

**Методические рекомендации по разработке
простого проекта виртуальной реальности (VR)**

“VR МУЗЕЙ: АСТРОНОМИЯ”

Концепция/цель: создание своего первого простого проекта с использованием технологии виртуальной реальности (VR) на основе базовых объектов VR проекта с использованием 3D моделей, изображений, видео, аудио и визуально-блочного скриптинга.

Задачи:

- знакомство с программным обеспечением – конструктором проектов дополненной и виртуальной реальности EV Toolbox;
- знакомство с инструментами и базовым функционалом программы;
- знакомство с базовыми объектами VR проекта: манипулятор камеры, вьюер, VR контроллер, поиск пересечений;
- знакомство с объектами проекта виртуальной реальности: модель, изображение, аудио, текст.

Инструменты для работы:

- ПК/ноутбук, соответствующий минимальным требованиям (https://eligovision.ru/toolbox/docs/3.4/articles/system_requirements.html) или выше;
- Установленный на ПК конструктор проектов дополненной и виртуальной реальности EV Toolbox (для выполнения проекта можно использовать trial версию конструктора);
- VR шлем на базе OS Android для тестирования готового собранного приложения (приложение идет в комплекте с данными методическими рекомендациями).

Необходимые навыки для выполнения проекта: навыки работы с ПК на уровне “Пользователь”.

Ход выполнения: проект выполняется поэтапно в соответствии с описанными в методических рекомендациях шагами. Проект можно выполнить за один подход или разбить выполнение проекта на несколько занятий.

Уважаемый пользователь!

Мы приветствуем вас в методических рекомендациях по разработке проекта **“VR МУЗЕЙ: АСТРОНОМИЯ”**. Это простой проект с использованием технологии виртуальной реальности, который поможет вам познакомиться с миром разработки VR проектов. Здесь вы узнаете насколько удобной может быть работа по созданию своего собственного VR проекта с использованием конструктора проектов дополненной и виртуальной реальности EV Toolbox, и что необходимо для создания подобного проекта.

Если вы еще не установили конструктор на свой ПК, сейчас самое время это сделать!

- Загрузить конструктор EV Toolbox можно на внутреннем портале разработчика конструктора: <https://eligovision.ru/portal>
- Подробную инструкцию по установке конструктора EV Toolbox можно найти в текстовом виде в документации по программе:
<https://eligovision.ru/toolbox/docs/3.4/articles/installation.html>
- Или в виде видеоинструкции на YouTube канале разработчика:
<https://youtu.be/g5WFafixRZ0>

Для работы над проектом на данный момент является актуальной версия конструктора **EV Toolbox 3.4.6.**, однако вы также можете использовать более ранние или более новые версии.

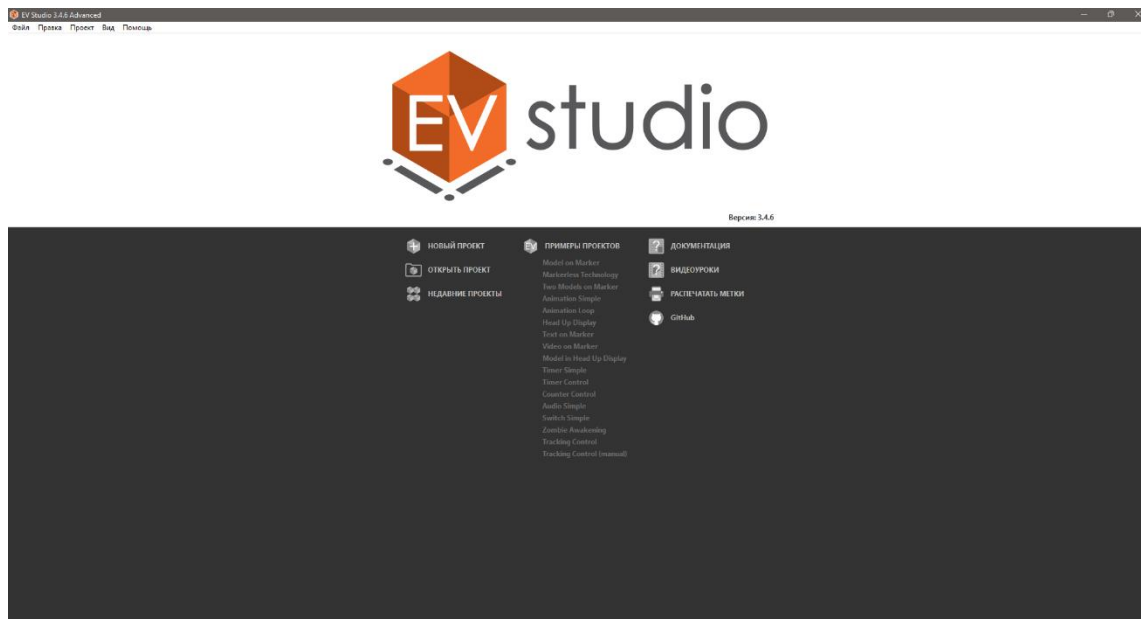
Для удобства работы над проектом мы подготовили для вас ресурсы, которые вы можете использовать при разработке. Загрузить ресурсы можно по ссылке:
<https://clck.ru/pdDWU> (архив .rar)

Приглашаем приступить к разработке и желаем успехов!

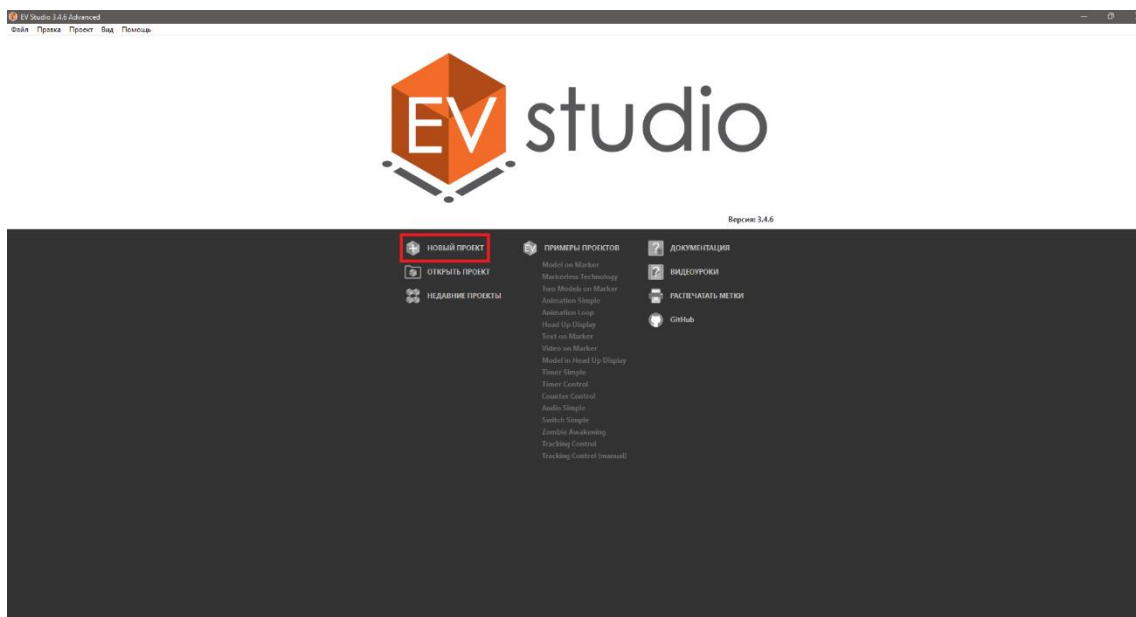
Этап 1. Начинаем разрабатывать VR проект. Создаем простую VR сцену. Добавляем в проект базовые объекты: Манипулятор камеры и Viewer

(время работы 15 минут)

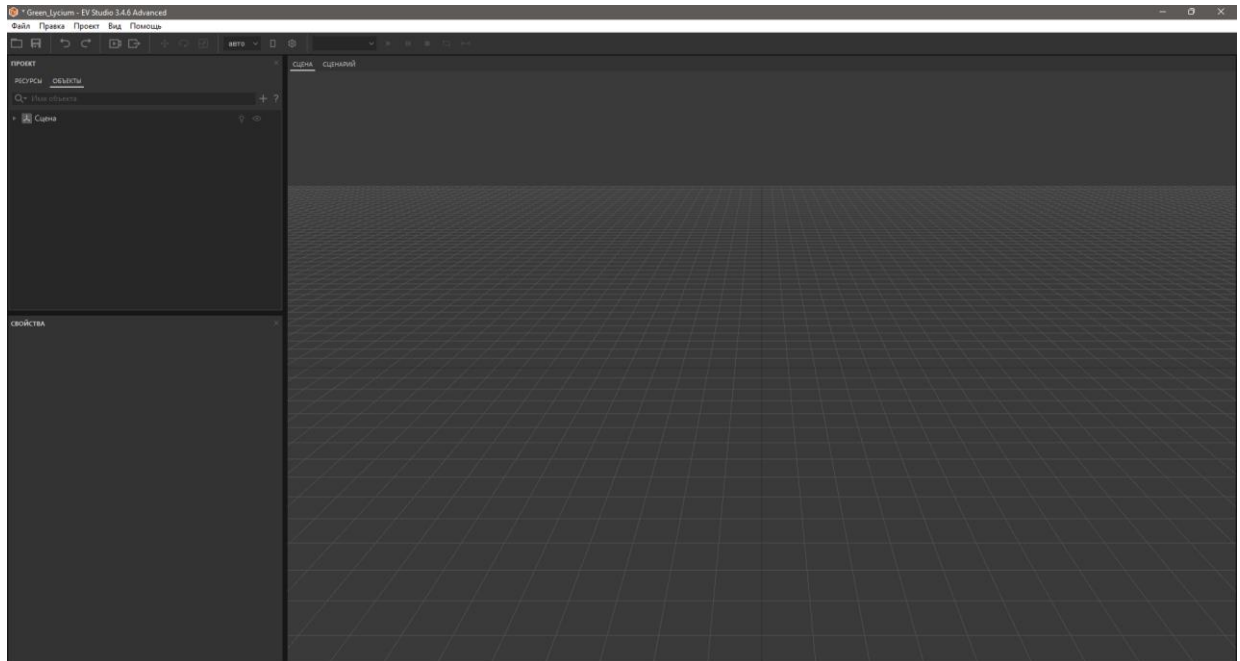
Запустите программу конструктора EV Toolbox на вашем ПК.



Перед нами приветственное окно программы с базовым набором функций для старта разработки нового проекта или проекта на базе встроенных примеров. Так как мы будем разрабатывать новый проект с нуля, нам необходимо выбрать пункт “Новый проект”.



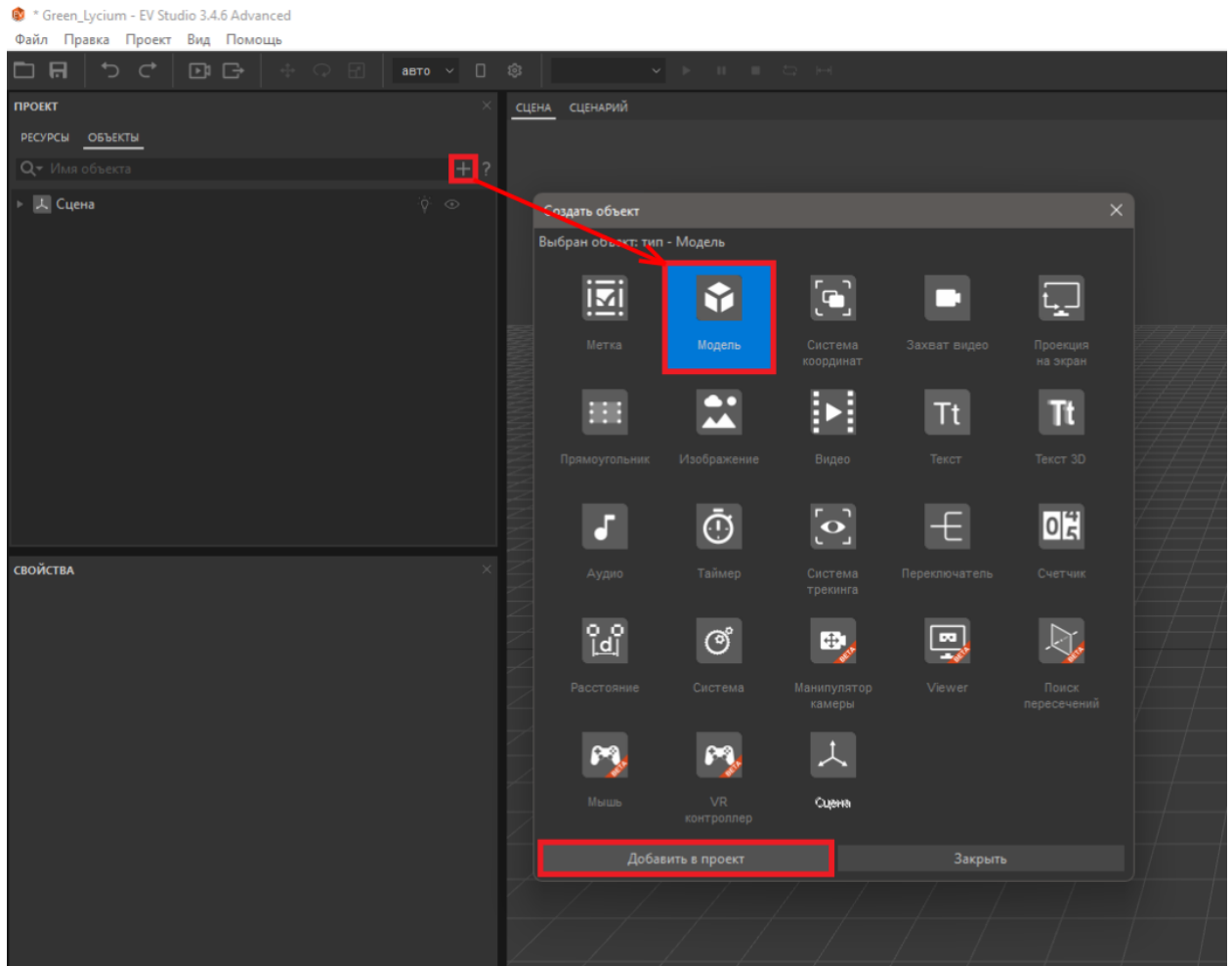
Перед нами появляется пустая виртуальная сцена, которую нам предстоит наполнить визуальной составляющей проекта.



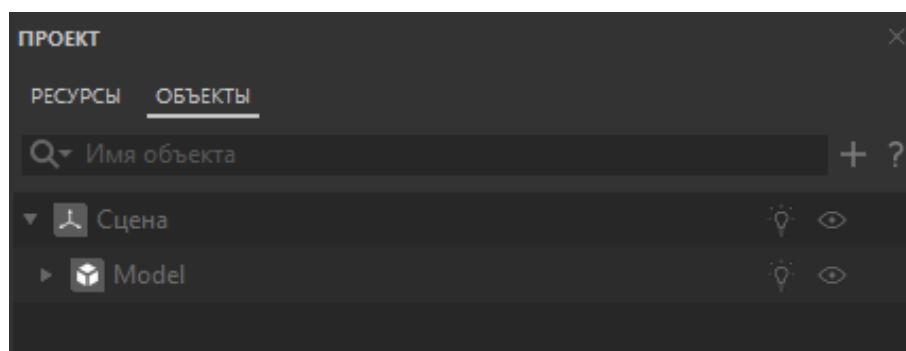
Как и для проектов с технологией дополненной реальности, для проектов с технологией виртуальной реальности необходимо использовать базовые объекты. Они позволяют разрабатывать и настраивать VR проекты так, чтобы они корректно запускались и работали на целевых носимых устройствах – VR шлемах.

Для наглядности работы базовых объектов VR проекта добавим в проект 3D модель музейной комнаты.

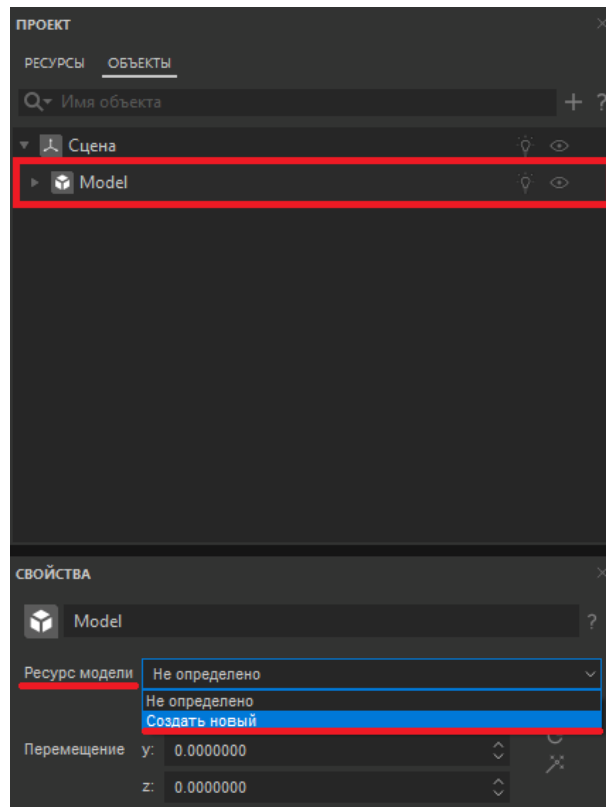
Для этого откроем окно добавления объектов и выберем объект Модель.



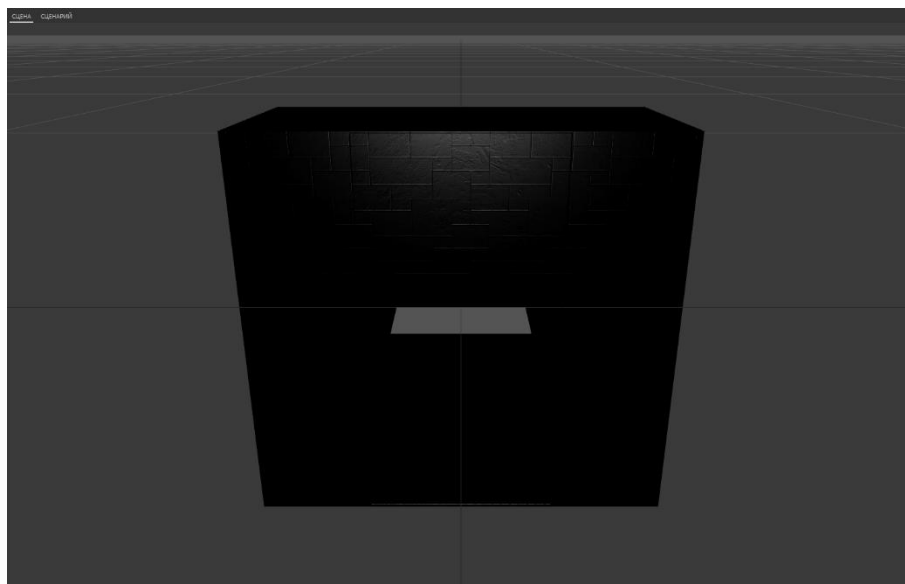
Модель появилась в дереве объектов, однако в сцене ничего не изменилось.



Чтобы модель комнаты появилась в сцене, для нее необходимо определить ресурс. Выделяем новый добавленный объект в дереве объектов и в панели свойств для параметра ресурс в выпадающем списке выбираем "Создать новый".

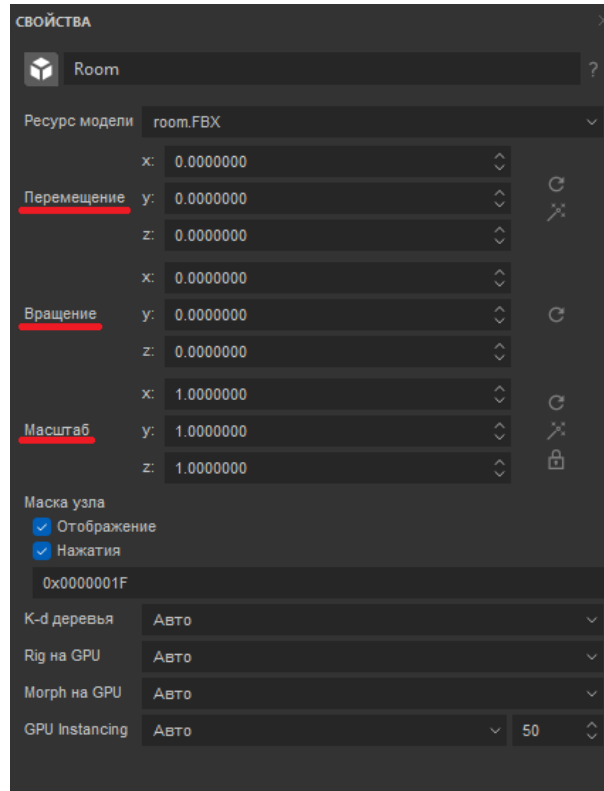


В папке с ресурсами для проекта выбираем папку Models, модель Room и нажимаем Открыть. В сцене проекта появилась модель комнаты.

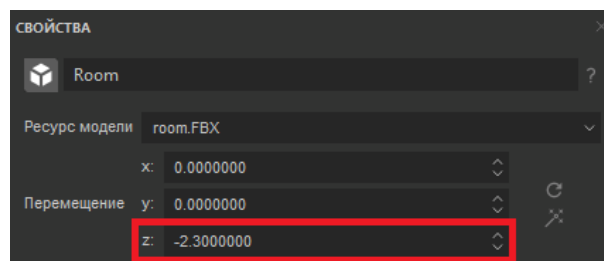


Так как в проекте будет достаточно большое количество различных объектов, то для каждого будет удобно задать различные оригинальные названия. Переименуем модель комнаты и назовем ее Room.

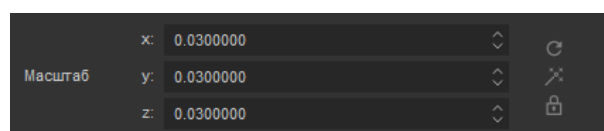
Корректно разместим модель в сцене проекта. Для этого воспользуемся параметрами трансформации в панели свойств объекта.



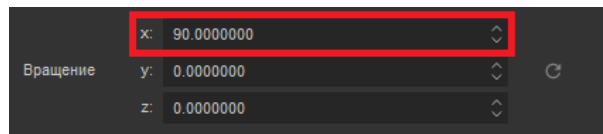
Сместим комнату по оси Z на -2,3.



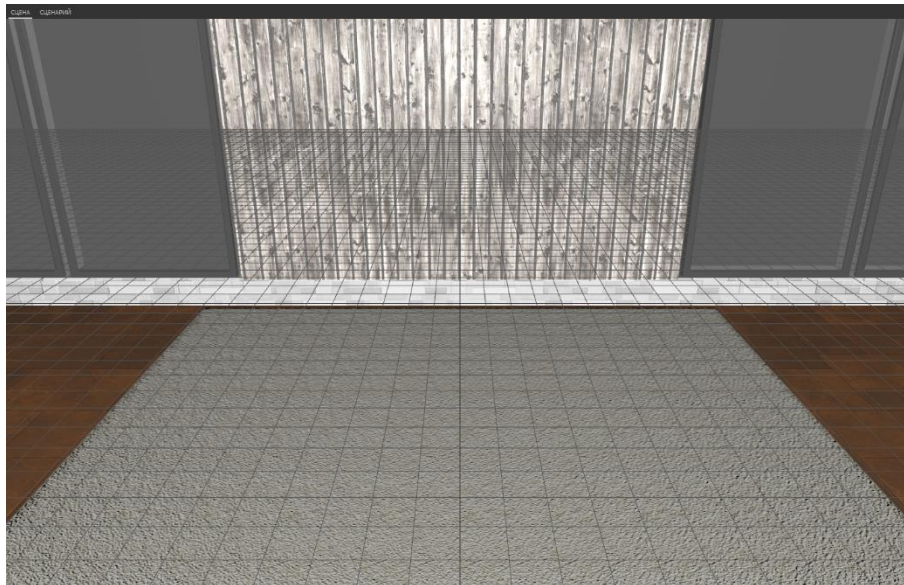
Уменьшим комнату, так как сейчас ее размеры очень большие для проекта. Изменим масштаб по оси X на 0.03, две другие координаты будут изменены автоматически.



Осталось повернуть комнату, так как сейчас мы видим внешнюю часть ее пола. Повернем комнату по оси X на 90°.



Теперь модель музейной комнаты расположена корректно относительно виртуальной сцены.

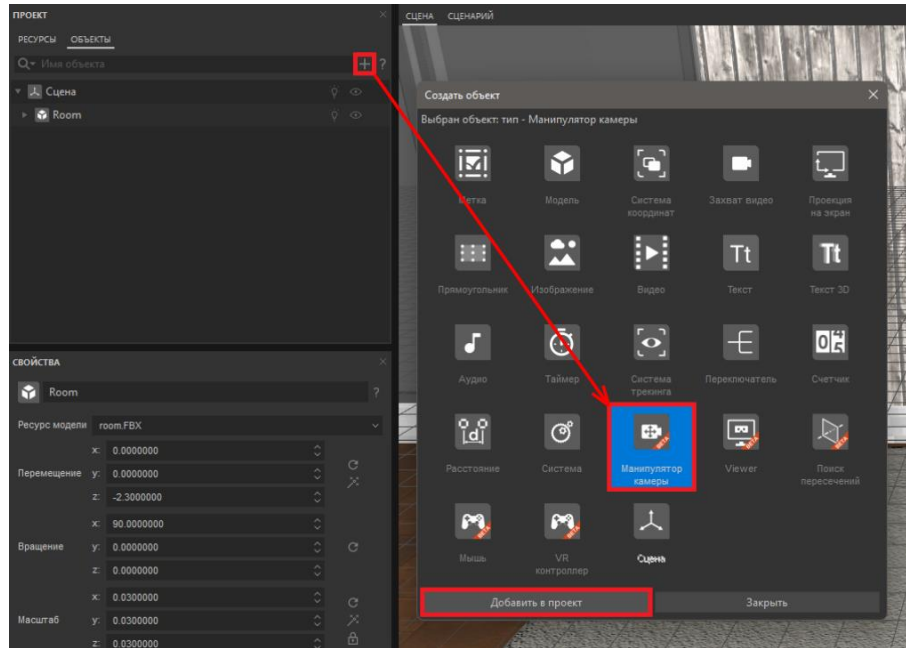


Запустим предпросмотр, чтобы увидеть, как в текущий момент выглядит виртуальное окружение.



Все, что мы можем увидеть в режиме предпросмотра, – это статичное изображение пола музейной комнаты.

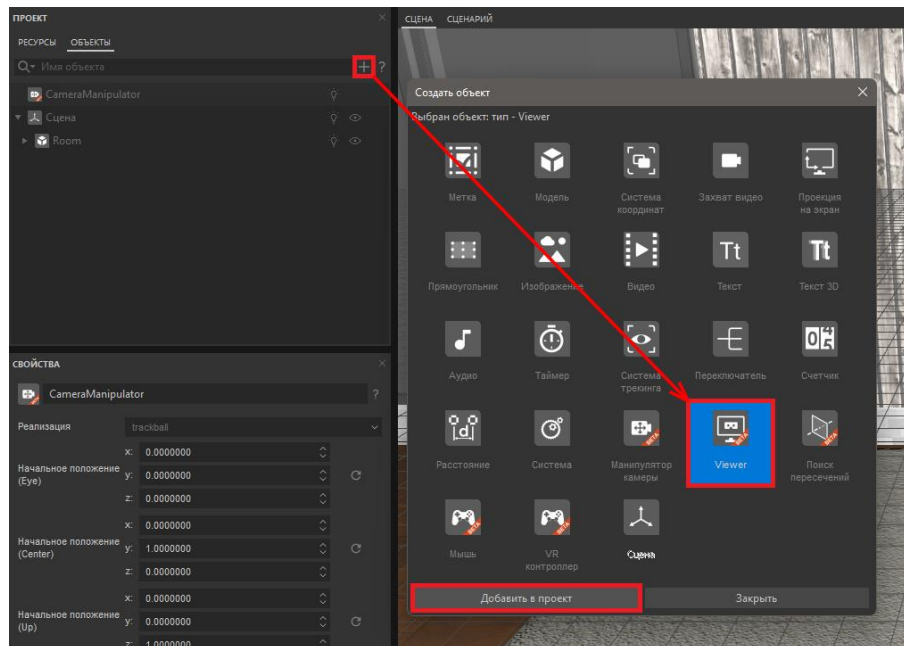
Все дело в том, что сейчас для работы с виртуальным окружением в проекте не хватает базовых объектов VR проекта. Это объект **Манипулятор камеры** и **Вьюер**. Добавим объекты в проект и подробнее познакомимся со свойствами каждого из НИХ.



Объект **Манипулятор камеры** – это базовый объект проектов с технологией виртуальной реальности, который используется для управления виртуальной камерой и определения положения пользователя в пространстве виртуальной сцены.

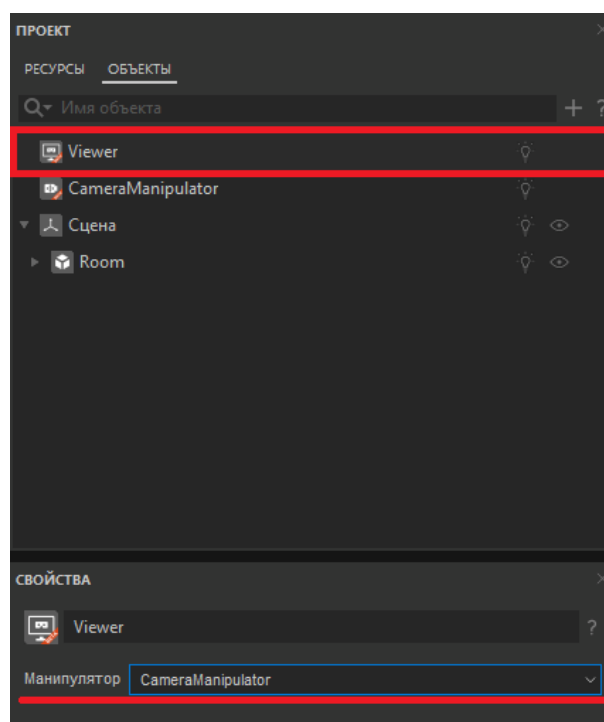
Вновь запустим режим предпросмотра и увидим, что после добавления нового объекта ничего не изменилось, мы по-прежнему видим только статичный пол комнаты.

Чтобы работать с визуальным представлением VR проекта в режиме предпросмотра, нам необходим еще один базовый объект проекта виртуальной реальности – **Вьюер** или **Просмотрщик**. Добавим его в проект.



Объект **Вьюер** – это второй базовый объект для создания проектов с технологией виртуальной реальности. Он позволяет просмотреть вид с выбранного (установленного) и активного в текущий момент времени манипулятора камеры.

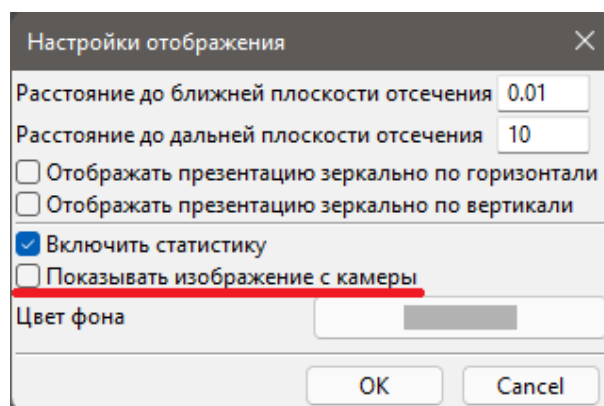
Чтобы увидеть модель окружения в режиме предпросмотра, нам необходимо в свойствах объекта Вьюер выбрать определенный манипулятор камеры. В нашем проекте только один манипулятор камеры, поэтому выберем его в свойствах объекта Вьюер.



Теперь при запуске режима предпросмотра мы сможем просмотреть созданное виртуальное окружение, используя левую кнопку мыши для поворота манипулятора и колесико мыши для приближения/отдаления объектов сцены.



Если вы работаете на ноутбуке со встроенной камерой или у вас подключена внешняя Web камера за окнами комнаты вы можете увидеть изображение с камеры. В проекте виртуальной реальности камера не используется, а значит изображение с камеры можно отключить. Для этого в главном меню найдем пункт “Проект” → “Настройки отображения” и снимем флажок в параметре “Показывать изображение с камеры”.

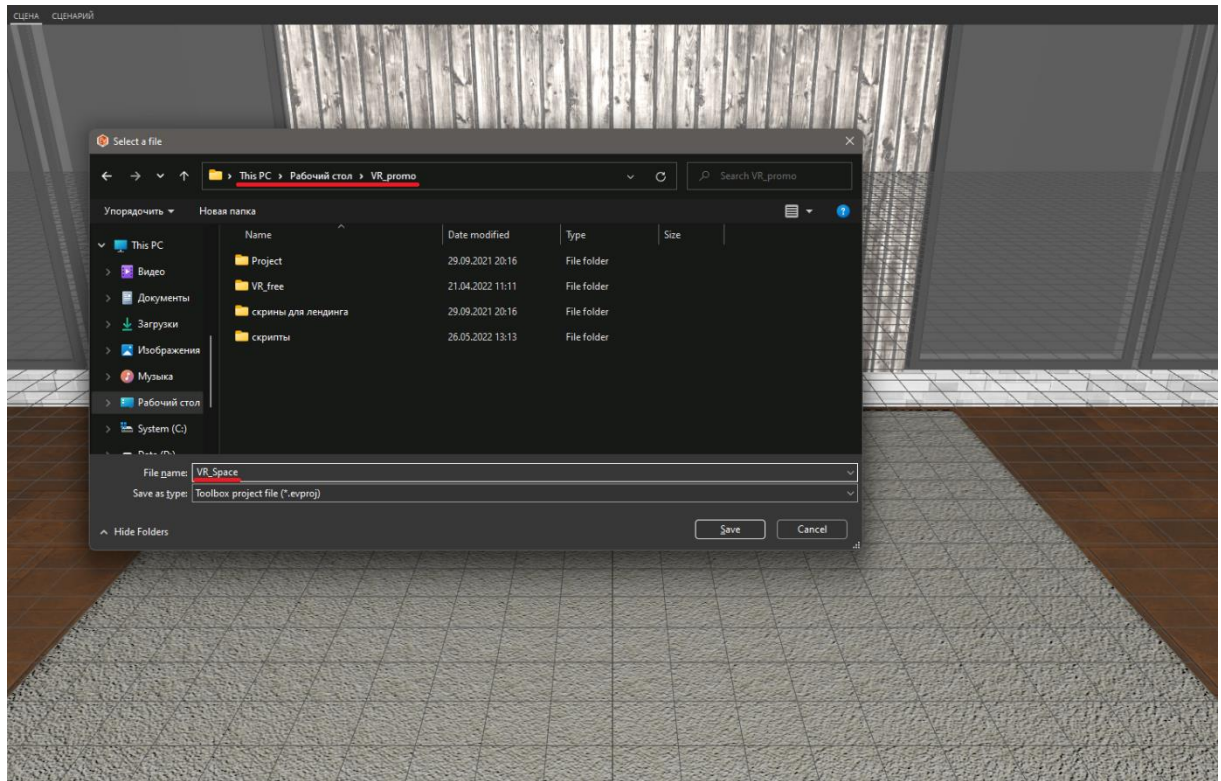


Теперь при запуске Предпросмотра через окно мы видим только одноцветное окружение сцены. Чуть позже мы вернемся к этому.

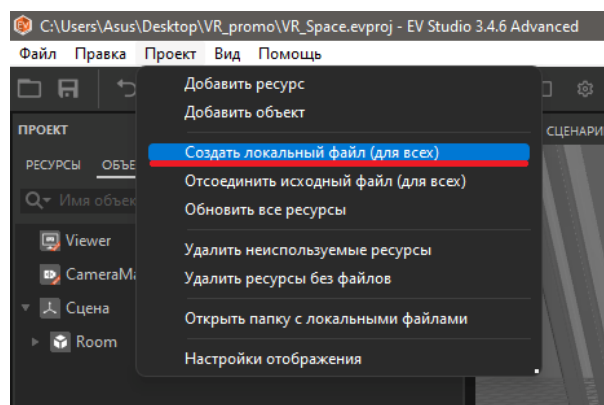
Обратите внимание на то, что все добавленные на данный момент базовые объекты VR проекта (Манипулятор камеры и Вьюер) **не имеют визуального отображения в сцене**. Они используются только для создания логики работы в сценарии проекта.

Для завершения работы с первым этапом нам необходимо корректно сохранить проект без потери ресурсов.

Для корректного сохранения проекта в главном меню необходимо выбрать **Файл**→**Сохранить проект/Сохранить проект как...**, указать место куда будет сохранен проект и задать для него имя.



После этого в главном меню необходимо выбрать пункт Проект→Создать локальный файл (для всех).



Проект корректно сохранен в выбранную папку без потери ресурсов.

Этап 2. Добавляем в виртуальное окружение 3D модели: проектор, экран, тумбы.

(время работы: 20 минут)

Запустим конструктор EV Toolbox на нашем ПК, найдем проект "VR_Space" в колонке "Недавние проекты" в окне приветствия программы и откроем его.



Когда мы говорим о технологии виртуальной реальности, то самое первое, что представляем – это реалистичное трехмерное окружение. Эффект погружения в созданный разработчиками мир достигается именно за счет максимально реалистичного трехмерного окружения вокруг пользователя. Поэтому важную роль при работе над проектом виртуальной реальности играет визуальный контент, в первую очередь 3D модели.

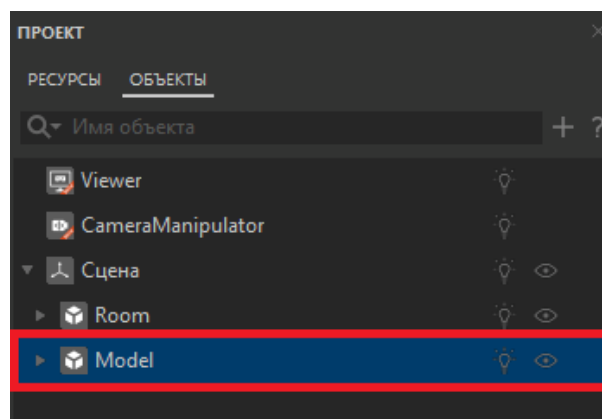
Что же делать, если вы хотите использовать в своем проекте 3D модели, но не являетесь при этом профессиональным моделлером? На помощь нам приходит сеть Интернет, где есть много ресурсов, которые как платно, так и бесплатно предлагают воспользоваться готовыми 3D объектами от различных авторов. Обычно мы рекомендуем проверенный сайт: www.sketchfab.com, где можно найти огромное количество моделей¹ различной сложности по всевозможным направлениям.

При загрузке моделей важно обращать внимание на **формат**. При создании проекта в конструкторе EV Toolbox можно использовать модели форматов ***.fbx** и ***.obj**. Модели формата *.fbx в свою очередь могут быть как с анимацией, так и без. Модели формата *.obj анимацию не поддерживают.

Первую 3D модель – модель музейной комнаты мы с вами добавили в прошлом занятии. В этой части мы продолжим добавлять визуальный контент, который позволит нам “оживить” виртуальное окружение.

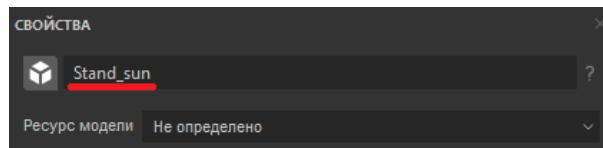
Сейчас перед нами стоит задача добавить в виртуальное окружение модели четырех тумб, на которых размещены музейные экспонаты (анимированные модели Солнца и планет).

Откроем окно добавления объектов, выберем объект Модель и добавим его в проект.



¹ Модели могут быть как платные, так и бесплатные и распространяются по различным типам лицензий. Подробнее см. www.sketchfab.com

Назовем наш новый объект "Stand_sun".

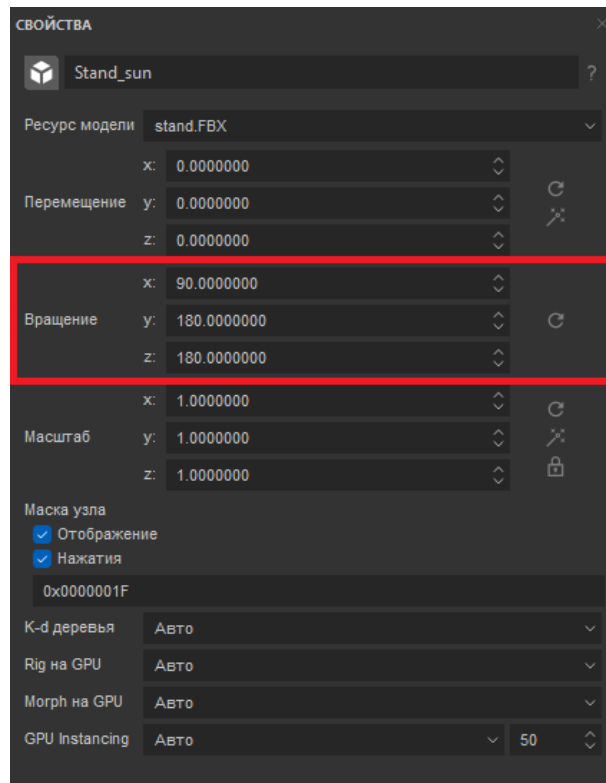


Определим ресурс для модели, выбрав в параметре Ресурс модели "Создать новый". В открывшемся окне находим папку с ресурсами к проекту, открываем папку Models и выбираем модель Stand. Модель тумбы подгрузилась в сцену.



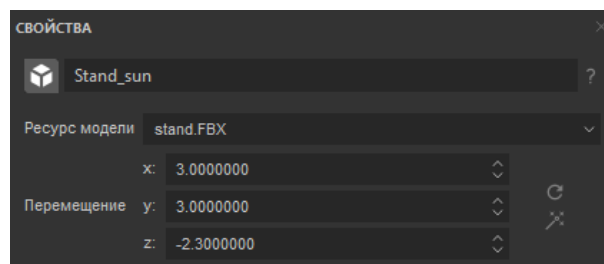
Теперь нам нужно корректно разместить ее в сцене и масштабировать. Для этого воспользуемся параметрами трансформации в панели свойств модели.

Повернем модель по оси X на 90°, по осям Y и Z на 180°.

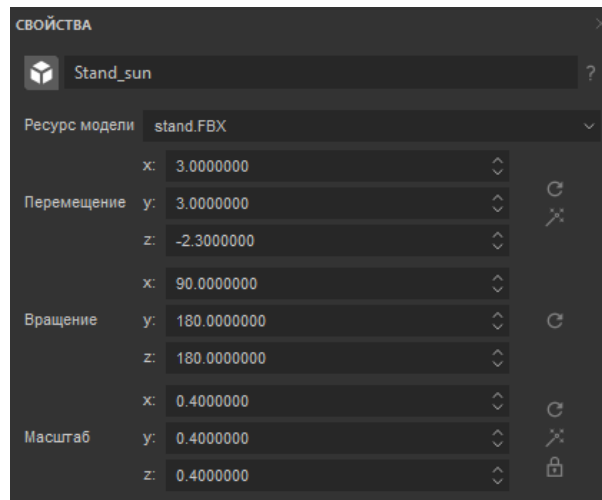


Поменяем положение модели в сцене при помощи параметра Перемещение. По оси X сместим тумбу на -3, по Y на 3, по Z на -2.3.

ВАЖНО: для корректного ввода значений необходимо использовать точки.



Уменьшим тумбу, так как сейчас ее размеры очень большие для проекта. Изменим масштаб по оси X на 0.4, две другие координаты будут изменены автоматически.

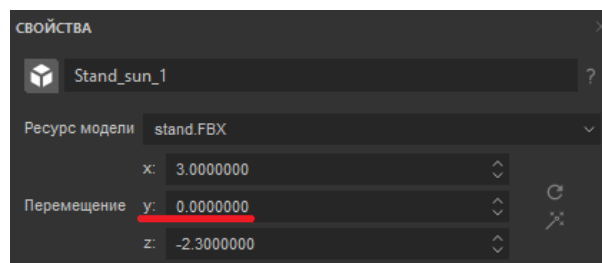


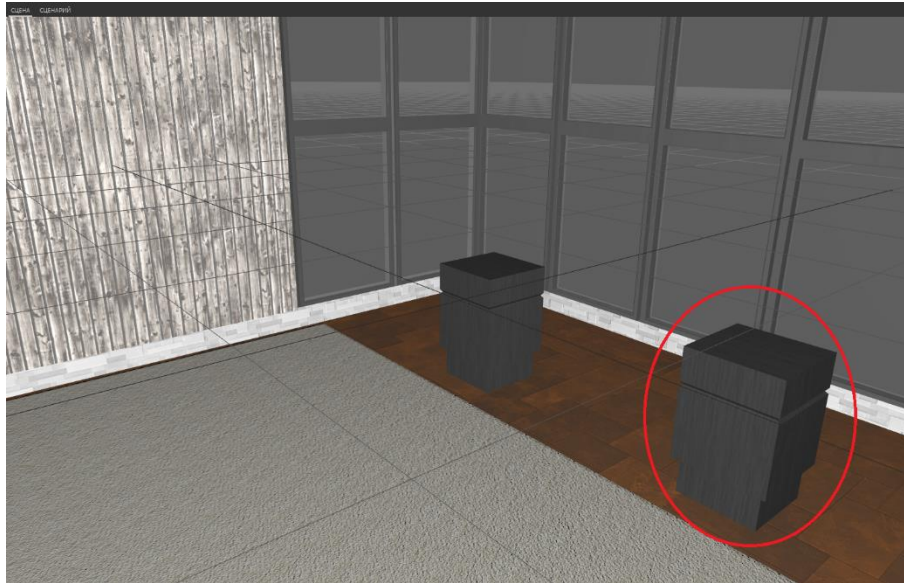
ПРИМЕЧАНИЕ: обратите внимание на особенности работы в пространстве виртуальной сцены в процессе разработки VR проекта. Чтобы переместить сцену в 2D пространстве (вправо/влево/вверх/вниз) можно зажать колесико мыши. Чтобы отдалить или приблизить виртуальное окружение, прокручиваем колесико мыши вперед-назад. Для поворота сцены в трехмерном пространстве достаточно зажать левую кнопку мыши и повернуть ее в нужную сторону.

Чтобы упростить дальнейшую работу с тремя другими моделями таких же тумб, воспользуемся функцией дублирования объектов. В дереве объектов правой кнопкой мыши нажимаем по модели Stand_sun и в выпадающем меню выбираем функцию **“Дублировать”** → **“Дублировать”**.

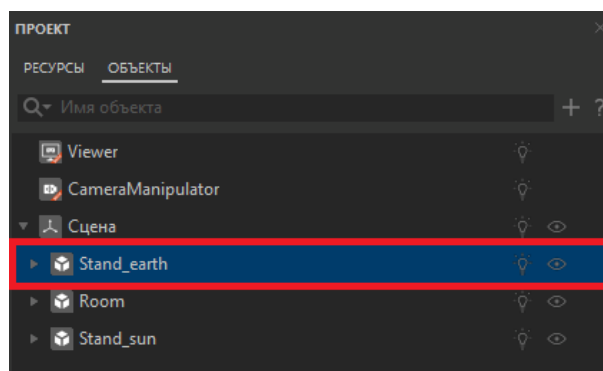
Копия создана, но мы ее не видим потому, что она находится на том же месте, что и оригинал. Для того, чтобы перенести вручную модель на другое место воспользуемся параметром Перемещение в свойствах новой модели.

Изменим координату Y на 0 для смещения вправо.

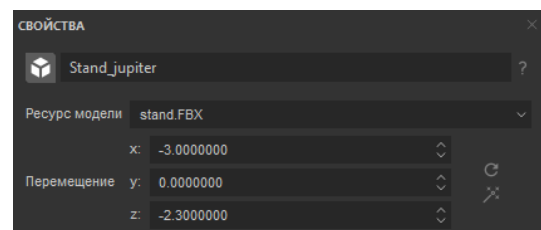
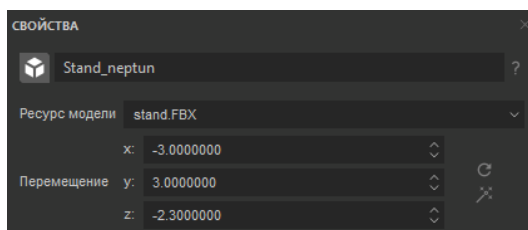




Обратите внимание на название новой модели-копии. Оно в дальнейшем может запутать нас при работе со сценарием проекта, поэтому переименуем модель, назвав ее "Stand_earth".

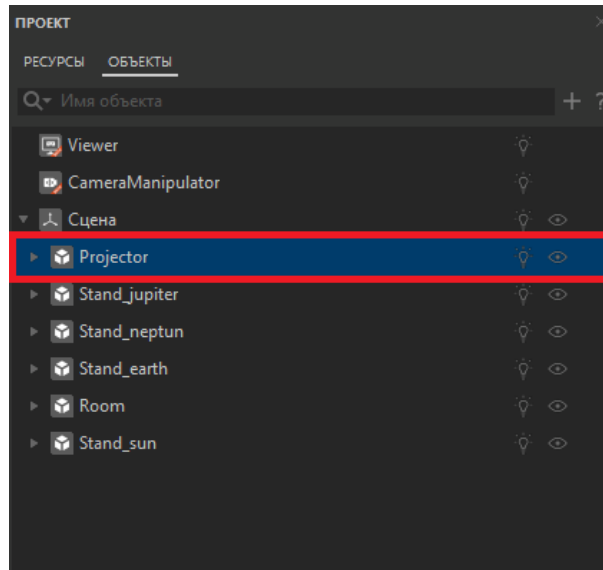


По аналогии с новой моделью создадим две копии тумб и разместим их напротив первых двух моделей. Меняем названия у созданных копий на "Stand_neptun" и "Stand_jupiter".

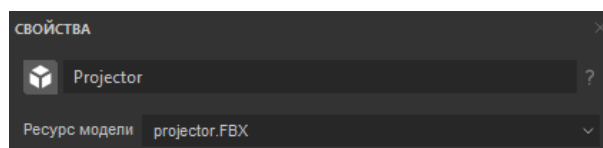


Теперь нам необходимо добавить в проект модели проектора и экрана, на которых по нашей задумке будет отображаться дополнительная информация о выбранных планетах. Начнем с модели проектора.

Открываем окно добавления объектов, выбираем объект Модель и добавляем его в проект. Меняем название новой модели на "Projector".

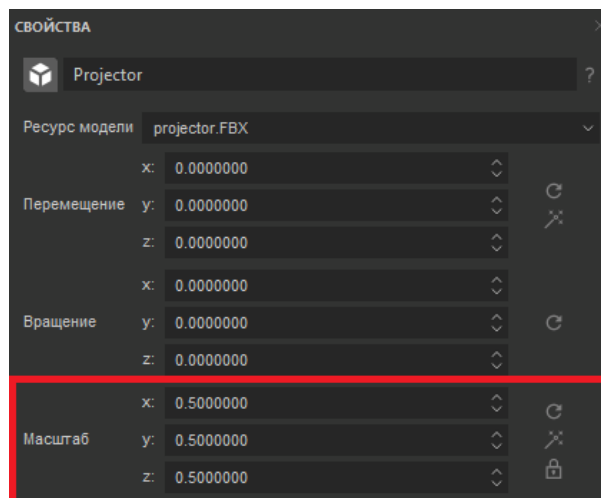


Определим ресурс для модели в панели свойств нового объекта: параметр Ресурс модели → Создать новый. В открывшемся окне ищем папку Models и выбираем модель Projector.

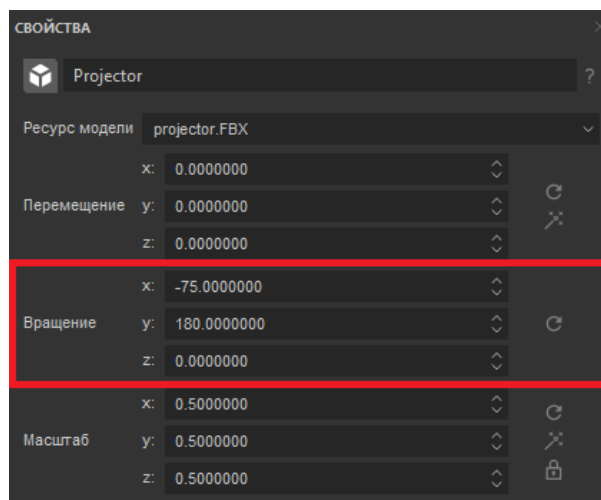


Модель проектора добавлена в сцену. Нам нужно ее уменьшить и корректно разместить в виртуальном окружении музейной комнаты. Для этого обратимся к параметрам Перемещение, Вращение и Масштаб в свойствах модели.

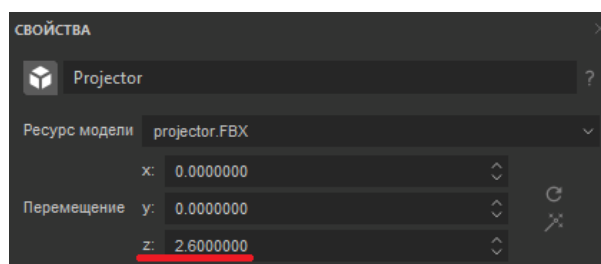
Начнем с масштабирования. Уменьшим модель проектора, так как сейчас ее размеры очень большие для комнаты. Изменим масштаб по оси X на 0.5, две другие координаты будут изменены автоматически.



Теперь повернем проектор таким образом, чтобы его объектив был направлен на стену перед нами. Для этого в параметре Вращение по оси X устанавливаем значение -75° , а по оси Y 180° .

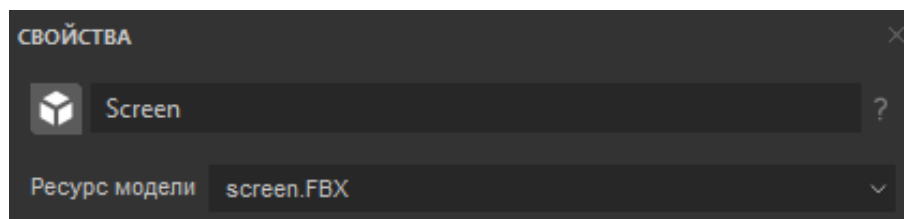


Осталось только "повесить" наш проектор на потолок. Для этого в параметре Перемещение поменяем положение модели по оси Z на 2.6.



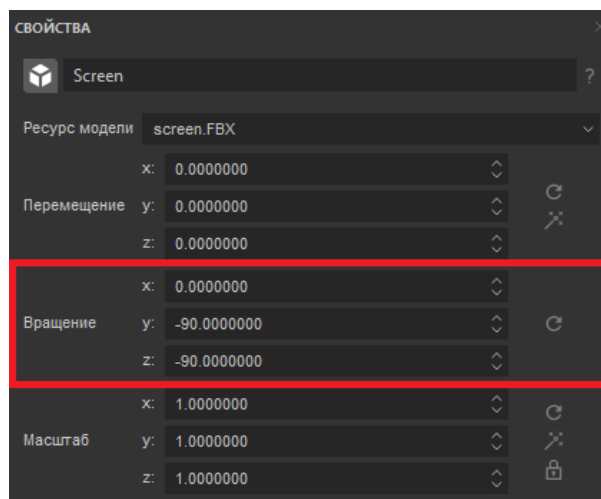
ПРИМЕЧАНИЕ: при работе с моделями в пространстве виртуальной сцены смещается начальное положение сцены. Чтобы в него вернуться, достаточно нажать левой кнопкой мыши на любое место внутри сцены и затем нажать клавишу “Пробел”.

Нам осталось добавить в проект модель экрана проектора. Открываем окно добавления объектов и выбираем объект Модель. В свойствах нового объекта меняем название модели на Screen и определяем для него ресурс. В папке с ресурсами ищем модель screen и открываем ее в проекте.

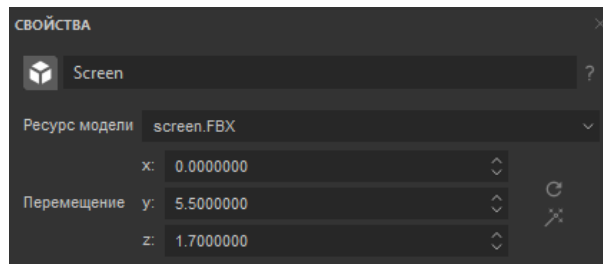


Модель появилась в сцене, но, как и предыдущие модели, нам нужно корректно ее расположить в пространстве виртуального окружения. А именно, повесить на стену перед нами между окон.

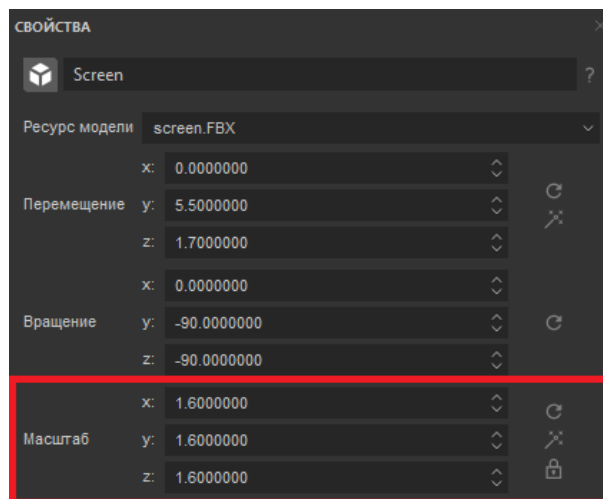
Первое, что нам предстоит сделать – это повернуть модель. В параметре Вращение меняем поворот по осям Y и Z на -90 градусов.



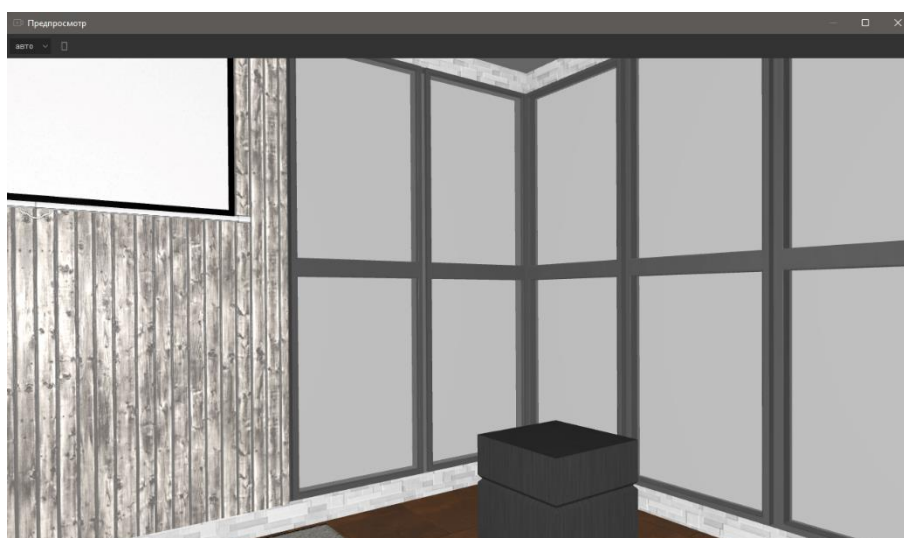
Переместим наш экран на стену. В параметре Перемещение меняем координату Y на 5.5, а Z на 1.7.



Увеличим модель экрана, так как сейчас ее размеры очень большие для комнаты. Изменим масштаб по оси X на 1.6, две другие координаты будут изменены автоматически.

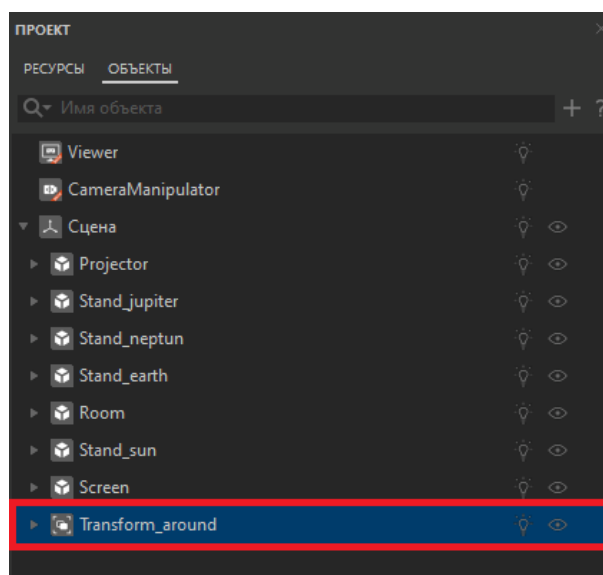


Теперь все добавленные модели органично вписываются в окружающее пространство. Запустим режим Предпросмотра и убедимся в этом.

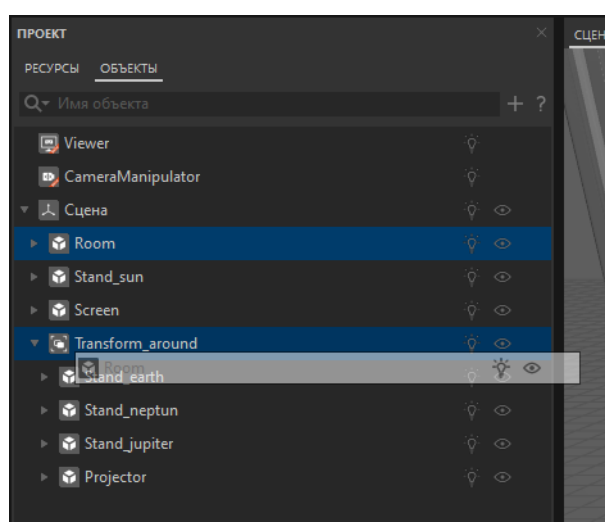


Как вы успели увидеть на протяжении всего этого занятия, работа с каждой моделью – это довольно трудоемкий процесс, который требует некоторого времени и привычки. Для удобства дальнейшей работы с функциями трансформации объектов оптимизируем наш проект и добавим еще один объект – **Систему координат**.

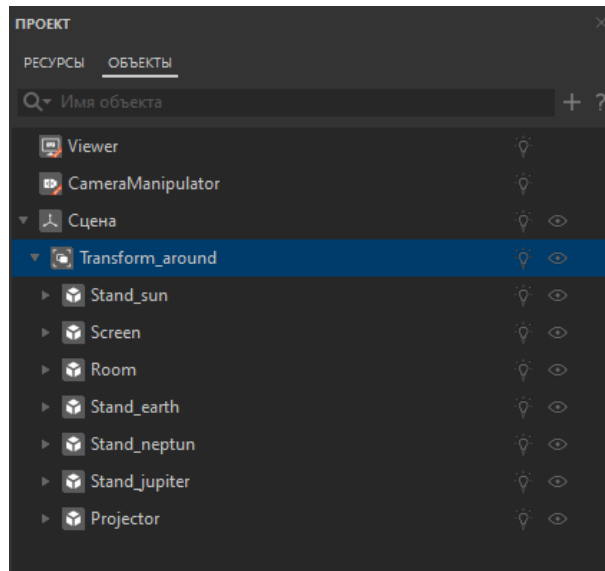
Так как у нас в проекте будет несколько систем координат, то нам нужно дать уникальное имя этому объекту. Назовем его “Transform_around”.



Перенесем все добавленные ранее модели в эту систему координат (сделаем дочерними объектами) с помощью левой кнопки мыши, таким образом сгруппировав их.



Теперь, если нам нужно будет трансформировать или перенести все 3D объекты сцены сразу, то мы можем трансформировать только систему координат, а не каждую модель отдельно. Это значительно сократит временные затраты.



Не забудьте корректно сохранить ваш проект без потери ресурсов.

Этап 3. Продолжаем работать над VR проектом. Добавляем 3D модели музейных экспонатов.

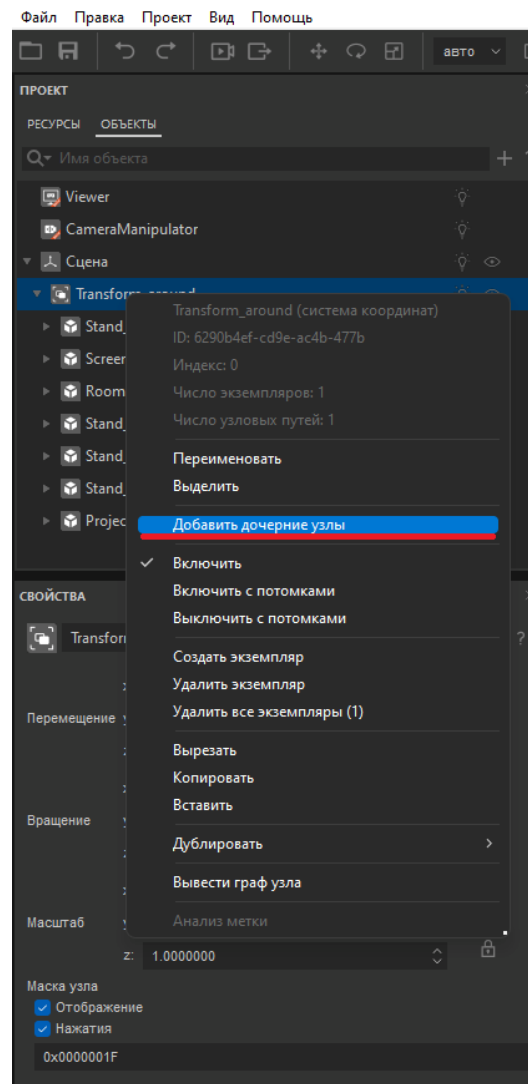
(время работы: 20 минут)

На этом этапе мы продолжаем разработку визуального представления простого VR проекта. В нашей музейной комнате не хватает экспонатов, которые будут размещаться на тумбах, которые мы уже добавили в предыдущем занятии. Так как мы разрабатываем музей астрономии, то на них будут размещаться анимированные модели планет солнечной системы.

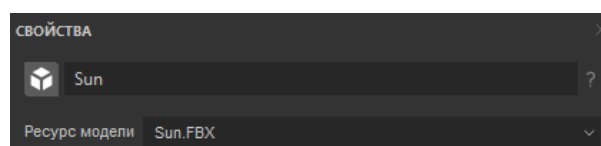
Начнем с модели Солнца.

В конце прошлого этапа мы для удобства и оптимизации временных затрат на разработку мы добавили в проект объект Система координат, в которую переместили все загруженные в сцену 3D модели. Новая модель будет также являться частью Сцены, поэтому добавим ее в созданную систему координат Transform_around.

Найдем в дереве объектов "Transform_around" и нажмем на нее правой кнопкой мыши. В выпадающем меню выберем пункт "Добавить дочерние узлы".

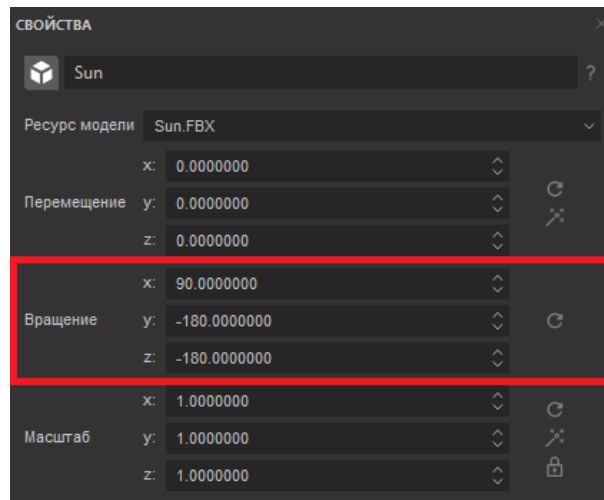


В открывшемся окне добавления объектов выбираем объект Модель и добавляем его в проект. Дальнейшая работа с этим объектом ничем не будет отличаться от работы с предыдущими моделями. В свойствах нового объекта меняем название модели на "Sun" и определяем для нее ресурс. В папке с ресурсами ищем модель Sun и открываем ее в проекте.



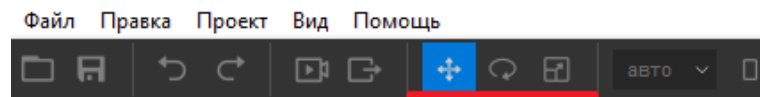
Корректно разметим модель в сцене над моделью Stand_sun. Для этого в свойствах модели Sun изменим параметр Вращения.

Для параметра Вращение меняем поворот по оси X на 90°, Y и Z на 180°.



В свойстве Вращение меняем координату X на 90 градусов, Y и Z на 180°.

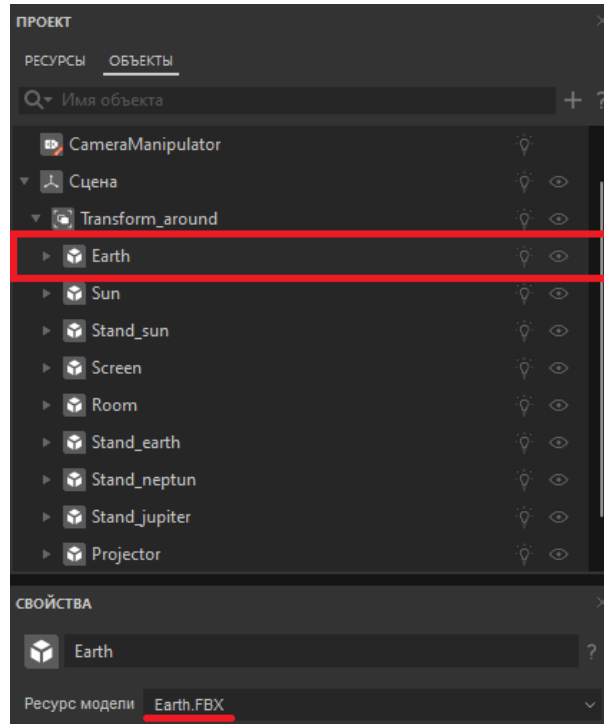
Теперь переместим модель. Отдадим сцену с помощью колесика мыши, чтобы полностью видеть модель. Так как модель будет перемещена на произвольное количество единиц, воспользуемся инструментом произвольного перемещения в панели быстрого доступа.



Поочередно потянем за оси координат, перемещая модель в пространстве сцены и размещая ее над соответствующей тумбой.

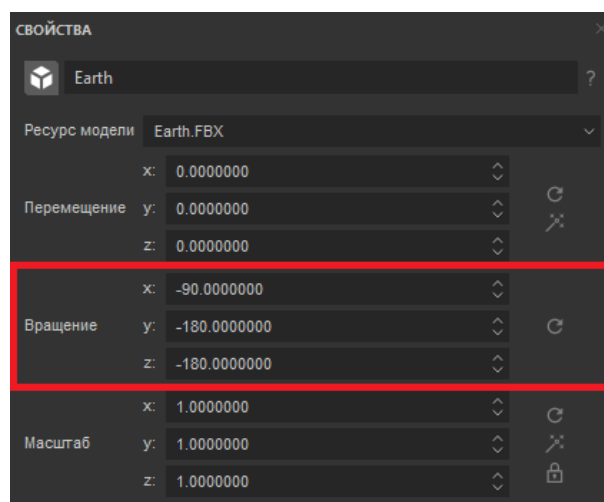


По аналогии с моделью Солнца добавим модель Земли как дочерний объект системы координат Transform_around.

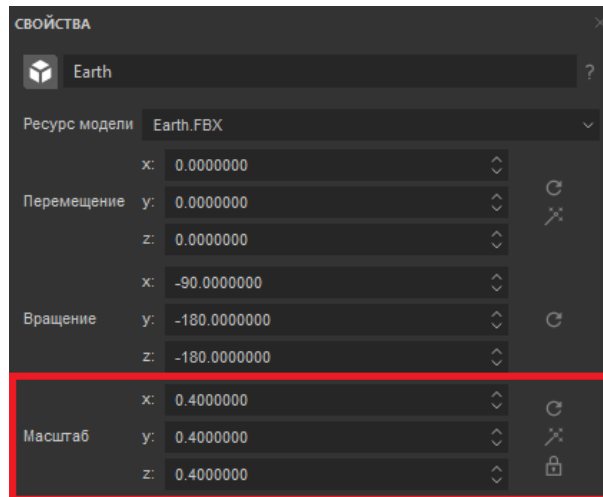


Корректно разметим модель в сцене над моделью Stand_earth. Для этого в свойствах модели Earth изменим параметры Вращения и Масштаба.

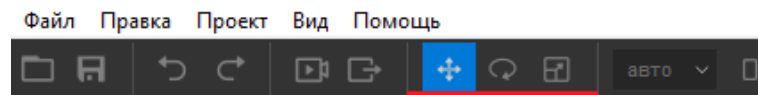
В свойстве Вращение меняем координату X на 90° градусов, Y и Z на -180°.



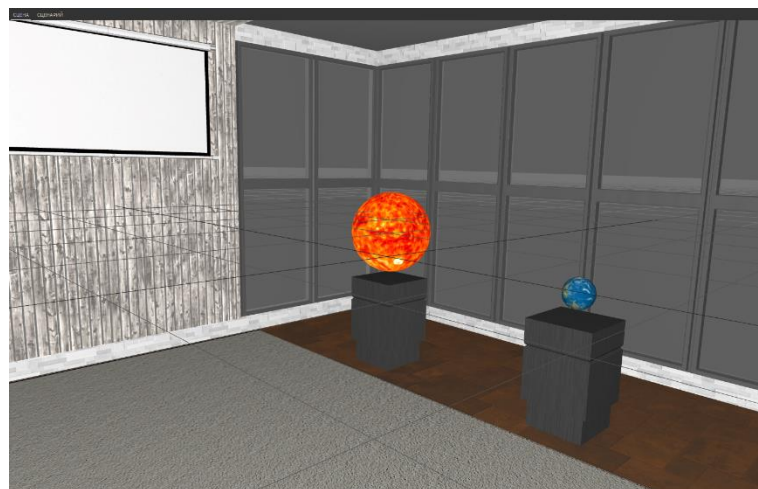
Так как модель Земли должна быть меньше модели Солнца для соблюдения соотношения масштабов планет в Солнечной системе, уменьшим ее масштаб по оси X до 0.4, две другие координаты будут изменены автоматически.



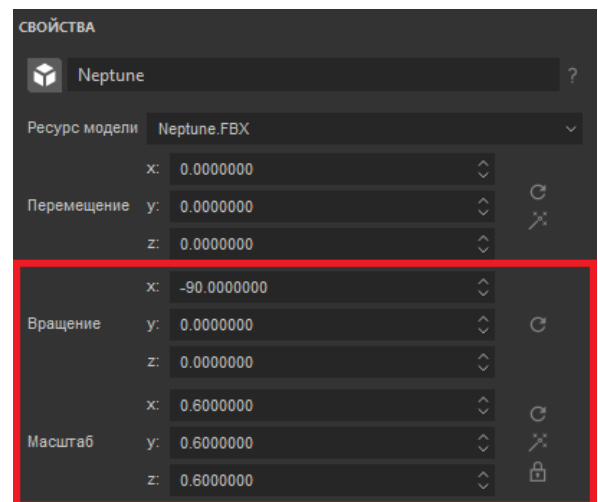
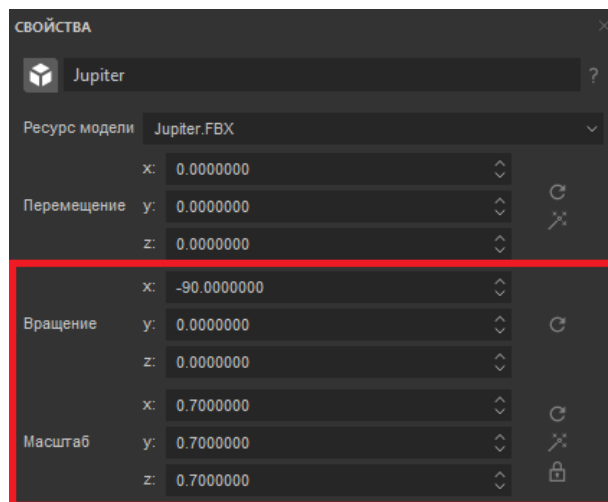
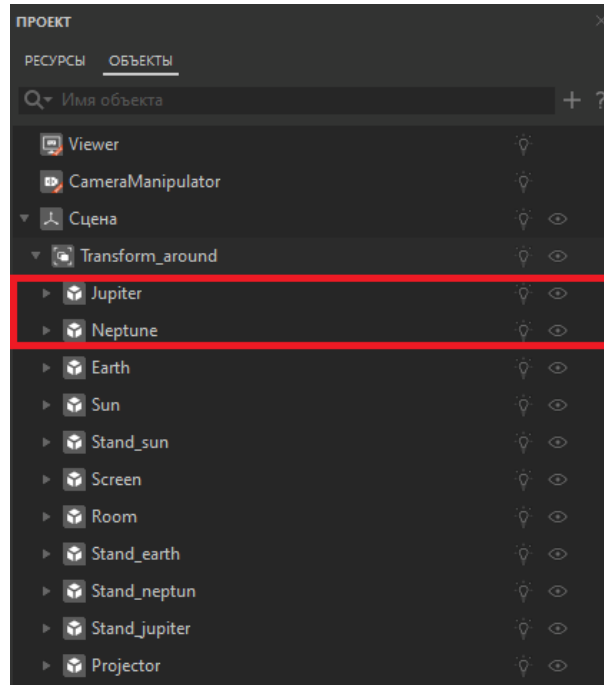
Переместим модель земли. Вновь воспользуемся инструментом произвольного перемещения в панели быстрого доступа.

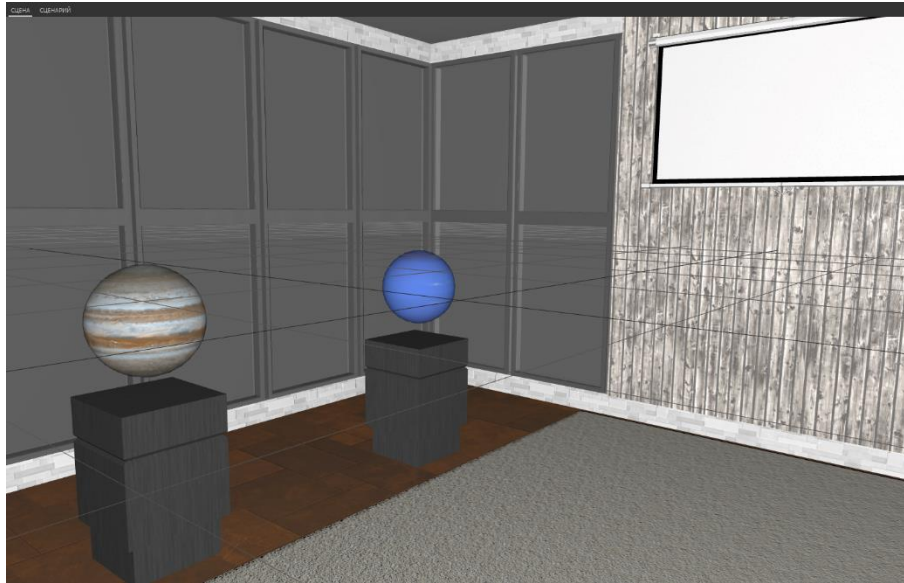


Поочередно потянем за оси координат, перемещая модель в пространстве сцены и размещая ее над соответствующей тумбой.



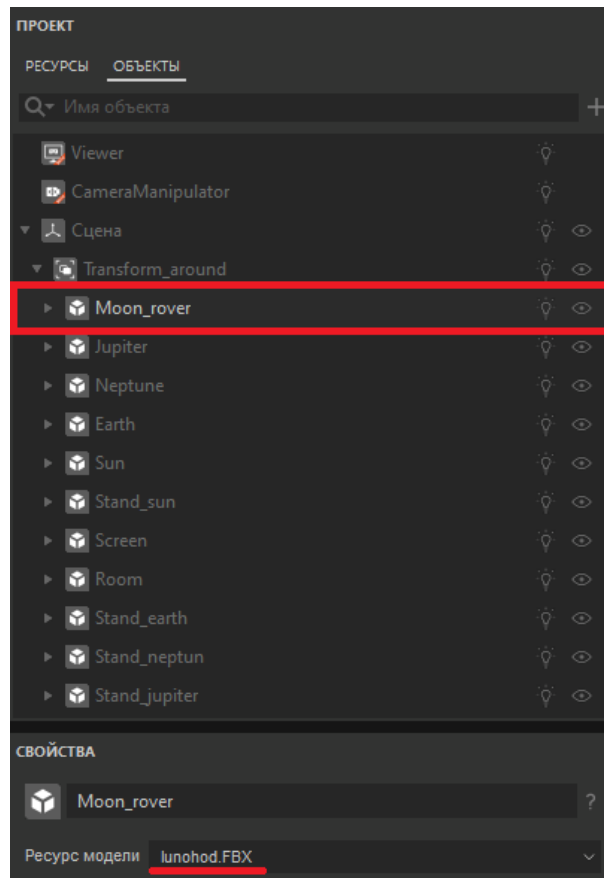
По аналогии с моделями Солнца и Земли, разместите в сцене модели планет Нептун и Юпитера в соответствии с параметрами ниже.



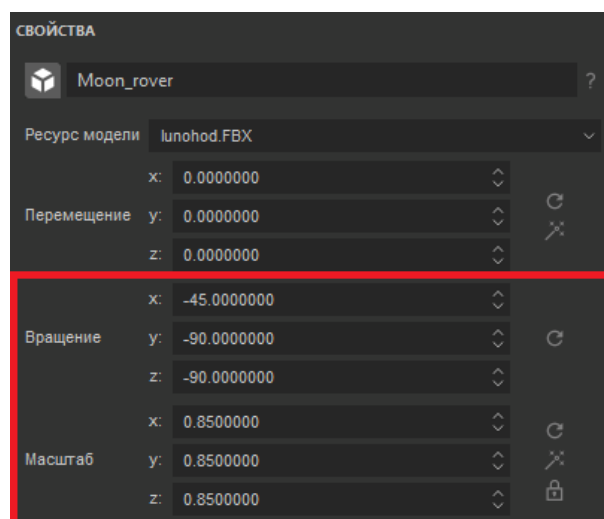


Модели экспонатов – планет Солнечной системы на своих местах. Осталось добавить последний экспонат - модель лунохода.

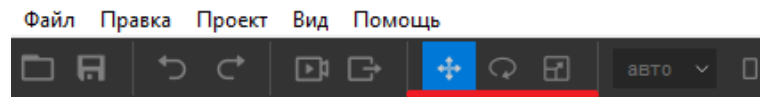
По аналогии с предыдущими моделями создаем дочерний объект для системы координат "Transform_around". В открывшемся окне добавления объектов выбираем объект Модель и добавляем его в проект. В свойствах нового объекта меняем название модели на "Moon_rover" и определяем для нее ресурс. В папке с ресурсами ищем модель lunohod.FBX и открываем ее в проекте.



Корректно разметим модель в сцене. Для этого в ее свойствах изменим параметры Вращения и Масштаба.



Переместим луноход чуть дальше от центра сцены и поставим его на пол. Воспользуемся инструментом произвольного перемещения в панели быстрого доступа.



Поочередно потянем за оси координат, перемещая модель в пространстве сцены.



ВАЖНО: не придвигайте модель лунохода к стене слишком близко, так как у этой модели есть анимация открытия верхней части. И при ее запуске может произойти ситуация, когда открывшаяся крыша лунохода может при открытии “утонуть” в модели окружения комнаты. Это будет выглядеть нереалистично, поэтому проверим, корректно ли расположена модель Лунохода. Для этого выделим модель лунохода в дереве объектов и запустим ее анимацию, нажав на кнопку Play в панели быстрого доступа.



Не забудьте корректно сохранить ваш проект без потери ресурсов.

Этап 4. Работаем с объектами VRcontroller и Поиск пересечений (Intersector).

(время работы: 20 минут)

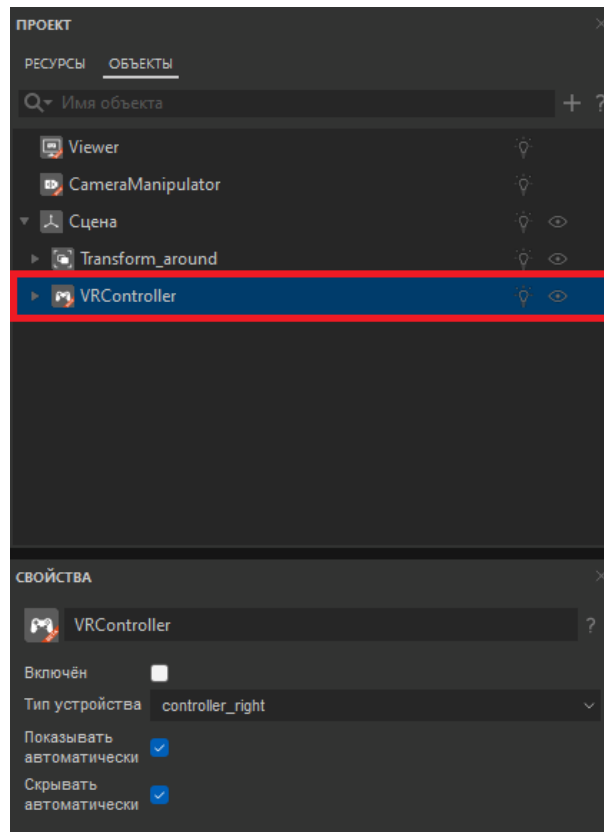
К концу работы на Этапе 3 мы получили визуальное представление простого VR проекта. Такой проект уже можно экспортировать на целевое устройство – автономный шлем виртуальной реальности и увидеть созданное виртуальное окружение, осмотреться в нем и пройтись. Однако взаимодействовать с каким-либо объектом в таком приложении пользователь не сможет.

Наверняка, вы уже знаете, что при приобретении VR шлема в комплекте с ним идут устройства, с помощью которых пользователь будет осуществлять взаимодействие с объектами виртуальной сцены – **VR контроллеры**.

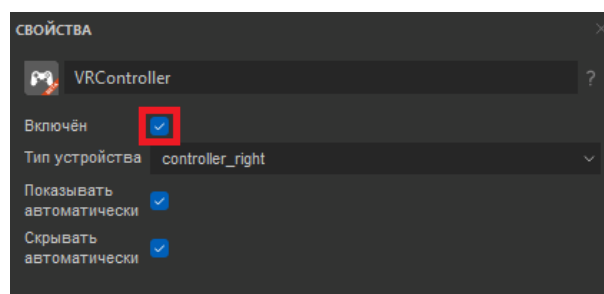


Для работы VR контроллеров шлема и организации возможности взаимодействия пользователя с объектами виртуального мира в проект необходимо добавить еще два базовых объекта VR проекта – **VR controller** и **Intersector**.

Откроем окно добавления объектов и добавим в проект объект **VRcontroller**. Благодаря этому объекту мы сможем настроить **логику работы** нашего проекта таким образом, чтобы пользователь мог взаимодействовать с окружающими объектами созданного виртуального окружения. Выберем VRcontroller в дереве объектов и перейдем в панель его свойств.

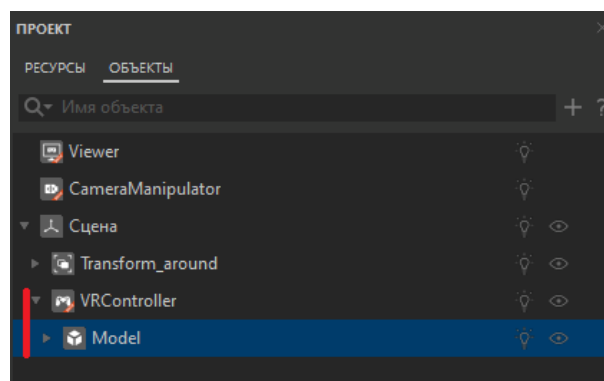
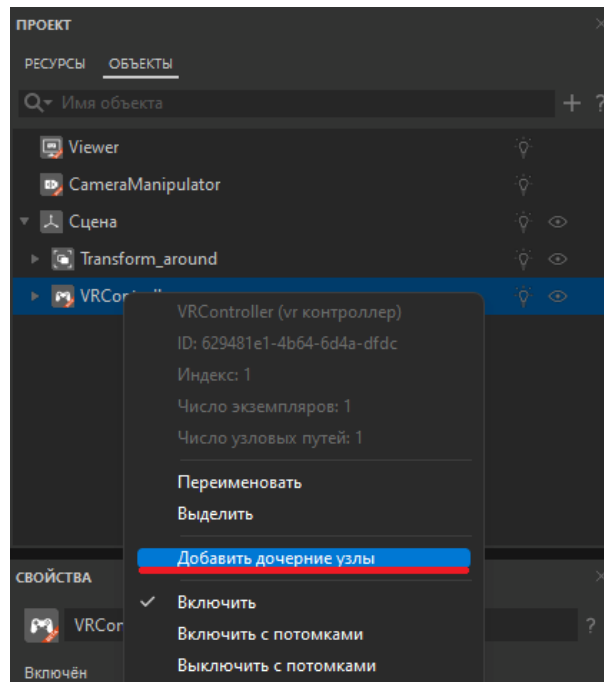


Так как в нашем проекте VR контроллер будет один, то можно не менять его имя. Обратим внимание, что в свойствах объекта не стоит флажок Включён – это означает, что этот объект сейчас выключен и работать в проекте не будет. Поставим флажок для этого параметра, чтобы наш контроллер работал в проекте. Проверим также, чтобы все остальные флажки в свойствах этого объекта были включены.

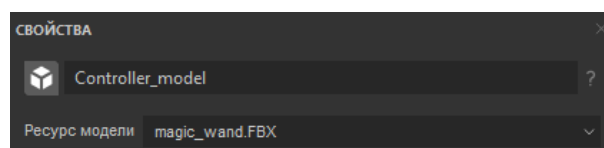


Как мы уже упоминали, при создании виртуальной реальности разработчику приложения важно помнить, что пользователь должен интуитивно понимать, каким образом он будет взаимодействовать с объектами виртуального мира. В связи с этим важно создать для объекта VRcontroller визуальное представление, которое пользователь также будет видеть, надев шлем. Многие разработчики используют различные 3D модели контроллеров VR шлемов, рук и других предметов.

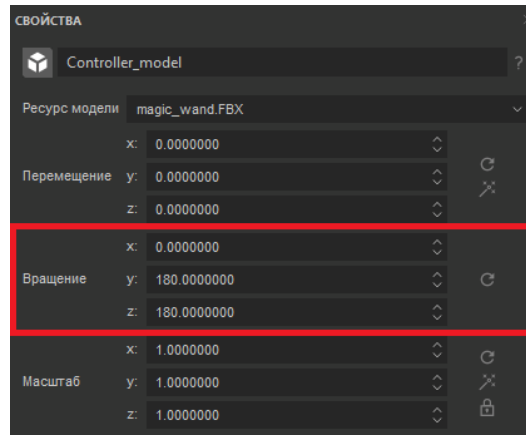
Мы в качестве контроллера решили выбрать для своего проекта модель волшебной палочки. Визуализировать контроллер можно также с помощью объекта Модель, который будет являться дочерним объектом для объекта VR controller.



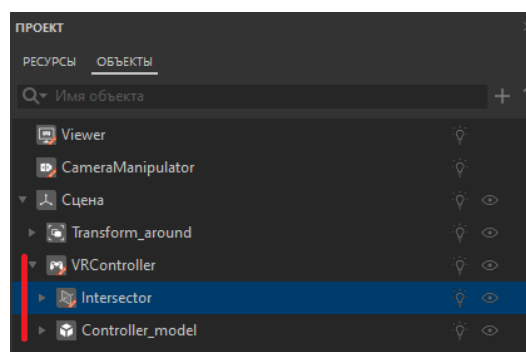
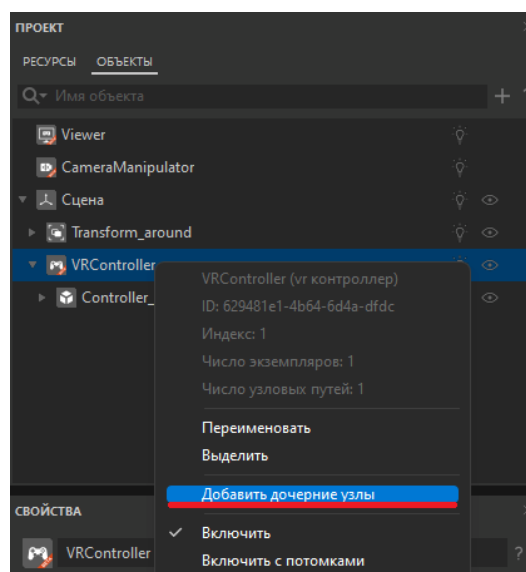
Переименуем модель и назовем ее Controller_model. Определим ресурс для этой модели - Magic_wand.fbx в папке Models в папке с ресурсами проекта.



В параметре Вращение в панели свойств модели меняем координаты Y и Z на 180°.

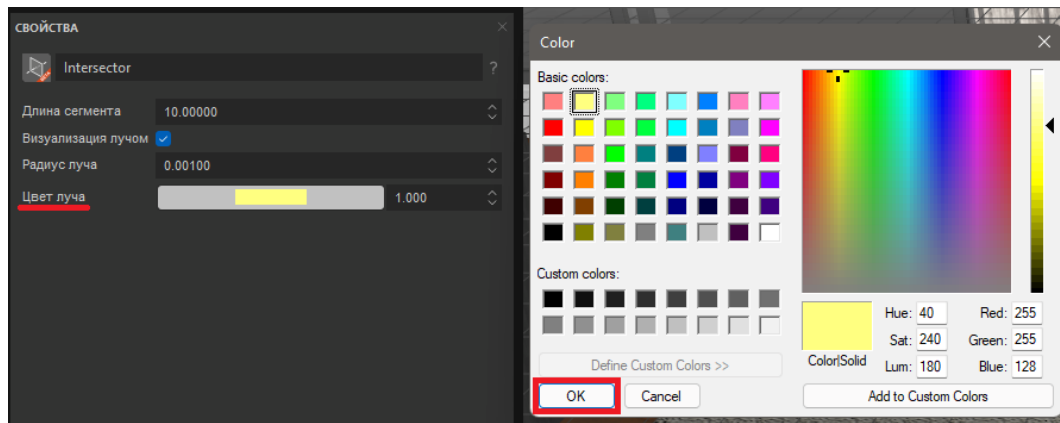


Однако, для взаимодействия с объектами виртуального мира одного объекта VR controller будет недостаточно. Для взаимодействия с объектами необходимо найти пересечение с каждым из таких объектов. В этом нам поможет объект – **Intersector** (Поиск пересечений). Эти объекты работают в связке друг с другом, так как поиск пересечений будет осуществляться по нажатию на кнопку контроллера, поэтому такой объект необходимо добавить как дочерний объект объекта VR controller.



Как только мы добавили Intersector в проект, в сцене появился луч поиска пересечений, выходящий из начала системы координат и из модели контроллера.

Так как конец волшебной палочки окрашен в желтый цвет, поменяем цвет луча на аналогичный цвет в панели свойств объекта Intersector. Выбираем один из базовых желтых цветов, представленных в конструкторе EV Toolbox.



Объекты для организации взаимодействия пользователя с объектами виртуальной сцены добавлены в проект и будут использованы при создании логики работы VR проекта.

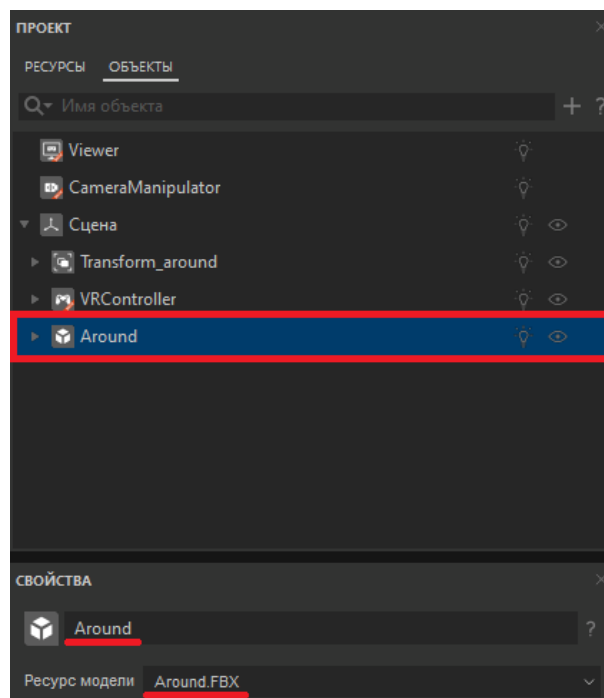
Не забудьте корректно сохранить ваш проект без потери ресурсов.

Этап 5. Работаем с 2D объектами при создании визуального представление VR проекта: объекты Изображение, Текст, Видео.

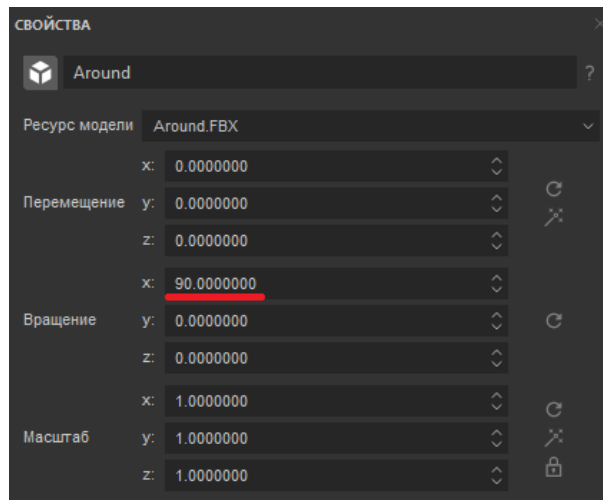
(время работы: 20 минут)

На этом этапе основная визуальная составляющая VR проекта практически готова. Нам осталось добавить в проект еще одну модель окружения, создав за окнами виртуального музея образ яркого звездного неба, а также дополнительный информационный контент в формате 2D объектов: изображений, видео и текста.

Начнем с окружения. Сейчас при запуске предпросмотра мы видим за окнами серый фон сцены. Для того чтобы это изменить, добавим в проект новую модель, назовем ее Around и определим для нее ресурс Around.FBX из папки Models с ресурсами проекта.



Корректно разместим модель в сцене. Для этого в ее свойствах изменим параметры Вращения.

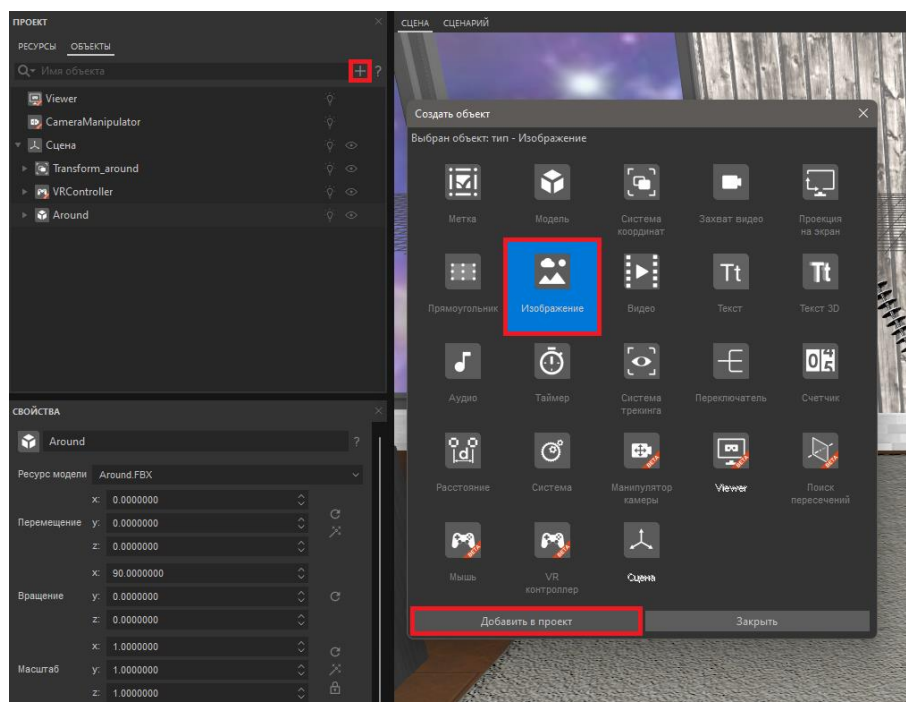


На этом работа с 3D моделями в рамках проекта завершена.

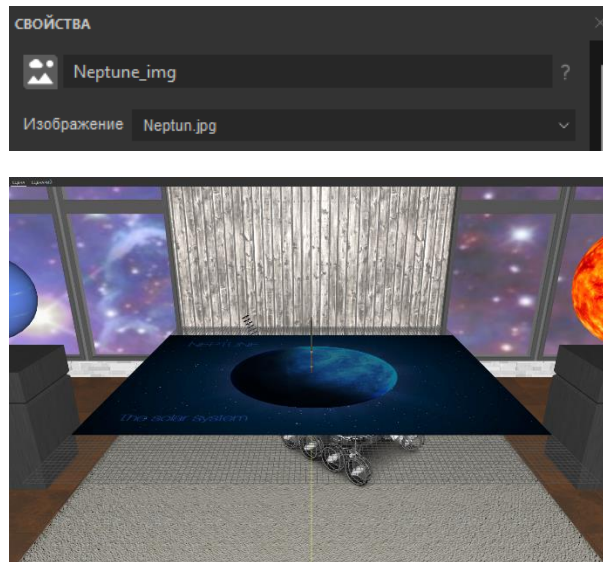
Перейдем к работе с дополнительным контентом в проекте. В наборе ресурсов у нас есть изображение планеты Нептун, небольшая текстовая справка о Юпитере, а также видео о Солнце. Все эти объекты появляются в определенный момент при взаимодействии пользователя с моделями Солнца или планет (по нажатию на них).

Весь дополнительный контент будет показан в тот или иной момент на экране проектора, поэтому все объекты будем размещать в области экрана.

Первое, что мы добавим, это изображение Нептуна. Для этого воспользуемся объектом **Изображение** в окне добавления объектов.



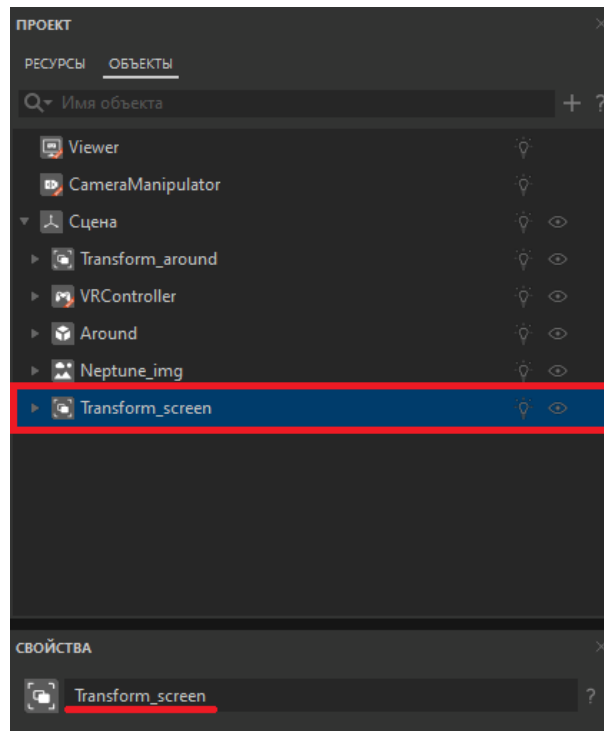
Выделим объект в дереве объектов проекта и перейдем в панель его свойств. Переименуем объект и назовем "Neptune_img". Определим ресурс для нашего изображения – "Neptun.jpg" из папки Images ресурсов проекта.



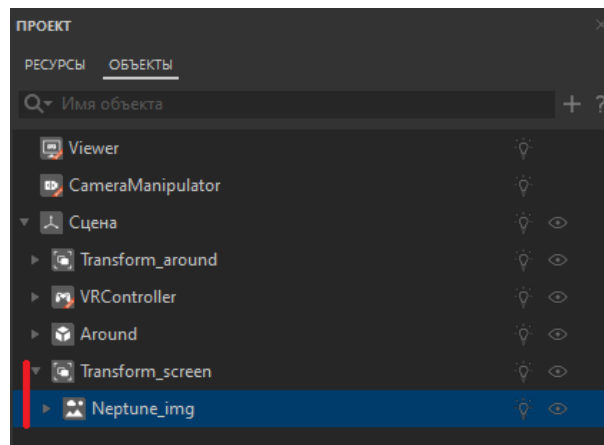
Изображение добавлено в пространство виртуальной сцены в начало системы координат. Так как изображение является двумерным объектом по умолчанию его можно перемещать только по двум осям: X и Y.

Однако сейчас мы работаем в пространстве трехмерной сцены (X, Y и Z), поэтому для трансформации (Перемещения, Вращения и Масштабирования) объектов типа Прямоугольник (Изображение, Текст, Видео и др.) в трехмерной сцене необходимо использовать объект **Система координат**. Добавим в проект еще один объект Transform в проект, найдем его в дереве объектов и перейдем в панель его свойств.

Переименуем объект и назовем "Transform_screen". В этой системе координат будут находиться все объекты дополнительного 2D контента.

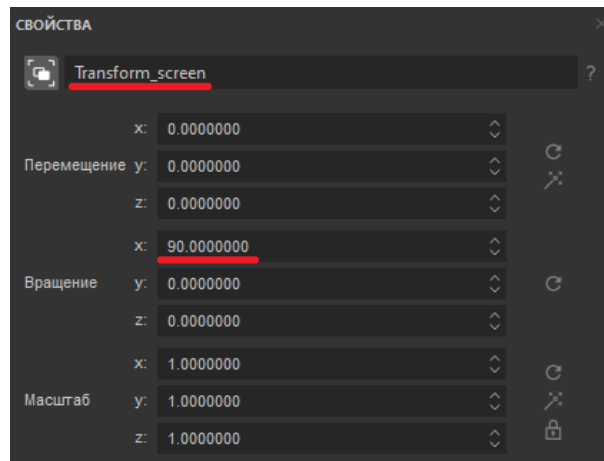


Сделаем добавленное ранее Изображение дочерним (зависимым) объектом для нашей Системы координат. Для этого в дереве объектов мы переносим изображение прямо под объект "Transform_screen".



Объект Изображение теперь "принадлежит" объекту Система координат "Transform_screen" (является его дочерним объектом). Теперь мы можем перемещать и вращать изображение по трем координатам с использованием Transform_screen.

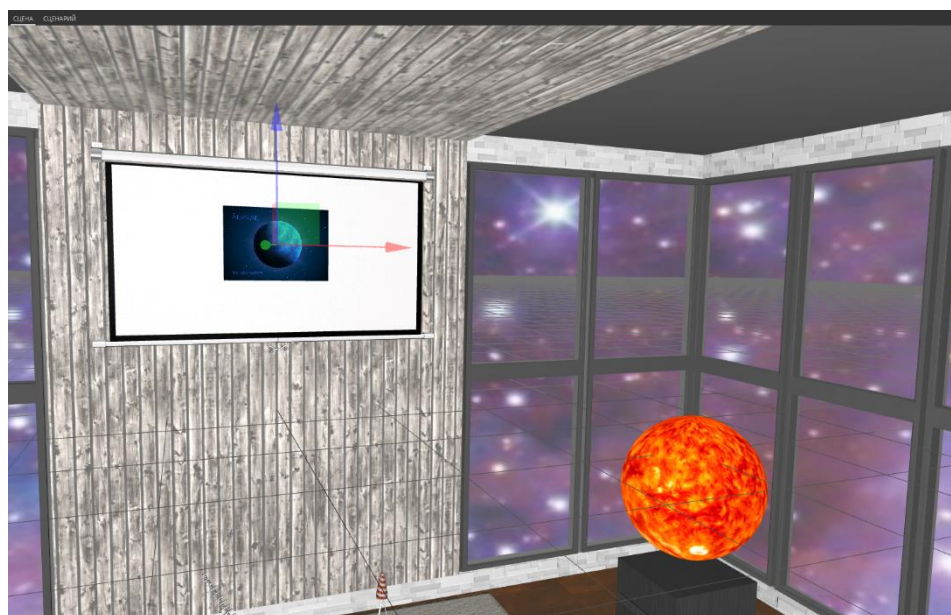
Выделим объект "Transform_screen" в дереве объектов и перейдем в панель его свойств. Повернем систему координат по оси X на 90о.

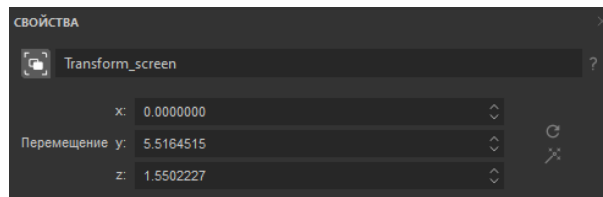


Затем воспользуемся инструментом произвольного Перемещения объектов в панели быстрого доступа.



Поочередно тянем за появившиеся оси координат и располагаем изображение произвольно по центру экрана. Не забудем отключить инструмент Перемещение, еще раз нажав на него.

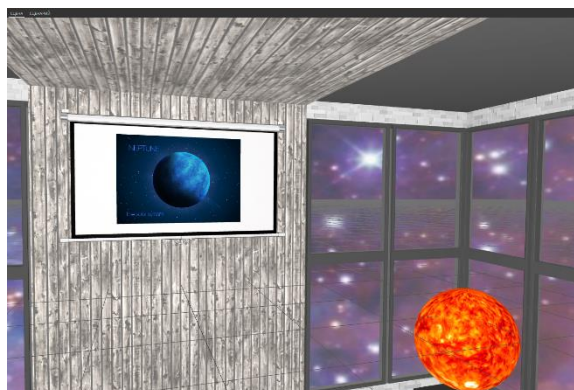




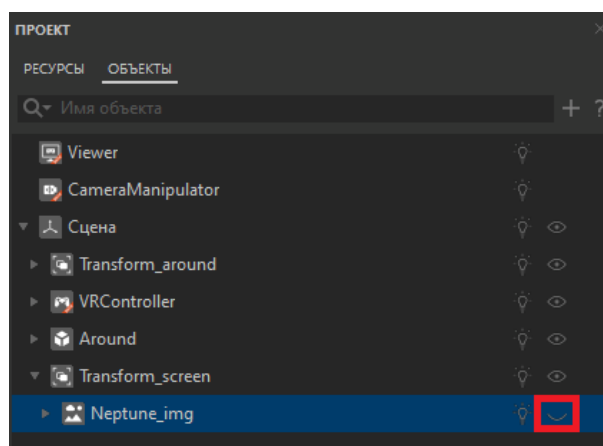
Увеличим изображение на экране проектора, используя инструмент произвольного Масштабирования в панели быстрого доступа.



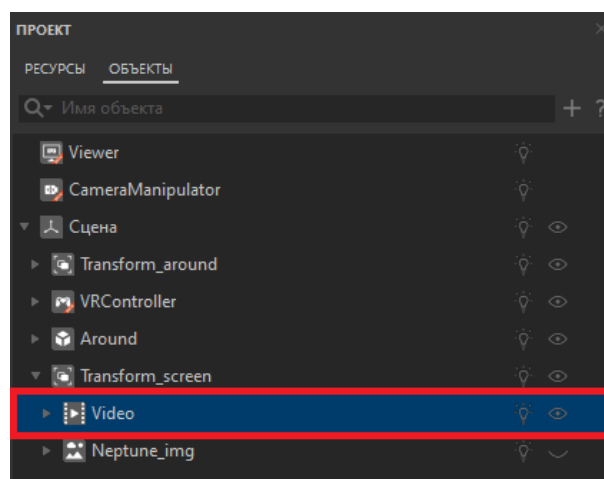
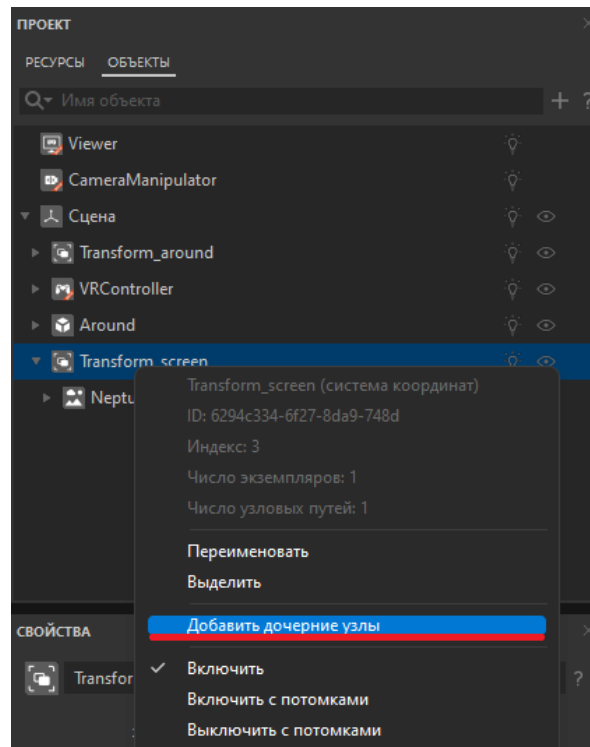
Также тянем за один из концов драггера, произвольно пропорционально увеличивая изображение. После увеличения можно еще раз выровнять изображение с помощью инструмента произвольного перемещения по центру.



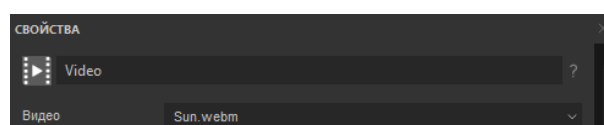
Так как изначально вся дополнительная информация в проекте скрыта, скроем объект Изображение в дереве объектов проекта с помощью функции "Скрыть/показать видимость объектов".



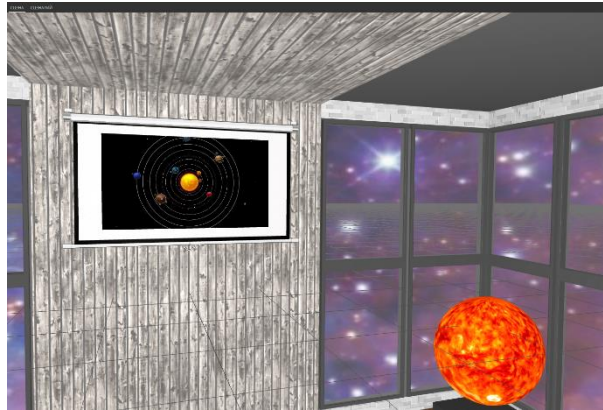
При взаимодействии пользователя с моделью Солнца (по нажатию на модель) на экране проектора должно быть показано обучающее видео об этой гигантской звезде. Создадим для системы координат "Transform_screen" еще один дочерний объект – **Видео**.



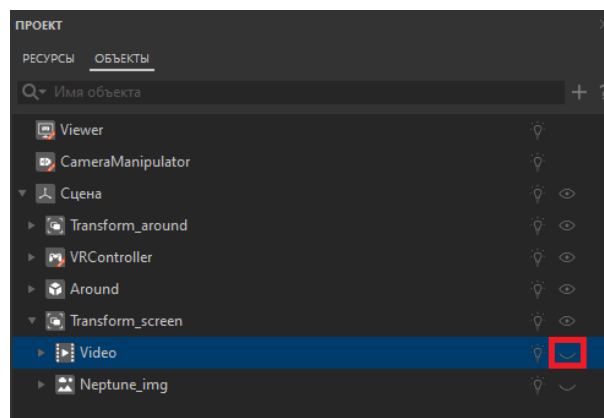
Назовем новый объект "Video_sun" и определим для него ресурс – видео Sun из папки "Video" в папке с ресурсами проектов.



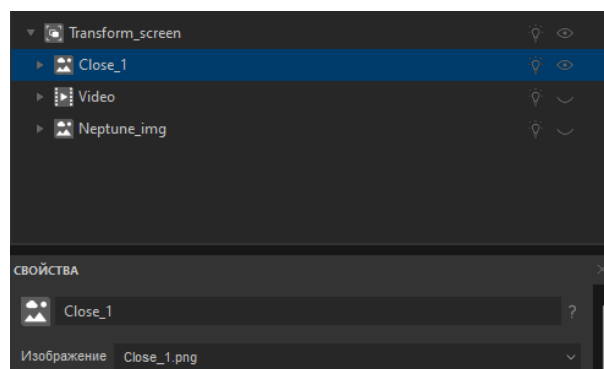
Так как для объекта "Transform_screen" уже определены настройки, то видео автоматически подстроилось под размер экрана проектора.



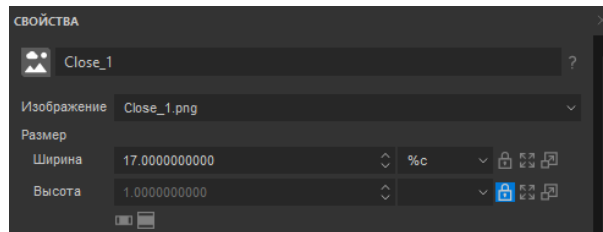
Также скроем видео в дереве объектов.



Добавим также кнопку, которая позволит закрыть видео в случае, если пользователь не захочет смотреть его до конца. Создадим для системы координат "Transform_screen" еще один дочерний объект – **Изображение**, который назовем "Close_1" и определим для него ресурс – изображение close_1 из папки с ресурсами проекта.

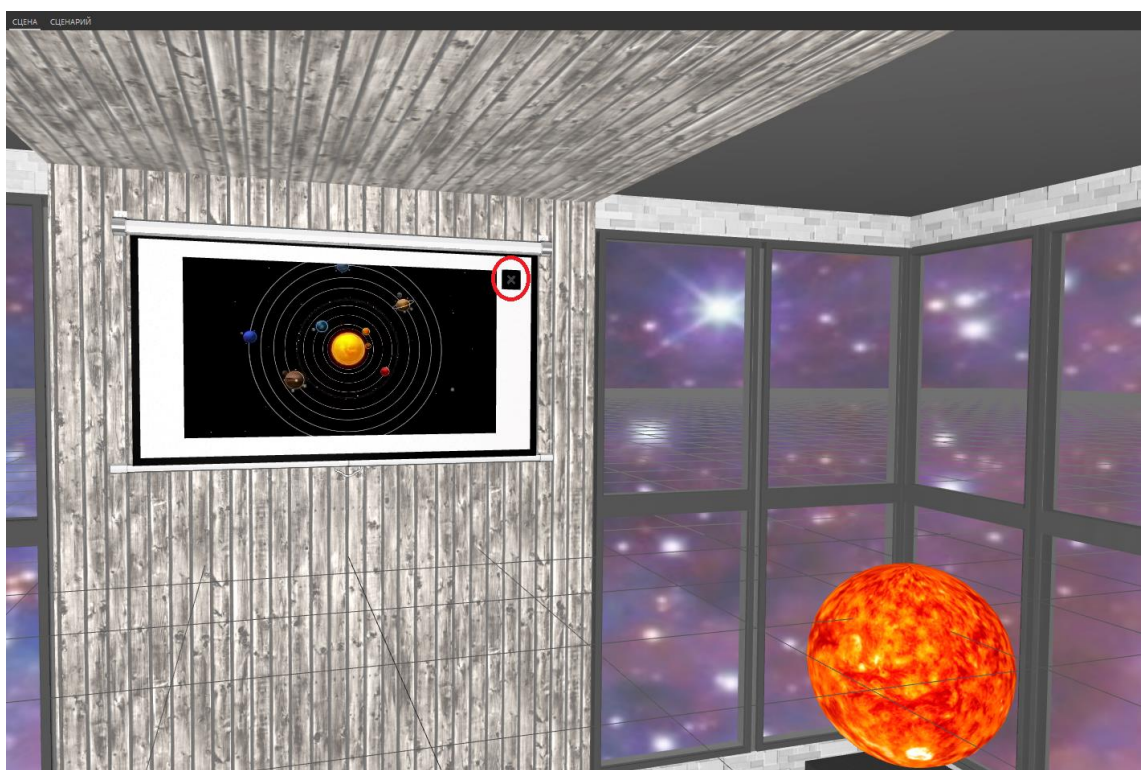


Установим ширину добавленной кнопки как 17% от контента, при этом оставив заблокированной высоту объекта для сохранения соотношения сторон изображения.

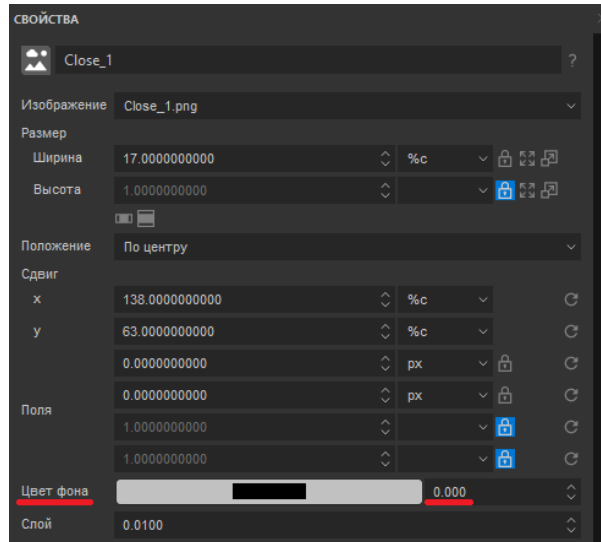


Сместим кнопку в плоскости экрана проектора. Для этого в панели свойств находим параметр "Сдвиг". Меняем единицу измерения по оси x и y с пикселей на процент от контента (%с). По оси X смещаем кнопку на 138 %с, по оси Y на 63 %с.

ПРИМЕЧАНИЕ: в зависимости от размеров видео, размер вашей кнопки и величина смещений может отличаться от приведенных.



Сделаем фон кнопки полностью прозрачным. Найдем в ее свойствах параметр Цвет фона и поменяем значение с 0,75 (то есть 75%) на 0.

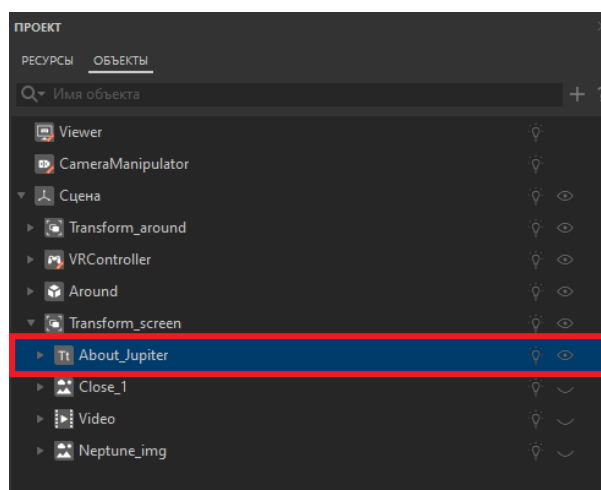


Не забывайте скрывать все дочерние объекты системы координат Transform_screen.

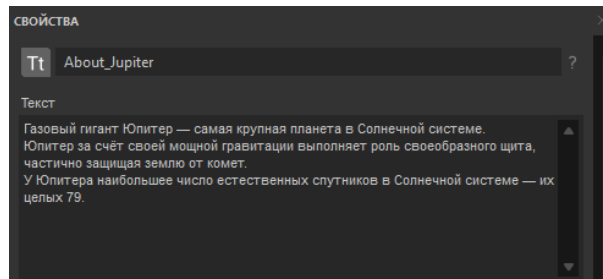


Добавим последний блок дополнительной информации в проект и кнопку для его закрытия. Это будет текстовая информация о планете Юпитер.

Для системы координат Transform_screen создадим еще один дочерний объект – **Текст** и назовем его "About_Jupiter".

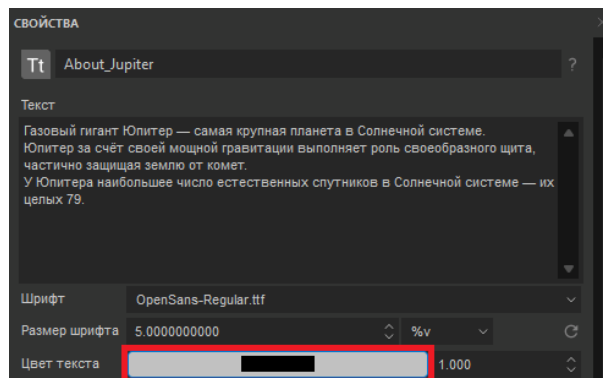


Изменим содержание текста в панели свойств. Копируем ее из doc-файла, расположенного в папке с ресурсами к проекту. Найдем его в папке "Text". Вставляем скопированный текст в поле Текст.

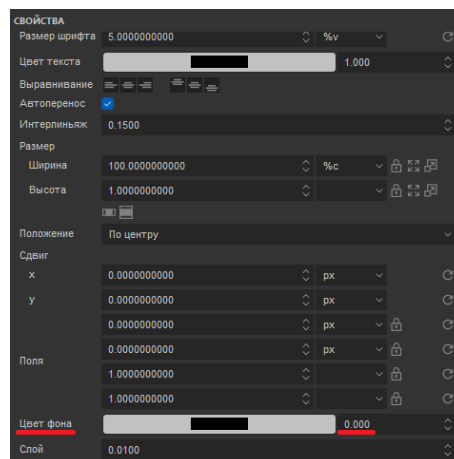


Ниже поставим флажок в параметре Автоперенос и в свойстве Размер шрифта увеличим размер шрифта с 4 на 5, оставив единицы измерения по умолчанию.

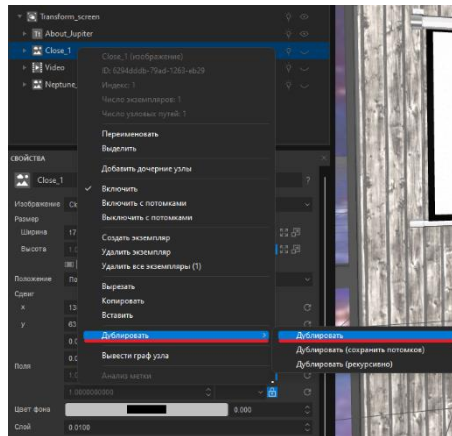
Изменим цвет фона и текста. Начнем с цвета текста: находим в панели свойств текста параметр Цвет текста и меняем его с белого на черный.



Спускаемся в панели свойств ниже, и находим свойство Цвет фона. Меняем его с черного цвета на белый.



После того как пользователь прочтет информацию о Юпитере, ее необходимо будет скрыть. Так как мы уже создали одну кнопку закрыть и настроили ее – Close_1, ее достаточно дублировать и переименовать. Нажмем правой кнопкой мыши на изображение Close_1 и в выпадающем меню выберем “Дублировать” → “Дублировать”.

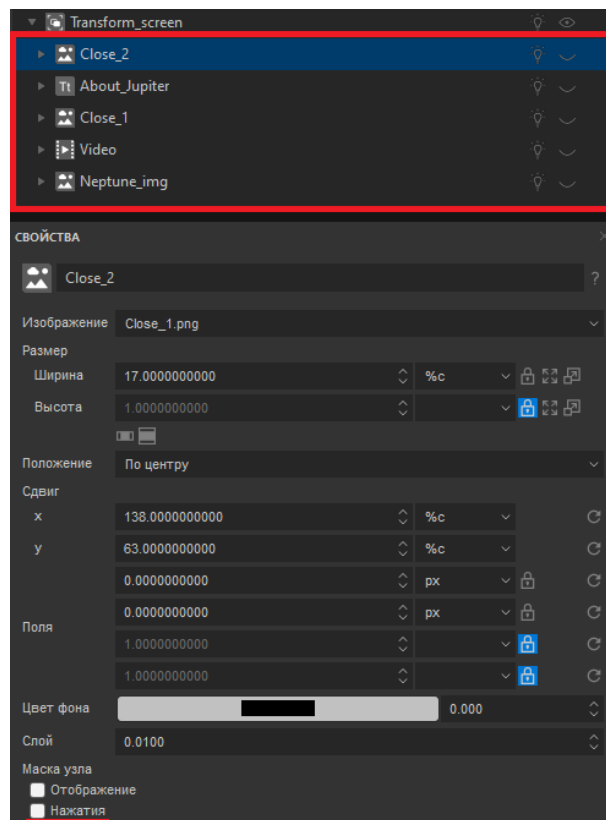


Переименуем новую кнопку и назовем ее Close_2. Все остальные настройки оставляем по умолчанию.

Проверим, чтобы все блоки дополнительной информации, которые являются дочерними объектами Transform_screen, были скрыты в дереве объектов проекта.

ВАЖНО: обратите, пожалуйста, внимание, что при работе в версиях программы 3.4.x в свойствах объектов, имеющих визуальное отображение в сцене, появились новые параметры маски узла: нажатие и отображение. Если ваш объект должен быть заранее скрыт в дереве объектов проекта, то в свойствах этого объекта также важно отключить нажатия на этот объект, чтобы не получилась ситуация, когда вы нажимаете на скрытый объект (если такой функционал не предусмотрен задумкой приложения). В этом случае включать нажатия на объекты мы будем уже в сценарии с использованием действия Объект – показать.

Отключите нажатия для всех скрытых дочерних объектов системы координат Transform_screen (Close_1, About_Jupiter, Close_2, Video, Neptune_img) в панели свойств этих объектов.



На этом этапе мы завершаем создание визуального представления простого VR проекта.

Не забудьте корректно сохранить ваш проект без потери ресурсов.

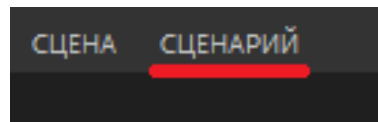
Этап 6. Начинаем работу над созданием сценария простого VR проекта.

(время работы: 30 минут)

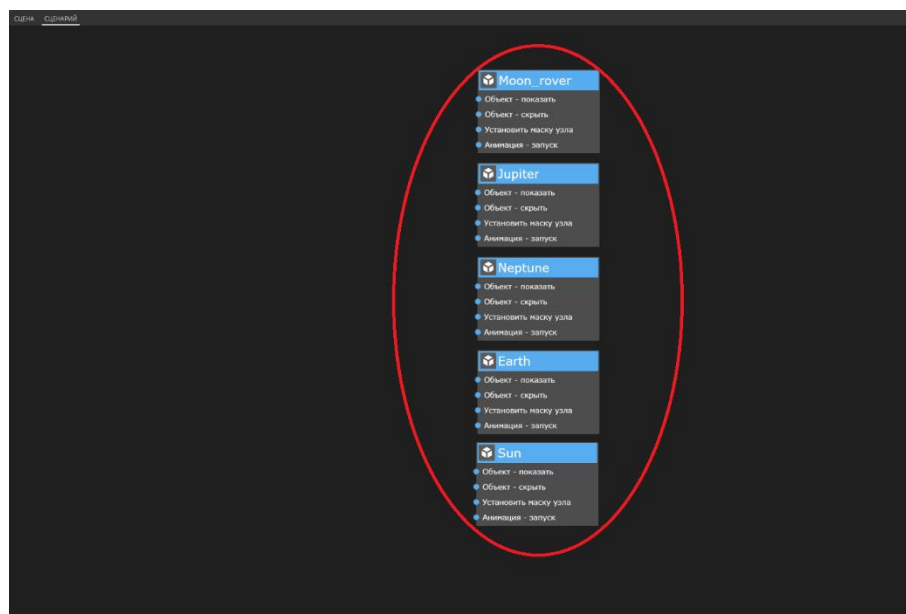
Еще один важный этап разработки VR проекта – это **создание логики работы** будущего VR приложения.

На этом этапе нам предстоит создать логику работы, в соответствии с которой сразу после загрузки приложения пользователь попадет в VR окружение музейной комнаты, наполненной анимированными моделями.

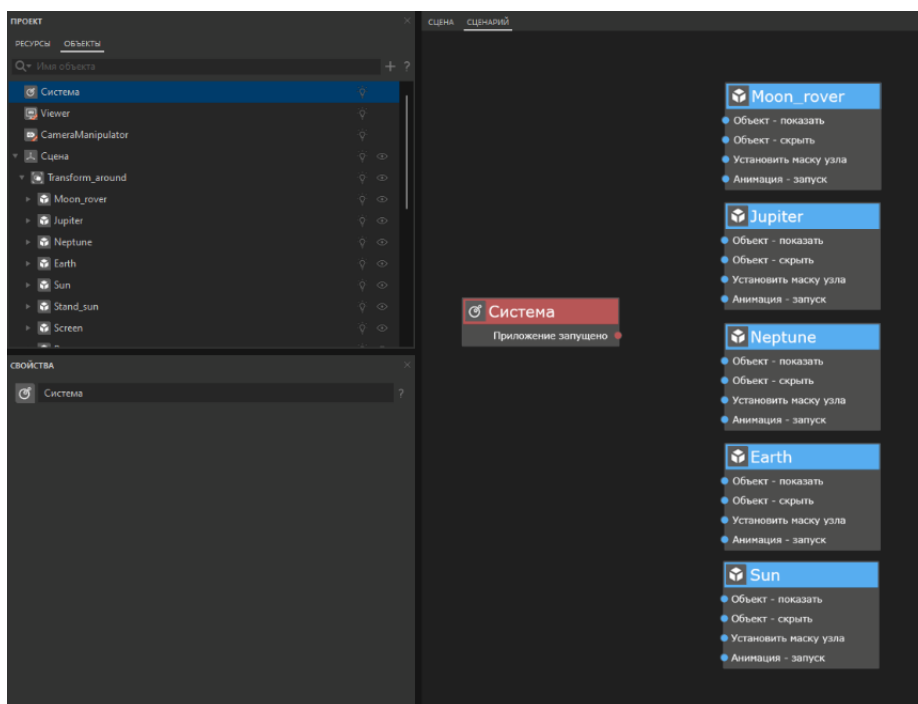
Для начала работы над сценарием (логикой работы) проекта перейдем на вкладку **СЦЕНАРИЙ** проекта.



Поочередно “берем” левой кнопкой мыши в дереве объектов все модели планет и лунохода и переносим их прямо в рабочую область сценария проекта.



Нам также понадобится еще один специальный объект. Открываем окно добавления объектов и выбираем объект **Система**. Сразу перенесем его в область сценария.

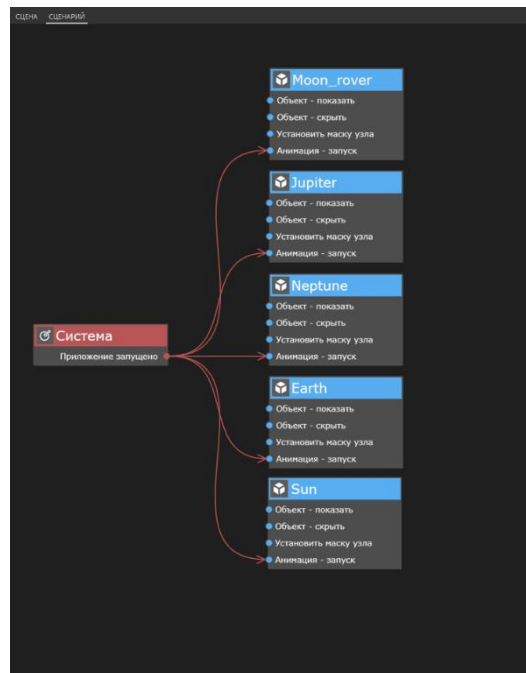


Сценарий проекта состоит из блоков объектов, для каждого из которых доступен список определенных событий и действий. **Событие** - это сигнал программы о том, что с объектом что-то произошло. **Действие** - это команда выполнить некоторую операцию. Чтобы открыть выпадающие списки доступных событий и действий для любого из объектов сценария, необходимо кликнуть правой кнопкой мыши на цветную плашку объекта.

Для удобства работы в сценарии можно несколько раз добавлять один и тот же объект. События и действия можно соединять между собой кривой линией. Создадим первое соединение.

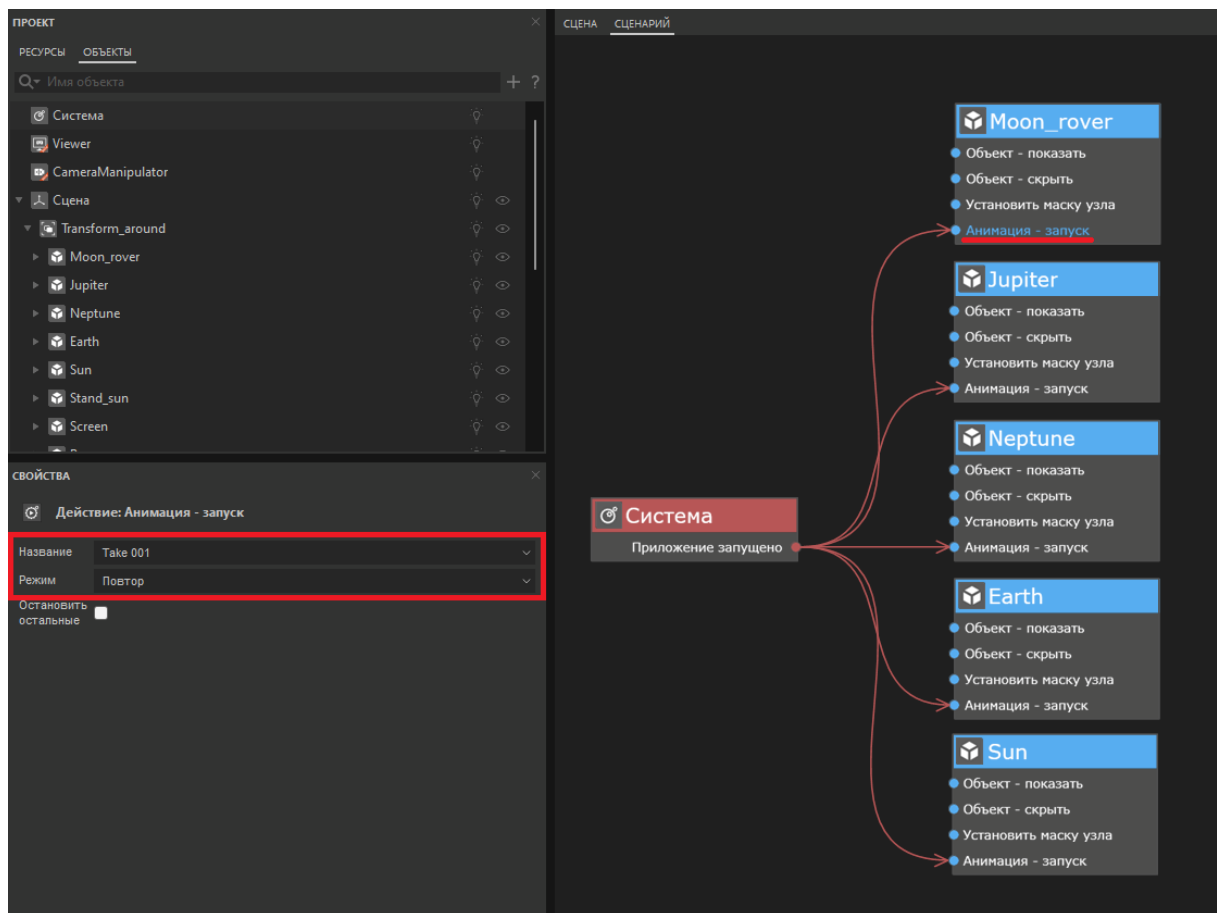
Для объекта Система нам необходимо событие "Приложение запущено". Оно уже добавлено в объект Система по умолчанию.

В объектах Модель нам необходимы действия "Анимация-запуск", они также уже добавлены по умолчанию. Соединим событие "Приложение запущено" у объекта Система и действия "Анимация-запуск" у всех моделей планет, Солнца и лунохода. Дважды кликаем левой кнопкой мыши по точке выхода у события, вытягиваем соединение и тянем ее к нужному действию. Нажимаем левой кнопкой мыши на действие или на точку входа в него – соединение создано. Соединяем таким образом каждую модель.



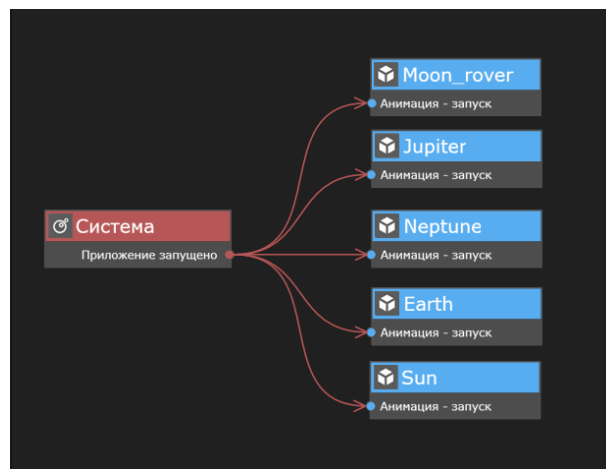
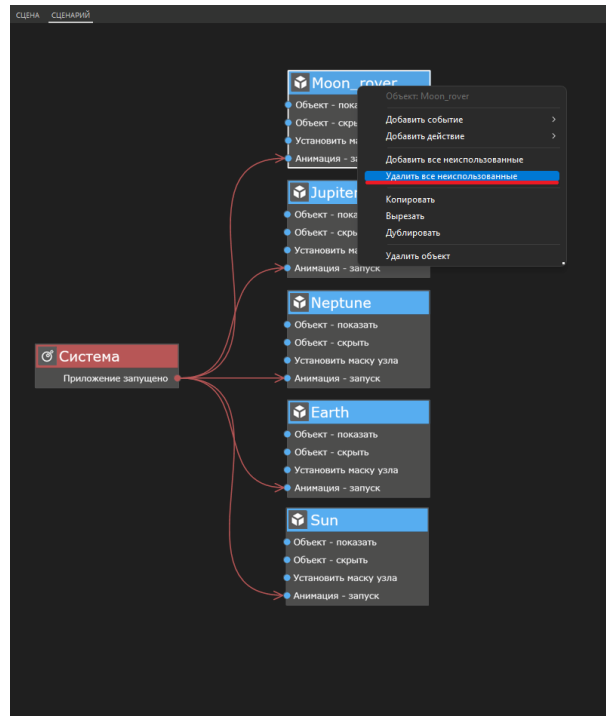
Как и у объектов, у некоторых событий и действий в сценарии проекта есть свои свойства. Для того, чтобы анимация у объектов модель была запущена в свойствах действий "Анимация - запуск" необходимо выбрать какую именно анимацию будет запускать программа.

Для выбора анимации поочередно выделим действие "Анимация – запуск" в каждом блоке модели в панели свойств действия выберем анимации "Take_001" для **каждой модели** в **режиме повтор**.

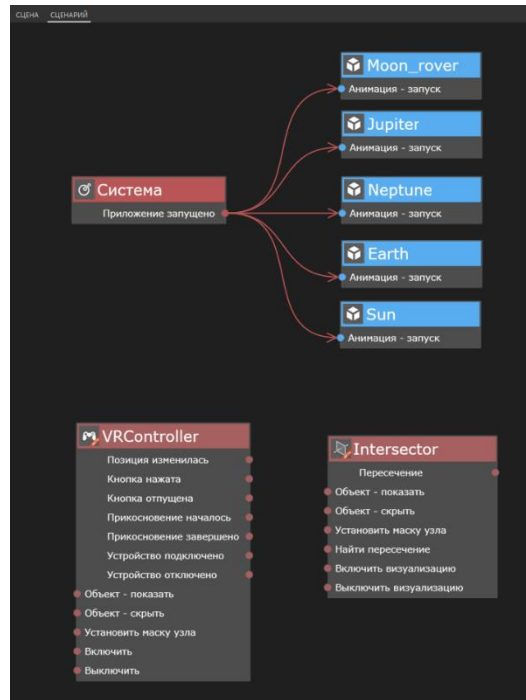


Запустите режим предпросмотра и проверьте запускается ли анимация моделей планет, Солнца и лунохода сразу после запуска приложения. Все должно работать корректно.

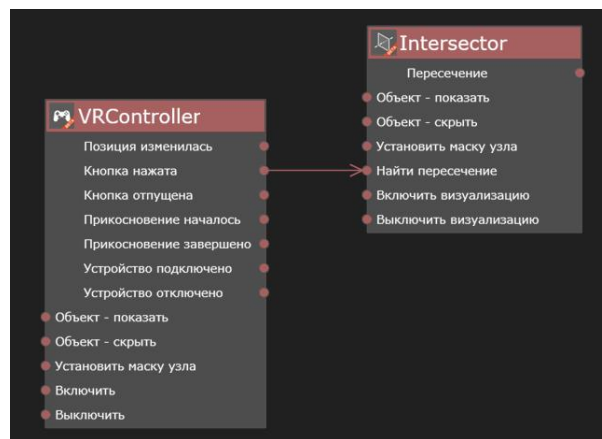
Однако, сейчас наш сценарий выглядит неаккуратным и перегруженным из-за того, что в блоках объектов есть неиспользованные события и действия (события и действия, у которых нет соединений). Удалим такие события и/или действия. Для этого необходимо нажать правой кнопкой мыши по блоку объекта, в котором они есть и в выпадающем меню выбрать "Удалить все неиспользованные".



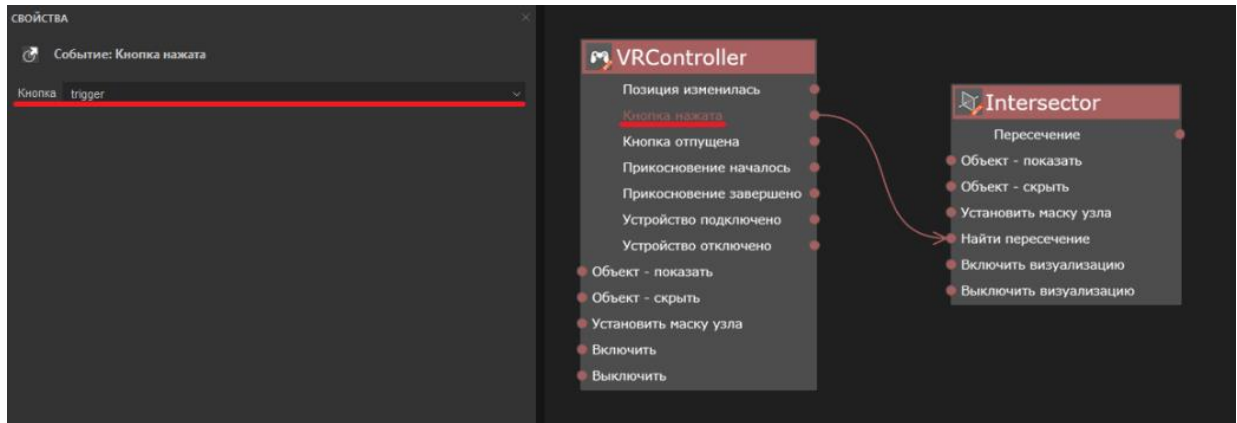
Теперь необходимо создать основу сценария для реализации взаимодействий пользователя с объектами виртуальной сцены. Для этого в рабочую область окна сценария необходимо вынести объекты VRController и Поиск пересечений (Intersector).



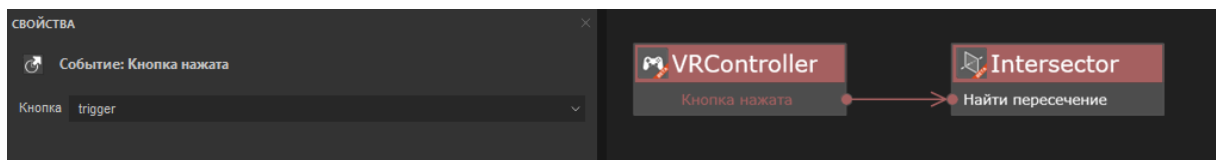
У объекта VRController будем использовать событие "Кнопка нажата", а у объекта Intersector – действие "Найти пересечение". Соединяем их между собой.



Выделим действие "Кнопка нажата" и перейдем в панель свойств этого действия, чтобы выбрать, на какую кнопку на контроллере должен нажимать пользователь, чтобы осуществить взаимодействие. В выпадающем списке выберем кнопку trigger (курок).

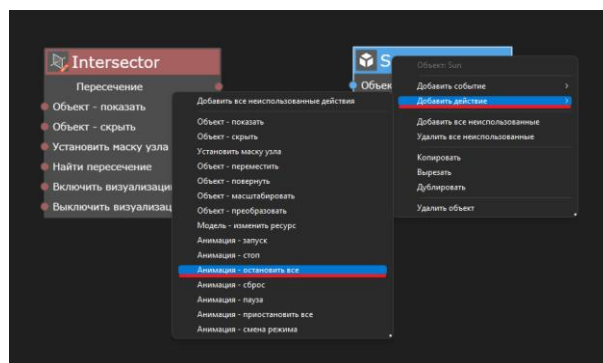


Перед завершением настройки этой части сценария удалим все неиспользованные события и действия. Основа VR проекта готова.

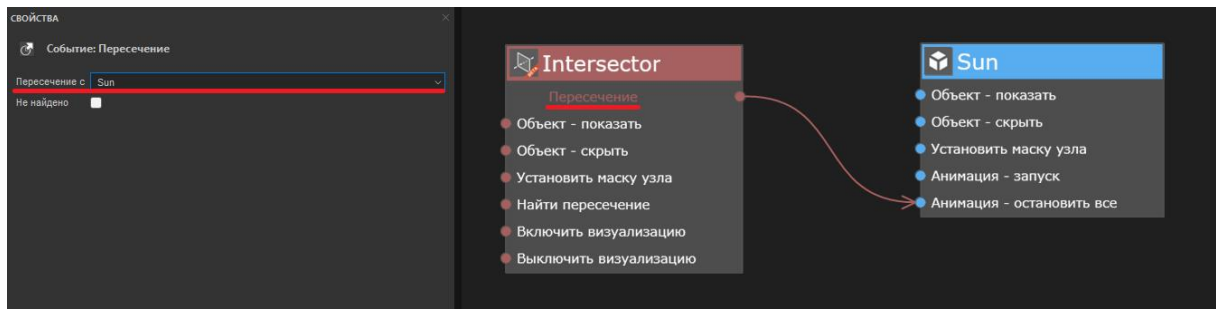


Перейдем к настройке взаимодействия пользователя с той частью объектов виртуального окружения, которая относится к Солнцу: по наведению и нажатию контроллером-волшебной палочкой по модели Солнца анимация модели останавливается, а на экране проектора будет показано и запущено видео о Солнце, а также показана кнопка Close_1, которая позволит остановить проигрывание видео и скрыть его.

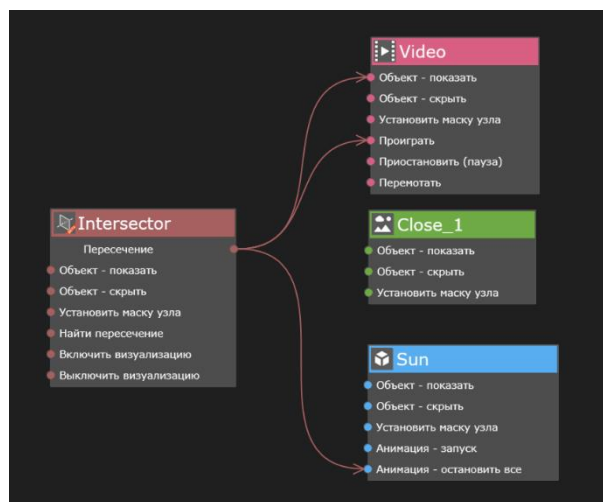
Еще раз отдельно перенесем в рабочую область окна сценария объекты Intersector и модель Солнца. У объекта Intersector нам нужно событие "Пересечение", оно добавлено в список событий по умолчанию. Для модели – действие "Анимация – остановить все". Добавим это действие. Для этого нажимаем правой кнопкой мыши по блоку модели и выбираем "Добавить действие" → "Анимация – остановить все".



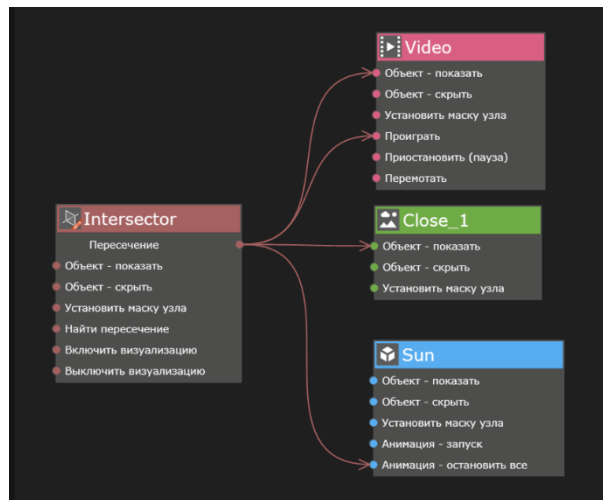
Соединяем между собой событие “Пересечение” у объекта Intersector и действие “Анимация – остановить все” у объекта Sun. Для события “Пересечение” важно выбрать с каким объектом должно быть найдено пересечение для остановки анимации модели Солнца – это сама модель Солнца. Выделим событие “Пересечение” и в панели свойств выберем необходимый объект.



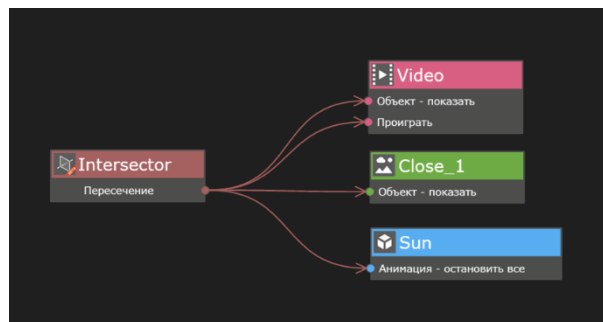
Перенесем в рабочую область окна сценария объекты Video и Close_1. У объекта Video нам нужно два действия: “Объект-показать” и “Проиграть”. Соединяем между собой событие “Пересечение” у объекта Intersector и действия “Объект-показать” и “Проиграть” у объекта Видео.



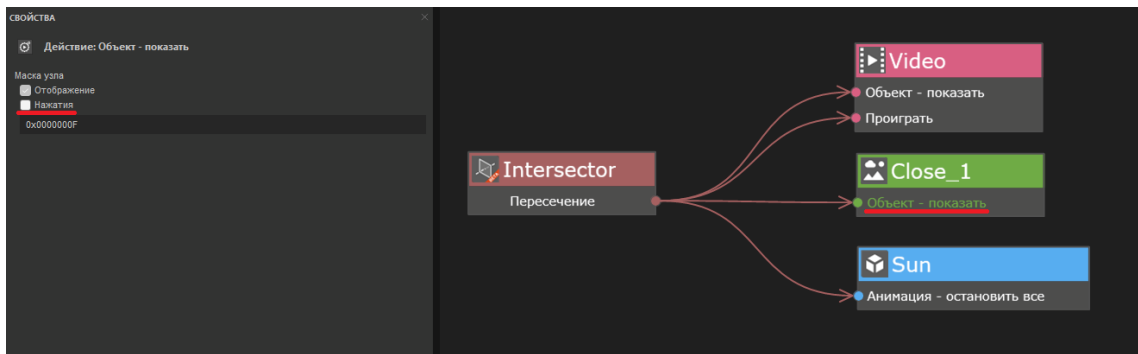
У объекта Close_1 нам нужно действие “Объект-показать”. Соединяем между собой событие “Пересечение” у объекта Intersector и действия “Объект-показать” у объекта Close_1.



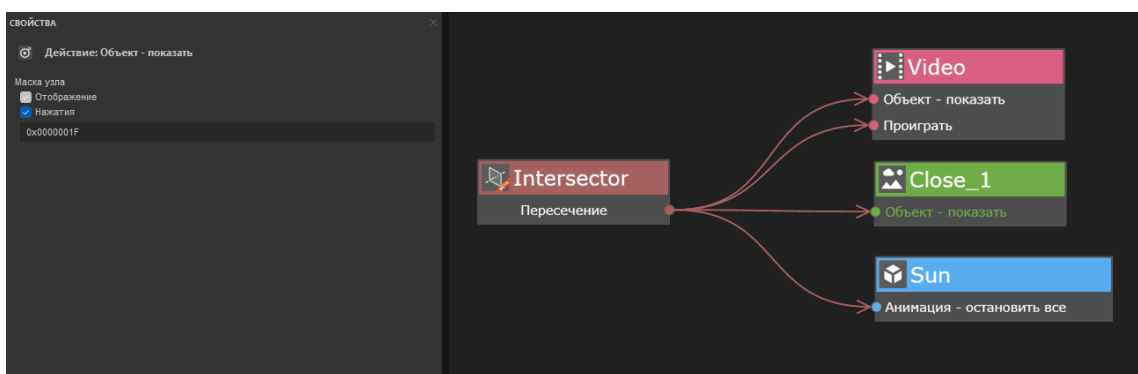
Не забываем удалять неиспользованные события и действия в сценарии проекта.



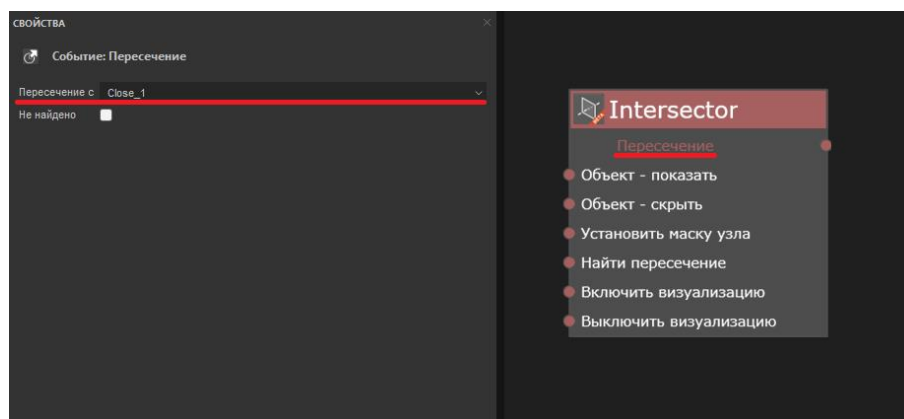
ВАЖНО: обратите, пожалуйста, внимание, что при работе в версиях программы 3.4.x для действий “Объект – показать” и “Объект – скрыть” появились новые свойства, благодаря которым можно отдельно управлять видимостью и обработкой нажатий для объектов, имеющих визуальное представление в сцене, таких как текст, изображение, видео и др. