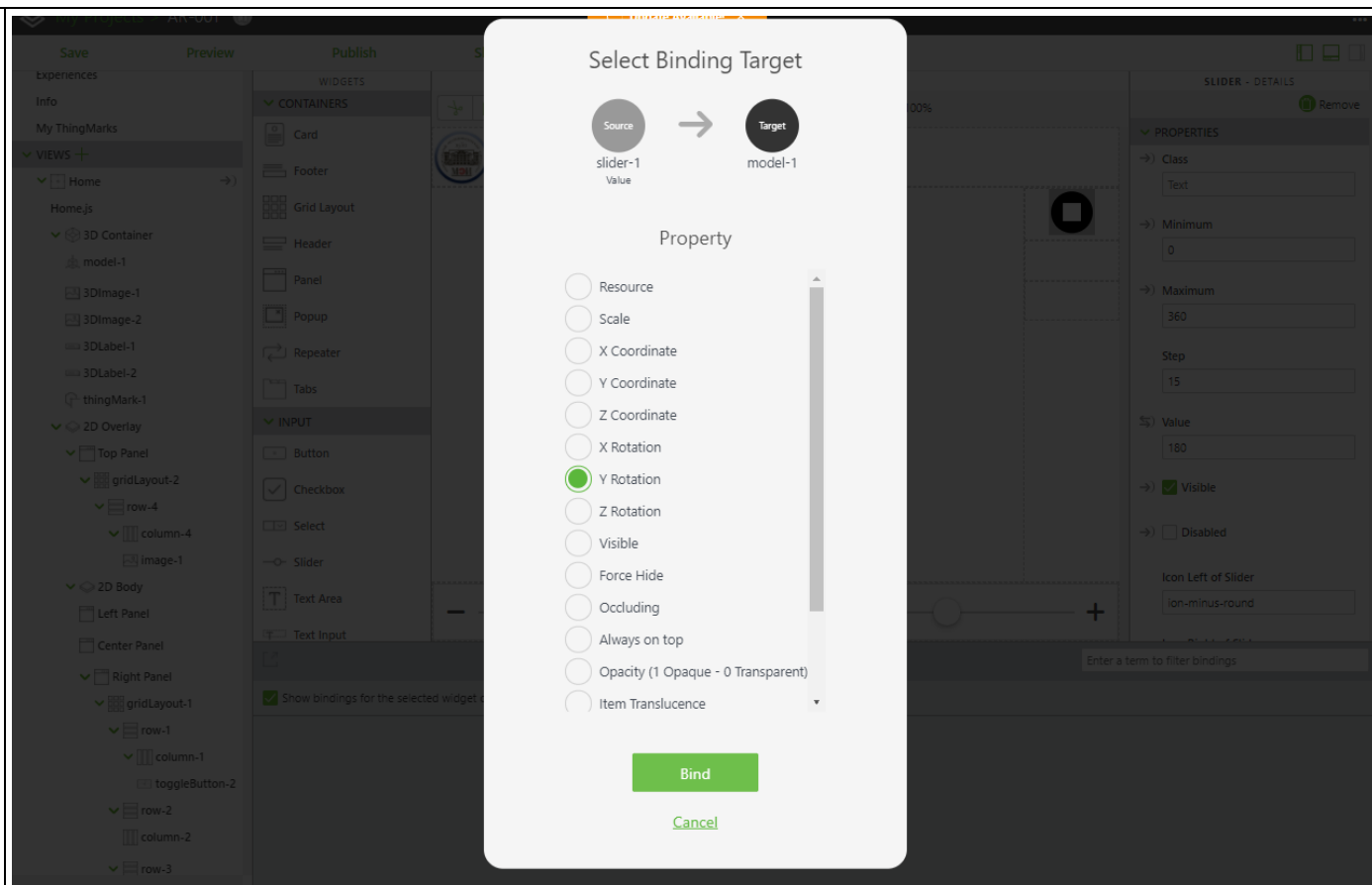


Шаг 62.

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**.

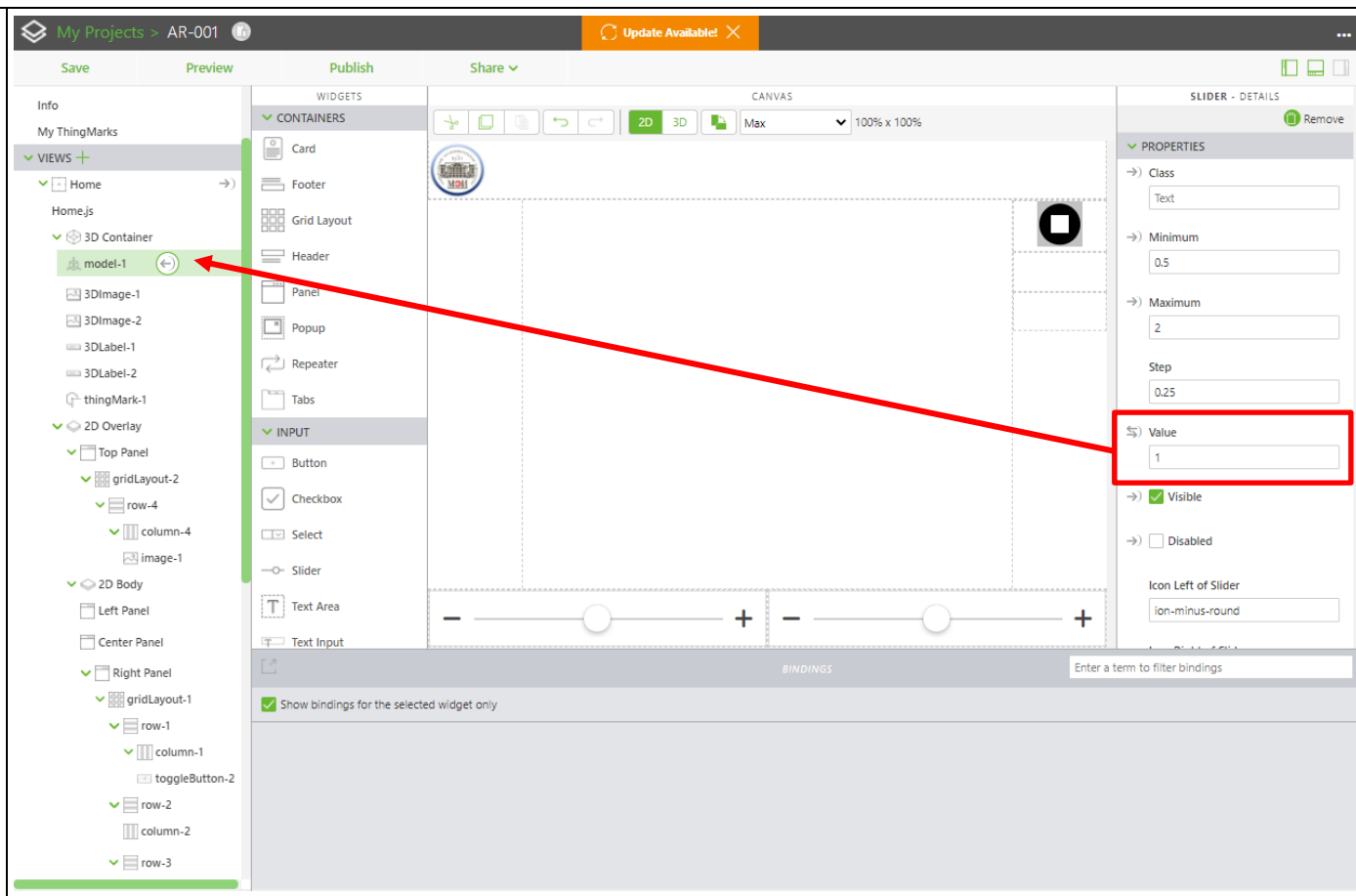
В нашем случае это должно быть свойство **Y- Rotation** (Вращение вокруг оси Y).

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.

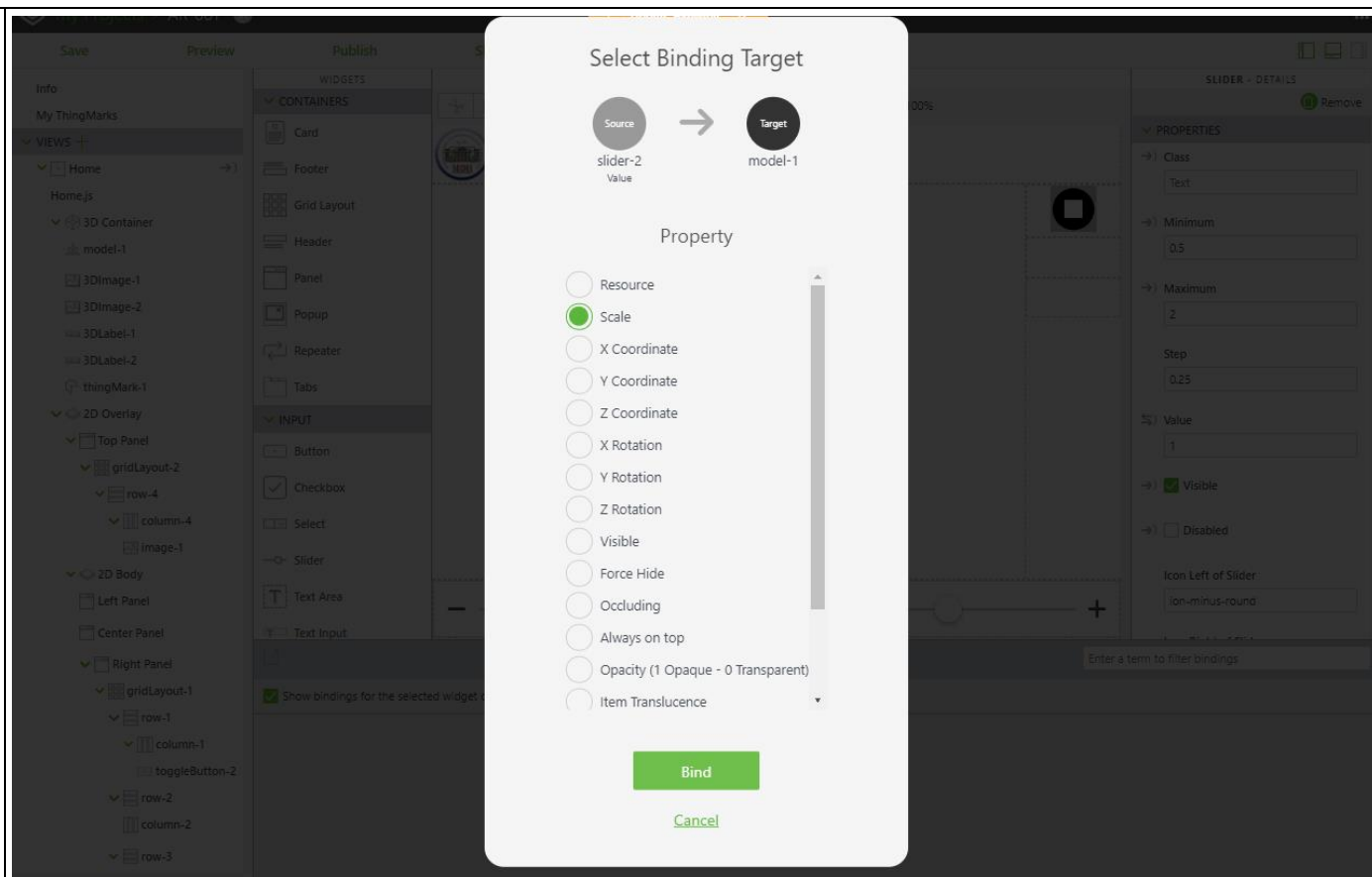


Шаг 64.

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 shows the Vuforia Studio interface with the following components:

- Info Panel:** Displays the project structure under 'My ThingMarks' > 'VIEWS' > 'Home' > 'Home.js' > '3D Container' > 'model-1'. It lists various 3D and 2D elements like 3DImage, 3DLabel, thingMark, 2D Overlay, Top Panel, GridLayout, row, column, image, 2D Body, Left Panel, Center Panel, Right Panel, and toggleButton.
- WIDGETS Panel:** Lists various UI components under 'CONTAINERS' (Card, Footer, Grid Layout, Header, Panel, Popup, Repeater, Tabs) and 'INPUT' (Button, Checkbox, Select, Slider, Text Area, Text Input).
- CANVAS:** A 3D view of the device model with a 'Max' zoom level and a 100% x 100% scale. It shows a 3D container with a 2D overlay and a slider.
- MODEL - DETAILS Panel:** Shows properties for the selected widget, including Resource (TestV1.pvz), Scale (1.0), X Coordinate (0), Y Coordinate (0), Z Coordinate (0), X Rotation (0), Y Rotation (90), and Z Rotation (0).
- BINDINGS Panel:** A table showing two new bindings added for the selected widget. The table is highlighted with a red border.

Show bindings for the selected widget only	
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.

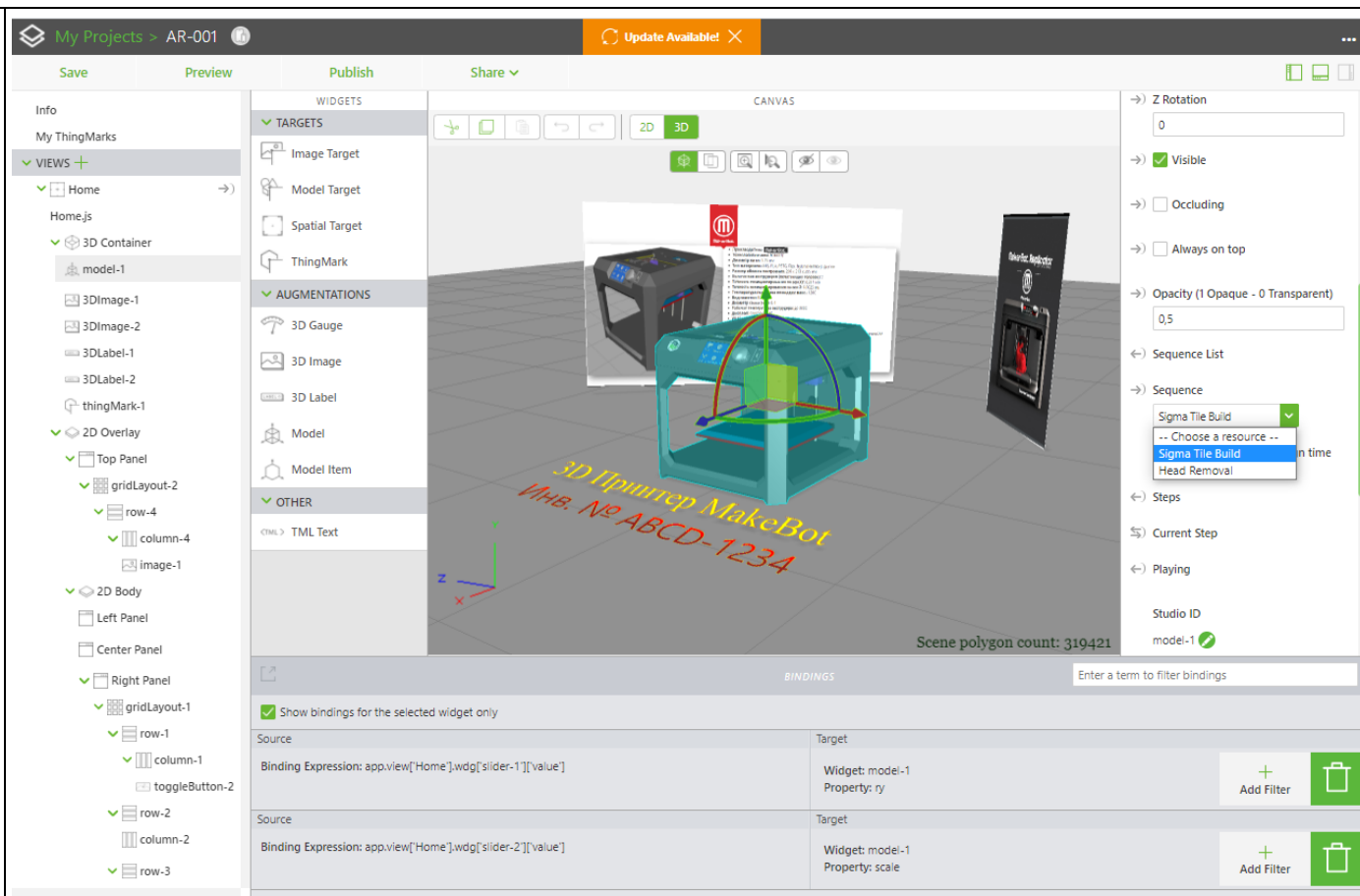
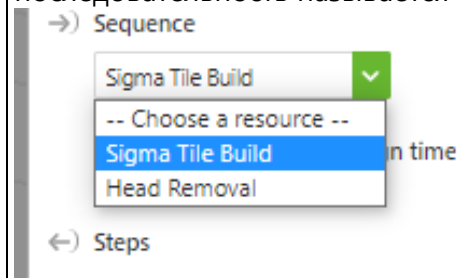
Посмотрите промежуточный результат в режиме **Preview**.



