



ЛАБОРАТОРНАЯ РАБОТА №3

доп. 2. Vuforia Studio: Создание приложения дополненной реальности. Разработка многоуровневого AR-Приложения, снабженного меню для навигации между сценами (часть II из III)

Оглавление

1. Введение.....	2
2. Разработка Иерархического многоуровневого меню.....	6
3. Практическое задание	52

Дата	Автор	Версия	Описание
25.11.2020	Пирогова М.А.	1	Разработка Дополнения 2. к ЛР № 3

1. Введение

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.

AR-приложения, разработанные в **Vuforia Studio**, размещаются в «облаке» и загружаются на устройство пользователя только в момент использования. Какие ограничения на размеры (объем) **AR**-приложения в этом случае имеют значение? Прежде всего, сам размер публикации, то есть тот размер программы, которую необходимо разместить на сервере **TWx ES** и для передачи которой на устройство пользователя необходимо использовать ресурсы каналов интернета. Однако реальная промышленная потребность в **AR**-приложениях рабочего места подразумевает принципиально большие размеры самих приложений. **Так например, если AR-приложение разрабатывается как вспомогательный информационный ресурс выполнения рабочих или сервисных операций (сборка, ремонтные операции, инструкции для специалистов соответствующего профиля, инструкции по эксплуатации, передача опыта внутри корпорации и пр.), то, как правило, AR-поддержка требуется для сложных узлов, агрегатов или приспособлений (сложная большая 3D-модель сборки), и при этом такое приложение может включать в себя несколько (десятков) шагов - элементарных действий и большое число справочной информации.**

Соответственно, размер **AR**-приложения, в котором пошагово рассмотрены операции по работе со структурно сложным **3D**-объектом при этом становится значительным и в отдельных случаях может требовать существенных (временных) затрат уже только на загрузку. Удобство применения **AR**-приложения в таком случае становится весьма сомнительным, поскольку вместо ускорения времени выполнения рабочих операций применение **AR**-приложения вносит задержки тем значительнее, чем сложнее (а, следовательно подробнее) само это **AR**-приложение.

Выходом из такой ситуации может служить декомпозиция одного сколь угодно большого («тяжелого») **AR**-приложения на несколько связанных между собой **AR**-приложений в рамках одного **AR**-проекта, разрабатываемого в **Vuforia Studio**.

При этом хотелось бы иметь такую структуру основного **AR**-приложения (проекта), при которой загрузка его из облака могла бы происходить не сразу (всего проекта целиком), а «по частям», отдельно для каждого из связанных в проекте [под-]**AR**-приложений. Таким образом, скорость загрузки **AR**-контента для выполнения отдельных шагов поддерживаемой приложением деятельности и общая эффективность применения **AR**-приложения на рабочем месте возрастает, нагрузка на ресурсы устройства пользователя снижается, и, как следствие, удобство применения **технологии дополненной реальности** для конечного пользователя становится очевидной.

С точки зрения разработки в **Vuforia Studio** разбиение проекта на связанные **AR**-приложения означает наличие не одной сцены, а нескольких «сцен».

В процессе выполнения всех предыдущих ЛР обучающие разрабатывали в **Vuforia Studio** проекты **AR**, которые принципиально состояли только из одной «сцены», то есть, проект не разбивался на несколько связанных [под-]AR-приложений – «сцен».

При этом вся структура контента - модель, таргет, виджеты **2D** и **3D**, т.е. все наполнение контента проекта - формировалось в одной сцене, объекте типа **VIEW** (вспомним структуру редактора **Vuforia Studio**). Для рационализации, о которой говорилось выше, необходимо разбить большой и сложный проект на какие-то осмысленные части, каждую из которых размещать со своей структурой (**3D модель+таргет+2D и 3D виджеты**) в отдельном **VIEW** («сцене») проекта. Проект таким образом становится контейнером («сценарием», последовательностью «сцен») для нескольких «сцен» (структурных объектов типа **VIEW**), каждая из которых публикуется на сервере **TWx ES** в один прием, а востребована может быть конечным пользователем (**Vuforia View-пользователь**) «по частям», отдельно для каждой из «сцен», что снижет, как и предполагалось, нагрузки и ресурсные требования.

Разработчику **Приложения ДР** необходимо предусмотреть в случае создания такого мультисценарного **AR**-проекта добавление в структуру Проекта **элементов управления передвижением по сценарию, то есть встроить в AR-проект элементы навигации по его структуре.**

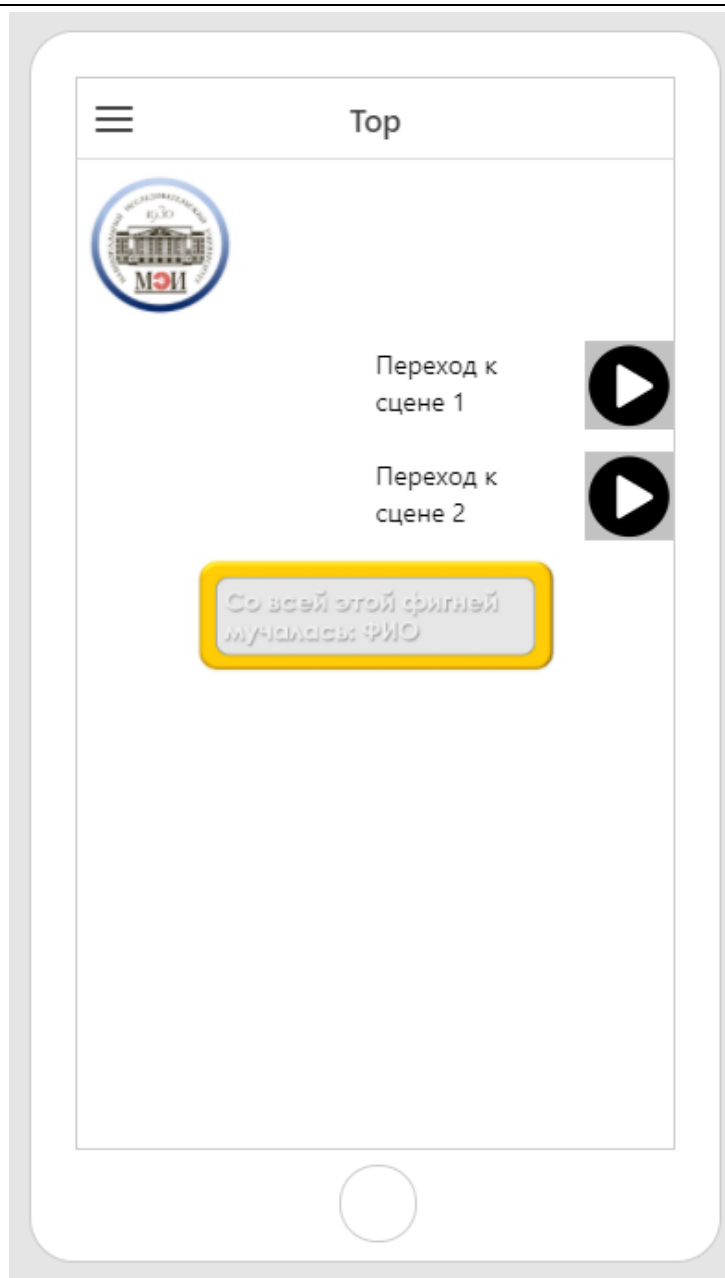
При этом разработчик в **Vuforia Studio** сам может определять такую организацию расстановки элементов навигации по сценарию, которая наиболее полным образом отвечает задачам проекта («сценарному плану»).

В данном **дополнительном пункте ЛР №3** требуется разработать **AR**-Проект, который состоит из нескольких пошаговых описаний различных операций и дополнительно включает **Титульную страницу**. Контентом такой страницы может быть, например, начальное описание Приложения, логотип компании, сведения о разработчиках т.п. При этом каждая из сцен разрабатываемого Проекта - объектов **VIEW, сцен** - и заглавная страница, и оба **AR**-Приложения – также должна включать элементы управления навигацией по сценарию **AR**-Приложения.

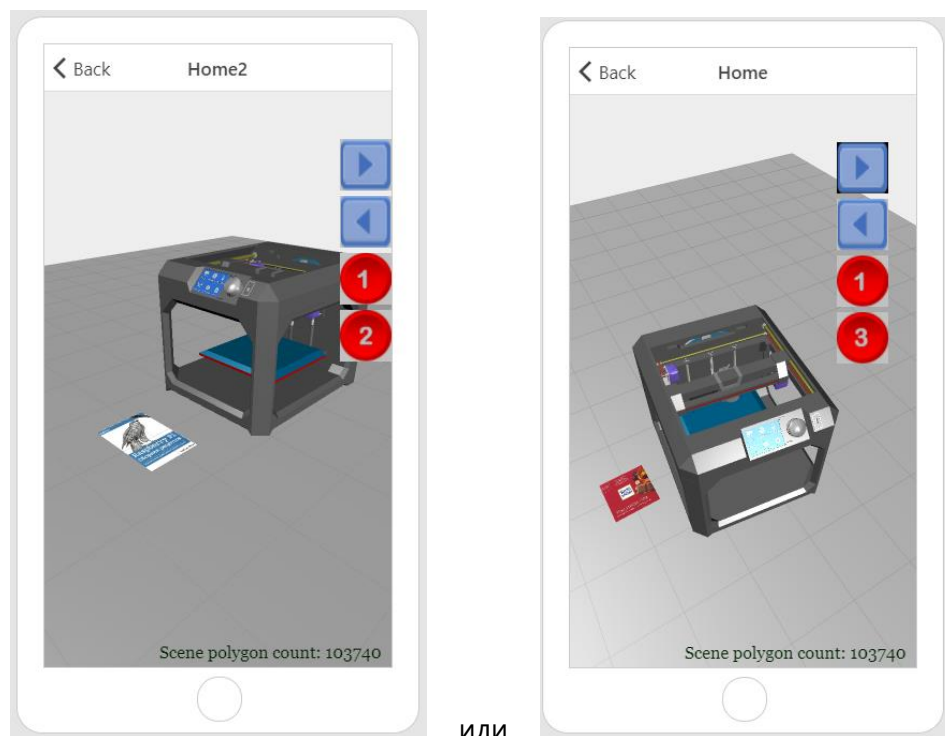
Ниже приведен возможный внешний вид **Титульной страницы** такого **Проекта**. Здесь размещается информация о разработчике, приводится логотип организации, а также корневое меню, инициирующее переход к остальным страницам «стекла». В разрабатываемом **AR**-Приложении, основанном на **Проекте AR**, созданном в ЛР №2, к уже созданной для базовой модели **3D-Принтера MakerBot Replicator 2** сцене предлагается добавить еще одну. В новой сцене к модели будут привязаны анимационные последовательности по замене у принтера рабочего органа (головка) – **Head Removal**.

Корневое меню должно состоять из кнопок навигации с пометками их предназначения - переход при нажатии к соответствующей сцене.

Разместить их предлагается с привязкой по правой границе **2D** Канвы («Стекла»).



Каждая из страниц, соответствующих содержательным сценам Проекта (отдельным [под-]AR-Приложениям, образующим его структуру и реализующим сценарий), имеет свое собственное, уже ставшее привычным в этом **Лабораторном Практикуме**, оформление элементами управления контентом внутри сцены (скроллеры, кнопки запуска анимированных последовательностей визуализации технологических процессов, Связанных с моделью, кнопки включения справочной информации и выносок и пр.). Это оформление также дополняется элементами управления **навигацией по Проекту** – переходы между сценами и возврат на **Титульную страницу**. Например:



ВАЖНО!! Поскольку для каждой из сцен Проекта может быть задействован свой набор компонент контента, **таргетирование**, необходимое для позиционирования этого контента в реальном окружении, также должно быть индивидуальным (различным) для каждой из сцен. Структура **AR-Проекта**, разрабатываемого в **Vuforia Studio** с использованием таргетирования по методу **ThingMark**, стандартно допускает использование только одного экземпляра **ThingMark** на весь **AR-проект**. Это означает, что для создания многоуровневых сценариев **AR-проектов** использование метода **ThingMark** не подходит. В данной лабораторной работе для индивидуального таргетирования каждой сцены будем применять метод таргетирования **Image Target**, для которого нет ограничений применения, типичных для таргетов типа **ThingMark**.

Т.о. в данном пункте ЛР №3 рассматривается пример создания мультисценарного мультитаргетного Приложения AR.

2. Разработка Иерархического многоуровневого плоского меню

Как и в предыдущих Лабораторных работах, будем считать, что тренировочный сервер **ThingWorx Experience Service (TWx ES)** находится по адресу (получить у преподавателя):

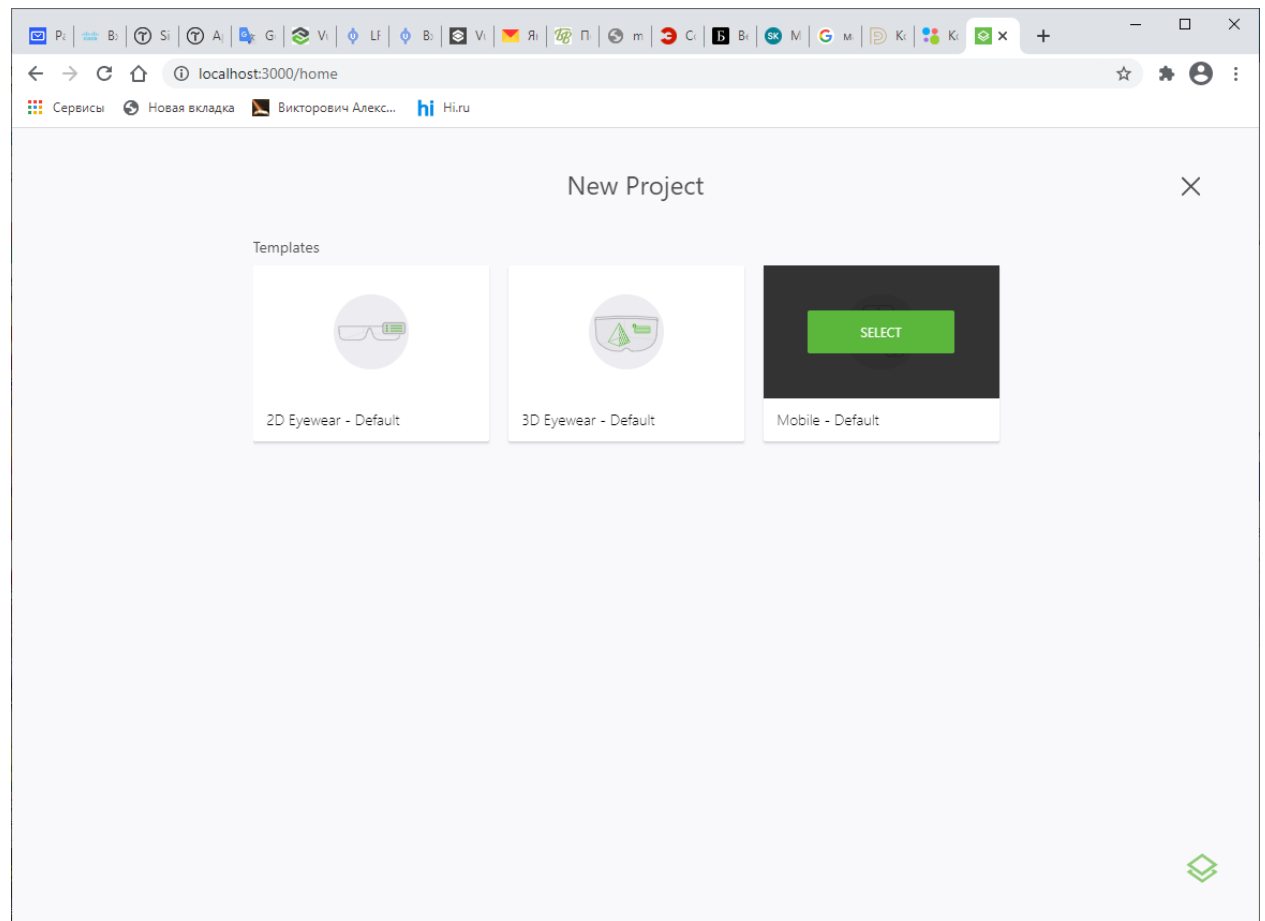
<http://rrr.kkk.com:2019>

Шаг 1.

В данной ЛР предлагается создать новый **Проект**. Подробное описание всех шагов, которые необходимо выполнить для этого, приведены в ЛР №2 →

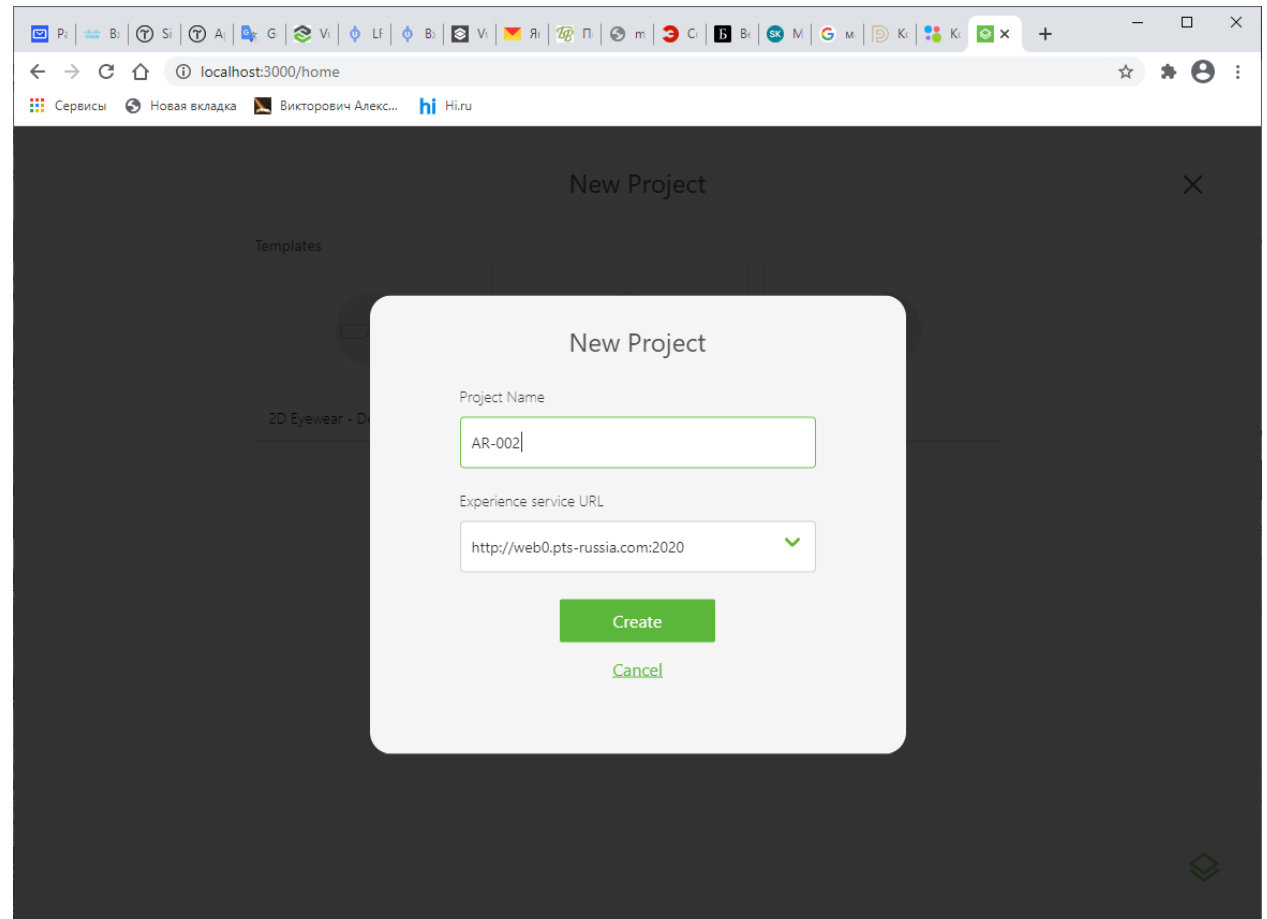
http://a0601.narod.ru/LR02_ARforIIoT.pdf

Новый Проект будет упрощен по сравнению с ранее разрабатываемыми и модифицируемыми. Возможен другой вариант → дублирование и модификация уже созданного проекта (см. ЛР №3), однако в любом случае в этом Проекте придется добавлять новую сцену.



Шаг 2.

