

# Документация для партнеров Sense Capital

В этой книге собраны статьи, которые помогут в работе партнерам Sense Capital

- [Разработка пространств](#)
  - [Как создать сцену для Sense Tower - видео](#)
  - [Процесс разработки сцены партнером Sense](#)
  - [Требования, предъявляемые к сцене для добавления в SenseTower](#)
  - [Рекомендации к моделям для сцен Sense Tower](#)
  - [Формирование архива с моделями и сценой](#)
  - [Требования к загружаемым объектам](#)

# Разработка пространств

Как добавлять свои пространства в Sense Tower

Разработка пространств

# Как создать сцену для Sense Tower - видео

Наглядный подробный пример создания сцены\*:

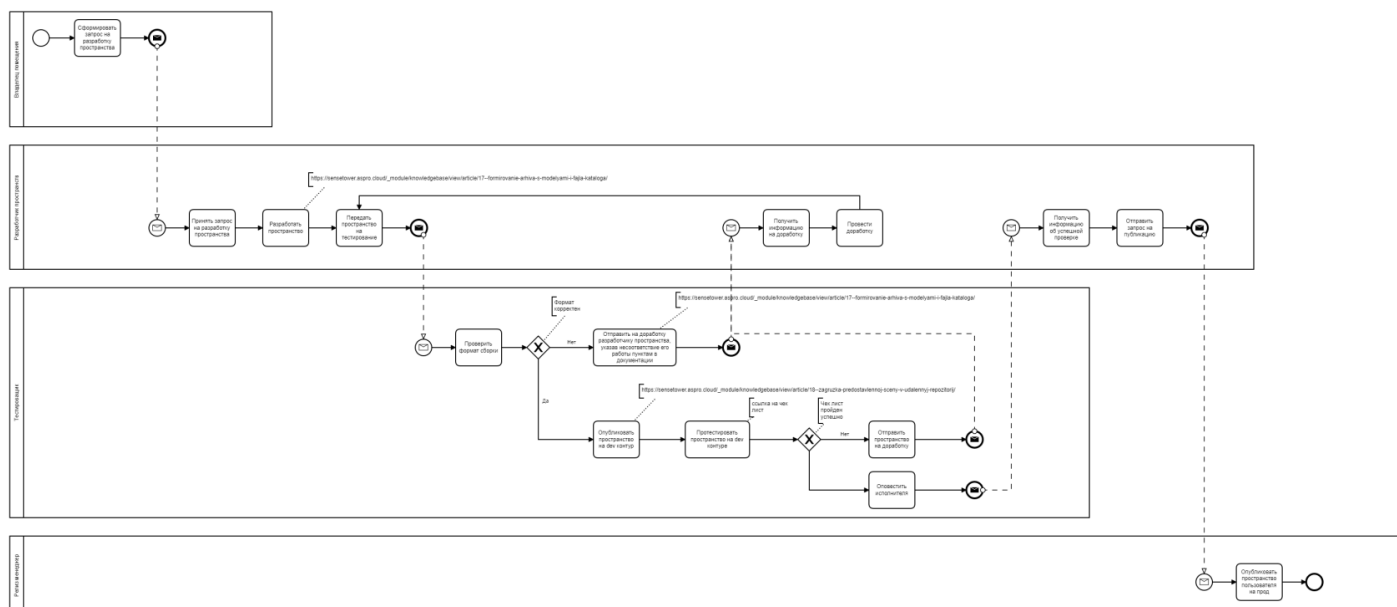
<https://www.youtube.com/watch?v=B6VO32ky7eI>

\* - В видео указана версия Unity 2021.3.16, но в данный момент применяется версия Unity 2021.3.25

# Процесс разработки сцены партнером Sense

В деталях процесс можно посмотреть здесь: <https://stormbpmn.com/app/diagram/5f875d97-a585-418a-9ce8-5609811d64d0>

Будем благодарны за рекомендации по его улучшению и замеченные неточности.

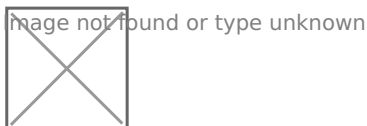
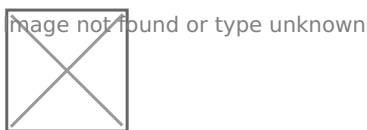


# Требования, предъявляемые к сцене для добавления в SenseTower

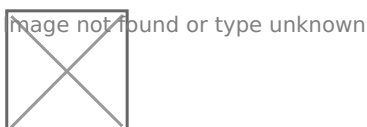
Все сцены и объекты собираются в версии [Unity 2021.3.25f1](#) с использованием URP с подключенными модулями Android и Linux сборки

При установке необходимо выбрать устанавливаемые модули (указаны на скриншотах ниже):

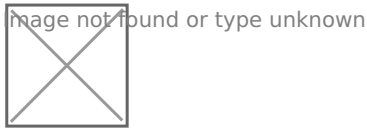
- Android Build Support
- OpenJDK
- Android SDK & NDK Tools
- Linux Build Support (IL2CPP)
- Linux Build Support (Mono)
- Linux Dedicated Server Build Support
- Windows Build Support (IL2CPP)
- Windows Dedicated Server Build Support



Если редактор Unity уже установлен, необходимо убедиться, что у него установлены все указанные модули и, при необходимости, установить отсутствующие. Чтобы это сделать, нажмите на шестерёнку и выберите AddModules:



Если для создания сцены используется пустой проект (не из репозитория <https://github.com/Sense-Capital/RemoteContentSample>), необходимо создать новый проект, выбрав шаблон 3D (URP):



## Быстродействие

Быстродействие сцены напрямую зависит от её загруженности.

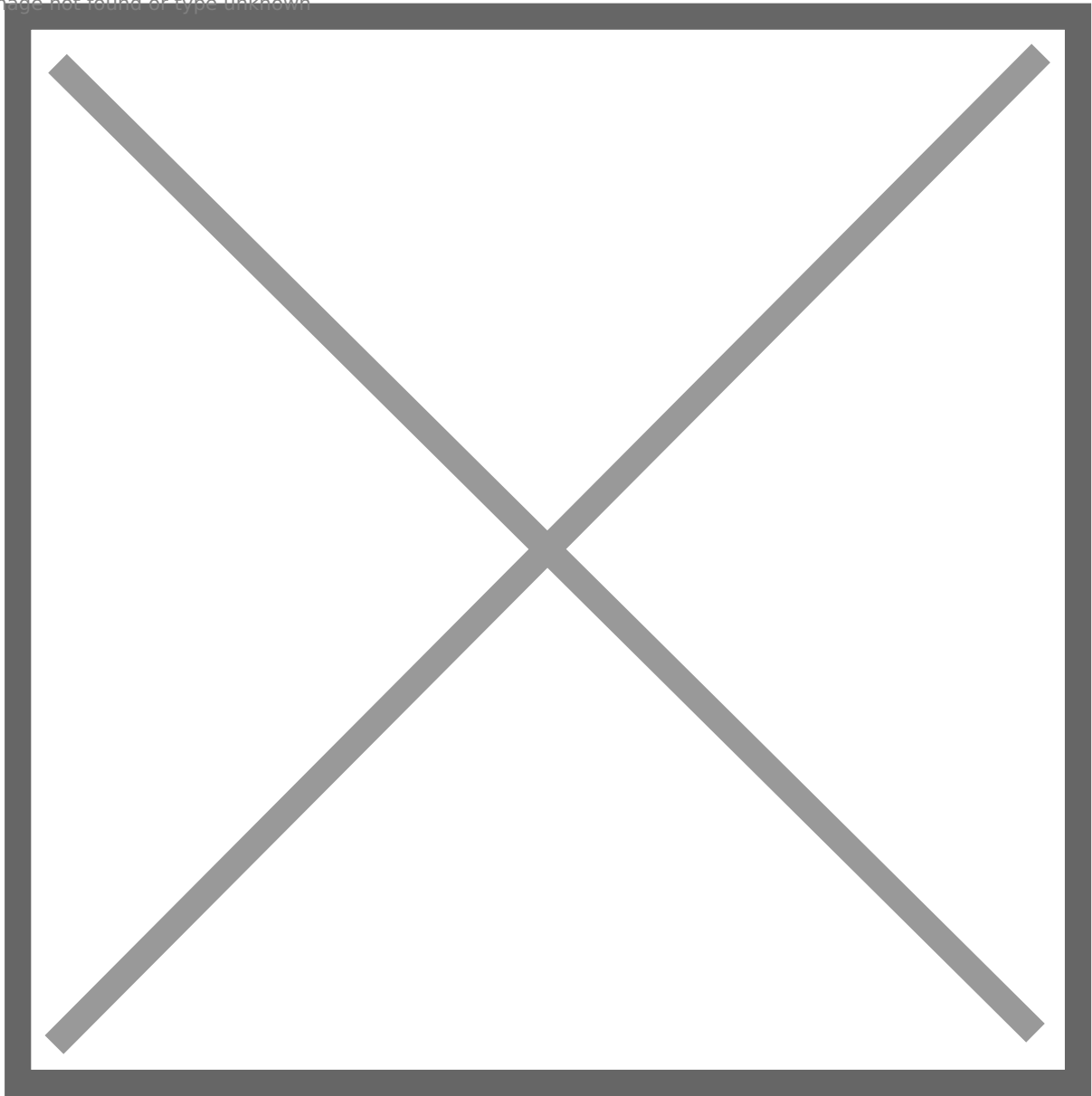
Общая цель — достичь приемлемой частоты кадров на разных устройствах. В Oculus Quest 1 и 2 в идеале сцена будет работать со скоростью 72 кадров/с со всех точек зрения, когда сцена заполнена, хотя диапазон 60–72 кадров/с часто является более реалистичной целью.