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**.

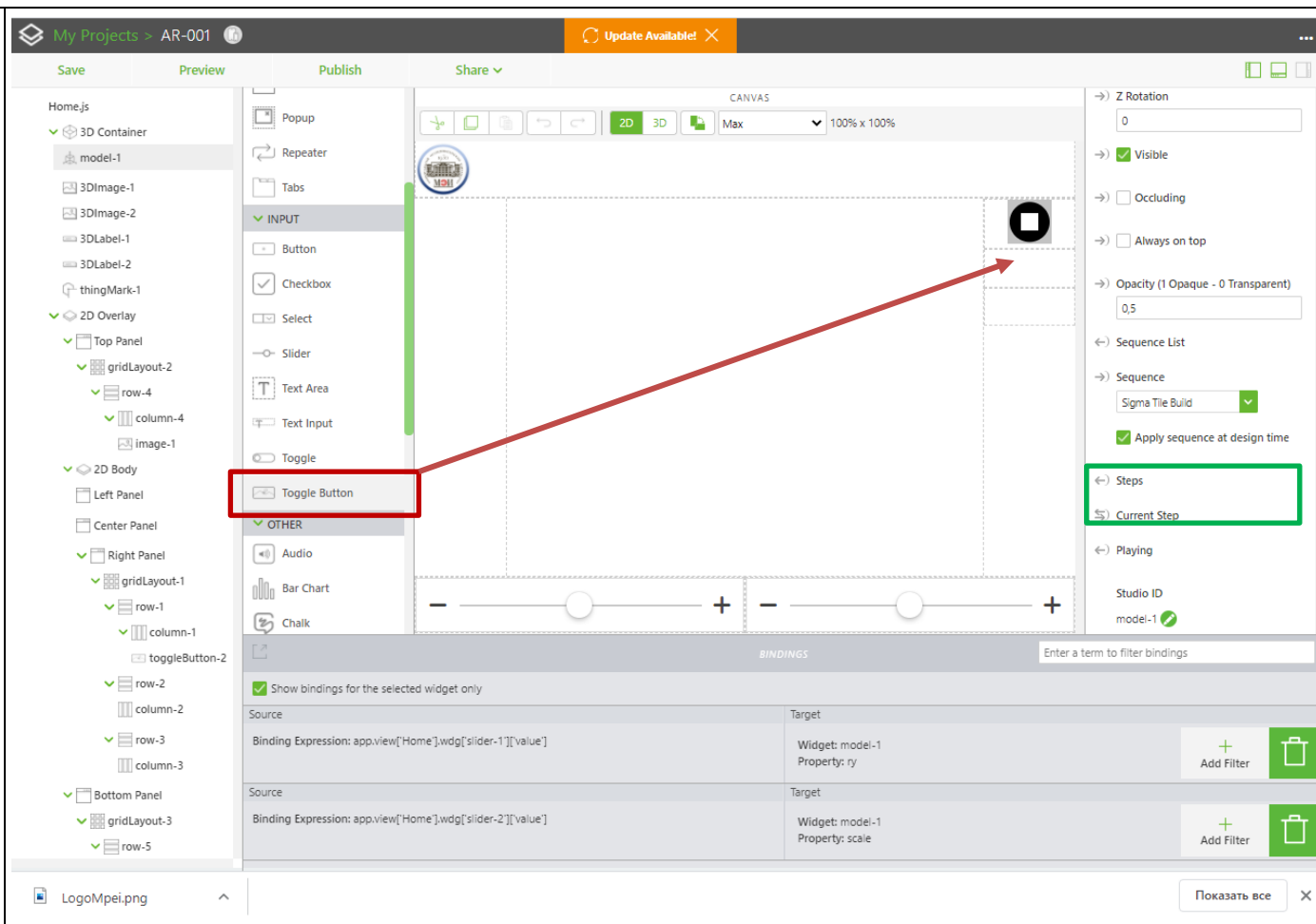
Вначале определим наличие **sequences** в архиве модели. Для этого в зоне «**Компоненты**» выбираем нашу модель, а в зоне «**Свойства компонентов**» найдем свойство **sequences**. В открывшемся списке, если он не пуст, выберем **нужную нам sequence**. В нашем примере эта последовательность называется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



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**.

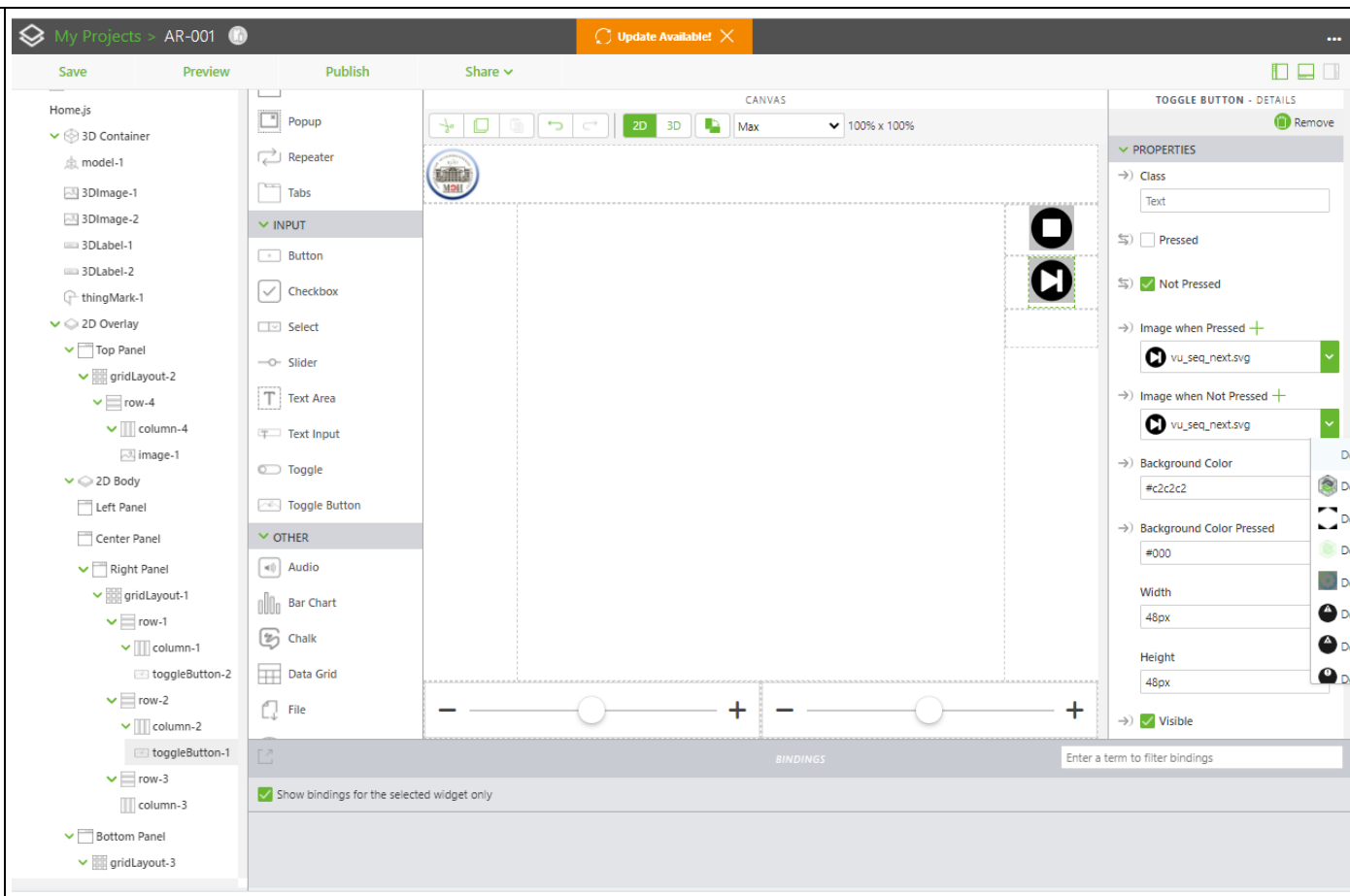
Шаг 68.

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».



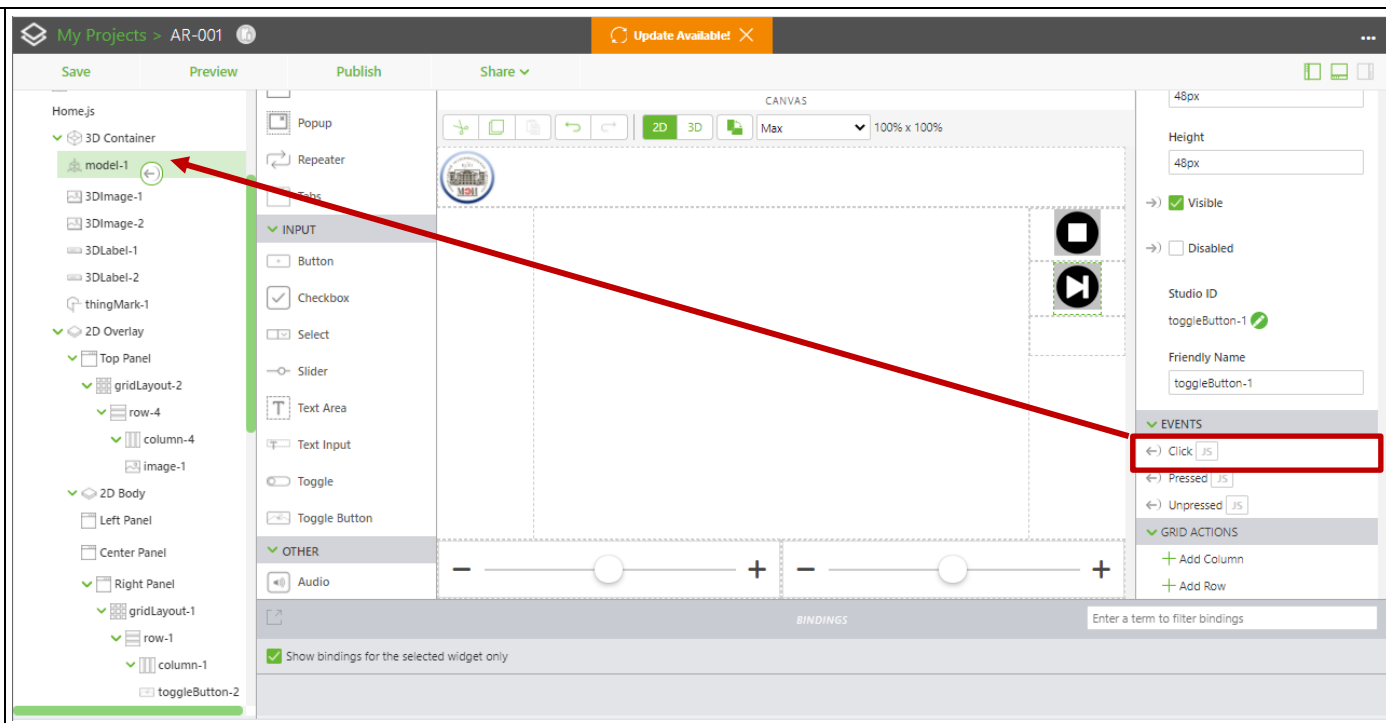
Шаг 69.

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.

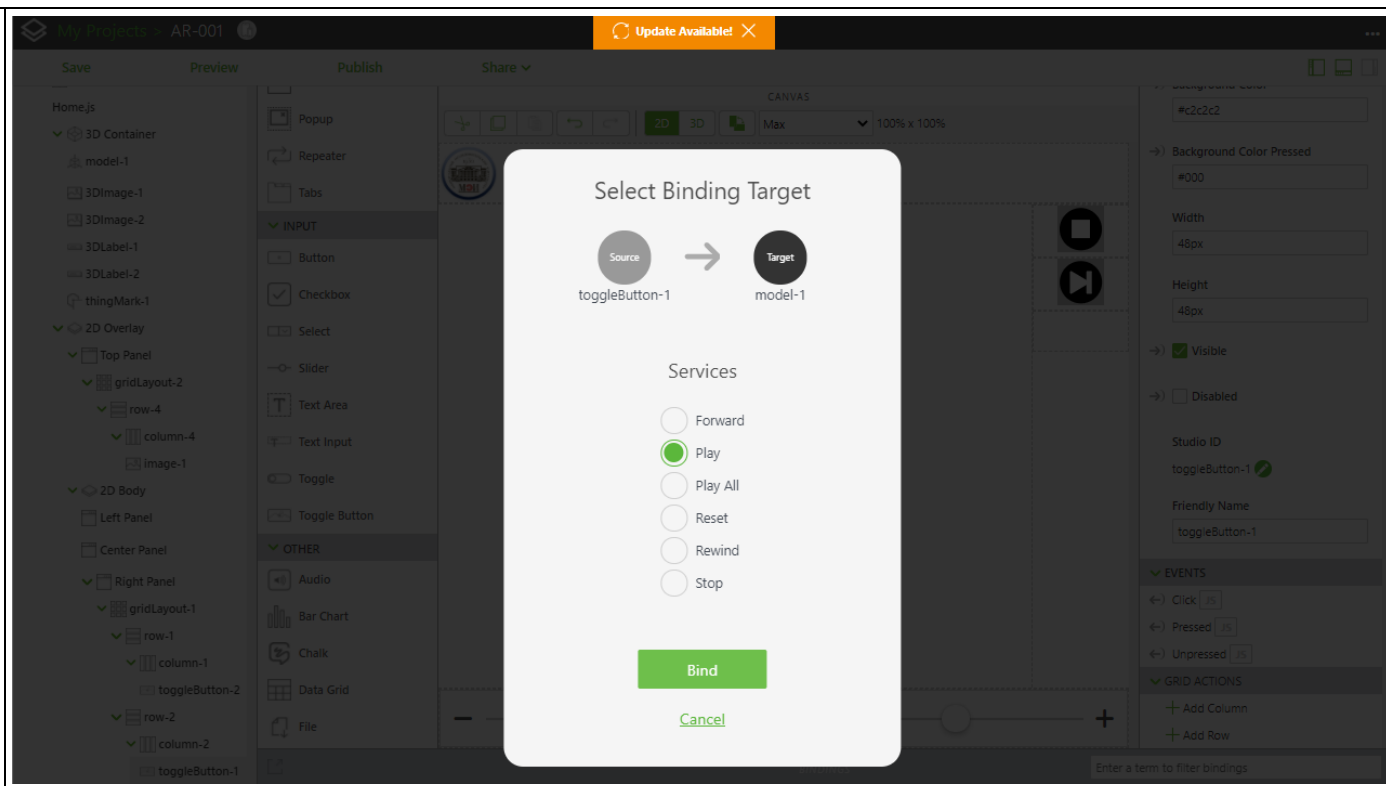


Шаг 71.

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**.

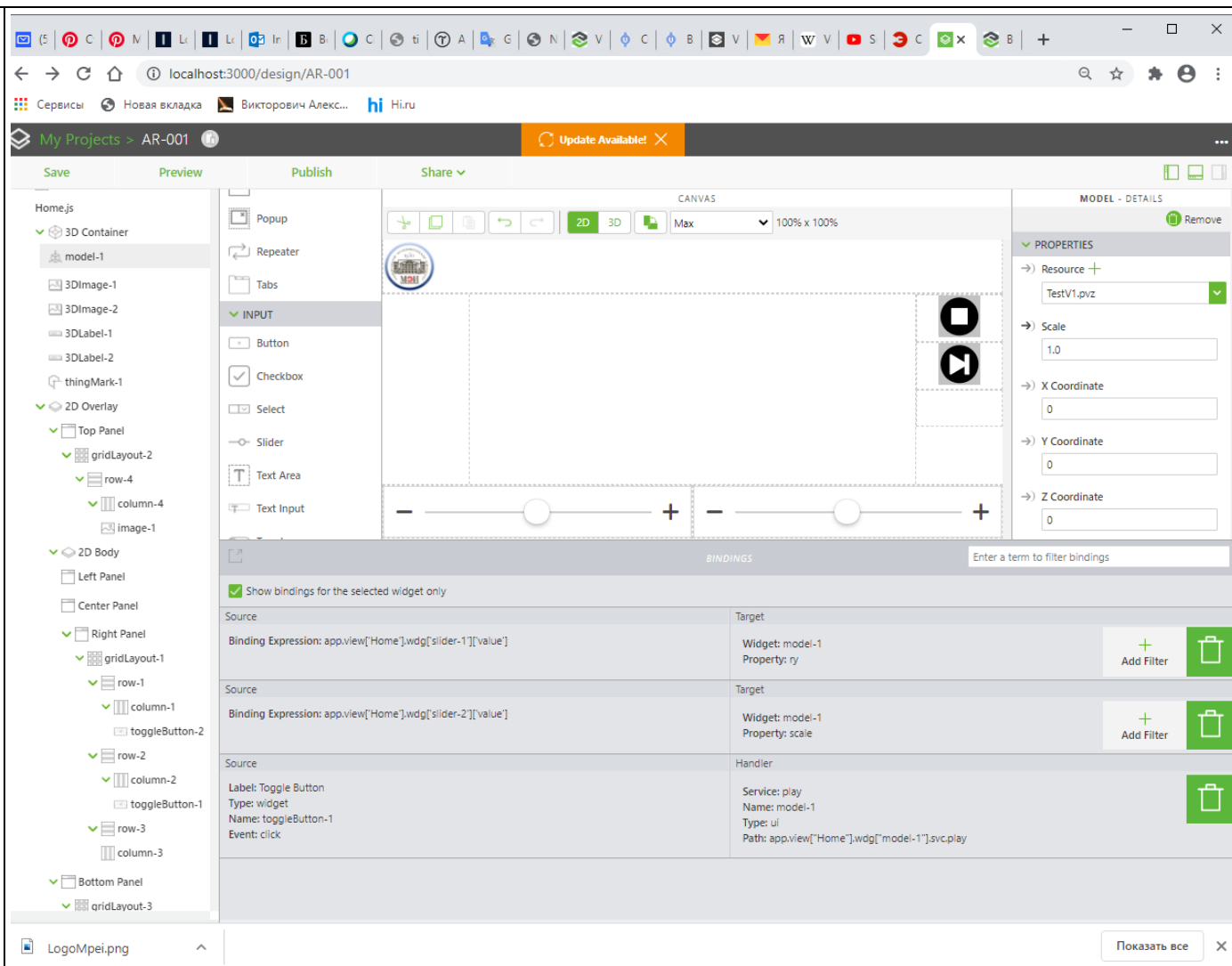
В нашем случае это должен быть **СЕРВИС Play**. Обратите внимание — **СЕРВИС Play All** мы не выбираем потому, что он реализует полный просмотр последовательности без разделения на шаги.

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



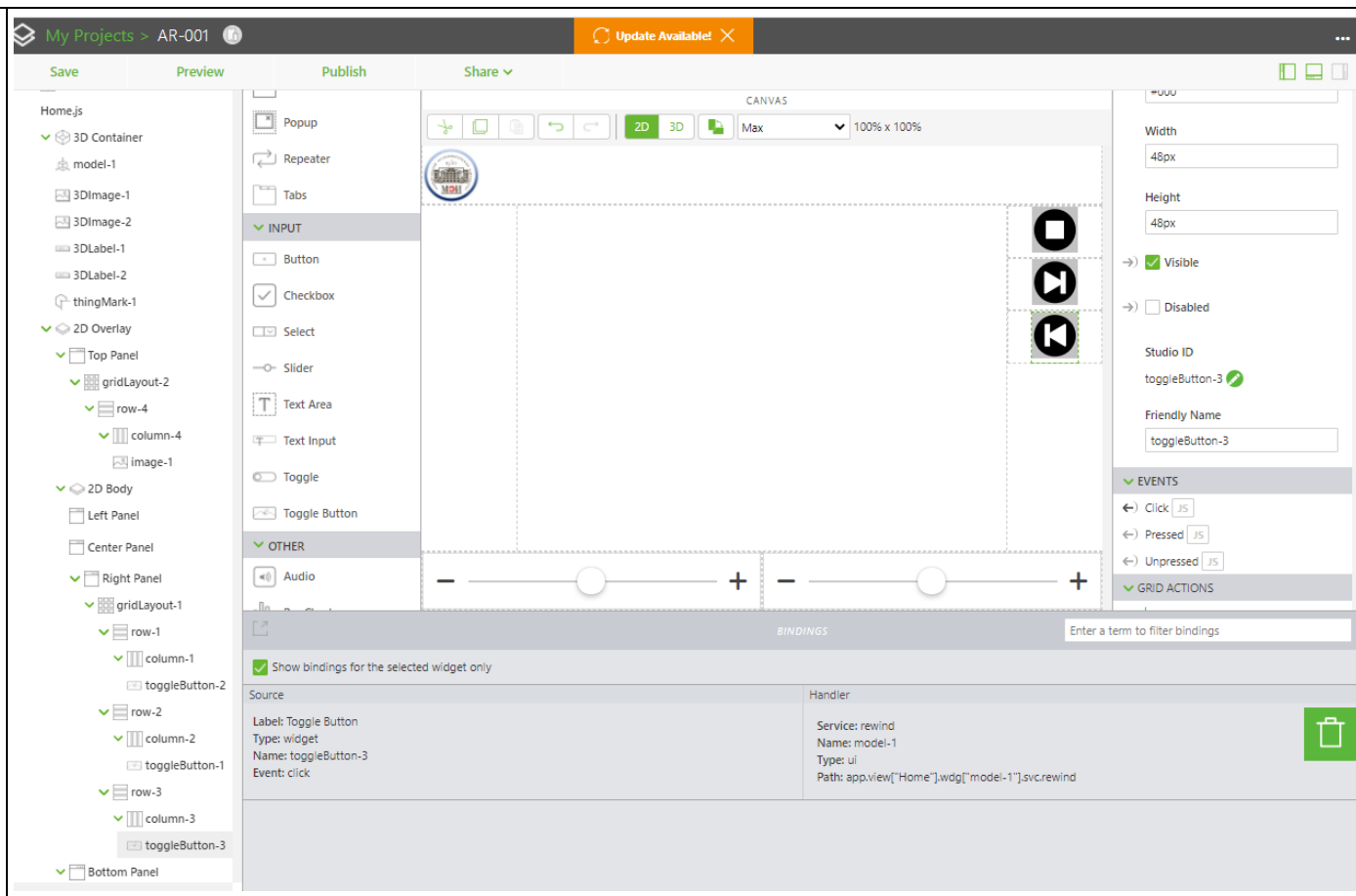
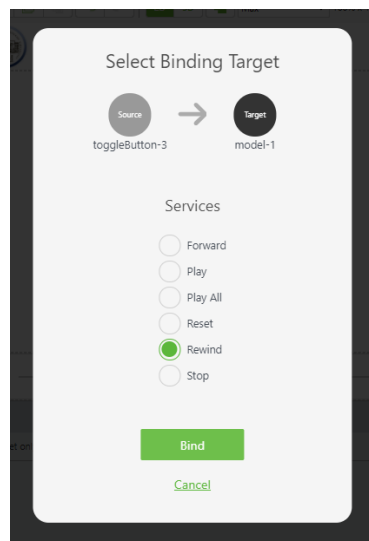
Шаг 72.

Обратите внимание – в результате количество связей у модели постоянно увеличивается. Это можно проконтролировать в зоне «Связи» для выбранной модели в зоне «Компоненты». Связи можно удалять, переделывать и т.д.



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**):



Шаг 74.

Проверяем полученный результат в режиме Preview.



Откат на шаг назад →

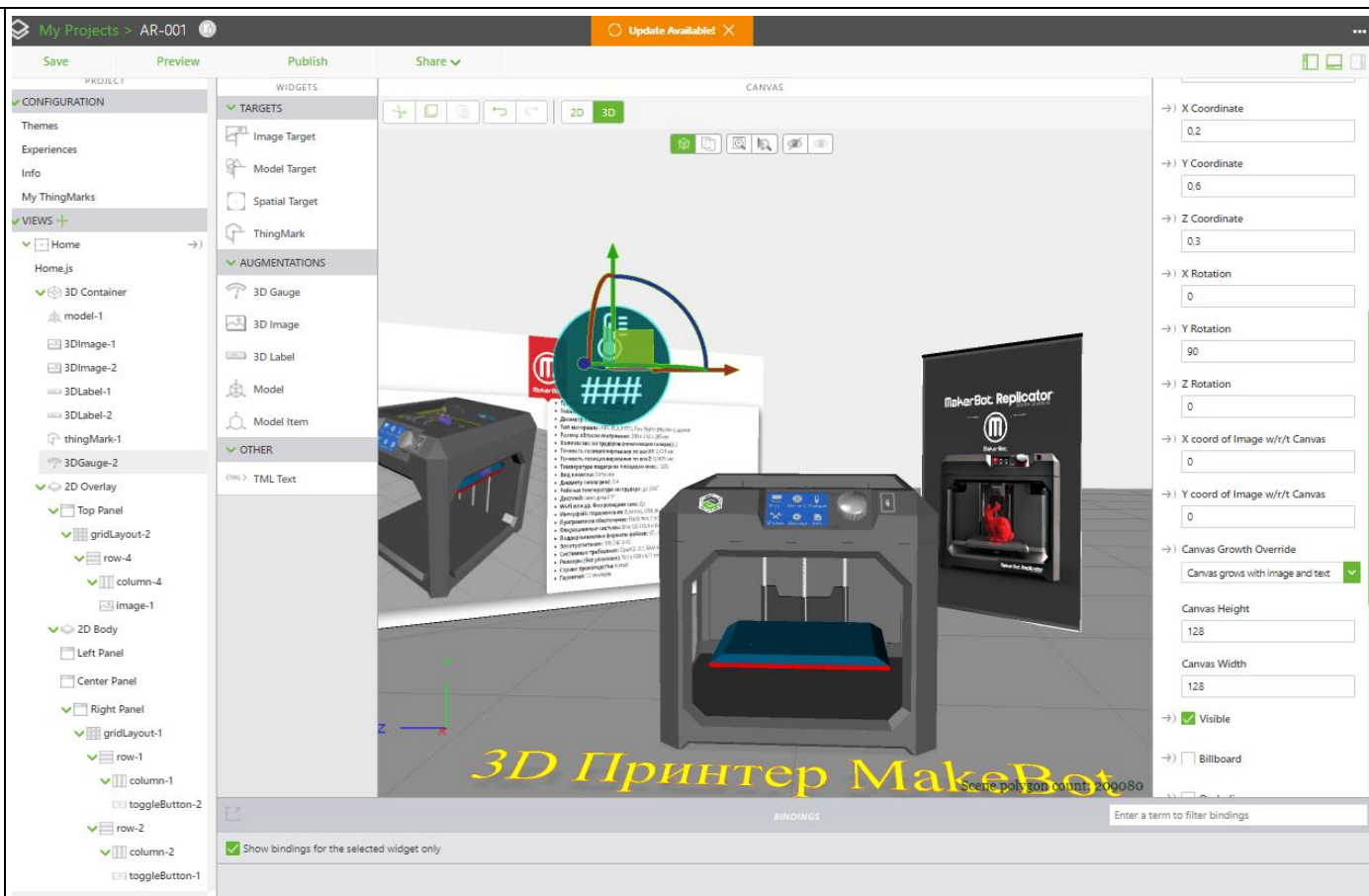


Шаг 75.

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 следующей 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** режиме реального времени.



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

МОЖНО РАЗМЕСТИТЬ ТОЛЬКО ОДИН 3D GAUGE



Шаг 77.

Проверяем полученный результат в режиме Preview.



4. Практическое задание

Шаг 1. Создайте Project – AR-Проект (Experience) . Имя «_____» Тип таргетинга - ThingMark	
Шаг 2. Разместите в проекте 3D-модель (архив .pvz)	
Шаг 3. Используйте в проекте 3D-виджеты : • 3D-Image • 3D-Label	
Шаг 4. Используйте в проекте 2D-виджеты : • Grid (Grid Layout) • Toggle Button для управления свойствами 3D-виджетов; • Slider для управления свойствами 3D-виджетов; • Image для размещения логотипов разработчика.	
Шаг 5. Выполните превью проекта, убедитесь в правильном выполнении всех предыдущих шагов.	
Шаг 6. Продемонстрируйте результат – просмотр проекта средствами Preview (в рамках последующих ЛР после полноценной регистрации)	

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