Присвоим новому Проекту имя **AR-002**. Как уже отмечалось выше, работа над данным дополнением к ЛР №3 ведется с использованием тренировочного сервера **TWx ES**, на котором каждый обучающийся уже имеет собственный аккаунт, а сам сервер размещен по адресу (см. выше) <http://rrr.kkk.com:2019>

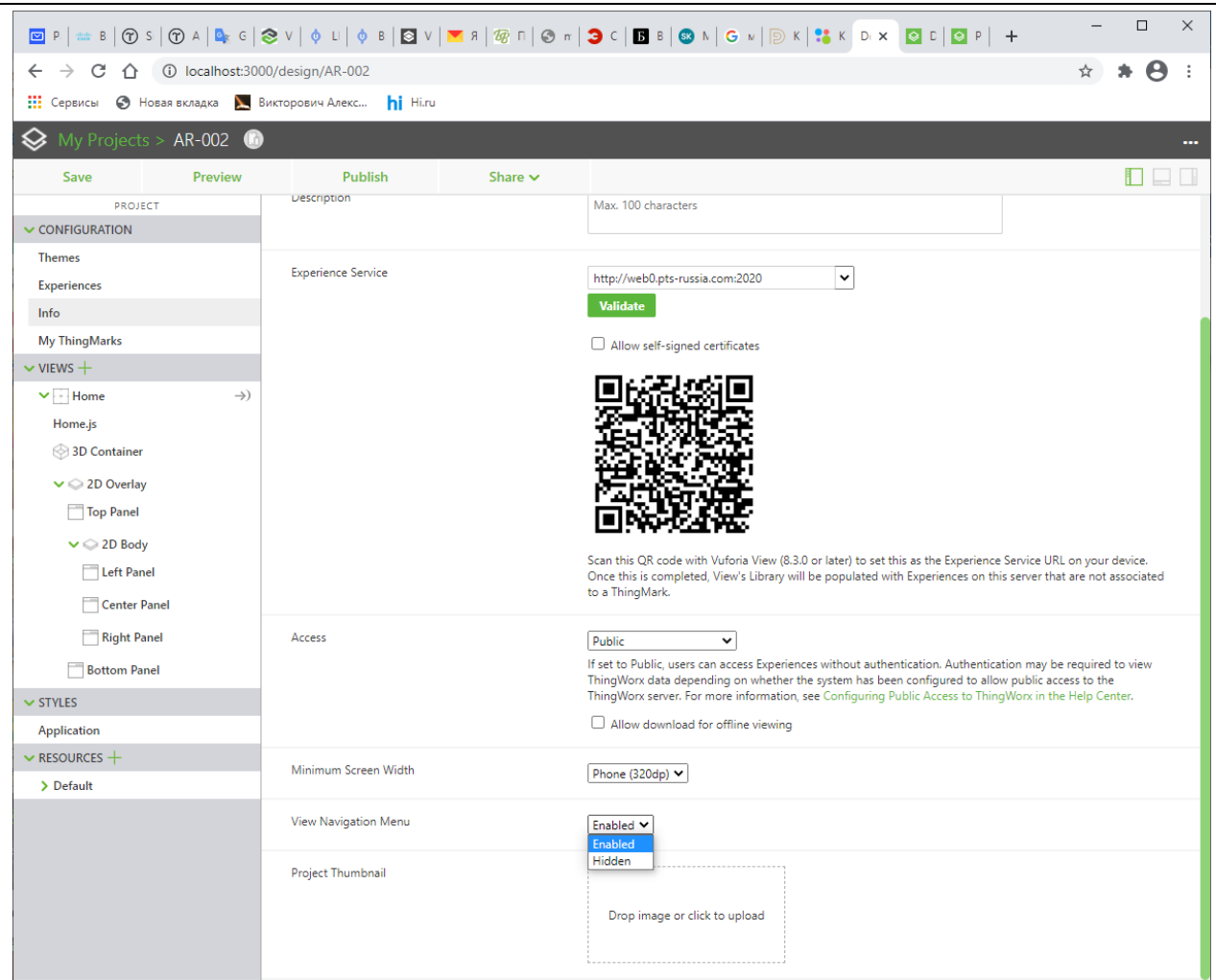


Шаг 3.

После входа в Проект выполняем необходимые проверки в поле редактора **CONFIGURATION** → **Info**: выбрана позиция **Public** в поле **Access**.

ВАЖНО!!! – **Vuforia View** предоставляет т.н. меню по умолчанию (базовое меню) – **Navigation Menu**, имеющее стандартное для современного стиля **GUI** оформление и кнопки управления, позволяющие осуществлять базовые функции навигации между сценами и возврата к началу **AR-Приложения**. На начальном этапе разработки многоуровневого меню, предусмотренного сценарием **AR-Приложения**, рекомендуется сохранить доступность меню по умолчанию:

View Navigation Menu → **Enabled**. После разработки собственного меню **Navigation Menu Vuforia View** можно отключить (**Hidden**)



Шаг 4.

Обратите внимание – В разделе **Edit Experience** указывается, какая Сцена (View) Приложения становится первой при старте **Experience** →

CONFIGURATION → **Experiences** → **Initial View**. Пока (и по умолчанию) таковой является пустая сцена **Home (шаблон)**, однако далее мы изменим это состояние, т.к. хотим, чтобы при старте разрабатываемого **AR-Приложения** пользователь попадал на **Титульную** страницу, которую только предстоит разработать.

Здесь же можно задать **Title** для просмотрщика **Vuforia Studio**, заполнить поле **Description**, а в разделе **CONFIGURATION** → **Info** добавить изображение для тумбнейла, но мы традиционно сделаем это в конце, перед публикацией Проекта.

The screenshot shows the Vuforia Studio web interface in a browser. The address bar indicates the URL is localhost:3000/design/AR-002. The interface is divided into a left sidebar and a main content area. The sidebar, under the 'PROJECT' heading, contains several sections: 'CONFIGURATION' with sub-items 'Themes', 'Experiences', 'Info', and 'My ThingMarks'; 'VIEWS' with a plus sign, containing 'Home' (selected with a right arrow), 'Home.js', '3D Container', '2D Overlay', 'Top Panel', '2D Body', 'Left Panel', 'Center Panel', 'Right Panel', and 'Bottom Panel'; 'STYLES' with 'Application'; and 'RESOURCES' with 'Default'. The main content area is titled 'Edit Experiences' and features a prominent green 'New Experience' button. Below this is a form for editing an experience. The form includes: 'ThingMark Association' (dropdown menu set to 'ThingMarks'), 'Title' (text input field containing 'AR-002'), 'ThingMark' (dropdown menu with the prompt 'Enter or select a ThingMark'), 'Initial View' (dropdown menu with 'Home' selected and a blue highlight), and 'Description' (text area). A red 'Delete' button is located at the bottom right of the form.

Шаг 5.

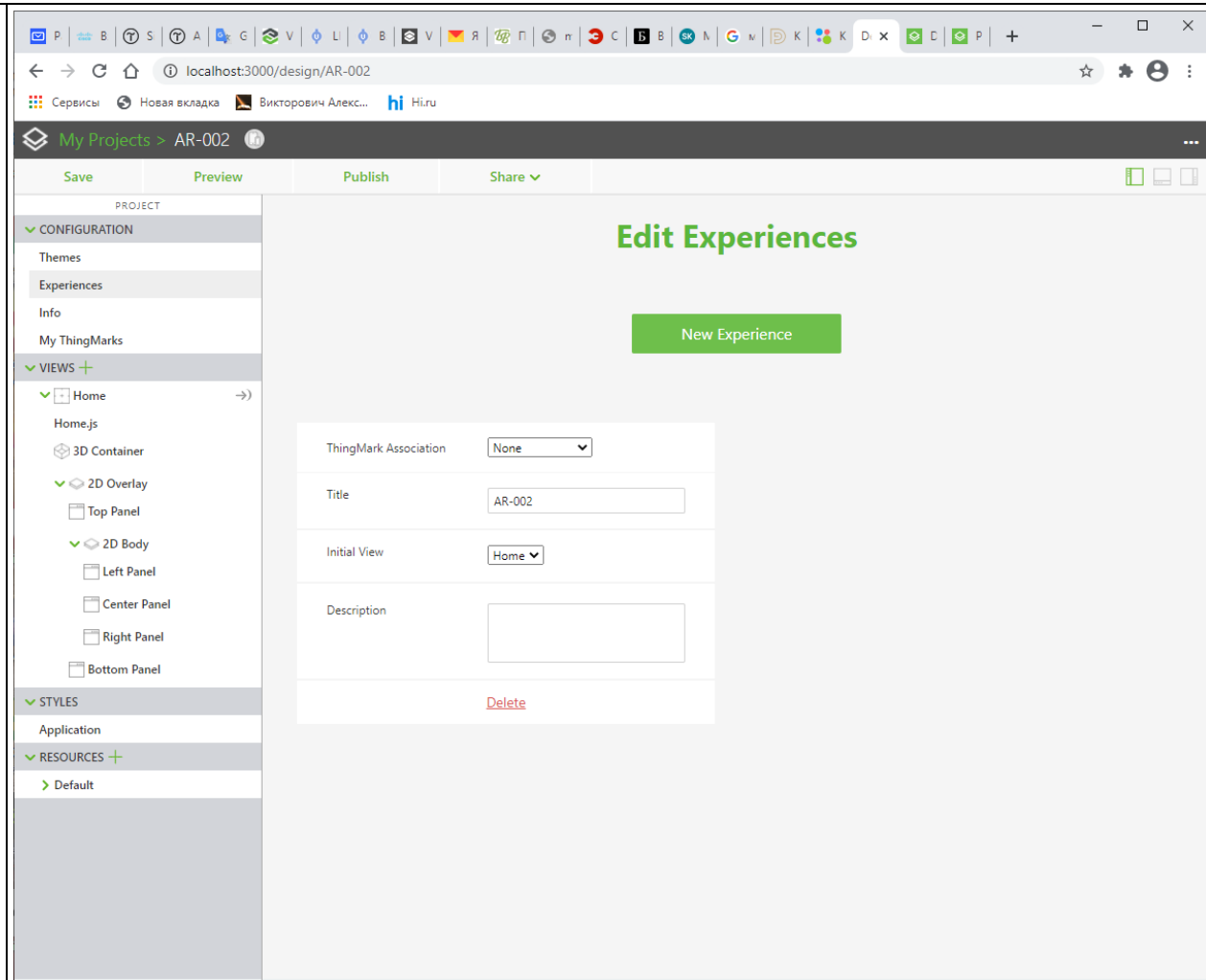
Не забудьте правильно выбрать для Проекта типа таргетирования, выбрав в поле **ThingMark Association** позицию **None** (см. пояснения во введении к данному описанию).

The screenshot displays the Vuforia Studio web interface for editing AR experiences. The browser address bar indicates the URL is localhost:3000/design/AR-002. The interface is divided into a left sidebar and a main content area. The sidebar shows a project tree with 'Home' selected under 'VIEWS'. The main content area is titled 'Edit Experiences' and features a 'New Experience' button. Below this, there is a form for editing an experience. The 'ThingMark Association' dropdown menu is open, showing options: 'ThingMarks', 'Thing Template', 'ThingMarks', and 'None', with 'None' selected. Other form fields include 'Title', 'ThingMark', 'Initial View' (set to 'Home'), and 'Description'.

Шаг 6.

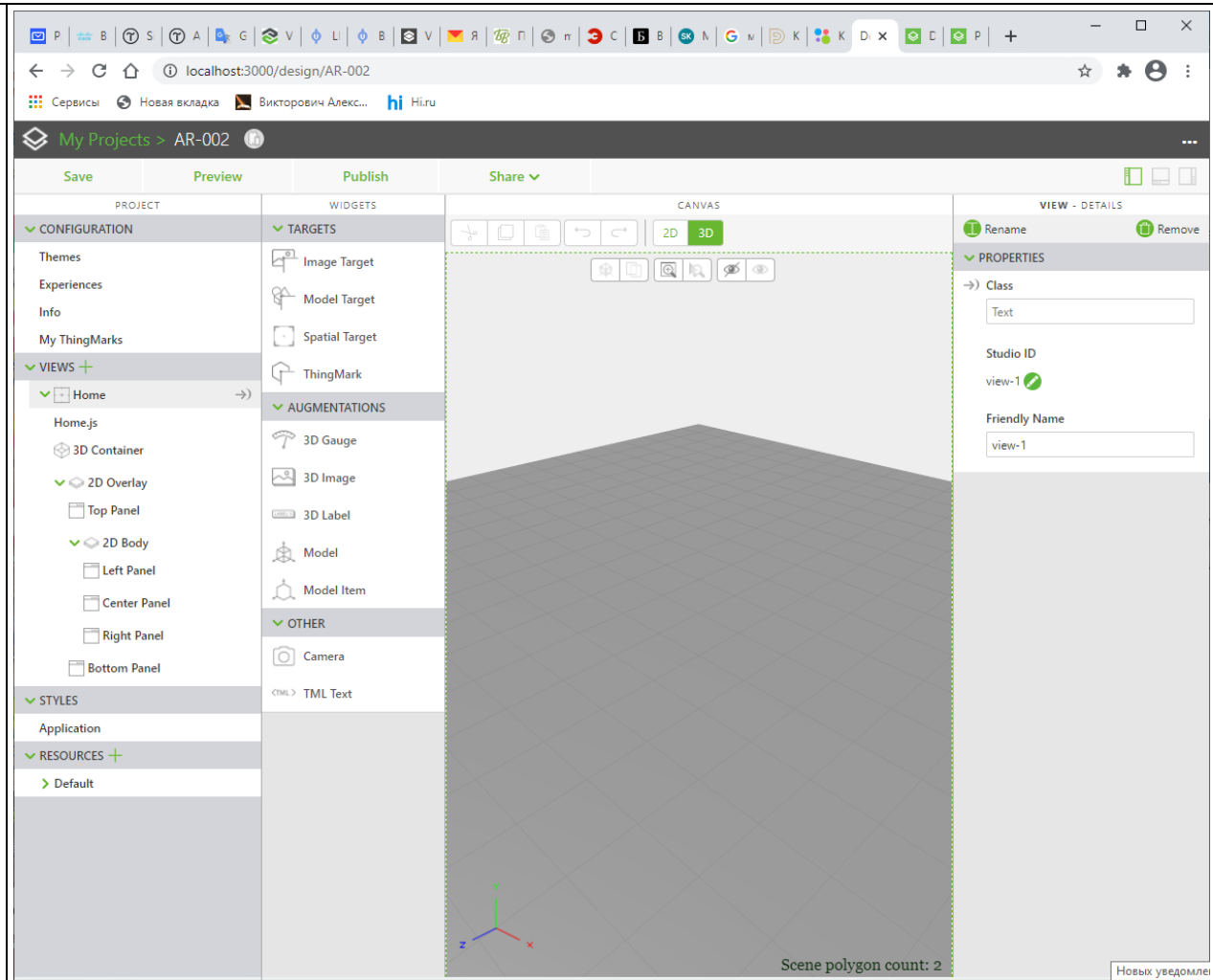
Переходим к разработке Проекта, которую начинаем с разработки сцен контента. Как было указано выше, первой страницей в данном Проекте д.б. **Титульная**, содержащая в том числе и корневое меню навигации по разрабатываемым сценам. **Титульная** страница (сцена → **IEWS**) является плоской, формироваться должна не в **3D**, а в **2D – Канве**.

Но у нас пока во **IEWS** нет ничего другого, кроме сцены-шаблона **Home**, куда мы и переходим, чтобы начать формировать новую сцену → **VIEW** → **Home**.



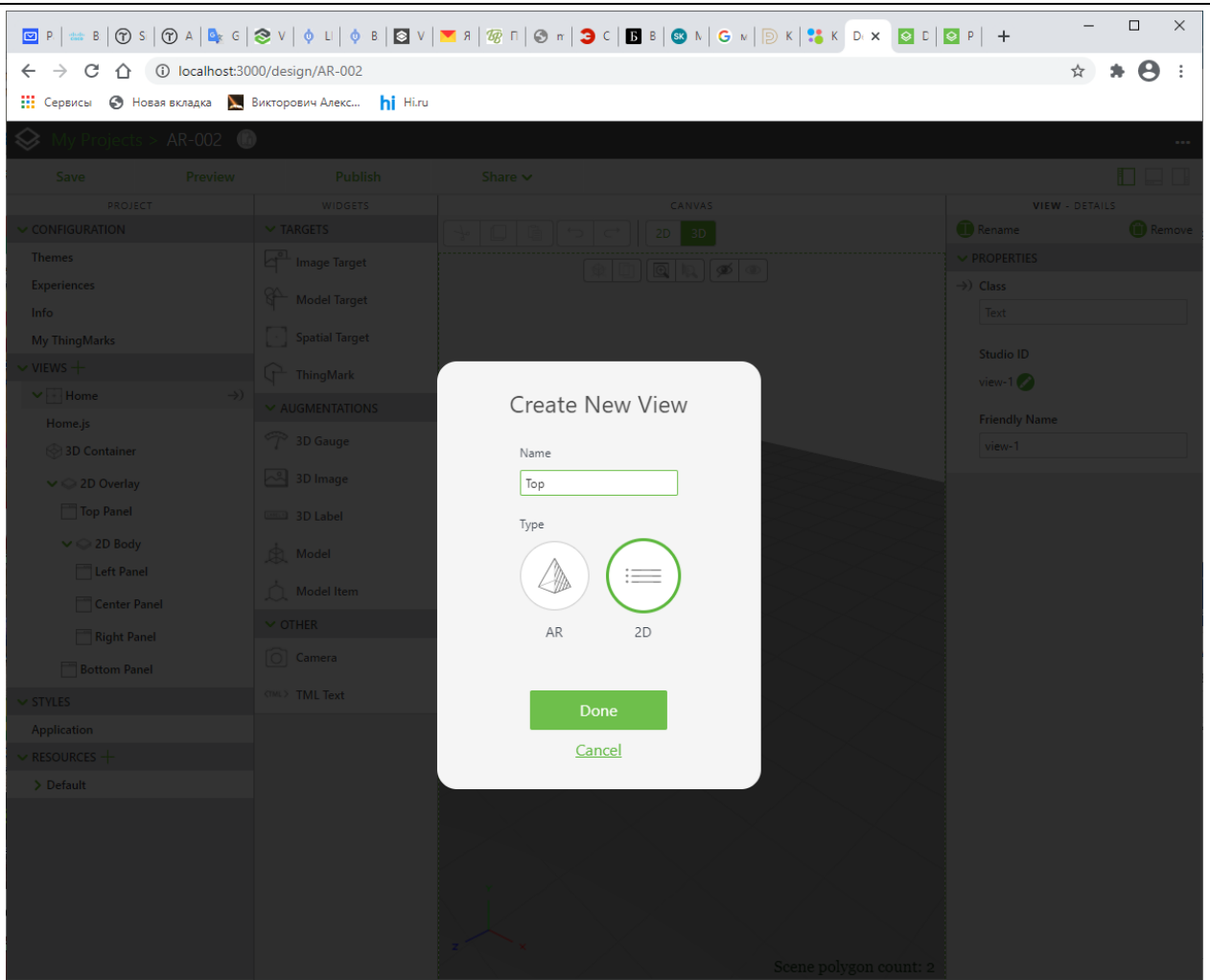
Шаг 7.

Шаблон (пустой) **Home** – это **3D** – сцена (канва), не то, что нам нужно, поэтому из под шаблона **Home** выбираем **VIEW+** → откроется меню для выбора новой сцены, которую предлагается в нашем Проекте назвать **Тор (<имя>: Тор)** для плоской сцены.



Шаг 8.

VIEW + → в открывшемся меню для создания новой сцены (присваиваем ей имя: **Top**) и выбираем канву разработки – **2D** → по нашему сценарию эта сцена должна быть плоской, верхним уровнем многоуровневого меню, должна содержать информацию о разработчике и меню навигации между сценами Проекта – корневое меню.

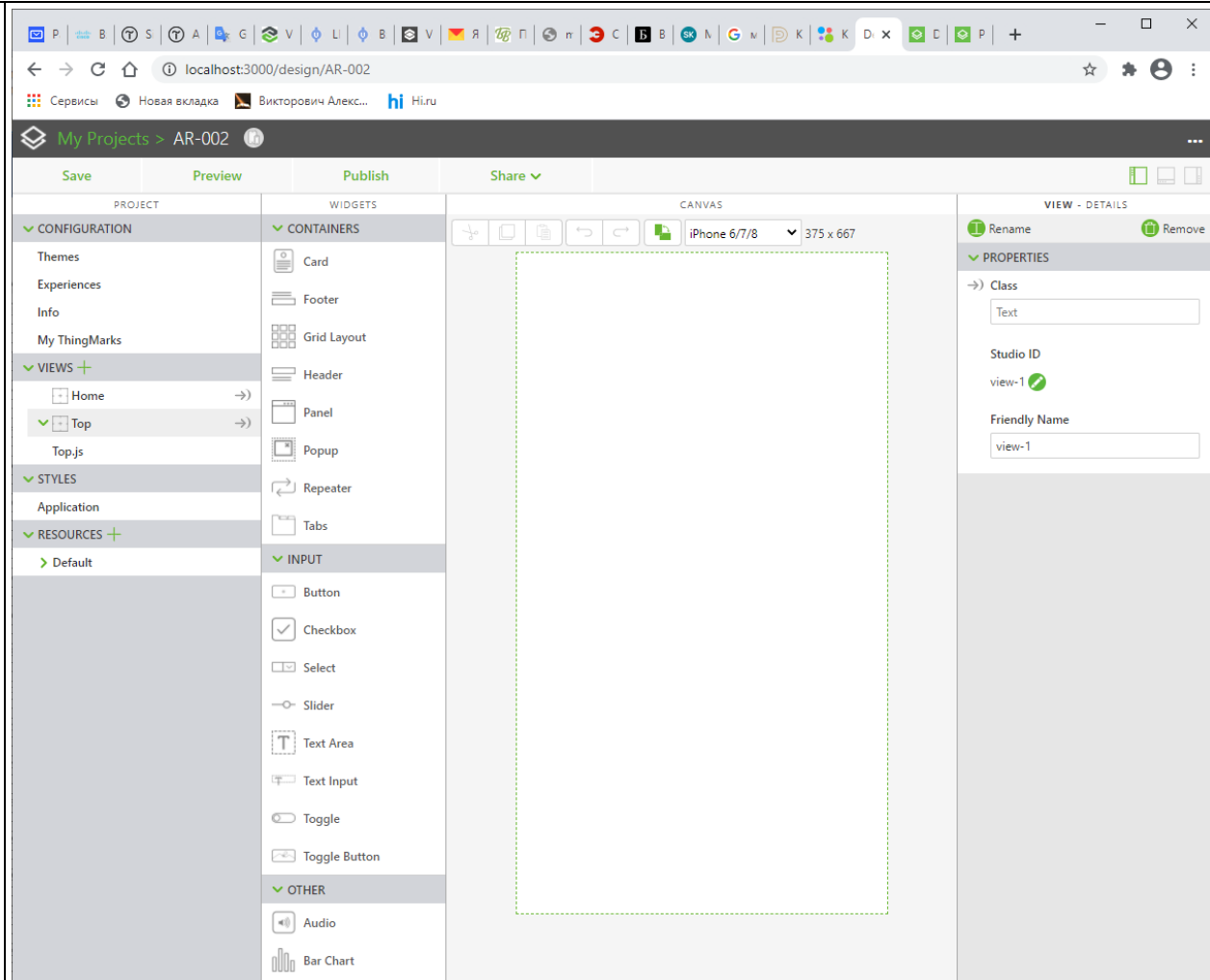


Шаг 9.

В этой сцене не будет **3D** контента, сразу открывается никак не размеченное пространство. Что несколько непривычно по сравнению с предыдущими опытами. И в этом пространстве нужно разместить плоские картинки. Подписи и управляющие элементы интерфейса и навигации по сценарию, т.е. переход к другим сценам, которых пока нет.

В первую очередь это пространство нужно разметить под это наполнение. Для разметки будем использовать контейнеры (из поля **WIDGETS**), пытаясь (например) воспроизвести разметку «стекла», на которую мы выходили при переключении в сценах между **2D** → **← 3D** канвой в первой ЛР. Например: панели снизу и сверху → в данной сцене – добавим **Panel** (сверху), затем - **Grid Layout** и опять **Panel** (как бы снизу), а между ними – в **Grid Layout** - добавляем несколько рядов, за счет чего вторая **Panel** смещается вниз. В рядах, уже размеченных под колонки (хотя бы одну), будем помещать элементы управления – кнопки.

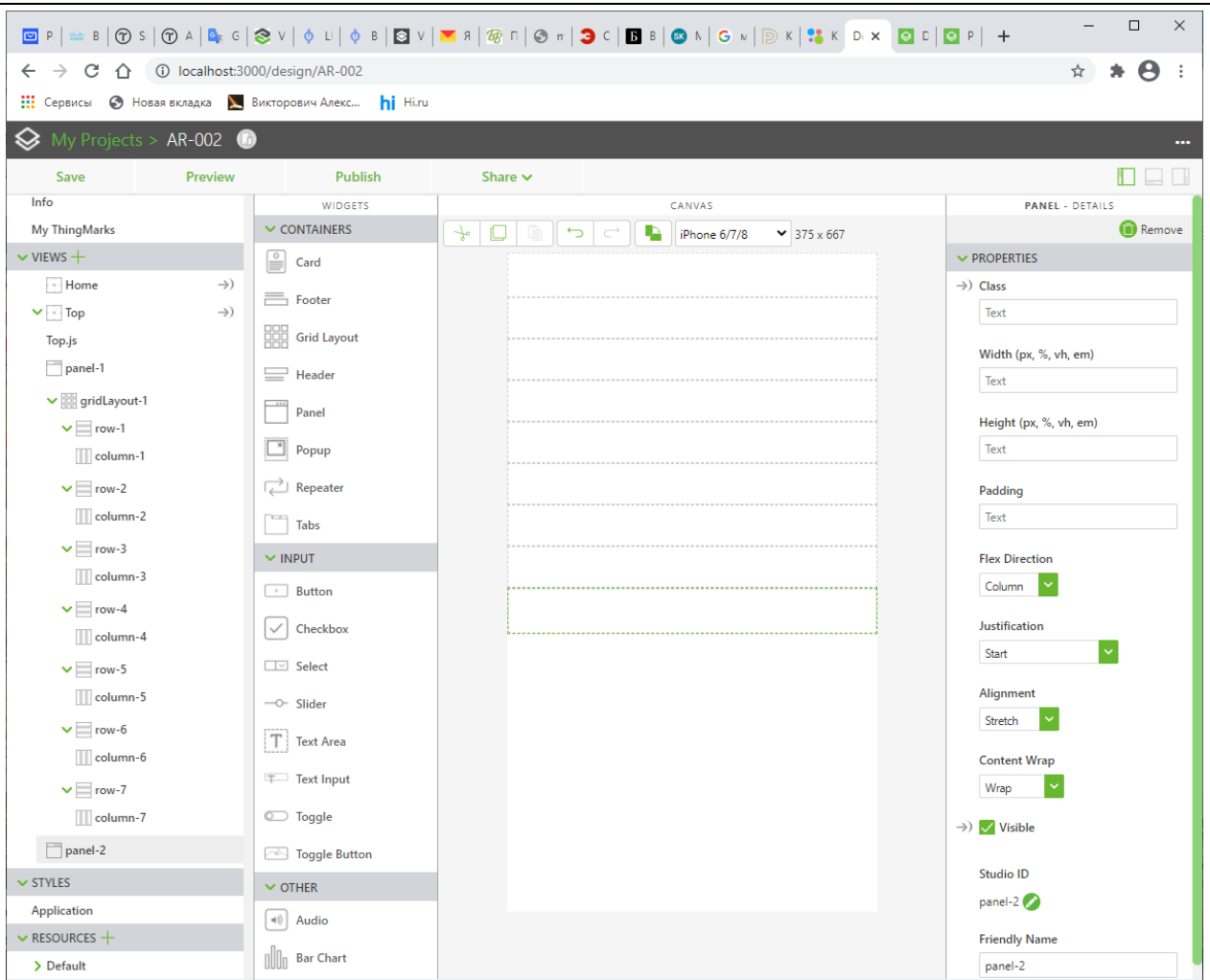
В частности, предусмотрим, что в сцене **Top** (считаем, что в Проекте есть 2 сцены) будут 2 кнопки навигации по сценам → **переход к сцене 1** и **переход к сцене 2** →



Шаг 10.

Начинаем размечать экран в **2D** - Канве.
Делаем разметку экрана (стекло) –
WIDGETS → Panel, Grid Layout,
последовательно заполняя экран, добавляя
ряды, добиваясь того, чтобы панель
сдвинулась вниз и в нужных ячейках можно
было добавить плоский контент.

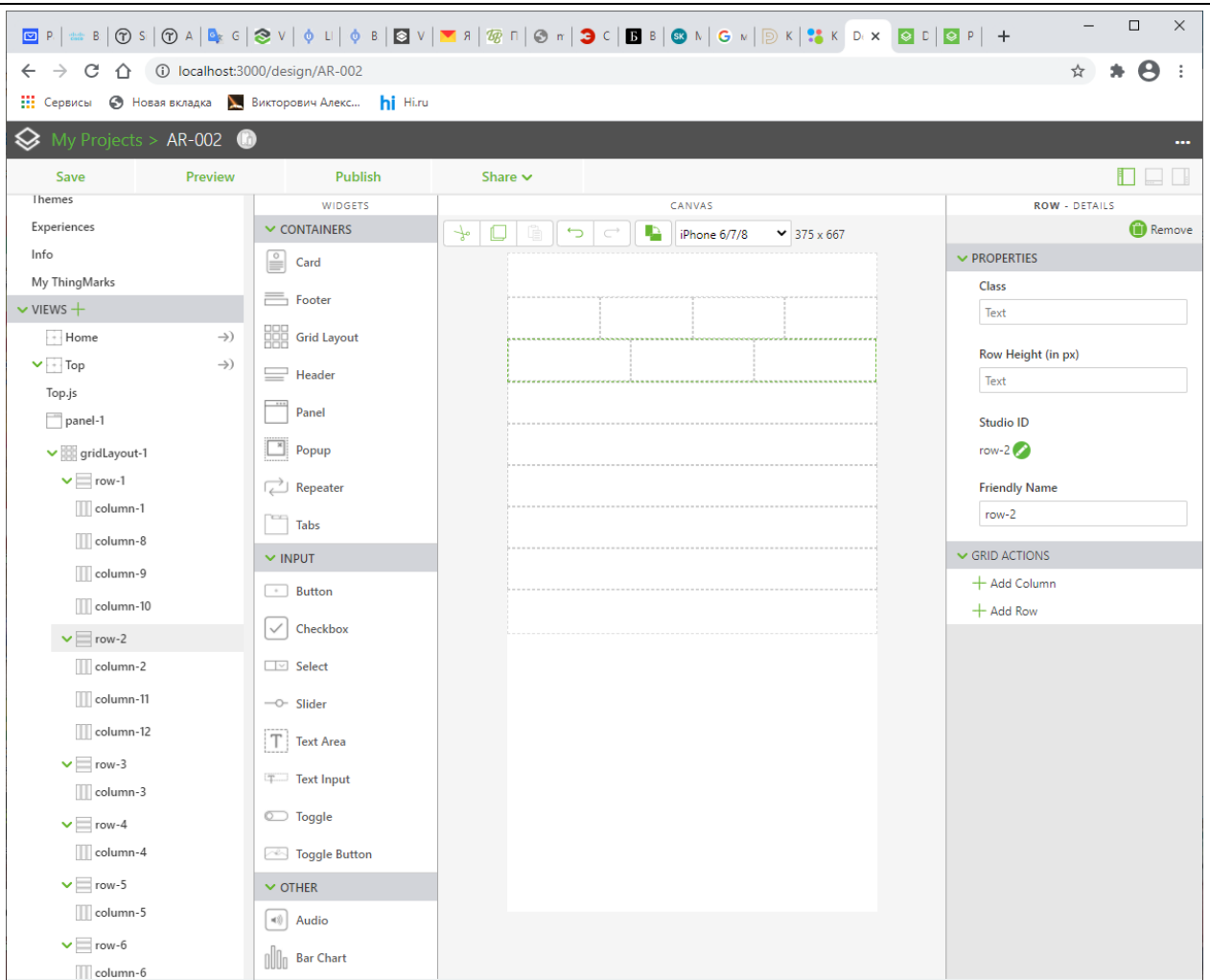
Следите за изменение иерархии объектов в
сцене в панели редактора **Vuforia Studio** –
CONFIGURATION/



Шаг 11.

Предлагается в верхней панели разместить логотип (например, МЭИ, подобно тому, как это было выполнено в **Проекте AR-001**, ЛР №2), а в колонках рядов контейнера **Grid Layout** - Управляющие элементы/кнопки. Для этого необходимо разметить подготовленные на предыдущем шаге ряды. Предлагается – 2 первых ряда разметить на 4 и 3 колонки соответственно. (см.

CONFIGURATOR) → **+ Add Column** в выбранном ряду (порядковые номера можно изменить). Элементы управления будут размещаться в разных рядах.



Шаг 12.

В качестве элементов управления выбираем уже знакомы нам – **Toggle Button** (все соответствующие контейнеры находятся в поле **WIDGET→ INPUT**).

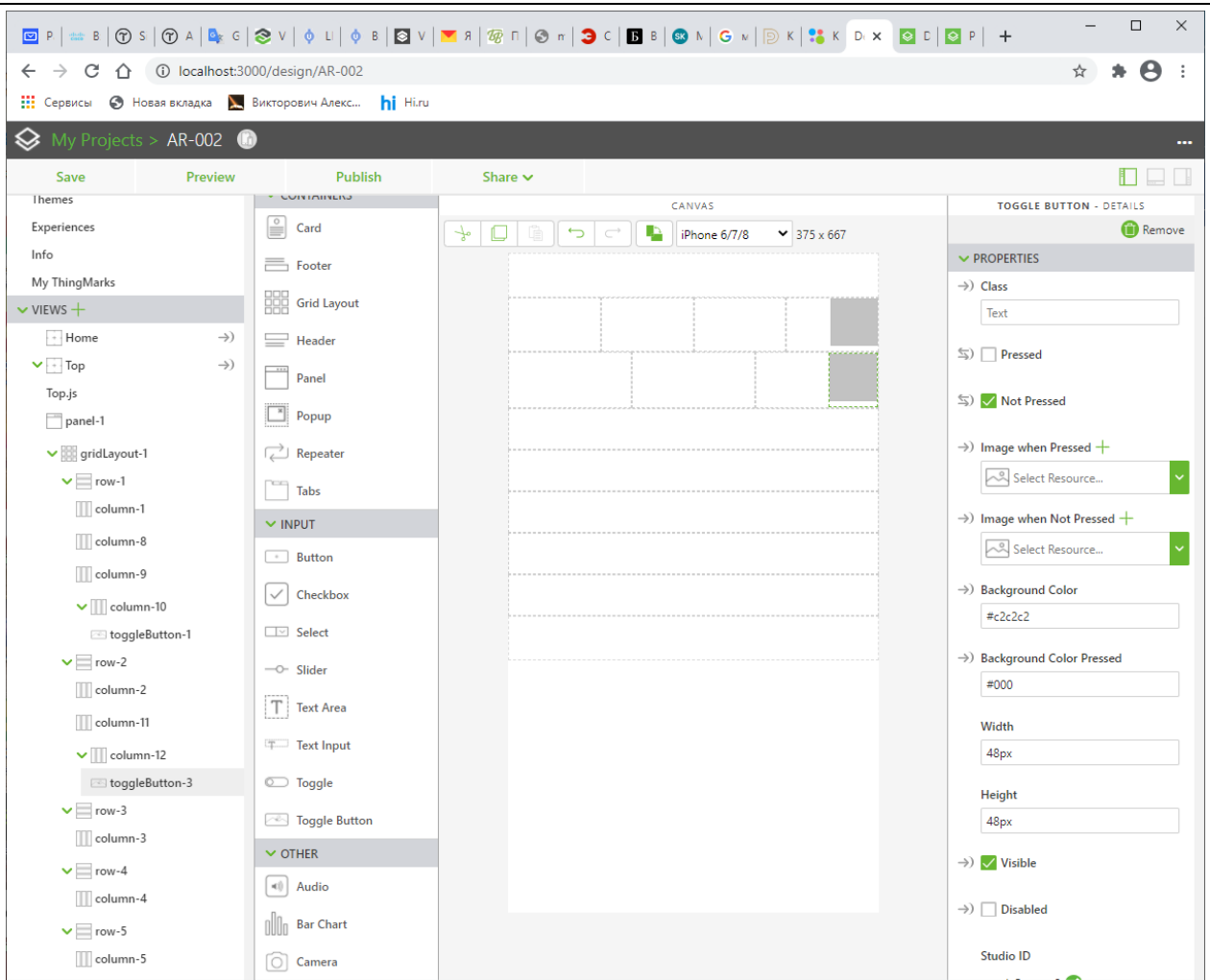
Помещаем выбранные виджеты в две правые колонки в двух верхних рядах **Grid Layout**.

Предварительно (!!) выравниваем колонки, в которых эти кнопки размещаются.

Пытаясь добиться желаемого эффекта. Например – выровнять по правому краю вплотную к границе или иное.

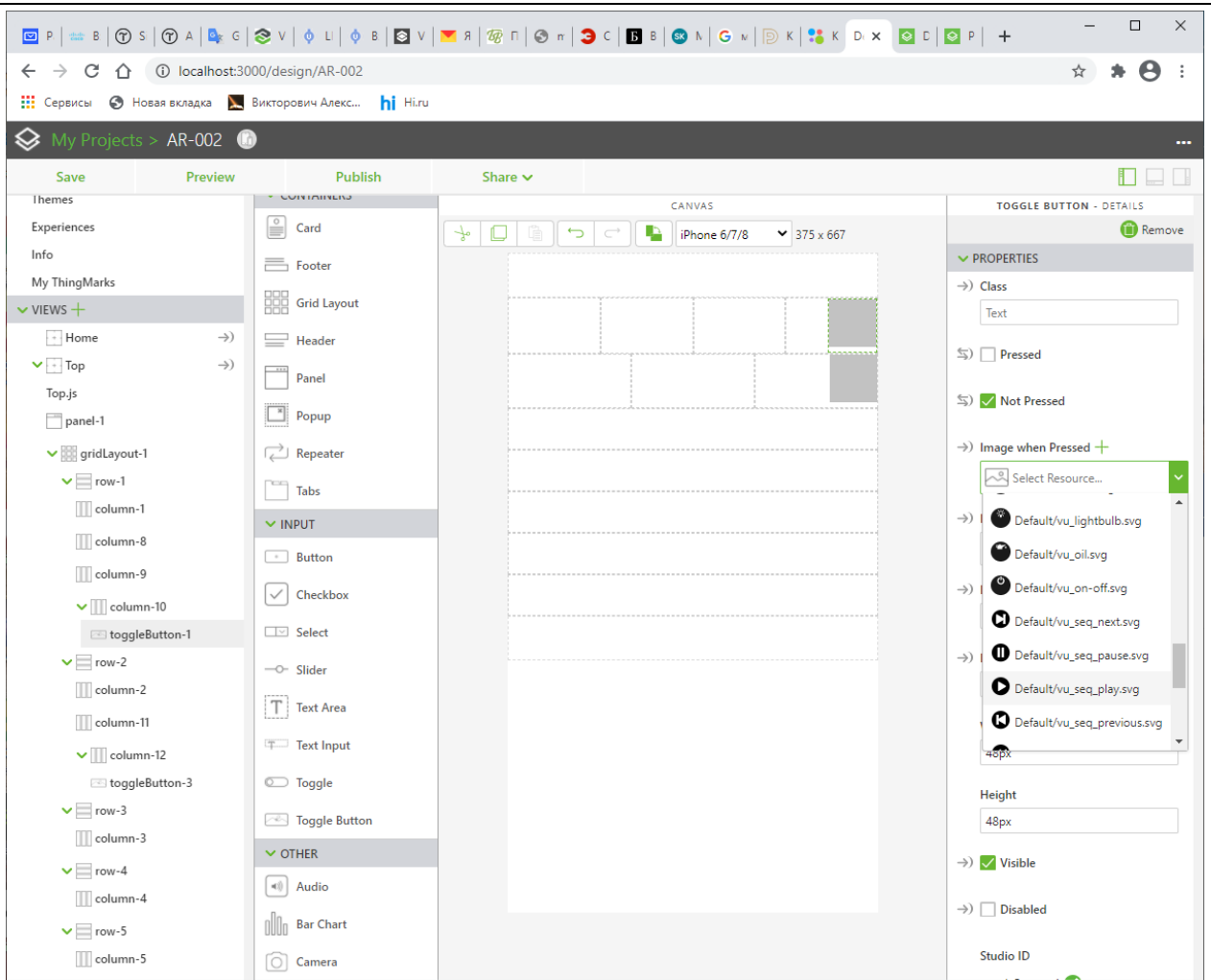
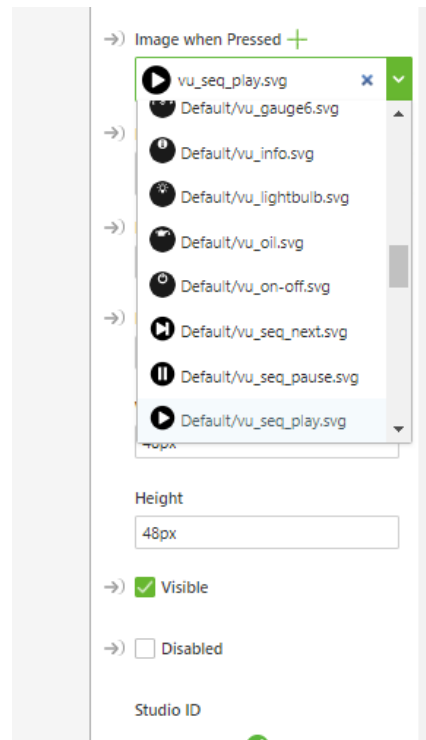
В нашем примере способ выравнивания колонок? в которых размещаются **Toggle Button** –

Justification/Alignment → End/End.



Шаг13.

На кнопки накладываем картинки из стандартного репертуара →



Шаг 14.

Выбор изображений производится для двух состояний (**Pressed/Not Pressed**) каждого виджета **Toggle Button** – выбираем одинаковые изображения:

The screenshot shows the configuration options for a Toggle Button widget. It includes:

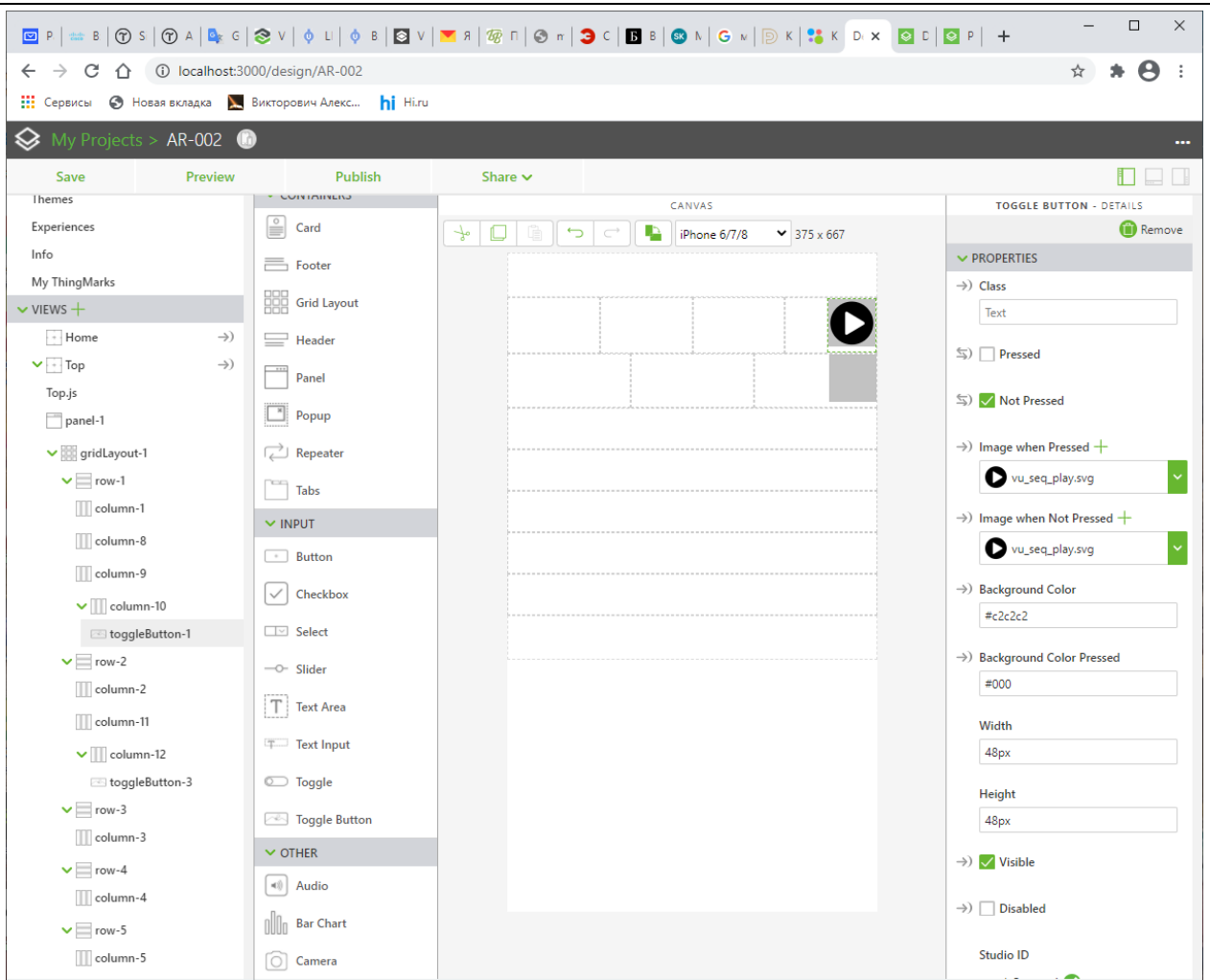
- Two radio buttons for states: Pressed and Not Pressed.
- Two sections for image selection:
 - Image when Pressed +**: A dropdown menu showing `vu_seq_play.svg`.
 - Image when Not Pressed +**: A dropdown menu also showing `vu_seq_play.svg`.
- Background Color**: A text input field.

The screenshot shows the Vuforia Studio interface with a 3D canvas and the properties panel for a Toggle Button widget. The canvas displays a grid layout with a play button icon on one of the cells. The properties panel on the right shows the following settings:

- CLASS**: Text
- Pressed
- Not Pressed
- Image when Pressed +**: `vu_seq_play.svg`
- Image when Not Pressed +**: Select Resource...
- Background Color**: `#c2c2c2`
- Background Color Pressed**: `#000`
- Width**: 48px
- Height**: 48px
- Visible
- Disabled
- Studio ID

Шаг 15.

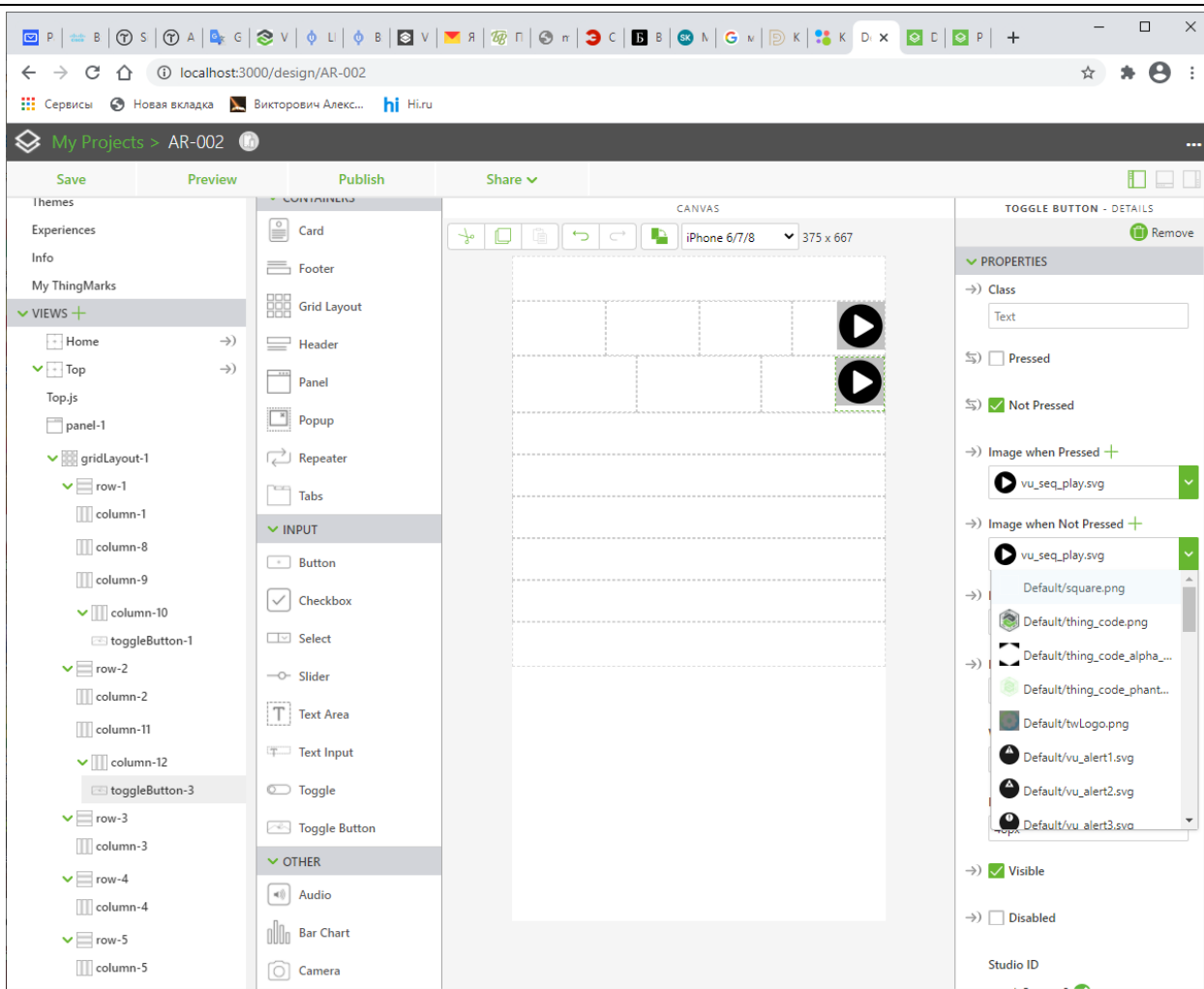
!!!! Для Not Pressed – тоже самое!!!!



Шаг 16.

Те же самые манипуляции производим и для второй кнопки.

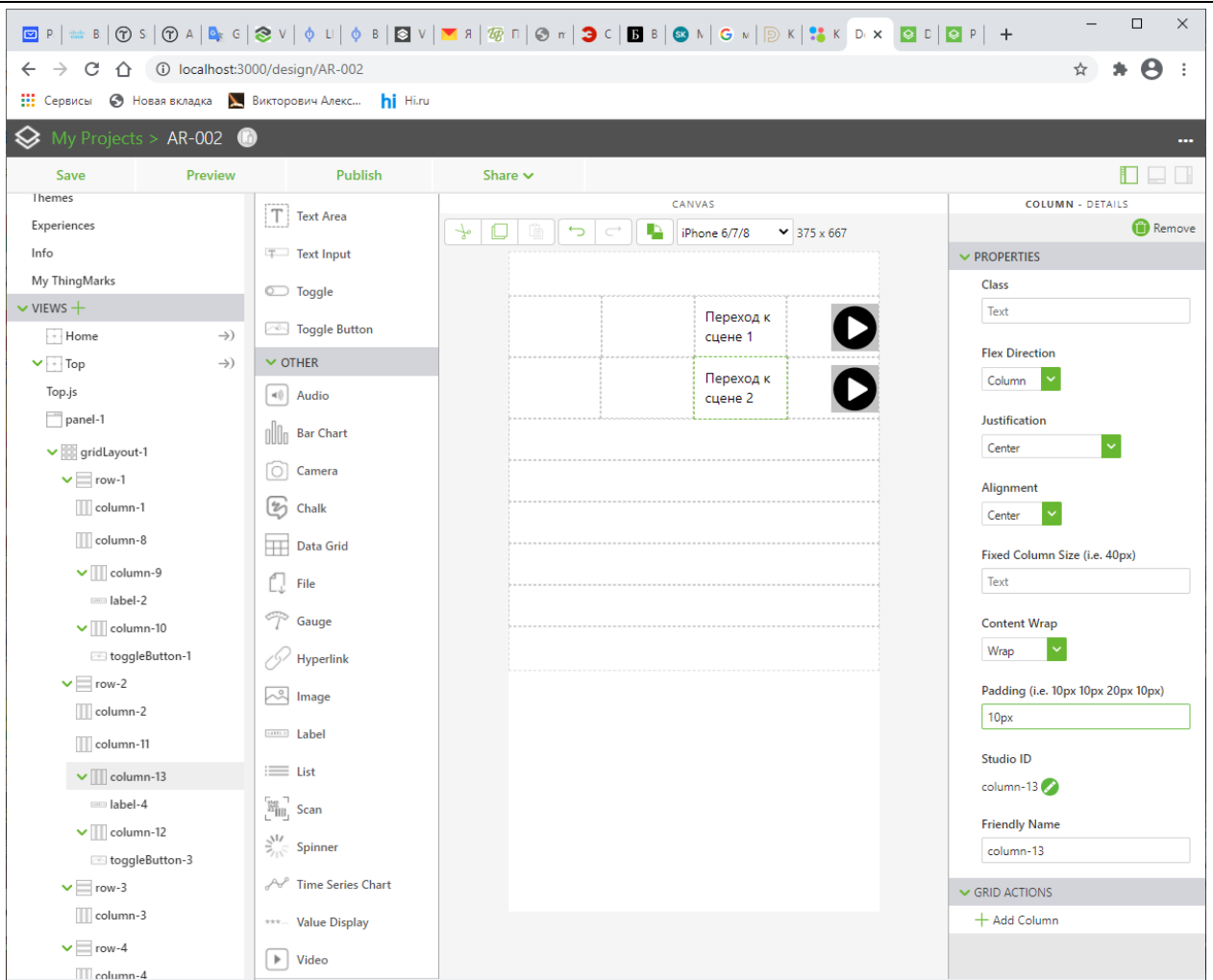
Кнопки предлагается снабдить текстовыми пояснениями в отдельной колонке.



Шаг 17.

Аналогично манипуляциям с изображениями кнопок, помещенных в колонки, в соседней (слева) от кнопки колонке (от каждой кнопки) добавляем текстовое описание с помощью **WIDGET'a Label**.

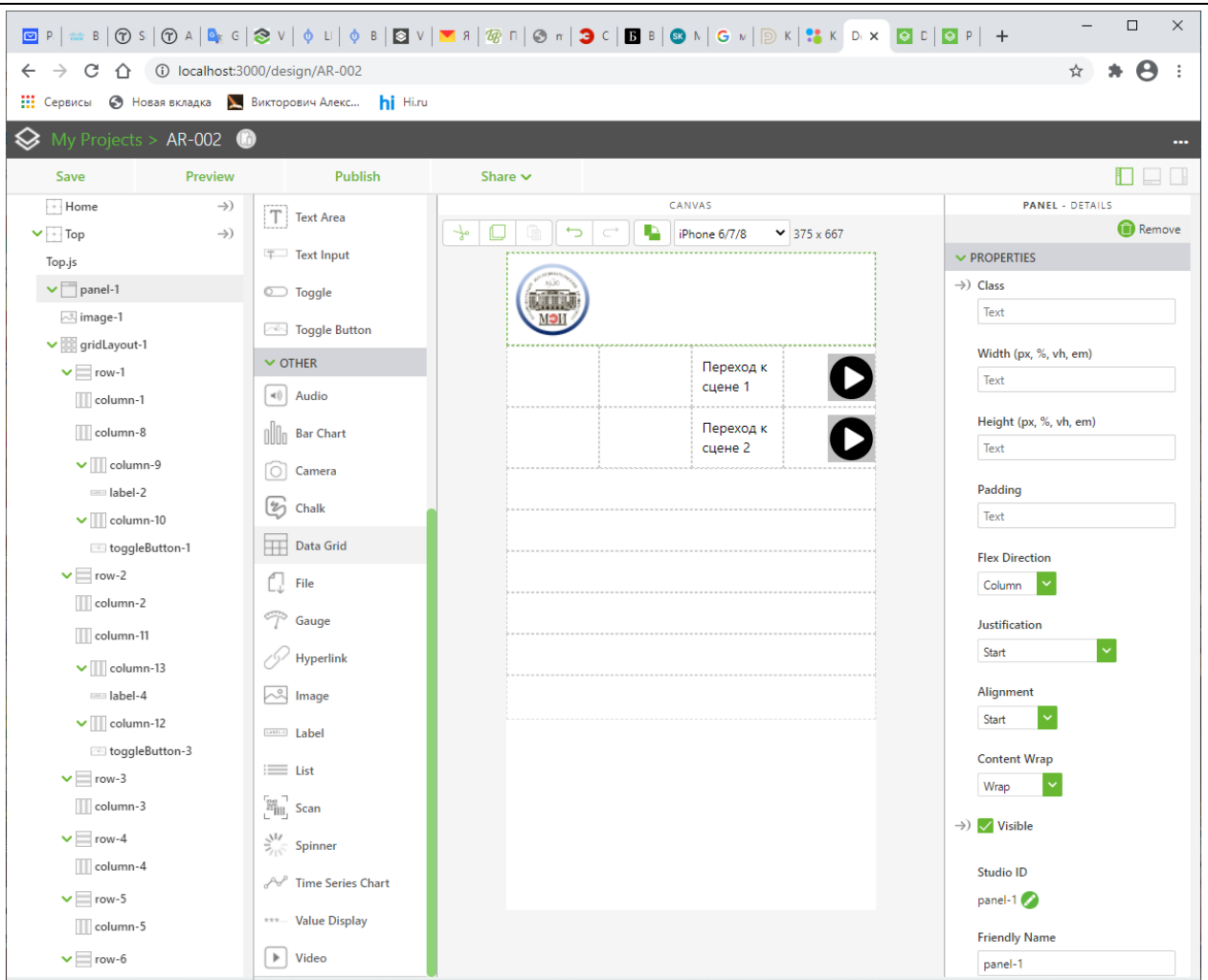
Настраиваем свойства соответствующей колонки для этого размещения и размещаемого в ней виджета **Label** (выравнивание, паддинг – можно попробовать разные значения, пробуя их влияние на результат).



Шаг 18.

Оформляем верхнюю панель **Титульной страницы** - добавим туда **Логотип МЭИ** с помощью **WIDGET → OTHER → Image**.

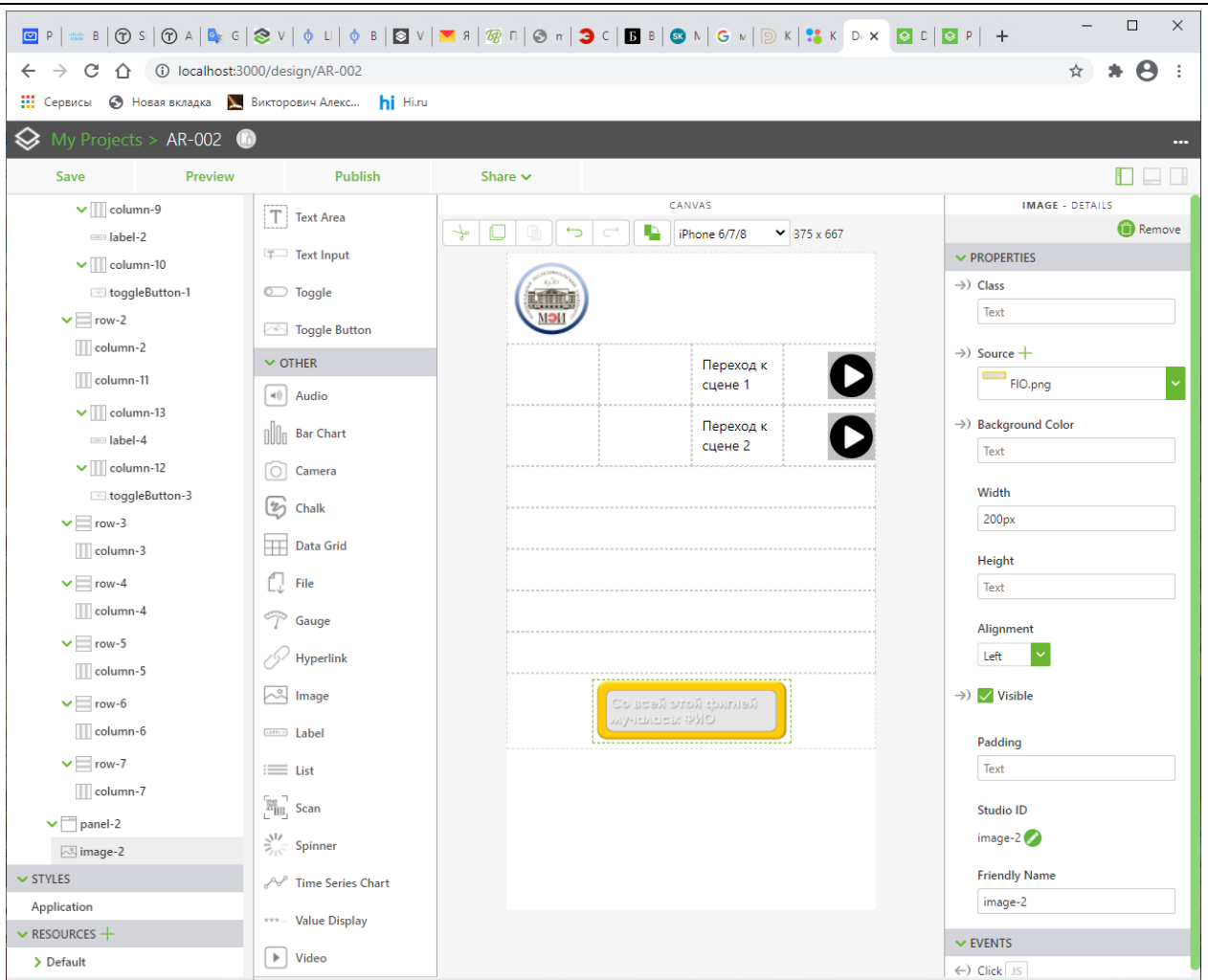
Логотип мог не попасть в ресурсы объектов (если вы заново формируете наполнение Проекта), в этом случае его нужно добавить извне известным вам способом.



Шаг 19.

Оформляем нижнюю панель **Титульной** **страницы**.

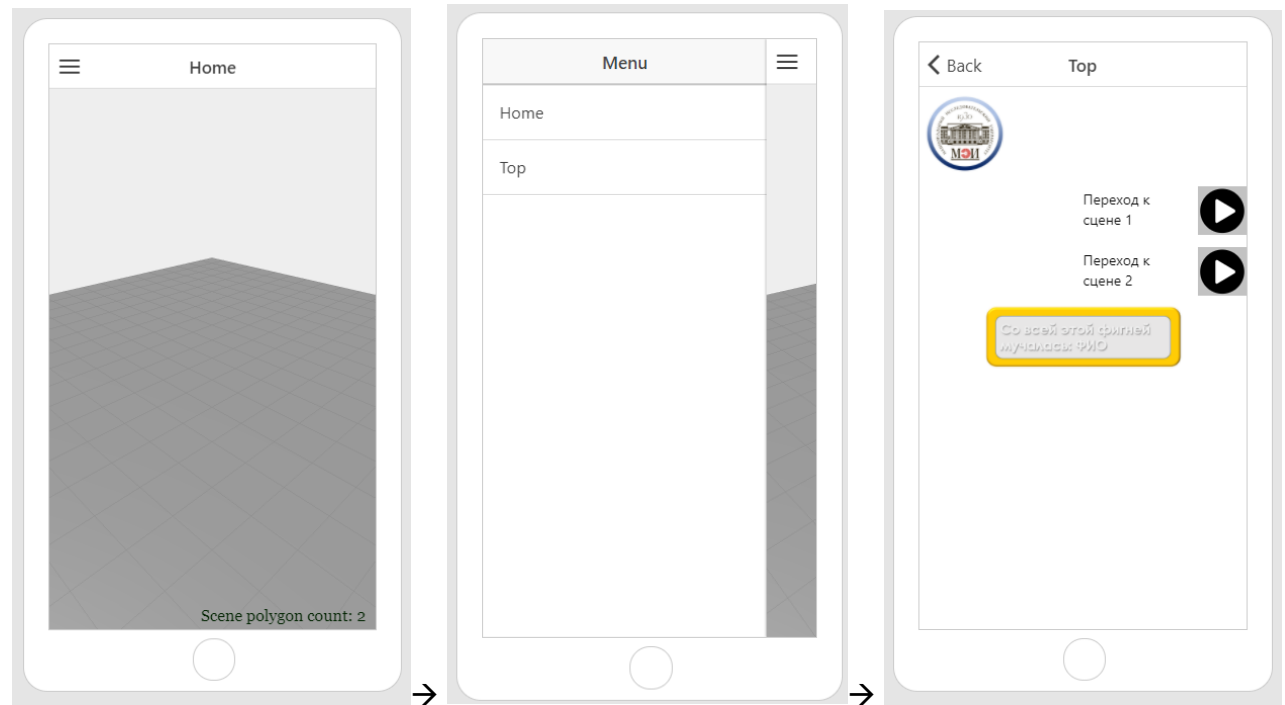
Это может быть информация об исполнителях или любая другая информация, оформленная также с помощью виджетов **Image** и **Label**.



Шаг 20.

Полученный результат можно просмотреть в режиме **Preview**, убедившись при этом в том, как работает **Navigation Menu**, предусмотренное в **Vuforia View** по умолчанию, как минимум для перехода между пустой сценой шаблона **Home** и **Титульной страницей Top**.

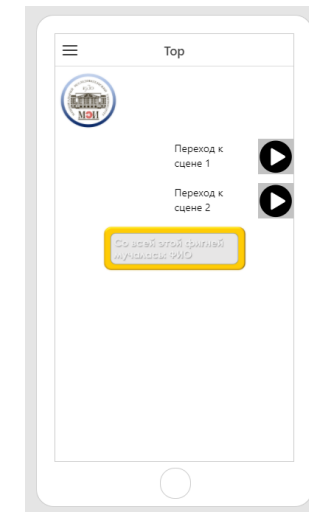
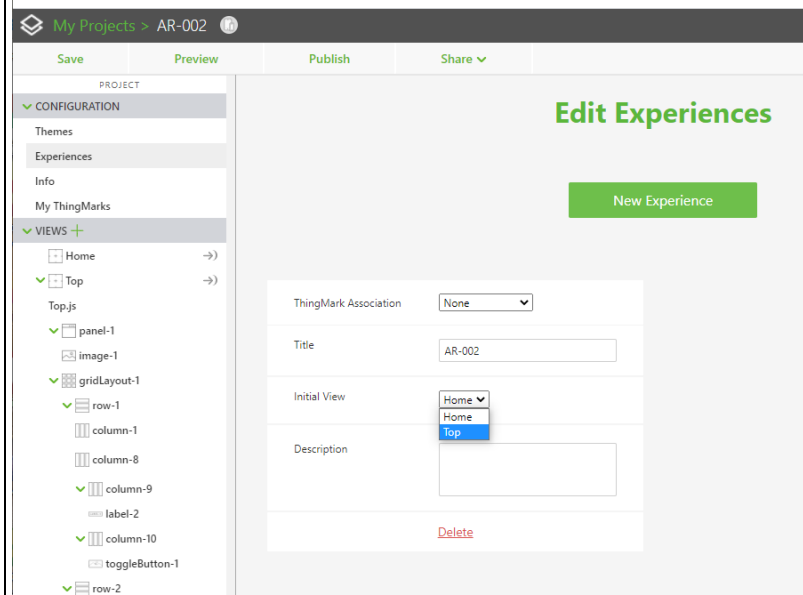
Если по каким-либо причинам полученный результат вас не устраивает, поменяйте соответствующие установки.



Шаг 21.

Теперь надо сделать эту плоскую сцену – первой, стартовой, **Титульной страницей**. Для этого в поле **CONFIGURATION** → **Experience** → **Initial View** → **выбираем Top**

Проверяем в режиме Preview видим:



Шаг 22.

Теперь на Титульной странице нужно организовать переход от сцены к сцене, связав соответствующие действия с размещенными кнопками (**Toggle Button**). В нашем Проекте пока ни одной из сцен нет, но для первой в поле **CONFIGURATION** есть заготовка – шаблон с именем **Home**. В ней мы будем формировать первую сцену (по аналогии с **Проектом AR-001**), а пока свяжем кнопки на титуле с выполнением навигации между сценами.

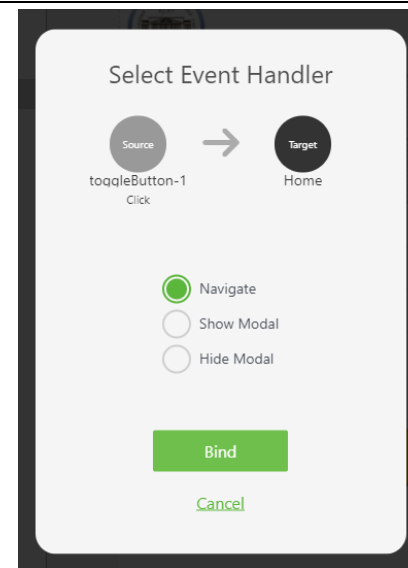
Связывание свойств виджета **Toggle Button** было подробно описано в ЛР №2 http://a0601.narod.ru/LR02_ARforIIoT.pdf

Нас в данном примере интересует связывание **EVENT Click** для **Toggle Button** с переходом (навигация) на сцену **1 VIEWS Home**→

The screenshot displays the Vuforia Studio interface for a project named 'AR-002'. The interface is divided into several sections: 'PROJECT', 'CONFIGURATION', 'OTHER', 'CANVAS', and 'EVENTS'. In the 'PROJECT' section, the 'VIEWS' list is expanded, showing a 'Home' view with a red arrow pointing to its 'toggleButton-1' widget. The 'CONFIGURATION' section shows the 'Toggle Button' widget selected. The 'OTHER' section lists various widgets like 'Text Area', 'Text Input', 'Toggle', 'Toggle Button', 'Audio', 'Bar Chart', 'Camera', 'Chalk', 'Data Grid', 'File', 'Gauge', 'Hyperlink', 'Image', 'Label', 'List', 'Scan', 'Spinner', 'Time Series Chart', and 'Video'. The 'CANVAS' section shows a preview of the application on an iPhone 6/7/8 screen, with a yellow box highlighting a button labeled 'Со всей этой информацией'. The 'EVENTS' section shows the 'Click' event selected, with the 'JS' script editor open, displaying the following code: 'Clicked JS', 'Pressed JS', and 'Unpressed JS'. The 'GRID ACTIONS' section shows 'Add Column' and 'Add Row' options.

Шаг 23.

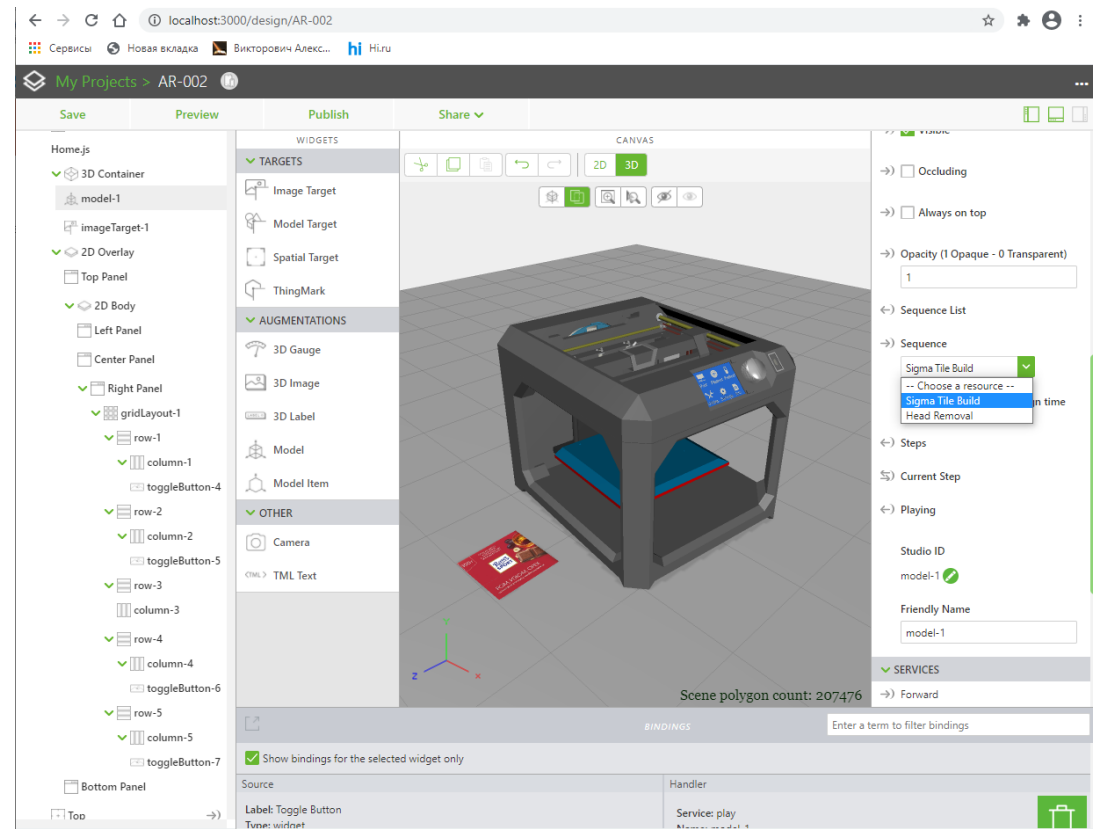
В открывшемся окне обработчика событий для **ToggleButton-1** (имя первой кнопки в иерархии **CONFIGURATION**) выбираем **Navigation**.



Шаг 24.

Теперь можно приступить к формированию первой сцены Проекта **AR-002**. Это будет в данном примере упрощенный вариант базового Проекта **AR-001 с моделью 3D-Принтера**. В данном примере формируем эту сцену заново, «вручную, в индивидуальном задании для каждого обучающегося возможен вариант модификации этого базового Проекта. Однако вторую сцену все равно придется делать вручную. Основные шаги по созданию 3D-Контента подробно описаны в ЛР №2:

- Помещаем модель в сцену;
- Добавляем таргет (**Image Target**):
- Выбираем первую анимированную последовательность – печать корпуса гайки;

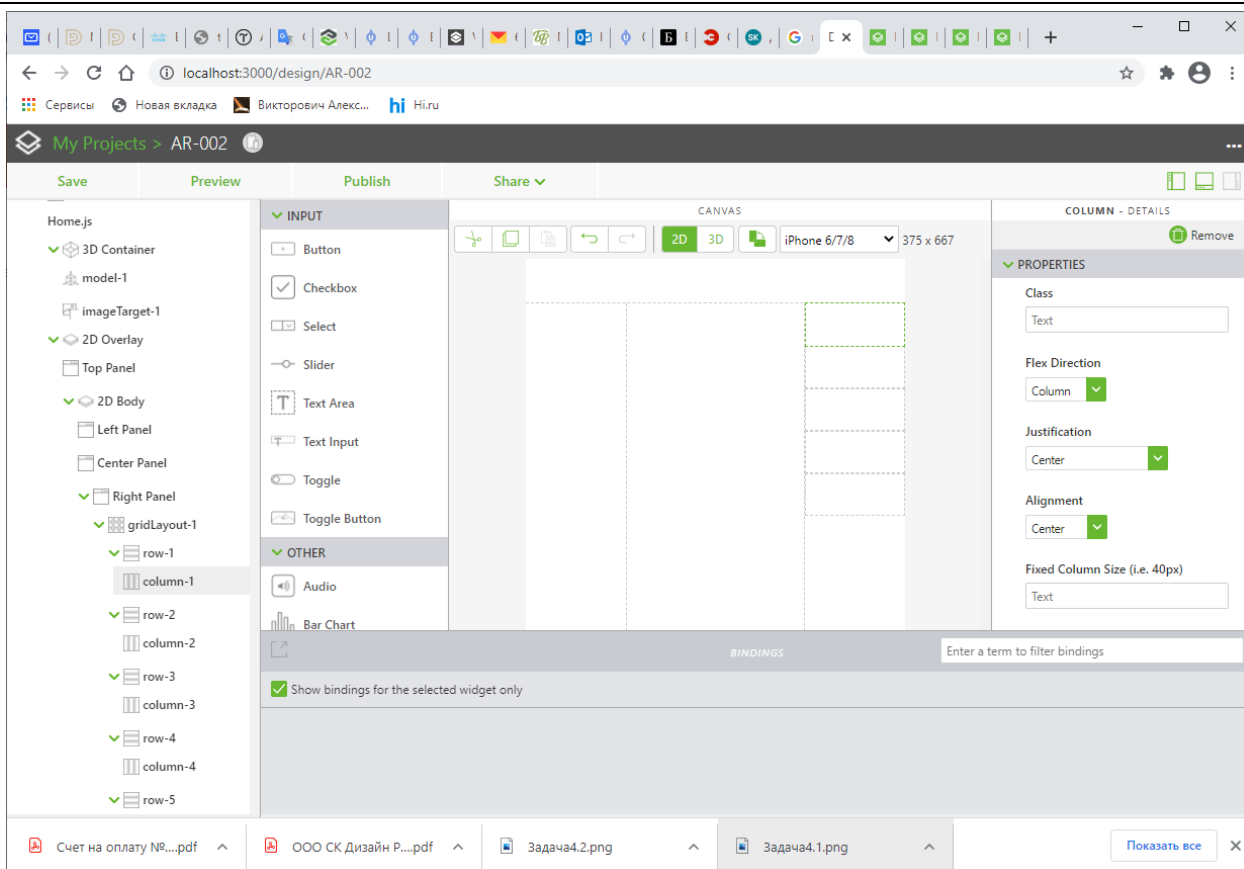


Шаг 25.

Результат можно проверять в **Preview**. Теперь на этой сцене надо добавить элементы управления – для переходов внутри сцены (запуск анимированной последовательности -пошагово - в обе стороны), и для перехода между сценами.

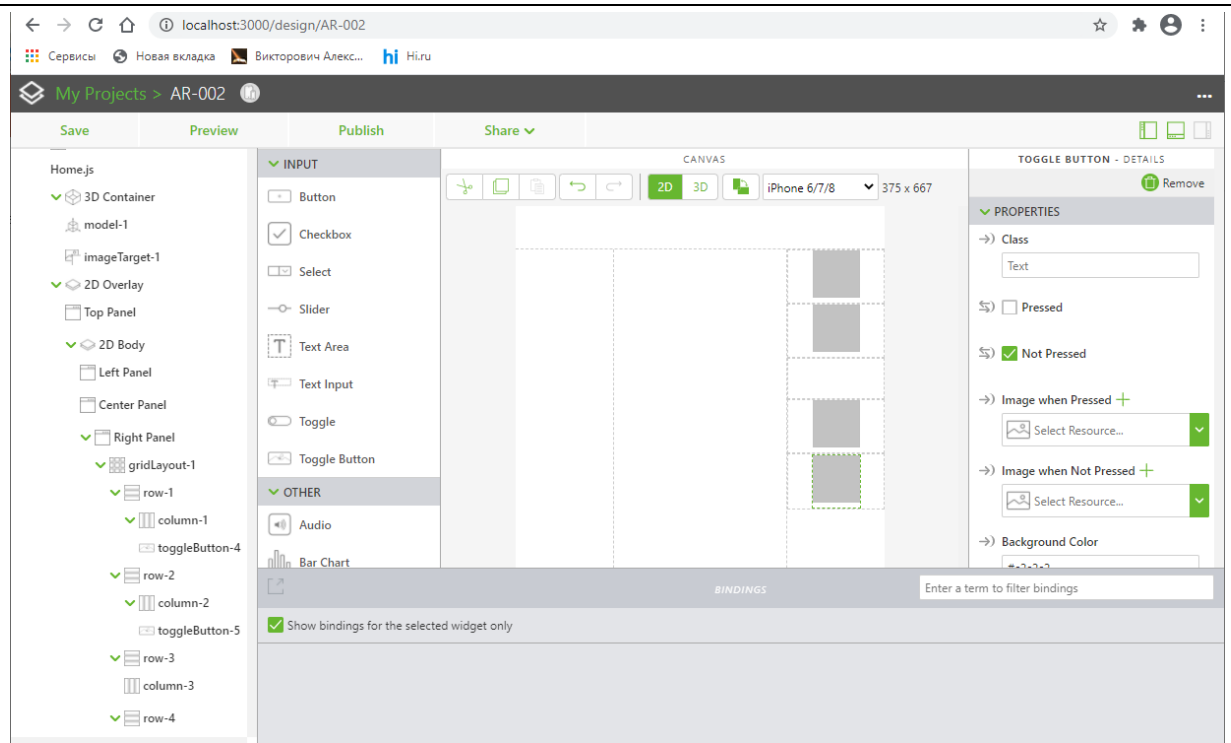
Переходим в **2D - Канву**, получаем размеченное по умолчанию пространство, в правую панель (все кнопки смещаем вправо) добавляем **Grid Layout** (плейсхолдер под виджеты), добавляем 5 рядов. Кнопки не будем снабжать текстом, как на титуле, потому что заранее у нас были созданы самостоятельно изображения для кнопок, снабженные текстом. Т.о. можно сэкономить пространство и задать свой стиль, преодолев ограничения РТС-шных кнопок по умолчанию.

Выравнивание внутри **Column** делаем Center/Center (в полях **Justification** и **Alignment**) и помещаем туда элемент управления **Toggle Button** (полезно вспомнить, что этот виджет что допускает гибкое управление и состоянием – **Pressed/Unpressed** и событиями – **Event**).





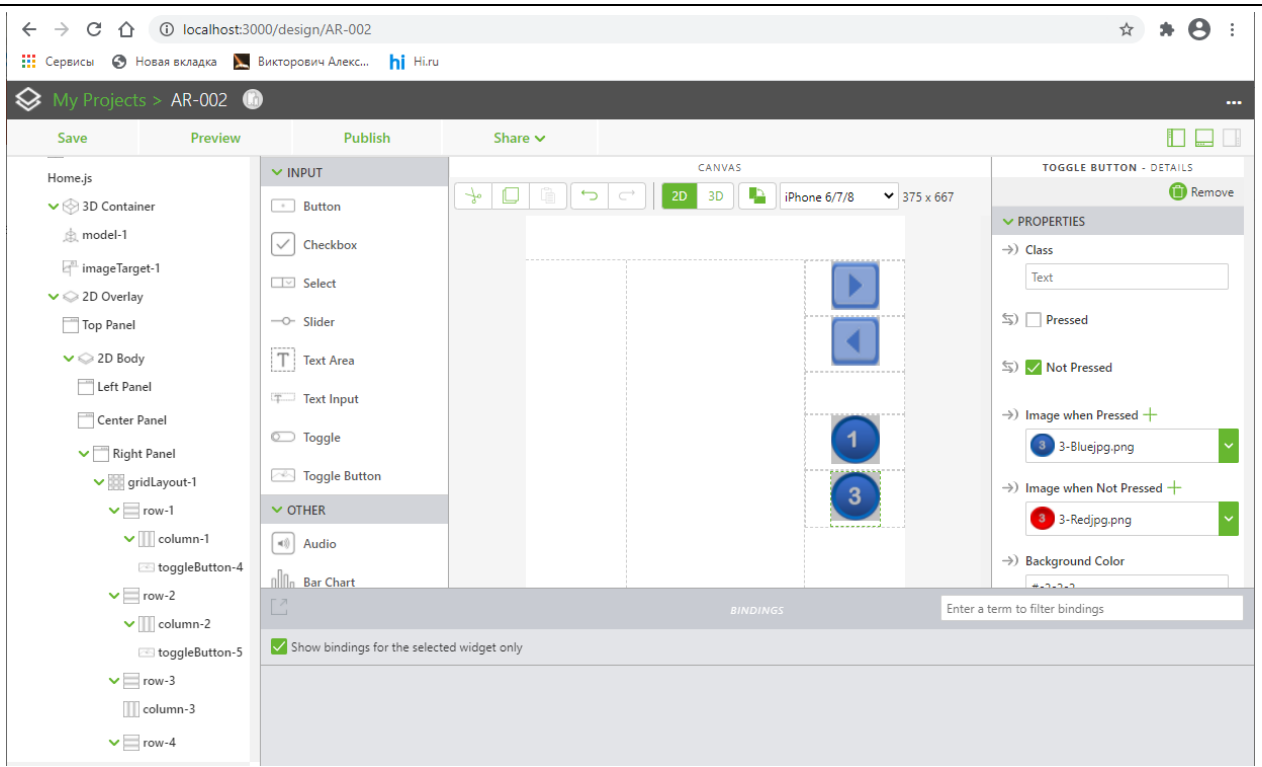
Шаг 26.

Повторим эту процедуру трижды, подготовив места для кнопок управления просмотром анимированной последовательности 3D-печати корпуса «гайки» - (вперед/назад, пошагово) – **Sigma Tile Build** - и переходов между сценами (вперед/назад) и **Титульной страницей**.



Шаг 27.

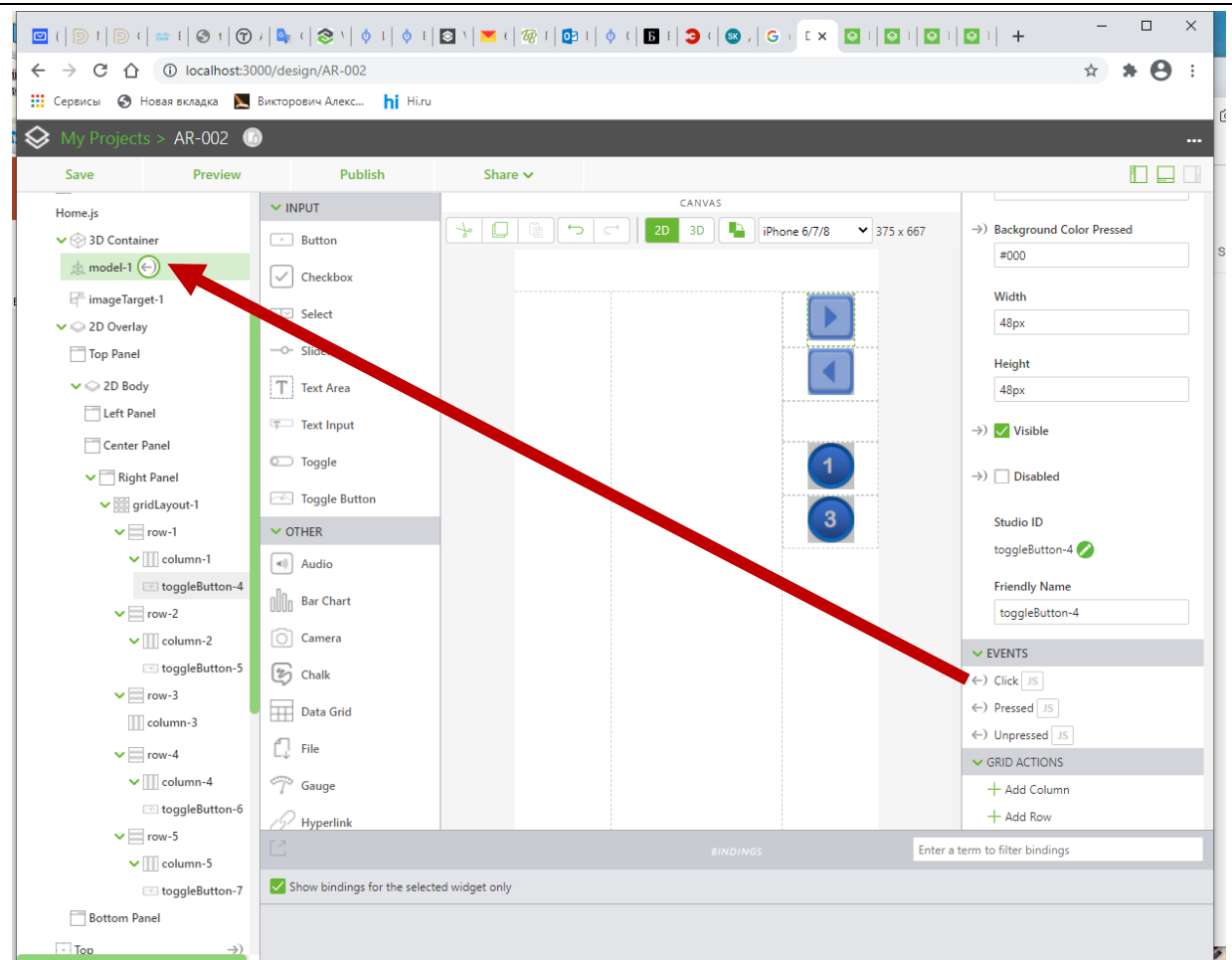
На каждую из кнопок назначим свои изображения: две верхние будут отвечать за переход внутри сцены ( , ), а нижние за переход на другие сцены ( – начало, т.е. **Титульная страница, «Тор»**; 2 – это сама текущая сцена в данном случае – **«Home»**;  – пока еще не созданная сцена, вторая, назовем ее **«Home2»**) →



Шаг 28.

Начинаем связывать заданные кнопки (те, что возможно) с имеющимися визуализациями и переходами.

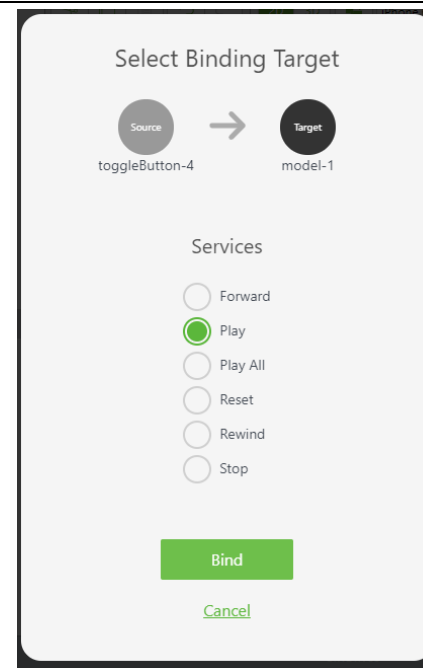
Первые две кнопки будут отвечать за переход между анимированными последовательностями в текущей сцене – печать коду **Sigma Tile Build** (пошагово – туда и обратно) →



Шаг 29.

Связываем кнопку для пошагового просмотра анимированной последовательности – т.е. связываем **toggleButton-4** с сервисом **Play**.

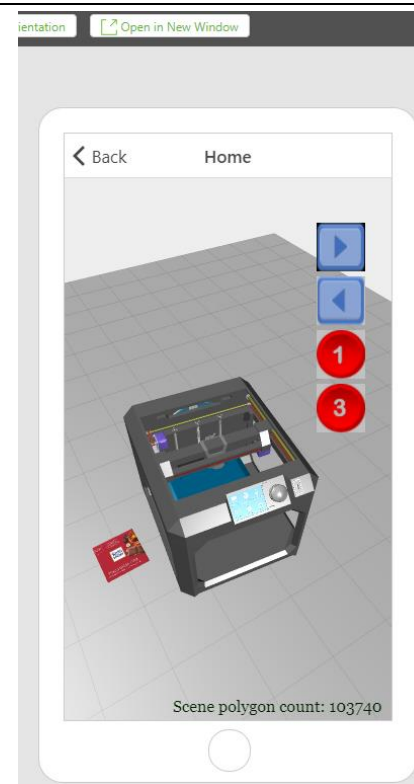
Анимированные последовательности (формат **.pvi**) входят в состав архива **.pvz 3D-Принтера**. Напомним, что **.pvz** – родной формат приложения **Creo Illustrate**.



Шаг 30.

Для второй кнопки повторяем те же самые действия, но связываем ее с сервисом **Rewind**.

Проверьте результат в **Preview** →



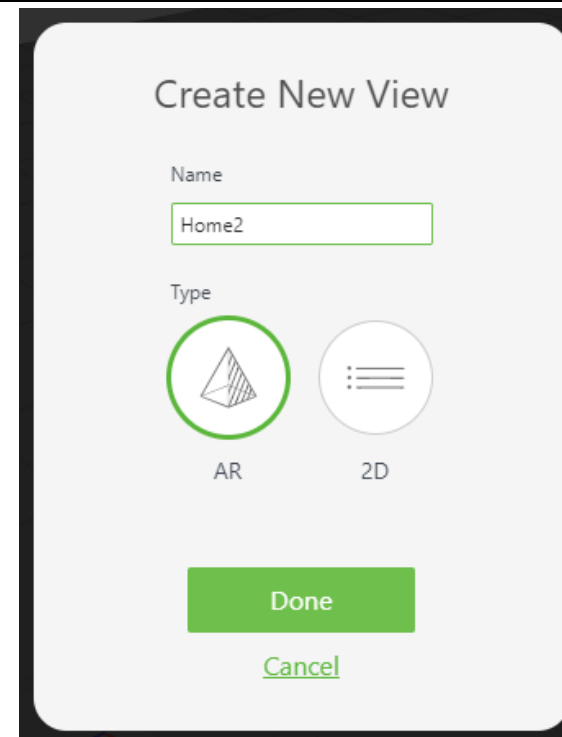
Шаг 31.

Начинаем выполнять вторую сцену.

Повторяем те же шаги, что и для создания предыдущей сцены. Новую сцену назовем **Home2**.

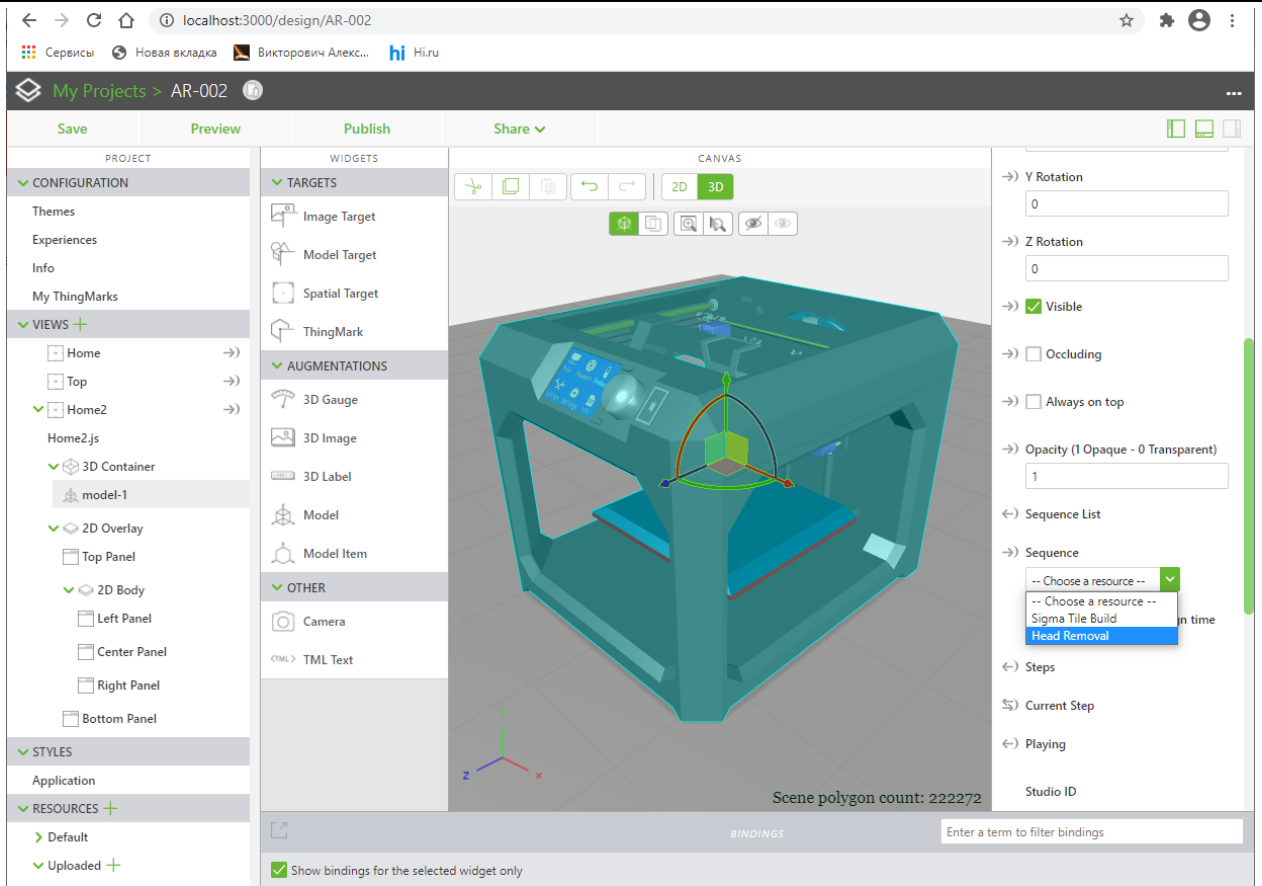
Разница контентов двух сцен в данном примере заключается в выборе другой анимированной последовательности (**Head Removal**) для той же самой модели. В общем случае каждая сцена может иметь принципиально свой контент, свой набор моделей, последовательностей и т.д. →

VIEW + →



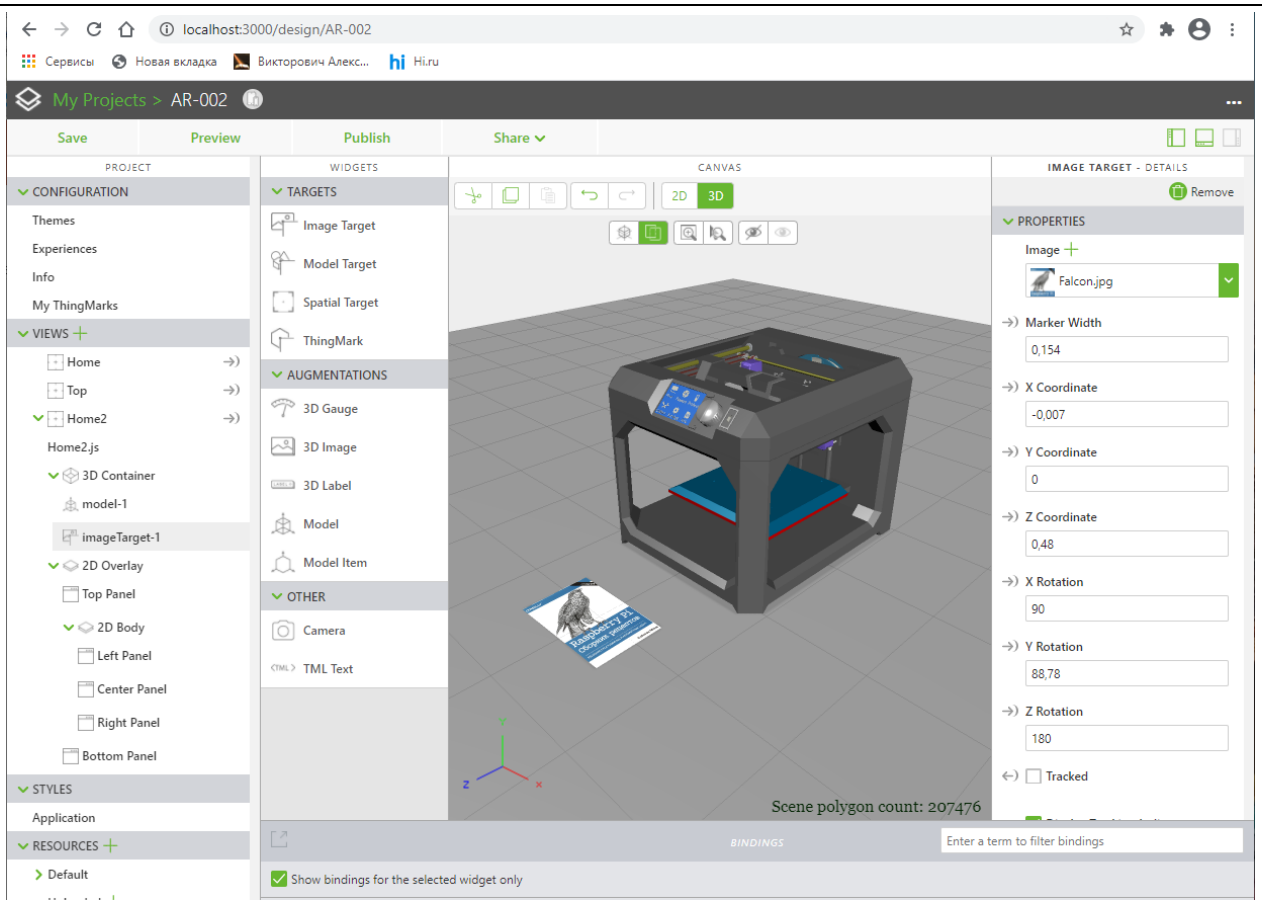
Шаг 32.

Начинаем формировать сцену → выполняем те же самые шаги, что и для сцены **Home**, за исключением выбираемой анимированной последовательности.



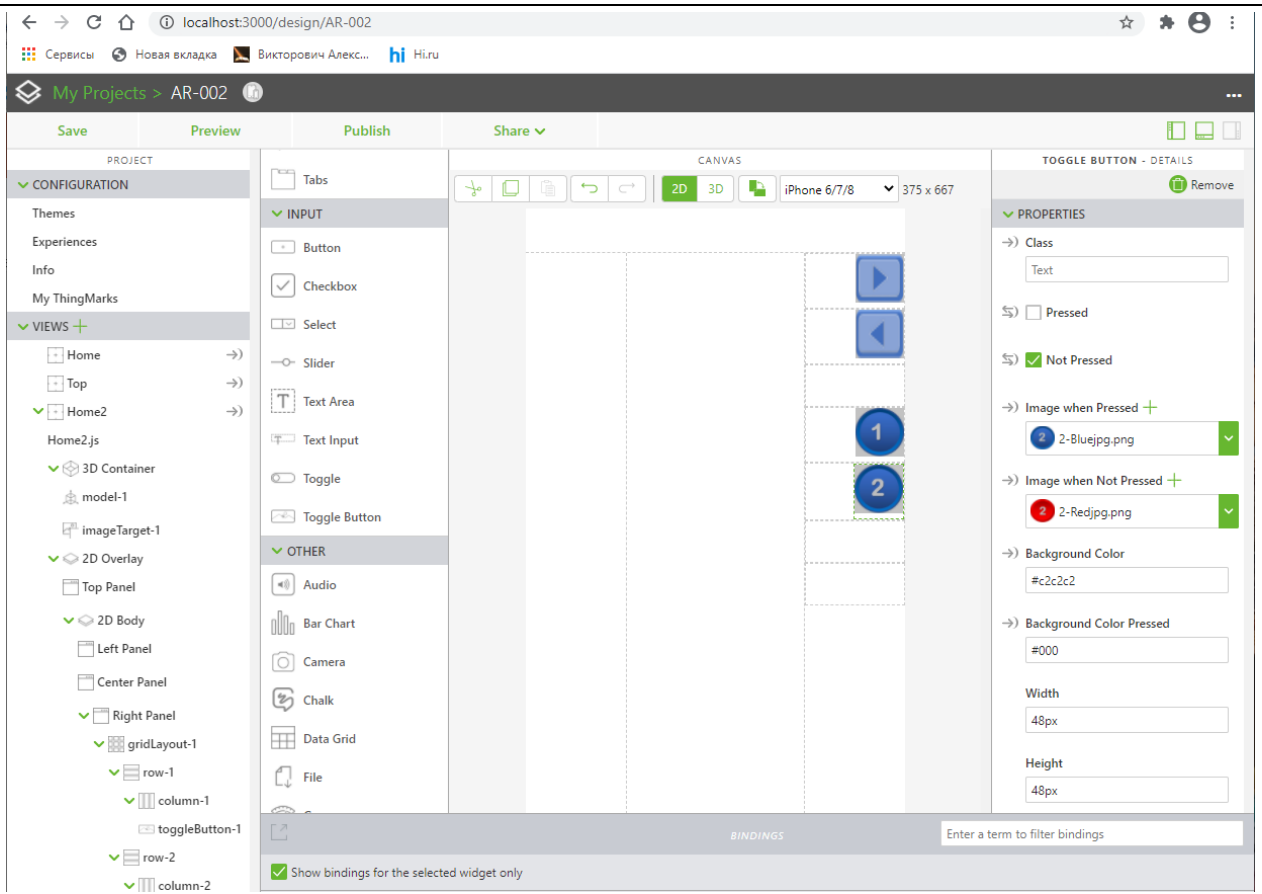
Шаг 33.

Таргет тоже надо заменить. Выбираем плейсхолдер **Image Target** и помещаем в него картинку:



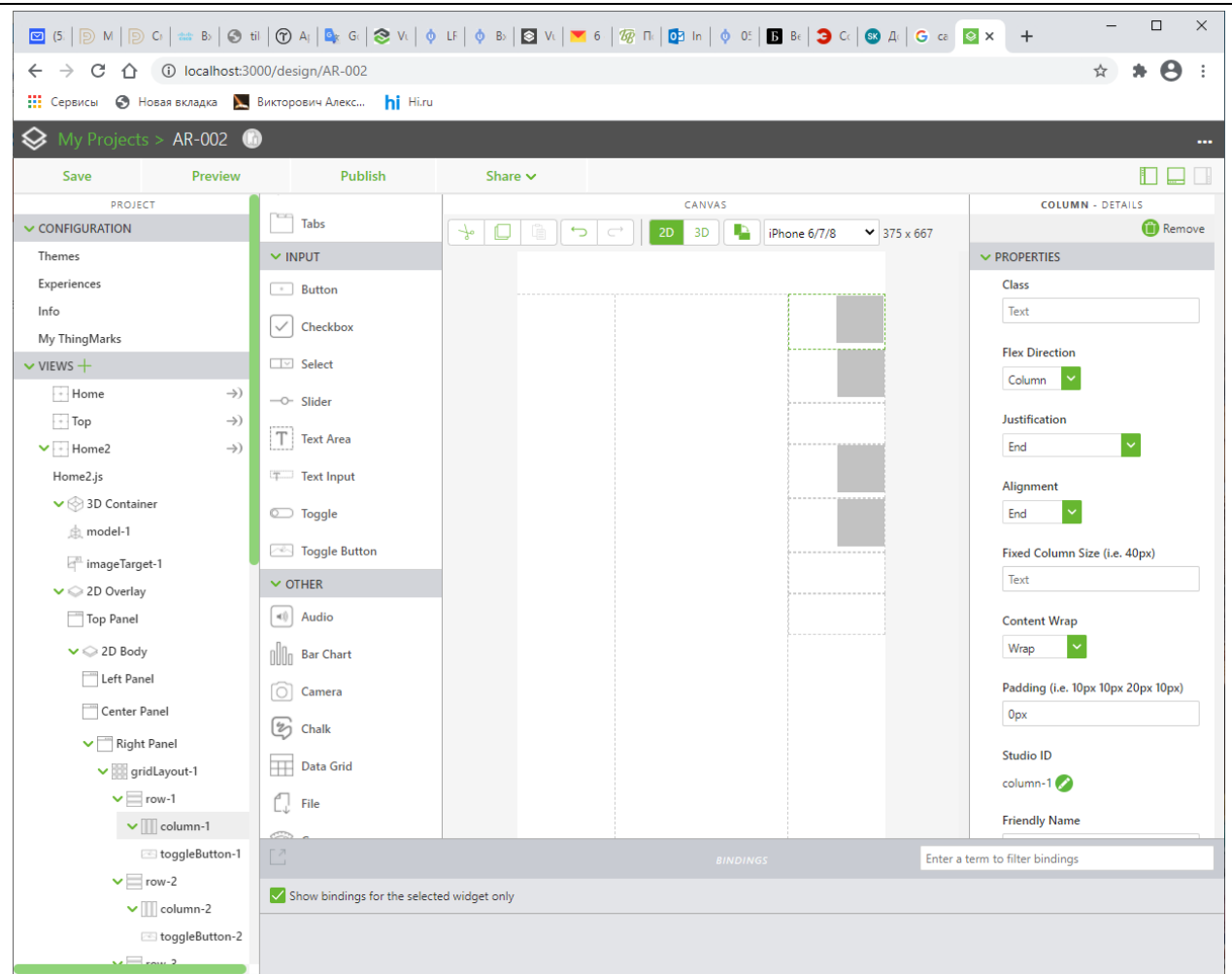
Шаг 34.

Оформляем меню для этой сцены → переходим в **2D Канву** → аналогично первой сцене в Home2 делаем разметку и размещение управляющих элементов – внутри сцены и перехода между сценами. Кнопки сместим к правому краю (**end/end**)



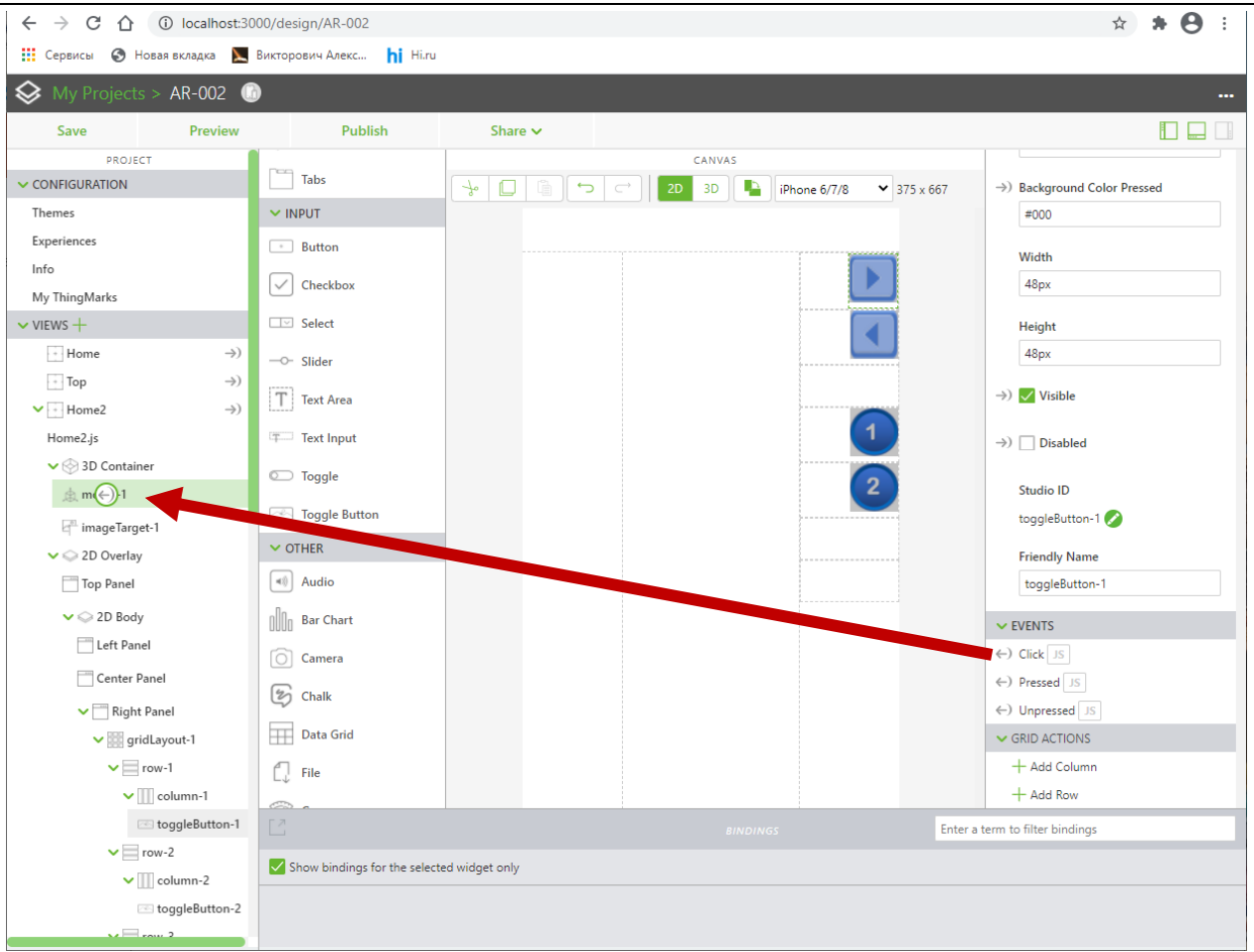
Шаг 35.

Первые две кнопки будут отвечать за переход между анимированными последовательностями в текущей сцене – Замена рабочего органа - **Head Removal** (пошагово – туда и обратно) →



Шаг 36.

toggleButton-1 → Home2 → model1

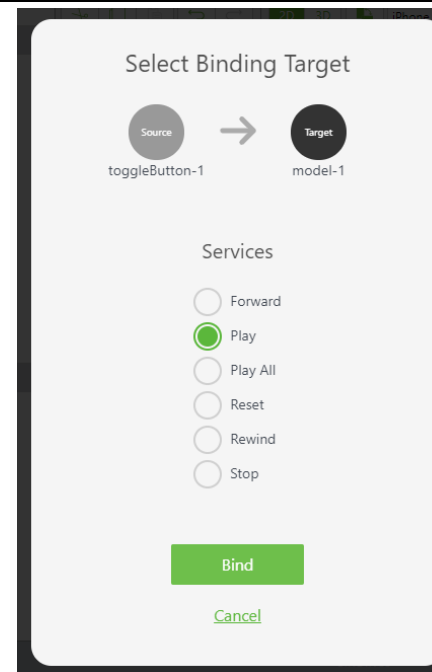


Шаг 37.

Кнопку

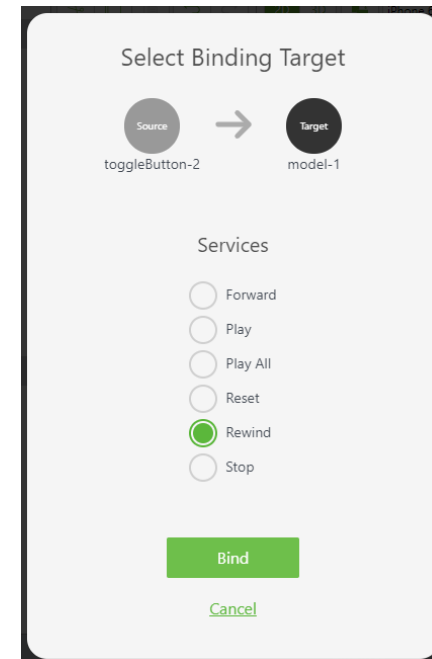
toggleButton-1 → Home2 → model1

связываем с сервисом прямого пошагового просмотра → **Play**



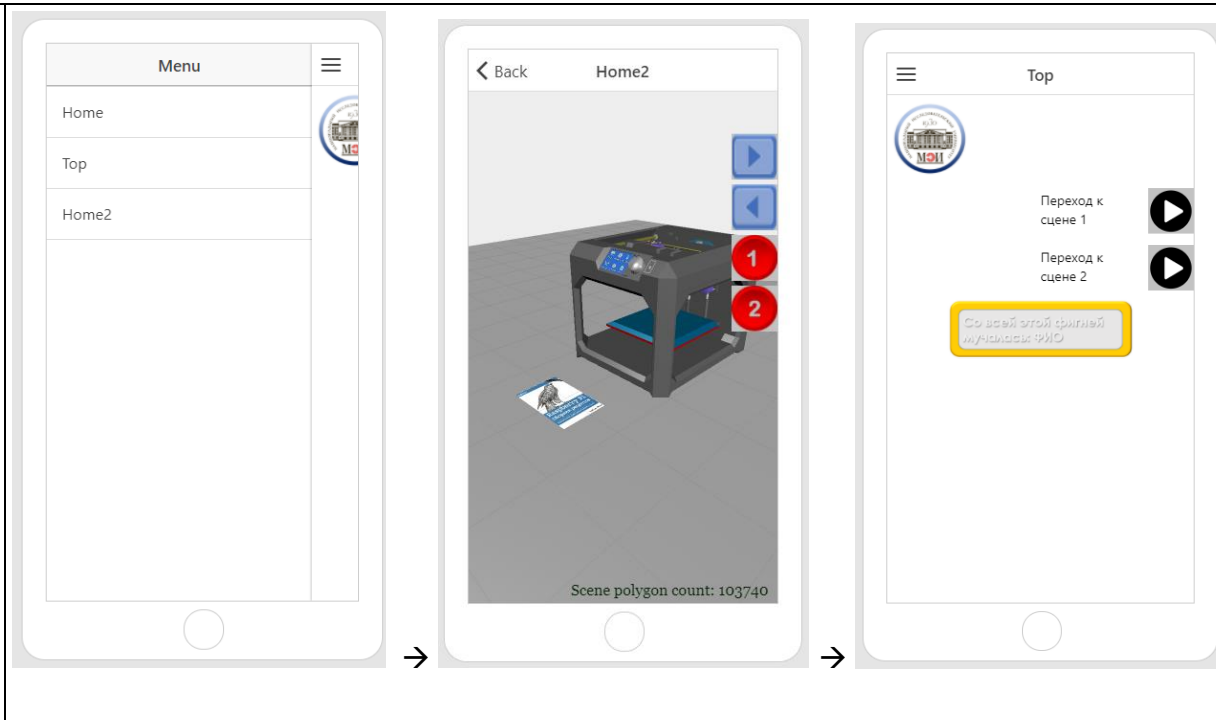
Шаг 38.

То же самое выполняем для обратного пошагового просмотра → сервис **Rewind**



Шаг 39.

Можно убедиться в работоспособности выполненных действий в режиме **Preview**. Обратите внимание, пока перехода между сценами **Home – Home2** и обратно; **Top – Home2** и обратно; **Home - Top** в нашем Проекте нет, получить необходимые переходы можно с помощью системного меню **Vuforia Studio**, меню по умолчанию, которое мы оставили доступным на время разработки и отладки **Проекта AR-002** →



Шаг 40.

В том, что с моделью связаны 2 действия по визуализации последовательности (пошагово вперед и пошагово назад) можно убедиться в режиме просмотра канвы 3D, открыв область просмотра связей:

The screenshot shows the Vuforia Studio interface in a 3D canvas view. The interface is divided into several sections:

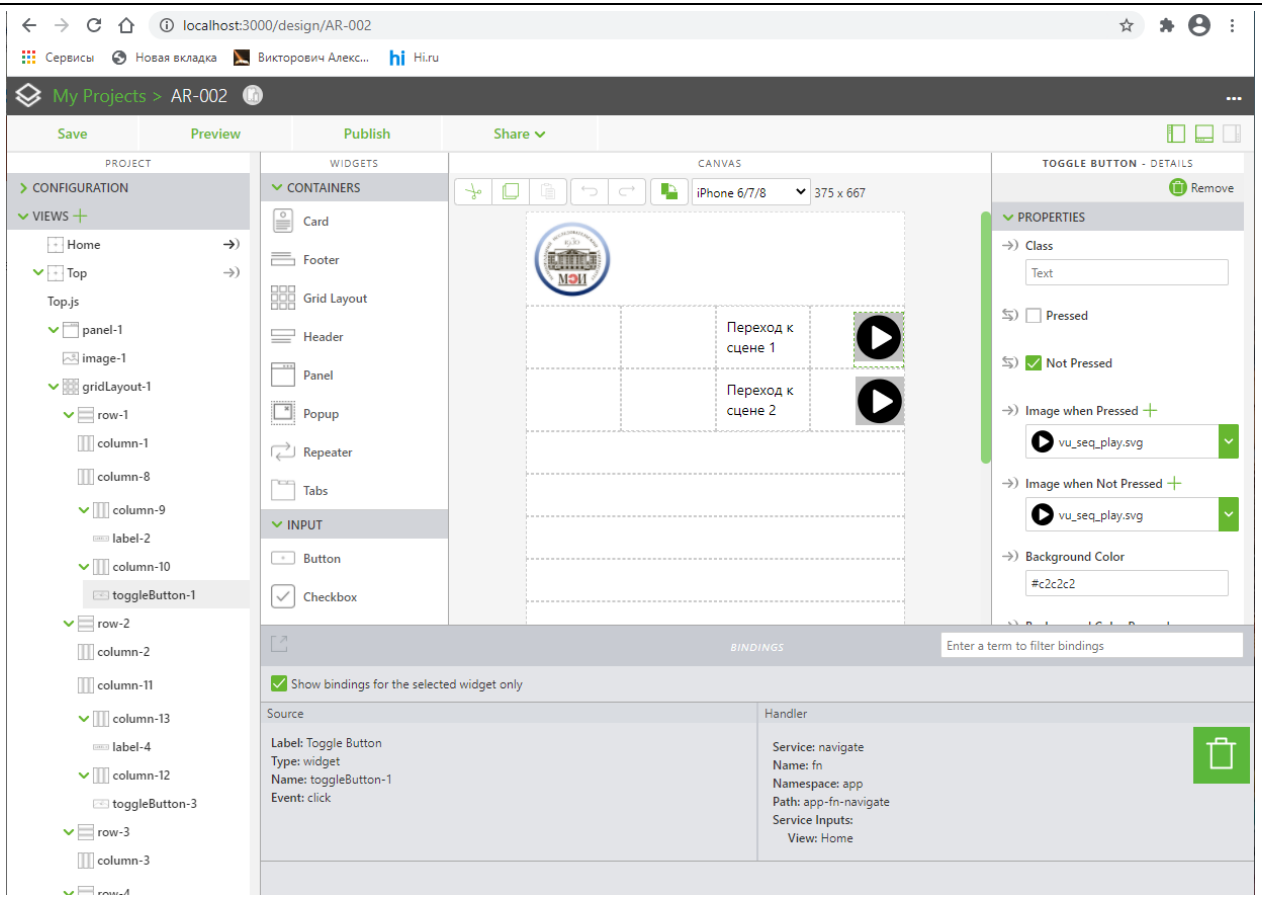
- PROJECT:** A tree view on the left showing the project structure. The selected item is "model-1" under "3D Container".
- WIDGETS:** A list of widgets available for use, including Image Target, Model Target, Spatial Target, ThingMark, 3D Gauge, 3D Image, 3D Label, Model, and Model Item.
- CANVAS:** A 3D view of a teal-colored device model on a gray grid floor. A small image target is visible on the floor. The scene polygon count is 319749.
- MODEL - DETAILS:** A panel on the right showing the properties of the selected model. The "Resource" is set to "TestV1.pvz". Other properties include Scale (1.0), X, Y, and Z Coordinates (0, 0, 0.002), and X Rotation (0).
- BINDINGS:** A table at the bottom showing the bindings for the selected widget. It lists two bindings for "Toggle Button" widgets, each with a "click" event and a corresponding service handler.

Source	Handler
Label: Toggle Button Type: widget Name: toggleButton-1 Event: click	Service: play Name: model-1 Type: ui Path: app.view["Home2"].wdg["model-1"].svc.play
Label: Toggle Button Type: widget Name: toggleButton-2 Event: click	Service: rewind Name: model-1 Type: ui Path: app.view["Home2"].wdg["model-1"].svc.rewind

Шаг 41.

Осталось организовать переходы между сценами.

Начинаем со сцены **Тор – Титульной страницы**. В меню на этой сцене пока сделан только переход к сцене 1.



Шаг 42.

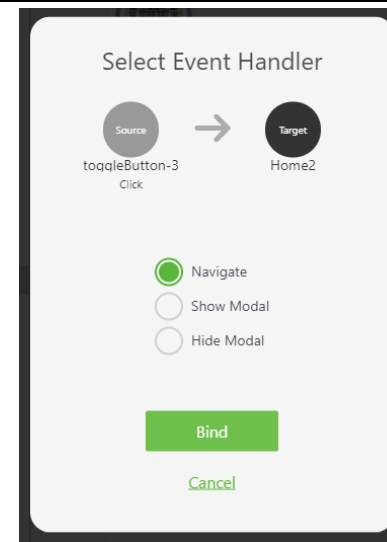
Выполним связывание второй кнопки со сценой 2.

The screenshot displays the Vuforia Studio interface for a project named 'AR-002'. The interface is divided into several panels:

- Left Panel (Project Tree):** Shows a hierarchy of widgets including rows, columns, panels, and images. The 'Home2' widget is selected, indicated by a red arrow pointing to its icon.
- WIDGETS Panel:** Lists various widget types such as Card, Footer, Grid Layout, Header, Panel, Popup, Repeater, Tabs, Button, Checkbox, Select, Slider, Text Area, Text Input, Toggle, and Toggle Button.
- CANVAS:** Shows a preview of the application on an iPhone 6/7/8 screen. It features a grid layout with two buttons labeled 'Переход к сцене 1' and 'Переход к сцене 2'. A yellow box highlights a text area at the bottom of the canvas with the text 'Со всей этой фигней нужно: FIO'. A red arrow points from this text area to the 'Home2' widget in the left panel.
- Right Panel (Properties):** Shows the configuration for the selected widget. It includes settings for background color, width (48px), height (48px), visibility, and disabled state. The 'EVENTS' section is expanded, showing actions like 'Click JS', 'Pressed JS', and 'Unpressed JS'. The 'GRID ACTIONS' section includes 'Add Column' and 'Add Row'.

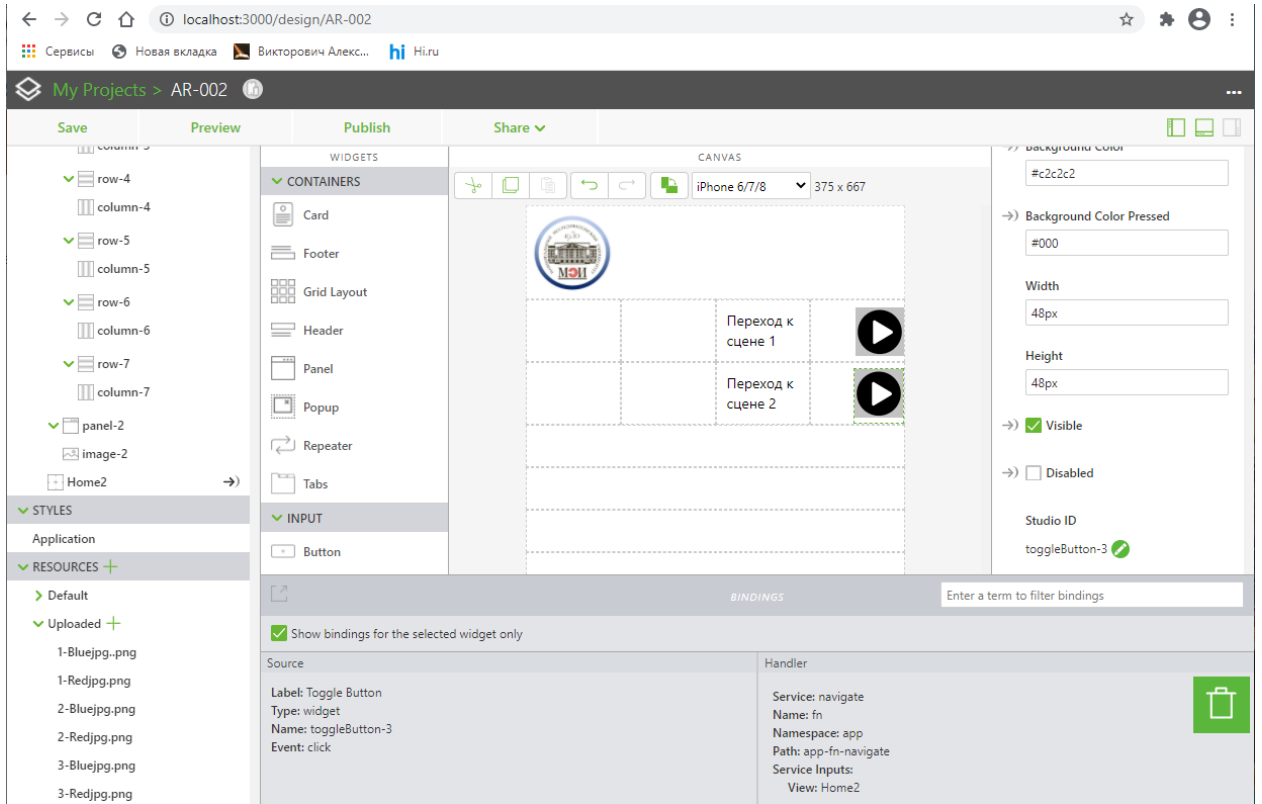
Шаг 43.

Обработчик событий для **toggleButton-3** → связываем событие **Click** с навигацией.



Шаг 44.

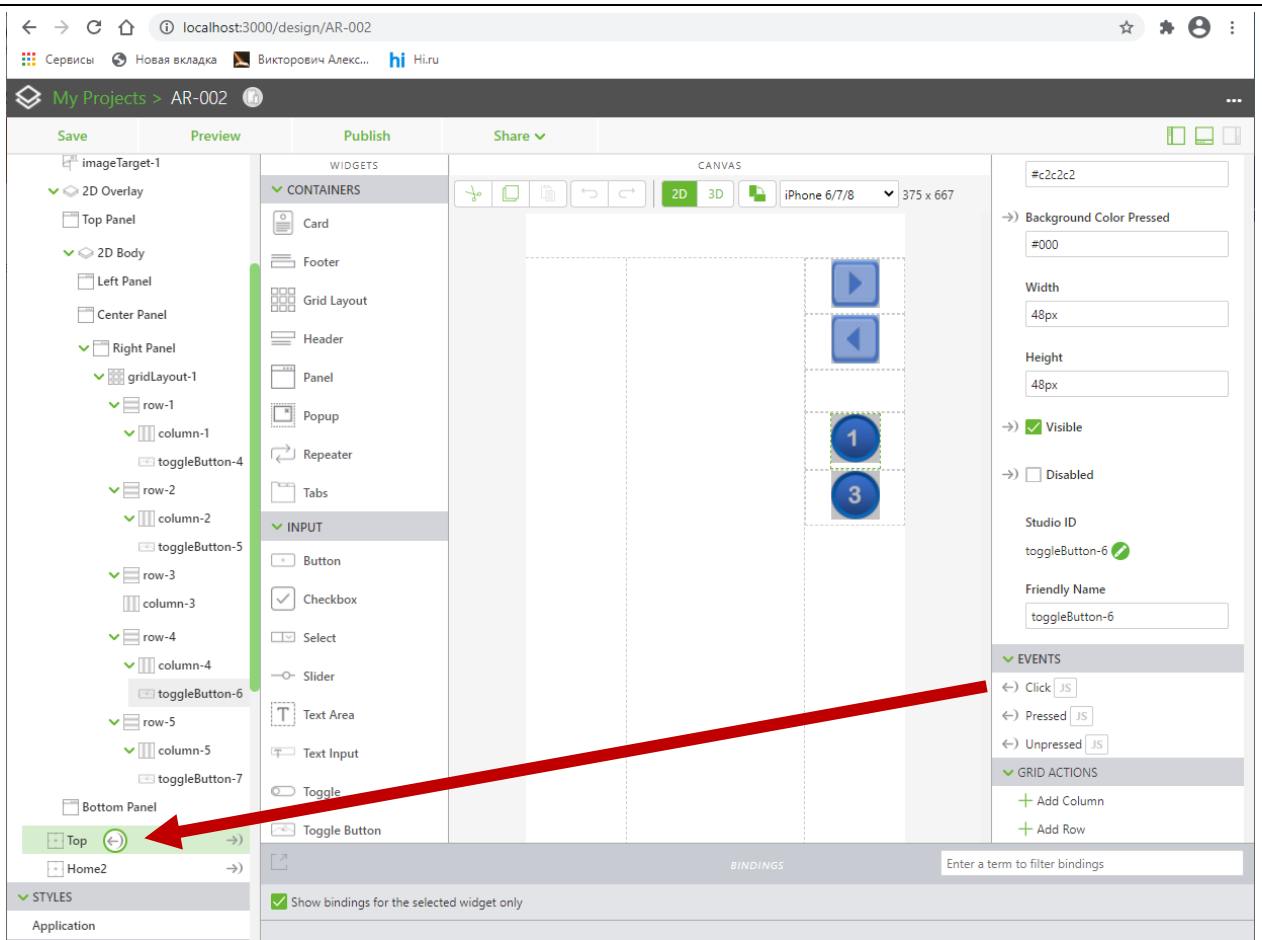
Проверяем наличие связи →



Шаг 45.

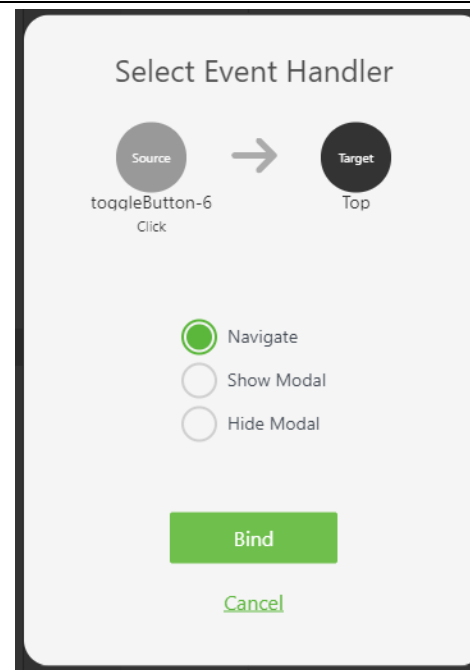
Переходим в сцену **Home** → выполняем назначение переходов на сцену **Top** и на сцену **Home2**.

Переходим в **2D-Канву**. Назначаем кнопки **toggleButton-6** и **toggleButton-7** для переходов на сцену **Top** и сцену **Home2** соответственно.



Шаг 46.

Обработчик событий для **toggleButton-6** → связываем событие **Click** с навигацией.



Шаг 47.

Аналогично связываем **toggleButton-7** с **Home2**.

The screenshot shows the Vuforia Studio interface for a project named 'AR-002'. The interface is divided into several panels:

- Left Panel:** A tree view showing the hierarchy of the 2D Body. It includes 'Left Panel', 'Center Panel', 'Right Panel', and a 'gridLayout-1' containing rows and columns. 'toggleButton-7' is selected in the 'column-5' of 'row-5'.
- WIDGETS Panel:** A list of available widgets. Under the 'INPUT' category, the 'Button' widget is selected.
- CANVAS:** A central workspace showing a mobile device view (iPhone 6/7/8) with a 375 x 667 resolution. Three blue buttons are visible on the screen, with the bottom one labeled '3' and highlighted with a dashed border.
- Properties Panel:** On the right, the properties for the selected widget are shown. The 'Image when Not Pressed' property is set to '3-Redjpg.png'. The 'Background Color' is '#c2c2c2' and the 'Background Color Pressed' is '#000'. The 'Width' and 'Height' are both '48px'. The 'Visible' checkbox is checked.
- BINDINGS Panel:** At the bottom, a table shows the binding configuration for the selected widget.

Source	Handler
Label: Toggle Button Type: widget Name: toggleButton-7 Event: click	Service: navigate Name: fn Namespace: app Path: app-fn-navigate Service Inputs: View: Home2

Шаг 48.

Переходим в сцену **Home2** и выполняем аналогичные действия для переходов на **Тор** и на **Home**.

Завершив эти действия, убедитесь в том, что у соответствующих кнопок появились связи по навигации между сценами, проверьте правильность реализации сценария в режиме **Preview**.

Ниже показаны все связи для сцены **Home**. Обратите внимание – для такого просмотра связей флажок в поле

Show bindings for the selected widget only

должен быть

СНЯТ.

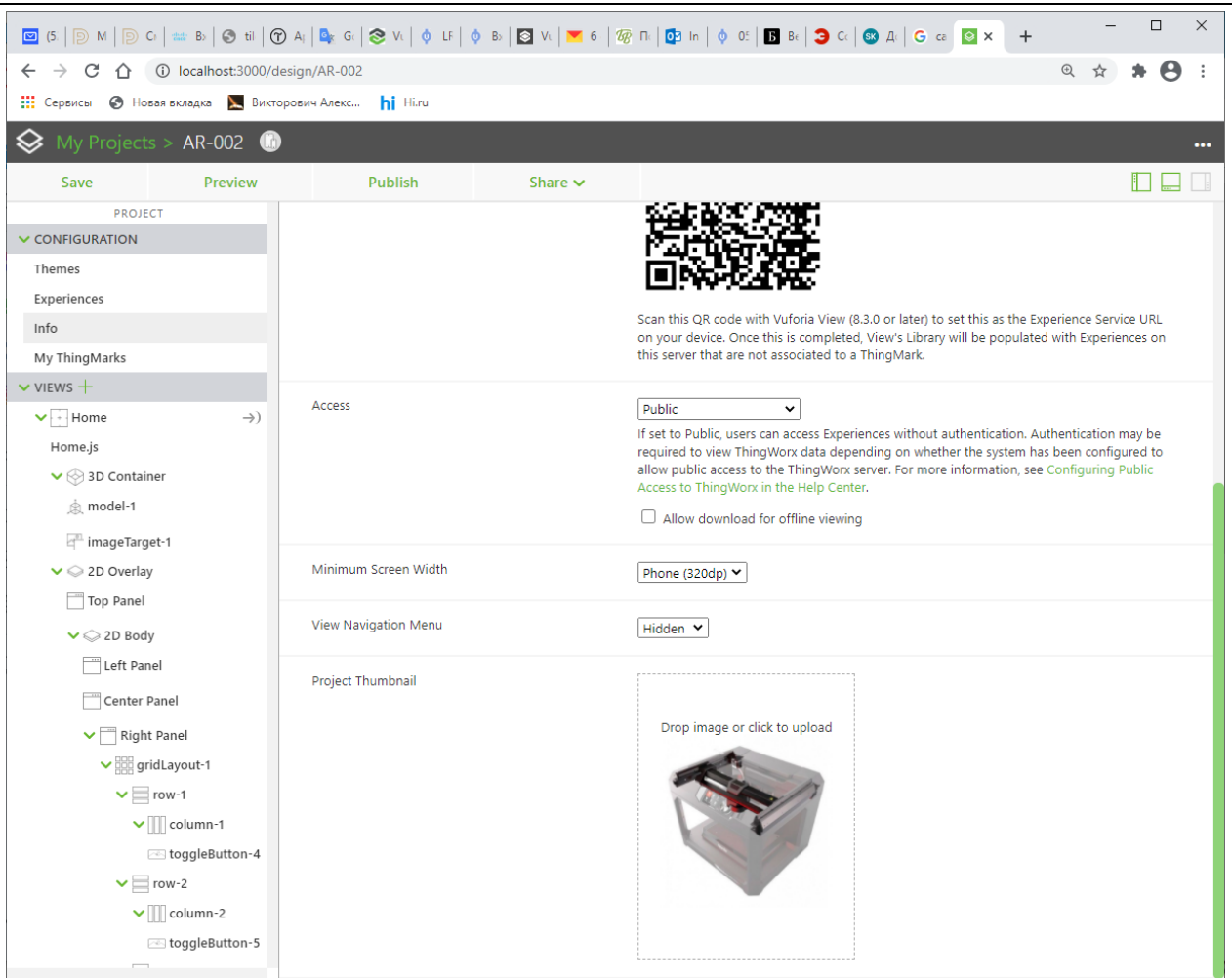
The screenshot shows the Vuforia Studio interface for project 'AR-002'. The 'BINDINGS' table is expanded to show the following configuration:

Source	Handler
Label: Toggle Button Type: widget Name: toggleButton-4 Event: click	Service: play Name: model-1 Type: ui Path: app.view["Home"].wdg["model-1"].svc.play
Label: Toggle Button Type: widget Name: toggleButton-5 Event: click	Service: rewind Name: model-1 Type: ui Path: app.view["Home"].wdg["model-1"].svc.rewind
Label: Toggle Button Type: widget Name: toggleButton-6 Event: click	Service: navigate Name: fn Namespace: app Path: app-fn-navigate Service Inputs: View: Top
Label: Toggle Button Type: widget Name: toggleButton-7 Event: click	Service: navigate Name: fn Namespace: app Path: app-fn-navigate Service Inputs: View: Home2

Шаг 49.

Переходим к публикации проекта. Перед публикацией полностью отлаженного Проекта **AR-002** рекомендуется в **CONFIGURATION** → **Info** → **View Navigation Menu** выбрать опцию **Hidden**, что означает скрыть меню «по умолчанию» - т.е. меню навигации **Vuforia View**, оставив только те возможности перехода между сценами, которые предусмотрены реализуемым сценарием.

Зададим специальный **Title (AR-002-Menu)** для публикуемого проекта и выберем картинку для его тумбнейла в списке доступных в библиотеке **Vuforia View**.



The screenshot shows the Vuforia Studio configuration interface for project AR-002. The interface is divided into a left-hand navigation pane and a main configuration area. The navigation pane shows a tree view of the project structure, including 'CONFIGURATION' and 'VIEWS'. The main configuration area displays various settings for the project, such as 'Access' (set to Public), 'Minimum Screen Width' (set to Phone (320dp)), and 'View Navigation Menu' (set to Hidden). A QR code is visible at the top right, and a 'Project Thumbnail' section shows a placeholder for an image with a 3D model of a toolbox.

Шаг 50.

Теперь Проект можно опубликовать и проверить работоспособность получившегося упрощенного **AR-Приложения** с помощью **Vuforia View**.



3. Практическое задание

В данном подпункте **Лабораторной работы № 3** предлагается разработать **Проект AR** для информационной поддержки процессов возможного сервисного обслуживания устройства **3D-Принтера MakerBot Replicator 2**.

Начальные условия:

1. **Архив моделей** для целевого устройства **TestV1.pvz** в формате **.pvz**, содержащий всю необходимую информацию для экспорта в **Vuforia Studio**, включая модель устройства и две анимированные последовательности (**sequences**) в формате **.pvi**. Последовательности определены на составе базовой целевой модели (сборки - **3D-Принтера MakerBot Replicator 2**), импортированы из **Creo Illustrate** и иллюстрируют две технологические операции:
 - **Sigma Tile Build** - пошаговая демонстрация процесса **3D принтинга** (аддитивного производства) кожуха (корпус и крышка) для размещения в нем микрокомпьютера **RPi** и блока датчиков **SenseHat**;
 - **Head Removal** - пошаговая демонстрация процесса замены рабочего органа (головки) **3D-Принтера MakerBot Replicator 2**;
2. **Разработанные** (и опубликованные) на предыдущих Лабораторных работах **AR-Проекты** с использованием различных типов таргетирования для базовой модели **3D-Принтера**. В базовом Проекте, в одной сцене – элементе типа **VIEW**, использованы различные элементы **3D** Контента: **3D**-модель устройства, анимированная последовательность **Sigma Tile Build**, информационное наполнение, связанное с данной моделью в виде справочной информации, наложенное на элементы типа **3DImage**, **3DLabel**, **3DGauge**; и - **2D** Контента: элементы управления компонентами контента внутри сцены – слайдеры, кнопки управления состоянием компонент и пр.

Задание на выполнение Дополнения № 2 ЛР №3:

1. **Необходимо разработать многоуровневое, мультитаргетное AR-Приложение** для информационной поддержки **сервисных операций** (например → сборка, ремонтные операции, эксплуатация) применительно к **целевому устройству 3D-Принтер MakerBot Replicator 2**, основываясь на Индивидуальном задании, разосланном каждому обучающемуся перед выполнением ЛР №2.
2. **Среда реализации:**
 - Редактор **Vuforia Studio** → **WEB-Просмотрщик localhost:3000**
 - Тренировочный сервер **ThingWorx Experience Service (TWx ES)**, который находится по адресу <http://rrr.kkk.com:2019>

3. Разрабатываемое AR-Приложение должно содержать несколько сцен (объектов типа **VIEW**):

- **Сцены**, содержащие **2D-** и **3D-** Контент, распределенный по страницам аналогично ранее созданным Проектам;
 - **К уже созданной** для базовой модели **3D-Принтера MakerBot Replicator 2** сцене предлагается добавить (как минимум) еще одну. В новой сцене к модели будет привязана вторая анимационная последовательность по замене у принтера рабочего органа (головка) – **Head Removal**.
 - **Разрабатываемый** Проект в обязательном порядке дополнительно включает **Титульную страницу**. Контентом такой страницы может быть, например, начальное описание Приложения, логотип компании, сведения о разработчиках т.п. и, самое главное, корневое меню, инициирующее переход к остальным страницам (сценам);
 - **При этом каждая из сцен** (объектов **VIEW**) разрабатываемого Проекта, не только заглавная страница, но и остальные сцены – [под-] **AR-Приложения** – также должны включать элементы управления навигацией по сценарию **AR-Приложения**;
 - **Рекомендуемый** метод таргетирования при разработке данного **AR-Проекта** – **Image Target**. Эксперименты/исследование по возможности использования метода таргетирования **Model Target** – приветствуются и поощряются.
4. **При разработке** данного Проекта обучающиеся могут в качестве шаблона использовать один из ранее разработанных Проектов (дублицирование → модификация, см. ЛР №3), или разрабатывать Проект заново, распределяя **3D-** и **2D-** контент и элементы управления компонентами контента внутри сцен между несколькими страницами.
5. **Обучающиеся** самостоятельно разрабатывают сценарий AR-Проекта, решая задачу предоставления гипотетическому заказчику **AR-Experience** всей необходимой информации для успешного и эффективного выполнения исполнителем сервисных операций по обслуживанию целевого устройства **3D-Принтера MakerBot Replicator 2**.
6. **Разработанный AR-Проект** должен быть опубликован на тренировочном сервере **TWx ES**, для проверки его работоспособности преподавателем. Не забудьте прислать ему свои **Image Target'ы!!!**