При формировании сцены для лучшего быстродействия необходимо учитывать требования предъявляемые устройством Oculus, они подробно описаны здесь - [Требования к сцене Unity](#) Из главных, это количество отрисовок в районе 80-200 и количества треугольников до 400к(чем меньше тем лучше).

Но следует понимать, что это общее количество треугольников, и рекомендуется формировать сцену таким образом, чтобы пользователь мог видеть в один момент не более 50 000 - 70 000 треугольников.

Количество треугольников можно проверить в окне Game, нажав на кнопку Stats. Предварительно необходимо направить камеру таким образом, чтобы вся сцена попала в обзор.

Image not found or type unknown



Проверять количество треугольников рекомендуется в Unity. В Blender возможны искаженные данные из-за применения модификаторов и инстансов.

Unity поддерживает импорт моделей напрямую из Blender. [Импорт модели из Blender в Unity.](#)

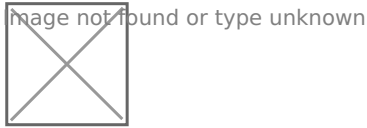
Текстуры должны иметь разрешение до 2к. Исключение - skybox-текстура окружающего ландшафта (для неё можно установить разрешение вплоть до 8k).

При соблюдении данных требований в приложении SenseTower будет поддерживаться необходимое количество FPS и соответственно максимальное удовольствие от нахождения в пространстве.

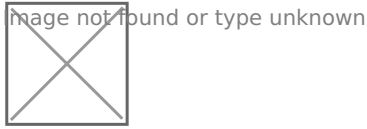
#### **Стандартные средства повышения быстродействия:**

**1) Occlusion Culling** это функция, отключающая рендеринг тех объектов, которые в данный момент не видит камера (они закрыты другими объектами). В компьютерной 3D графике это

не происходит автоматически. Для использования Occlusion Culling необходимо задать все неподвижные объекты на сцене как Static.



Далее вызвать окно Occlusion Culling в пункте меню Windows:



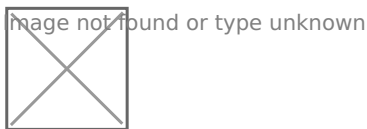
И запечь Occlusion Culling:



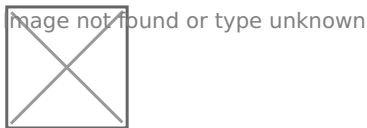
**Важно: При любом изменении объектов на сцене (добавление/изменение/удаление объектов) операцию запекания Occlusion Culling необходимо провести заново.**

**2) Запекание света** необходимо, чтобы зафиксировать освещение в текстурах и отключить источники света на сцене. Для этого нужно:

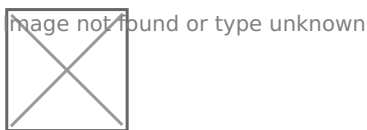
В окне Project найти используемые на сцене модели (в формате fbx или blend), включить и применить на них опцию Generate Lightmap UVs



Для корректной работы сцены необходимо все источники света сделать типа Baked

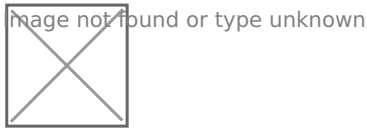


Далее вызвать окно Lighting:

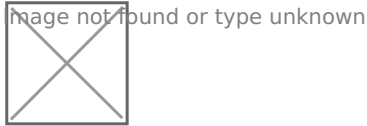


Рекомендуется выбрать следующие настройки запекания света:

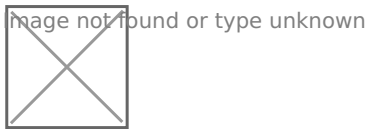




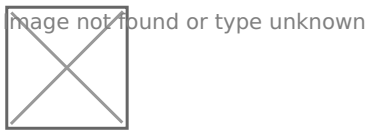
Также рекомендуется сменить настройки Source со Skybox на Color и Custom в указанных ниже пунктах для более корректного отображения цветов при открытии сцены через Sense Tower. Таким образом, освещение сцены не будет зависеть от Skybox Sense Tower.



После данных настроек нажать кнопку Generate Lighting внизу окна Lighting. Процесс запекания с данными настройками может несколько минут. После завершения запекания внизу окна появится надпись о количестве получившихся карт запекания. В данном случае видно, что получились две карты запекания размером 2048 пикселей, занимающие 21,3Мб, что является приемлемым результатом.



В случае, если карты запекания занимают более 30Мб, рекомендуется понизить разрешение карты запекания и перезапечь освещение заново:



# Рекомендации к моделям для сцен Sense Tower

## Рекомендации к моделям для сцен Sense Tower

### 1. Цветовое решение

Цветовое решение моделей должно соответствовать общей стилистике материалов проекта. Не должно кричаще выбиваться, из разработанного интерьерного решения: Возможна вариативность по цветам, однако материалы должны быть соответствующими: - дерево (панели, планки), бетон (светлый), ковролин (темный) мрамор — светлый/темный. Безусловно возможна акцентная вариативность, однако без ущерба общему соответствию стилистике башни.

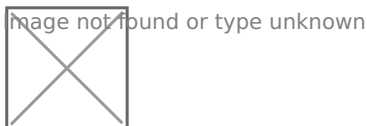
### 2. Требования к сценам и моделям

#### 2.1. Общие рекомендации.

Сцена разделена на 2 типа объектов — динамические и статические. Динамические объекты, - это объекты, которые должны интерактивно взаимодействовать с пользователем: стулья, столы, кресла, пуфики, иные объекты. Статические объекты — это объекты которые выполняют функцию фона: стены, окна, деревья, полы, потолки, часть обстановки за окнами.

#### 2.2. Динамические объекты

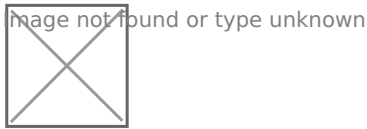
Про создании модели любого объекта для башни необходимо соблюдать необходимый и достаточный баланс между количеством полигонов и внешним видом модели. Примеры правильного и неправильного использования полигонов:



Для корректного отображения высокополигональной модели может потребоваться ее ретопология, и подготовка карт диффуза, АО, карты нормалей. Обязательно наличие карт диффуза и карт АО для всех динамических моделей. Карта диффуза должна быть не больше 1024\*1024 точек, должна иметь координаты UV0. Карта АО должна быть не больше

1024\*1024 точек, может иметь как идентичные карте диффуза координаты (UV0), так и иные координаты — UV1, UV2. Для однородных поверхностей предпочтительно использовать тайленные карты текстур разрешения до 512\*512 точек и карты AO и Normal для отображения псевдорельефа поверхности.

Пример использования тайленной карты диффуза размером 256\*256 точек, умноженной на цвет, с использованием карт AO и Normal для отображения рельефа



### 2.3 Статические объекты

Принцип построения статических объектов аналогичен динамическим, однако статические объекты подразумевают наличие карты освещения (lightmap). Lightmap – это запеченная карта диффузного освещения объекта. Важно именно diffuse не glossy, specular или другие каналы — важен именно канал diffuse. Lightmap должна использовать координаты UV1, поверхность использующая lightmap должна иметь тайленную текстуру diffuse до 1024\*1024 точек, возможно использование карт AO и Normal но важно чтобы карты AO и Normal не использовали слой UV1 – для статических объектов этот слой зарезервирован для карты освещения (lightmap).

## 3. Общие принципы организации сцены

3.1 Каждый динамический объект должен представлять из себя отдельный FBX файл — клонировать его нужно уже в Unity

3.2 Статические объекты могут быть одним файлом FBX, но внутри FBX не должны быть единым объектом. (не применять attach (max)/join (blender/maya))

## 4. Требования к текстурам

### 4.1 общие требования

Все текстуры должны быть в формате PNG 8bit, альфа канал должен быть только на тех текстурах, где он необходим художественно. Максимальный размер текстуры может достигать 2048\*2048 точек. Допустимо использование кратного масштабирования, например: 128\*128, 128\*64, 2048\*512. Эти требования обусловлены алгоритмами mipmapping'a.

### 4.2 Рекомендации по текстурированию

Необходимо соблюдать, как в случае с ретопологией, необходимое и достаточное разрешение текстур, стремясь к наименьшему без видимой художественной потери. То есть если модель удовлетворительно выглядит с текстурой 512\*512 точек — этого достаточно и нет необходимости использовать текстуру большего разрешения. Приветствуется

использование тайловых черно-белых текстур низкого разрешения для последующего придания цвета с помощью параметров материала в Unity и псевдорельефа с помощью карт AO и Normal.

# Формирование архива с моделями и сценой

## Инструкция по созданию сцены для загрузки в башню SenseTower

В текущей документации будет описан процесс создания проекта, который нужно будет передать вместе с сформированными в процессе действий из этой публикации файлами и папками.

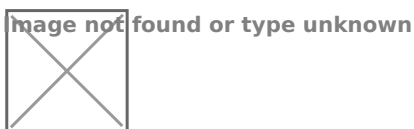
Ниже представлены алгоритмы действий при создании удаленной сцены в башне SenseTower в двух вариантах. Первый через клонирование проекта с репозитория SenseTower с заранее прописанными параметрами, второй через самостоятельное создание проекта и выставление необходимых параметров вручную.

### Первый вариант

В начале работы необходимо убедиться что у вас [установлена версия Unity 2021.3.25](#) с [подключенными модулями Android, Windows и Linux](#) сборки.

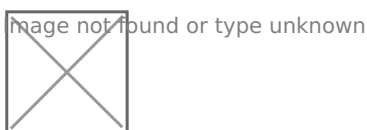
Проект с примерами хранится по пути. <https://github.com/Sense-Capital/RemoteContentSample>

**1)** Необходимо осуществить Fork проекта и скачать на ПК. В этом проекте заранее выставлены все параметры и настройки, необходимо будет только ввести свои параметры для идентификации папок и файлов

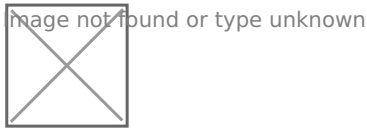


**2)** Для того, чтобы сцену можно было загрузить как удаленный объект, необходимо указать название проекта в настройках профиля Addressables. Делается это следующим образом.

Для начала перейдем в настройку профиля:



В появившемся окне Addressables Profiles необходимо выбрать профиль GithubPages:



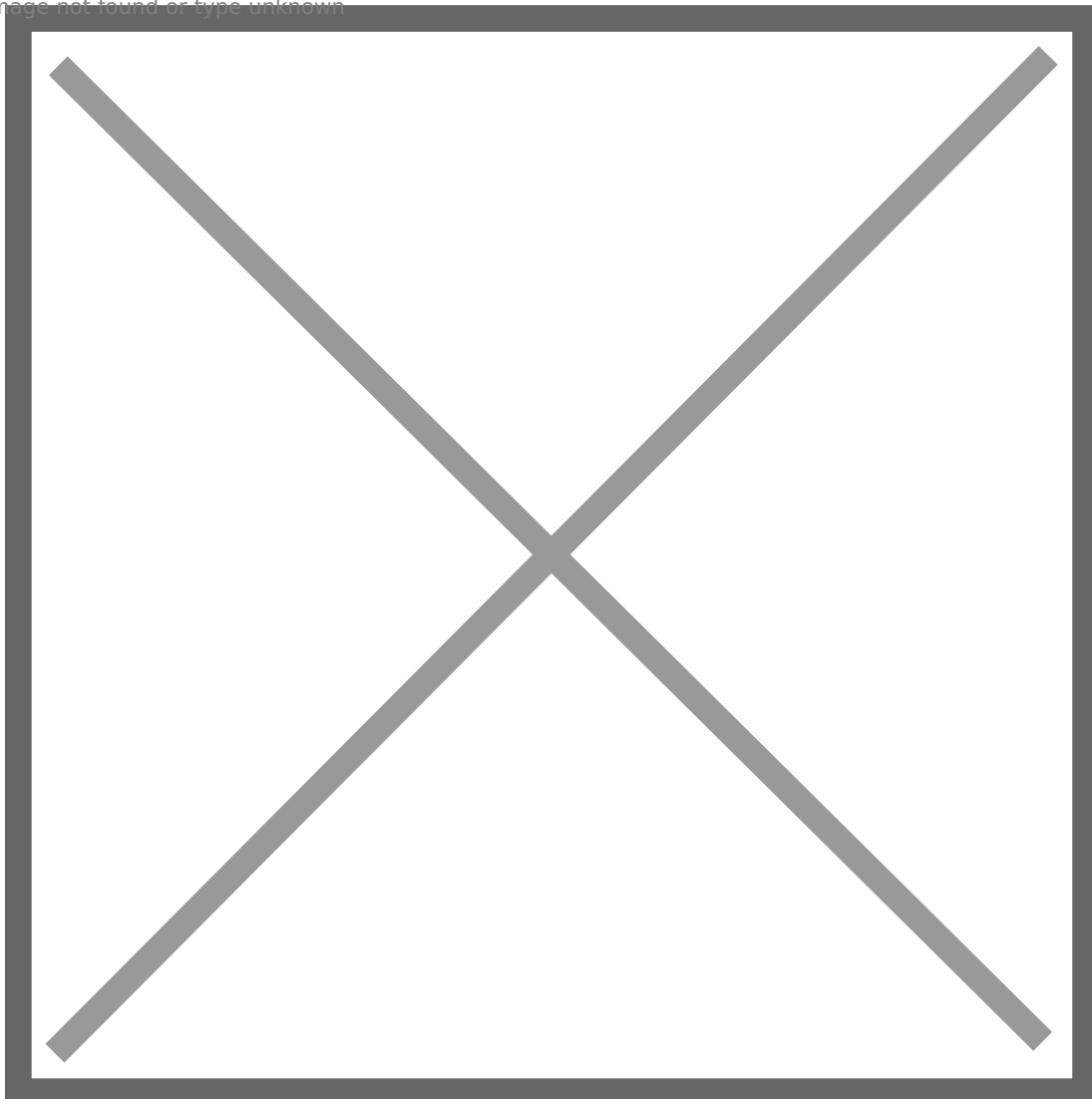
Далее необходимо придумать уникальное название для каталога вашего проекта на удалённом репозитории. Из названия должно быть ясно назначение загружаемой сцены. Оно должно состоять только из букв латинского алфавита, цифр, знаков подчёркивания, тире, и не иметь пробелов. Рекомендуется указывать имя в формате <Название компании>\_<Название сцены>. **В примере ниже используется название `OurBestCompany_RemoteScene1`, ваше название должно быть другим и отражать именно специфику вашей сцены!**

Это название необходимо указать в поле **Remote.BuildPath** и в поле **Remote.LoadPath** вместо присутствующего в данных строках параметра [ProjectName]. Пример:

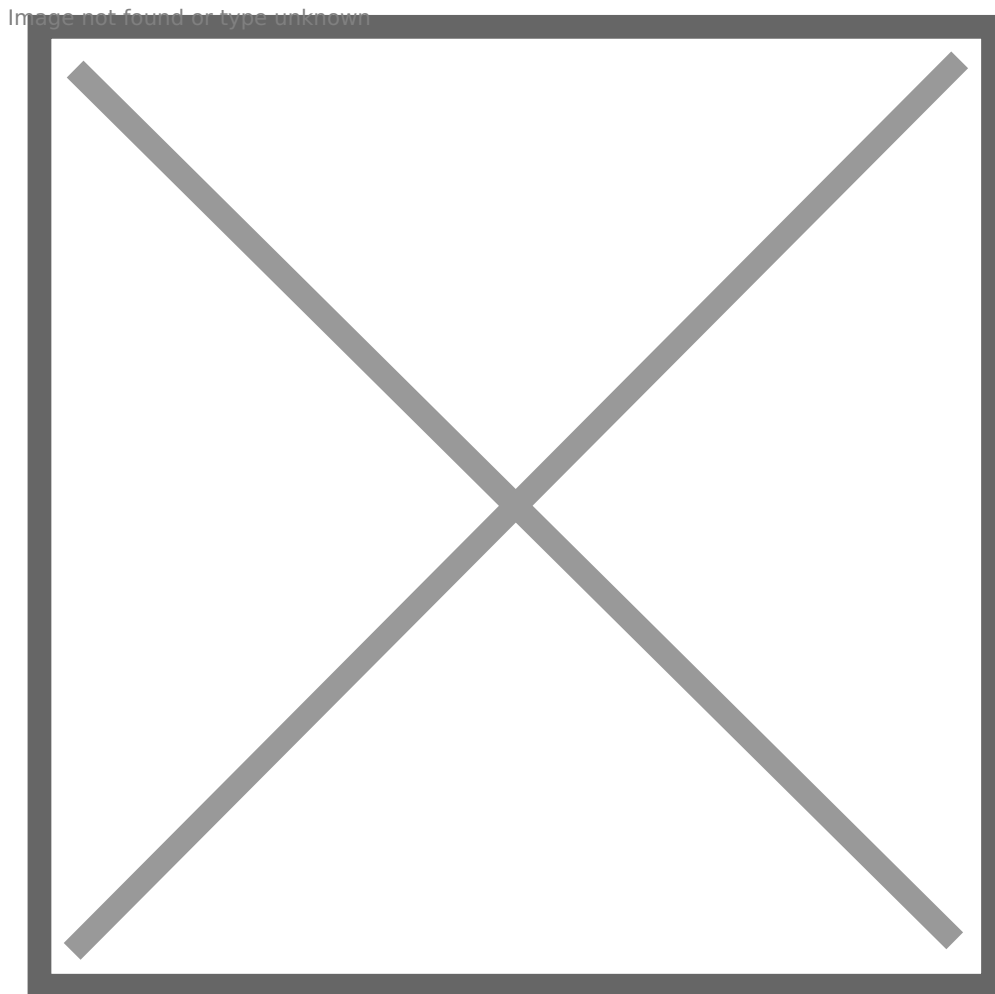


После этого перейдем в общие настройки Addressables

Image not found or type unknown



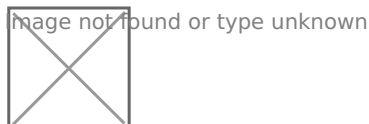
В открывшихся параметрах объекта Addressable Asset Settings в инспекторе необходимо также прописать указанное ранее название:



На этом настройка проекта завершена, теперь нужно при необходимости наполнить сцену возможностью перемещения и садиться на стулья и добавить файлы сцены в группу, предназначенную для удаленной загрузки в башню SenseTower. Рассмотрим эти шаги подробнее.

### 3) Наполнение сцены

В скачанном проекте есть пустая специально заготовленная сцена с настройками света и заготовленными настройками для формирования удаленной сцены, она находится по пути Assets/Scene

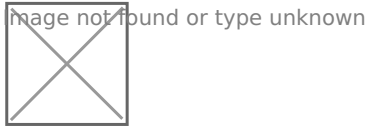


Эта сцена предназначена для наполнения объектами, формирующими пользовательское пространство. Необходимо все модели размещать в этой сцене.

Необходимо убедиться что в сцене присутствуют только объекты моделей

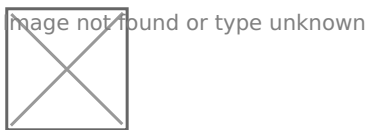
пользовательского пространства. На объектах не должно быть скриптов и компонентов Unity, выполняющих какую-либо дополнительную логику (например, объект камеры или Volume).

Также необходимо проверить коллайдеры. На сцене не должно быть лишних коллайдеров, перекрывающих большой объём пространства, в котором может перемещаться пользователь (пример на скриншоте ниже). В противном случае объекты в такой зоне могут вести себя некорректно.



После того, как на сцену были добавлены объекты-модели пользовательского пространства, сцена должна быть выровнена относительно точки (0, 0, 0).

Необходимо выравнивать сцену так, чтобы центральная точка объекта находилась в месте желаемого появления пользователя чуть выше уровня пола, а глобальный вектор forward (синий вектор) указывал бы туда, куда пользователь должен смотреть при появлении. Проще всего это будет сделать, добавив на сцену временный объект (например, сферу) в точке (0, 0, 0).



После всех выравниваний не забудьте удалить временный объект.

### Требования предъявляемые к сцене

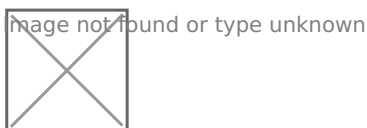
Быстродействие сцены напрямую зависит от её загруженности.

Рекомендации по настройке сцены приведены здесь:

[https://sensetower.aspro.cloud/\\_module/knowledgebase/view/article/11--trebovaniya-predyavlyaemye-k-scene-dlya-dobavleniya-v-sensetower/](https://sensetower.aspro.cloud/_module/knowledgebase/view/article/11--trebovaniya-predyavlyaemye-k-scene-dlya-dobavleniya-v-sensetower/)

**a)** В помещение необходимо обязательно поставить дверь выхода. Через дверь выхода пользователь сможет осуществлять переход из помещения в общий холл. Холл - точка доступа ко всем помещениям башни, включая и создаваемое помещение, поэтому необходимо дать возможность пользователю туда вернуться.

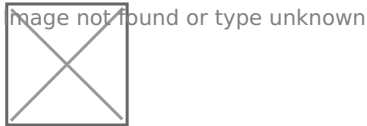
Для того, чтобы разместить дверь выхода, создаём на сцене новый пустой объект с названием **ExitDoor**.



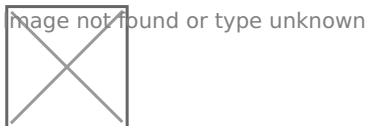


Объект размещаем в центре модели двери или в центре предполагаемого места выхода.

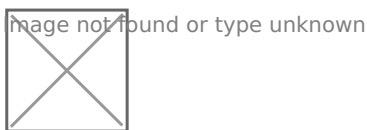
Необходимо повернуть объект двери так, чтобы локальный вектор Forward объекта (вектор синего цвета на скриншоте), указывал на фронтальную часть двери (как на скриншоте).



Для этого переключим инструмент вращения на локальный вектор и позиционируем объект.



В текущей реализации коллайдер двери, который будет создан в этой точке, имеет размеры: ширина = 0.8, высота = 2.45, глубина = 0.08. Чтобы активная зона двери начиналась от пола, объект **ExitDoor** необходимо разместить на высоте 1.225 единиц над уровнем пола (пример размещения показан на скриншоте ниже).



Если нет необходимости перемещаться по пространству, садиться на предметы или взаимодействовать с браузером, то следующие пункты можно пропустить.

**б) Создание зон телепортации.** В иерархии объектов на сцене создаем пустой родительский объект (окно Hierarchy -> Create Empty) с названием TeleportationArea. Внутри этого объекта в качестве дочерних необходимо разместить **невидимые** (с отключенным Mesh Renderer) объекты, коллайдеры которых будут зонами телепортации (например, пол или лестница). Обязательным условием для этих объектов является наличие компонента MeshCollider. Дочерние объекты TeleportationArea будут не видны, но по ним будет определяться зона телепортации. Видимые части модели, которые не должны быть зонами телепортации, размещать дочерними TeleportationArea не следует.

Image not found or type unknown



### Замечания при формировании объектов для телепортации

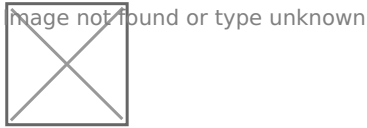
Телепортация в башне SenseTower происходит на основе меша в переданных объектах. Поэтому необходимо правильно его сформировать, чтобы не возникало ситуаций, когда пользователь может застрять. В случае если модель не позволяет такое сделать (например, когда на модель пола ставится модель стены). Рекомендуется использовать ProBuilder. В нем можно формировать отдельный меш, который будет удобен для телепортаций. В примерах проекта лежит сцена SampleScene. В данной сцене визуально показаны два варианта выделения меша для реализации телепортации.

**SampleScene(1 Variant with Probuilder)** - объекты для телепортации сформированы при помощи плагина ProBuilder. Зеленым цветом выделен колайдер, на который будет осуществлена возможность телепортации.

Image not found or type unknown



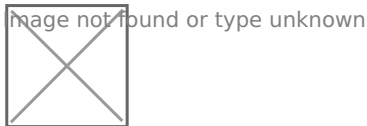
**SampleScene(2 Variant without Probuilder)** - объекты для телепортации сформированы при помощи стандартных Plane.



Зелеными линиями выделен коллайдер для телепортации.

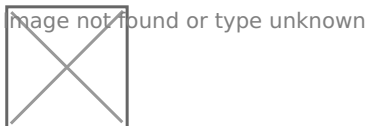
Как видно из двух примеров, объекты сформированные ProBuilderom имеют меньшее количество полигонов, соответственно этот метод оптимизированнее метода с Plane.

При формировании зон телепортации возможно потребуется запретить пользователю телепортироваться через стену, для этого необходимо на объект стены назначить Layer-> Walls. В таком случае мы запретим проходить лучу телепортации через стену. Layer Walls уже есть в проекте.

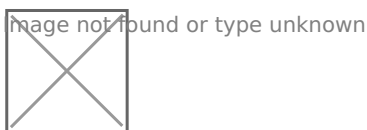


**в) Создание точек телепортации (стульев).** В иерархии объектов на сцене создаем родительский объект с названием LandingPlace.

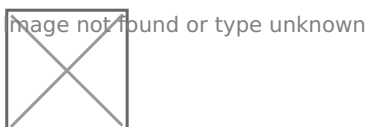
Внутри этого объекта в качестве дочерних необходимо разместить точки, в центр которых будет осуществляться возможность сесть.



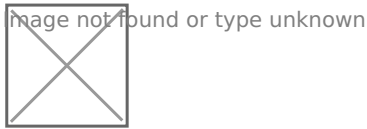
Она должна находиться в том месте, где будет пользователь садиться, примерно как на скрине



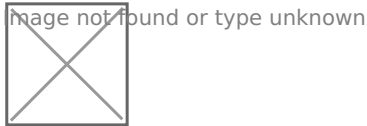
Необходимо повернуть объекты-точки так, чтобы локальный вектор Forward объекта (вектор синего цвета на скриншоте), указывал в направлении взгляда пользователя на стуле. Для этого переключим инструмент вращения на локальный вектор



**г)** В помещение имеется возможность поставить браузер. Для этого необходимо создать на сцене два пустых объекта: **BrowserAdminPlace, BrowserPlace.**

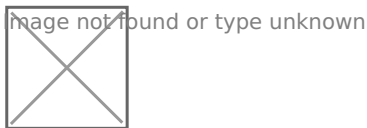


**BrowserAdminPlace** - объект, при нажатии на который осуществляется открытие браузера. Это должен быть пустой GameObject. Его располагаем на высоте 0.2 над уровнем пола, и смотрим чтобы локальный вектор Forward объекта (вектор синего цвета) был направлен в сторону стены. На скриншоте ниже показан пример, как будет выглядеть кнопка взаимодействия с браузером, сгенерированная в точке BrowserAdminPlace.



**BrowserPlace** - объект, браузера. располагается в центре предполагаемого места размещения браузера. Объект должен также находиться на расстоянии 0.35 от стены. Локальный вектор Forward объекта должен смотреть в сторону стены. В текущей реализации полотно браузера, которое будет создано в этой точке, имеет размеры: ширина = 4.1, высота = 2.3. Чтобы элементы управления браузером не находились под полом, объект **BrowserPlace** необходимо разместить на высоте 1.23 над уровнем пола.

На скриншоте ниже показан пример размещения браузера по объекту **BrowserPlace** и указаны размеры элементов.



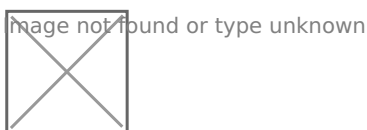
**Примечание:** названия TeleportationArea и LandingPlace и ExitDoor, BrowserAdminPlace, BrowserPlace должны быть именно такими, чтобы программа корректно определила области телепортации, места посадки, и дверь на выход.

На этом наполнение сцены для работы в башне SenseTower закончена.

Следующим шагом необходимо удостовериться или добавить сцену в группу Addressables, чтобы иметь возможность загружать её с облака в приложение Sense Tower.

### Добавление сцены в группу

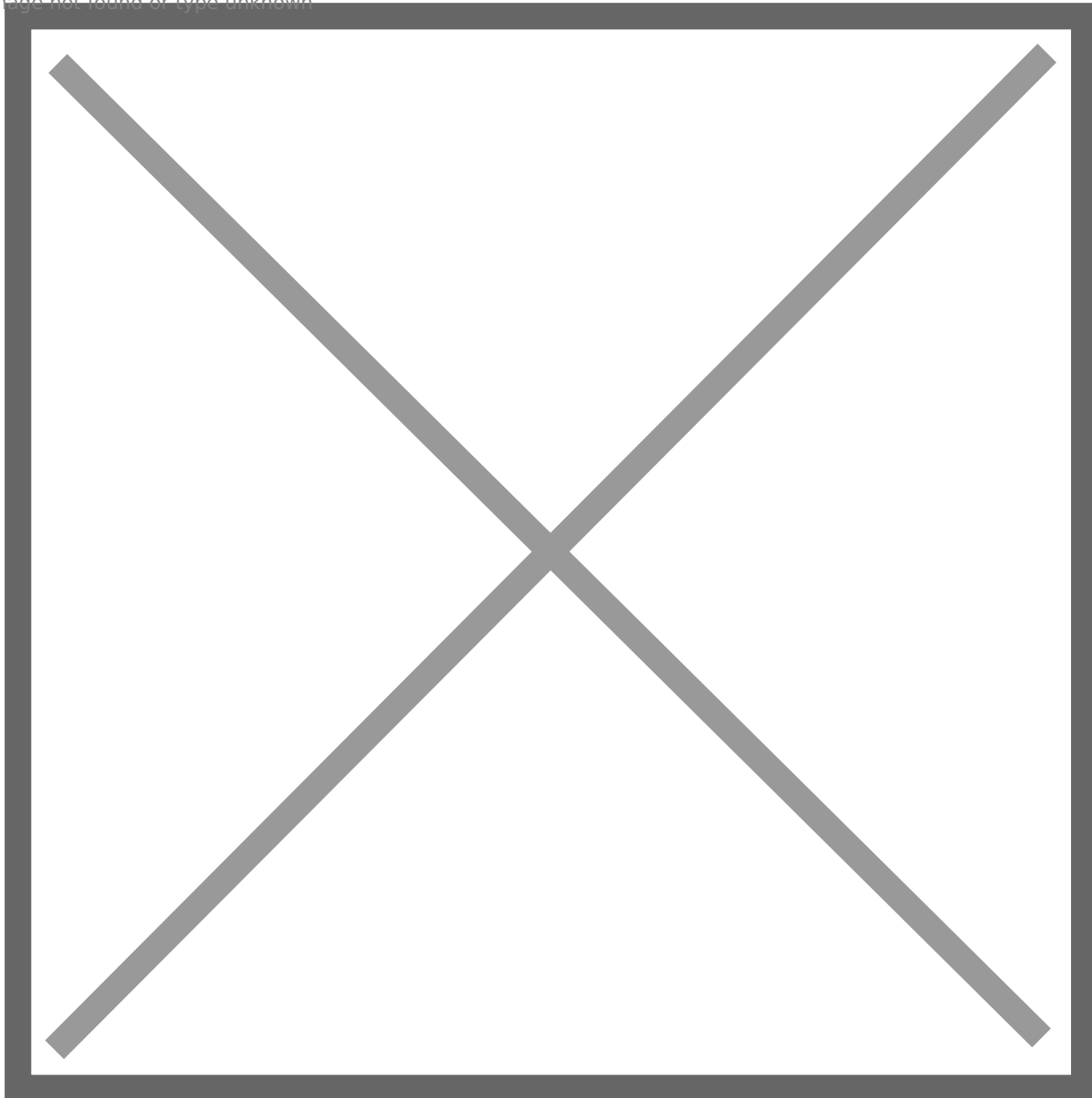
Изменим название сцены MyRemoteScene на такое же название, которое придумали ранее для каталога (в примере используется название OurBestCompany\_RemoteScene1, ваше название будет другим):



Теперь убедимся что наша сцена будет использоваться для удаленной загрузки.

Для этого находим сцену в папках проекта и в панели инспектора удостоверяемся что стоит галочка напротив Addressables:

Image not found or type unknown



Переходим в группы

Image not found or type unknown



Сцена появится в группе по умолчанию Default Local Group (если галочки напротив Addressables ранее не было) или в группе RemoteAssets:

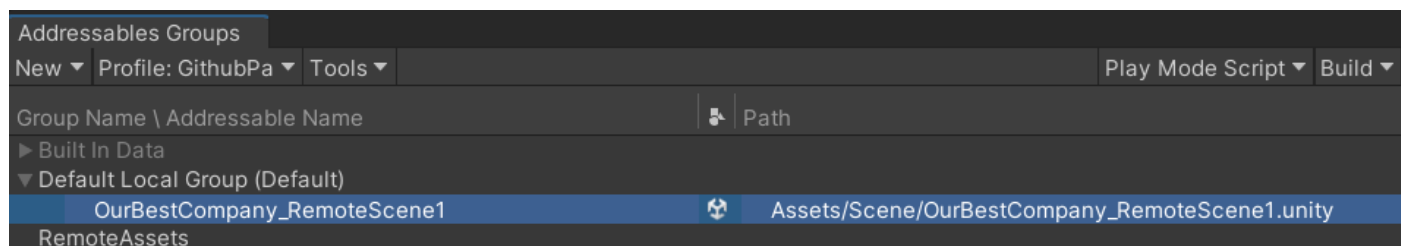
Image not found or type unknown



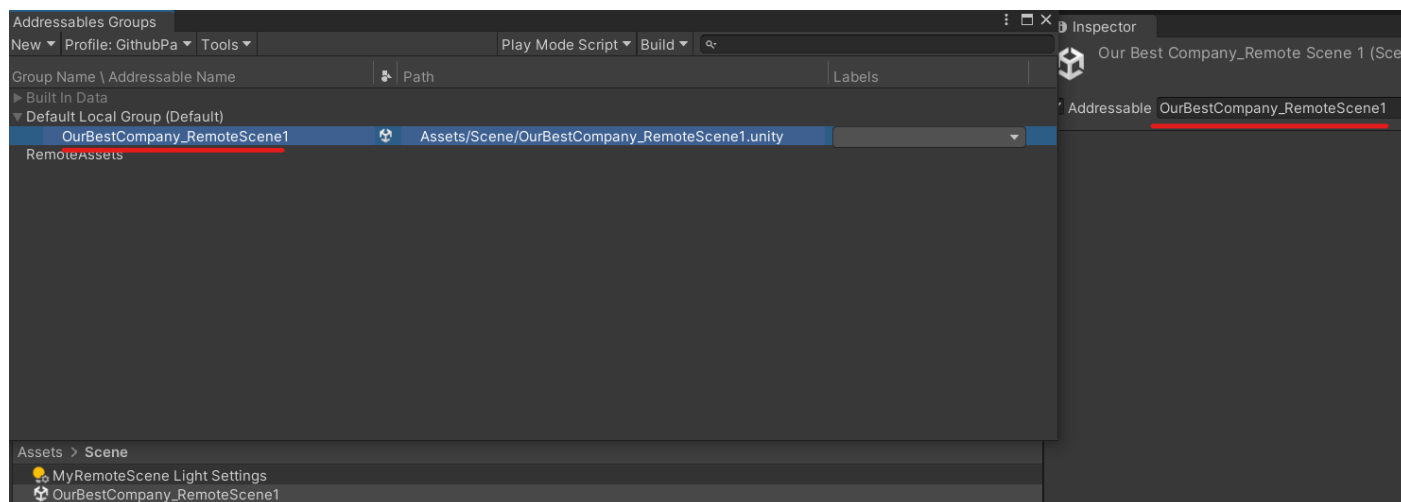
Кликаем правой кнопкой мыши на сцену и выбираем пункт Simplify Addressable Names. Это необходимо для того, чтобы задать для сцены упрощённое имя, по которому будет осуществляться поиск на облаке:



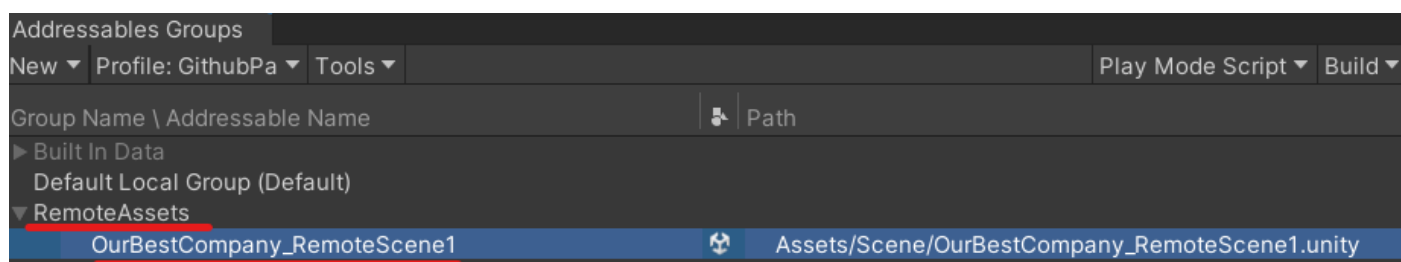
После этого для удалённого объекта сцены будет установлено упрощённое название, совпадающее с названием сцены:



**Важно:** Убедитесь, что название Addressable Name сцены совпадает с названием самой сцены. Если это не так, переименуйте Addressables Name сцены, нажав на неё в инспекторе объектов и задав правильное название в панели инспектора.



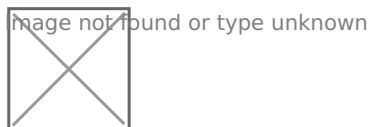
Перетаскиваем сцену из Default Local Group в Remote Assets(если её там не было). Если она уже там, то данный пункт пропускаем.



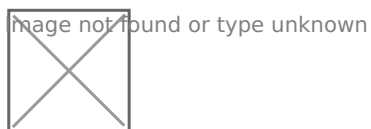
На этом настройка завершена. Осталось сформировать пакет для разных платформ.

## Формировка пакета

Пакет формируется отдельно под каждую платформу, для этого надо менять настройку Platform в BuildSettings через SwitchPlatform

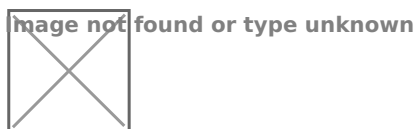


Пакет формируется через New Build->Default Build Script.

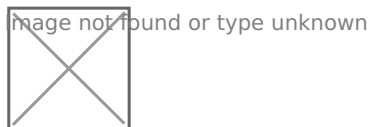


Пакет необходимо сформировать под три платформы Android, Windows, Linux. Для этого мы под каждую платформу меняем Platform и заново нажимаем New Build->Default Build Script.

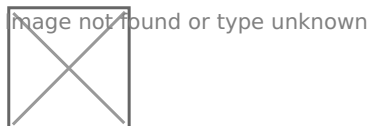
### Под Windows



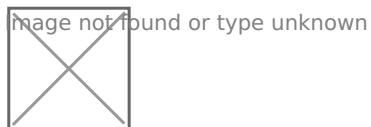
### Под Linux



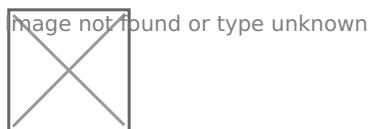
### Под Андроид



После того как создали все необходимые пакеты, у нас должно получиться примерно следующее. В папках проекта появится папка ServerData с подпапками для каждой платформы.



Внутри каждой подпапки.



Все пакеты подготовлены.

Теперь для загрузки сцены в систему SenseTower, необходимо запаковать в архив весь проект и передать сотруднику SenseCapital, ответственному за загрузку динамических сцен в SenseTower.

## Второй вариант

**1)** Создаём проект в версии Unity 2021.3.25 с использованием URP с подключенными модулями Android и Linux сборки



**2)** Создаём сцену и наполняем её объектами, формирующими пользовательское пространство. Если нет необходимости перемещаться по пространству или садиться на предметы, то шаги 3, 4 можно пропустить и приступить к установке Addressables.

## Требования предъявляемые к сцене

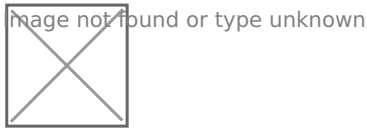
Быстродействие сцены напрямую зависит от её загруженности.

При формировании сцены для лучшего быстродействия необходимо учитывать требования предъявляемые устройством Oculus, они подробно описаны здесь - [Требования к сцене Unity](#)  
Из главных, это количество отрисовок в районе 80-200 и количества треугольников до 500к(чем меньше тем лучше) и текстуры разрешения до 2к.

При соблюдении данных требований в приложении SenseTower будет поддерживаться необходимое количество FPS и соответственно максимальное удовольствие от нахождения в пространстве.

**3)** В иерархии объектов на сцене создаем родительский объект с названием TeleportationArea. Внутри этого объекта в качестве дочерних необходимо разместить объекты, в пределах которых можно двигаться путём телепортации (например, пол или лестница). Обязательным условием для этих объектов является наличие компонента MeshCollider.



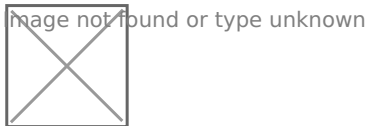


### Замечания при формировании объектов для телепортации

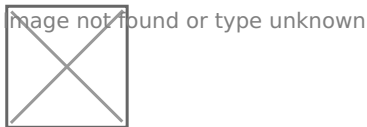
Телепортация в башне SenseTower происходит на основе меша в переданных объектах. Поэтому необходимо правильно его сформировать, чтобы не возникало ситуаций, когда пользователь может застрять. В случае если модель не позволяет такое сделать( например когда на модель пола ставится модель стены). Рекомендуется использовать ProBuilder. В нем можно формировать отдельный меш, который будет удобен для телепортаций.

**4)** В иерархии объектов на сцене создаем родительский объект с названием LandingPlace.

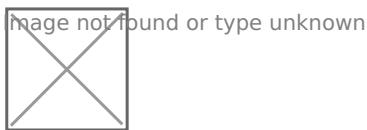
Внутри этого объекта в качестве дочерних необходимо разместить точки, в центр которых будет осуществляться возможность сесть.



Она должна находиться в том месте, где будет пользователь садиться, примерно как на скрине

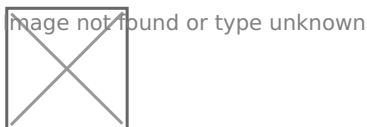


Необходимо повернуть объекты-точки так, чтобы локальный вектор Forward объекта (вектор синего цвета на скриншоте), указывал в направлении взгляда пользователя на стуле. Для этого переключим инструмент вращения на локальный вектор.

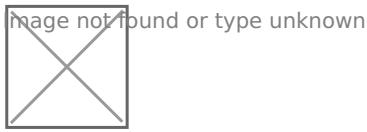


**в)** В помещение имеется возможность поставить дверь. При нажатии на которую, осуществляется переход в холл. Объект который за это отвечает должен присутствовать на сцене и называться ExitDoor. Объект размещаем в центре модели двери, или в центре предполагаемого места выхода.

Необходимо повернуть объект двери так, чтобы локальный вектор Forward объекта (вектор синего цвета на скриншоте), указывал на фронтальную часть двери, как на скрине.



Для этого переключим инструмент вращения на локальный вектор и позиционируем объект.



**Примечание:** названия TeleportationArea и LandingPlace и ExitDoor должны быть именно такими, чтобы программа корректно определила области телепортации, места посадки, и дверь на выход.

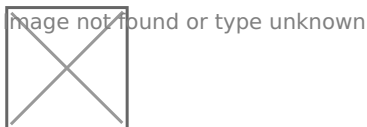
Примечание: названия TeleportationArea и LandingPlace должны быть именно такими, чтобы программа корректно определила области телепортации.

На этом наполнение сцены для работы в башне SenseTower закончена.

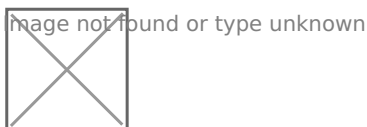
После подготовки сцены и после того как убедились что сцена подходит под наши требования, необходимо приступить к формированию сцены в необходимом формате, для этого используется Addressables.

### Установка Addressables

Установка пакета Addressables происходит через PackageManager.



Ищем пакет в списках Unity Registry



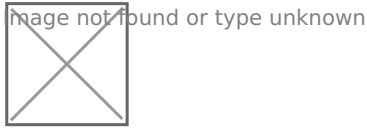
Текущая используемая версия 1.19.19. Устанавливаем эту версию.

### Установка поддержки VR

Так же помимо Addressables необходимо установить XR Plugin Management

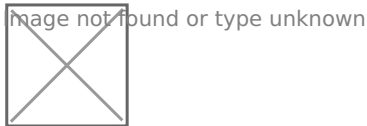


## И Oculus XR Plugin

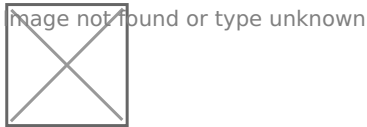


Поскольку при компиляции моделей и сцен механизм компиляции опирается на настройки проекта, нам необходимо указать, что проект будет использоваться для VR устройств Oculus.

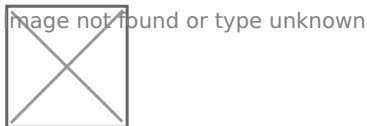
Включаем поддержку VR и Oculus в настройках проекта. Это делается через File->Build Settings-> Player Settings-> Xr Plug-in Management и включить поддержку Oculus



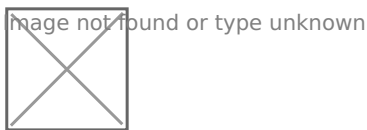
со следующими настройками



Чтобы начать пользоваться Addressables необходимо создать дефолтные настройки. Для этого переходим в Addressable->Groups.

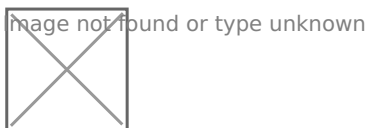


И нажать кнопку создания настроек

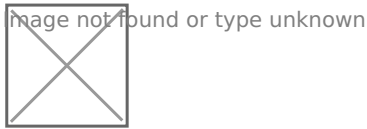


## Настройка профиля

После установки необходимо указать куда сохраняются файлы содержащие сцену. Это делается через настройки профиля



Создаем новый профиль через Create-> Profile. Задаем имя новому профилю и выбираем его активным через контекстное меню



Теперь необходимо прописать две переменные. Предварительно указав в Remote в выпадающем списке Custom( тогда откроется возможность изменять нужные переменные)

**Remote.BuildPath** - путь куда будут складываться наши файлы архивов со сценами и моделями внутри репозитория. По дефолту можно указать. ServerData/[Название проекта]/[BuildTarget] - это означает что файлы архива будут сохраняться в папке ServerData, и внутри этой папки создается подпапка в зависимости от платформы(Android, StandaloneWindows64), для которой формируется пакет.

Например: ServerData/AddressablesContent/[BuildTarget]

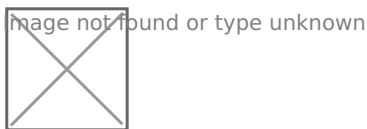
**Remote.LoadPath** - путь, указывающий на место, где будут храниться файлы каталога. Указываем

[https://sense-capital.github.io/RemoteContent/ServerData/\[Название проекта\]\[BuildTarget\]](https://sense-capital.github.io/RemoteContent/ServerData/[Название проекта][BuildTarget])

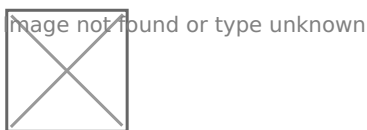
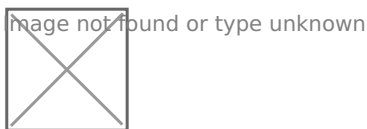
Например: [https://sense-capital.github.io/RemoteContent/ServerData/AddressablesContent/\[BuildTarget\]](https://sense-capital.github.io/RemoteContent/ServerData/AddressablesContent/[BuildTarget])

[Название проекта] - прописываем без пробелов

Должно получится как здесь.



После того как заполнили переменные в настройках необходимо указать что системе нужно собирать удаленный каталог это делается через AssetManagement->Addressables->Settings

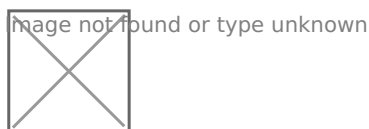


Добавить в Player Version Override добавочное слово, для идентификации каталога. Оно может быть любое.

На этом настройка профиля завершена. Теперь система будет понимать куда сохранять полученные в результате сборки файлы, и куда надо ссылаться внешнему источнику для скачивания необходимой сцены.

## Настройка группы

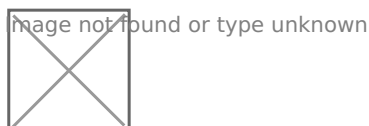
После того как профиль настроен необходимо настроить группы, где будут сохранены необходимые объекты.



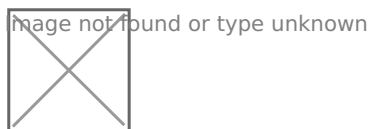
Открываем вкладку настроек группы и создаем новую через New->PackageAssets, можно переименовать на имя, которое подходит



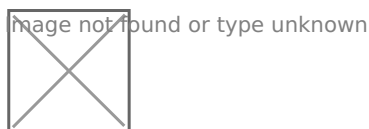
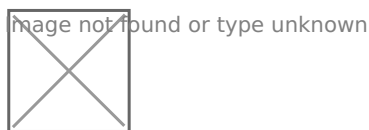
Созданную группу можно переименовать для удобства



необходимо убедиться что в профиле используется профиль, который мы создали.



Теперь в ново созданной группе и в DefaultLocalGroup надо указать что мы будем использовать удаленный путь загрузки,



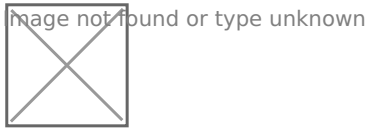
Выставляем во вкладке Play Mode Script-> Use Existing Build. Это нужно для того, чтобы материалы нашей сцены передавались вместе с каталогом.



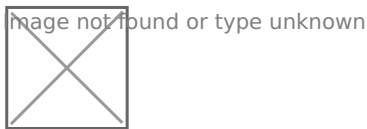
Настройка групп завершена. Теперь необходимо добавить сцену в эту группу.

### Добавление сцены в группу

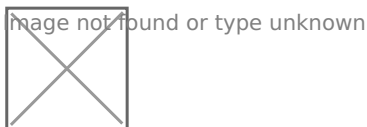
Для этого надо указать нашей сцене, что она будет использоваться для загрузки, находим сцену в папках проекта и ставим галочку напротив Addressables



Она появится в дефолтной группе (если мы указали новую группу как дефолтную, то сцена сразу появится в этой группе)



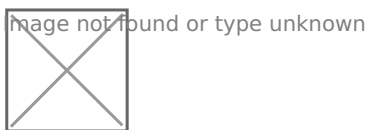
Через правую кнопку мыши нажимаем на Simplify Addressable Names, и у нашей сцены удобочитаемое имя, перетаскиваем её в ново созданную группу(если мы указали новую группу как дефолтную, то сцена сразу появится в этой группе)



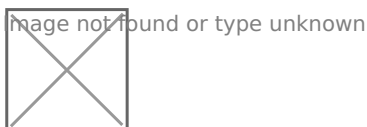
На этом настройка завершена, мы настроили профиль, группу и добавили сцену для удаленной загрузки. Осталось сформировать пакет.

### Формировка пакета

Пакет формируется отдельно под каждую платформу, для этого надо менять настройку Platform в BuildSettings через SwitchPlatform



Пакет формируется через New Build->Default Build Script.



Пакет необходимо сформировать под три платформы Android, Windows, Linux. Для этого мы под каждую платформу меняем Platform и заново нажимаем New Build->Default Build Script.

### Под Windows



image not found or type unknown

## Под Linux



image not found or type unknown

## Под Андроид



image not found or type unknown

После того как создали все необходимые пакеты, у нас должно получиться примерно следующее. В папках проекта появится папка ServerData с подпапками для каждой платформы.



image not found or type unknown

Внутри каждой подпапки.



image not found or type unknown

Все пакеты подготовлены.

Теперь для загрузки сцены в систему SenseTower, необходимо запаковать в архив весь проект и передать сотруднику SenseCapital, ответственному за загрузку динамических сцен в SenseTower.

Так же вместе с архивом сотруднику SenseTower надо сказать ключ сцены, который будет использоваться в башне, его можно найти по пути



image not found or type unknown

И имя ключа присвоенное сцене отмечено на скрине

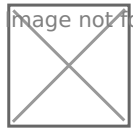


Image not found or type unknown



# Требования к загружаемым объектам

Чтобы обеспечить наилучшее качество и совместимость с VR метавселенной SenseTower, ваши загружаемые модели должны соответствовать следующим требованиям:

1. **Формат файла:** Модели должны быть в формате .FBX.
2. **Текстуры:** Все текстуры должны быть прикреплены к модели и включены в ту же папку, что и файл модели. Они должны быть в формате .PNG. Если текстуры не включены, это может вызвать проблемы с отображением модели.

## **Рекомендация:**

Обязательно наличие карт диффуза и карт АО для всех динамических моделей. Карта диффуза должна быть не больше 1024\*1024 точек, должна иметь координаты UV0. Карта АО должна быть не больше 1024\*1024 точек, может иметь как идентичные карте диффуза координаты (UV0), так и иные координаты — UV1, UV2. Для однородных поверхностей предпочтительно использовать тайленные карты текстур разрешения до 512\*512 точек и карты АО и Normal для отображения псевдорельефа поверхности.

Все текстуры должны быть в формате PNG 8bit, альфа канал должен быть только на тех текстурах, где он необходим художественно. Максимальный размер текстуры может достигать 2048\*2048 точек. Допустимо использование кратного масштабирования, например: 128\*128, 128\*64, 2048\*512. Эти требования обусловлены алгоритмами mipmapping'a.

Исключение - skybox-текстура окружающего ландшафта (для неё можно установить разрешение вплоть до 8k).

3. **Шейдеры.**  
Если используются какие-то кастомные шейдеры, то необходимо проверить их на соответствие работоспособности в unity при URP.
4. **Размер файла:** Общий размер файла модели и текстур должен быть оптимизирован, чтобы обеспечить низкую задержку загрузки и эффективное использование памяти. Рекомендуемый максимальный размер файла - 5 МБ.
5. **Полигоны:** Модель должна содержать разумное количество полигонов, чтобы обеспечить хорошее качество отображения и производительность. Рекомендуется, чтобы модель не содержала более 1 000 полигонов.

6. **Анимация:** Если модель содержит анимации, они должны быть включены в файл модели в виде встроенных клипов анимации. Анимации должны быть оптимизированы, чтобы обеспечить плавное движение без существенного влияния на производительность.
7. **Масштаб:** Модели должны быть масштабированы правильно, чтобы они соответствовали реальному миру в метавселенной SenseTower.

**Рекомендация:**

Для масштабирования используется обычный unity куб, он соответствует 1 юнити метру,

8. **Поддержка VR:** Модели должны быть оптимизированы для использования в VR, включая правильную настройку коллизий, поддержку головного отслеживания и контроллеров VR, если это применимо.
9. **Содержание:** Все модели должны соответствовать политике контента SenseTower. Они не должны содержать обидного, непристойного, насильственного, неправомерного или ненавистнического контента.

Пожалуйста, обязательно проверьте свои модели на соответствие этим требованиям перед загрузкой, чтобы обеспечить наилучший опыт для всех пользователей SenseTower.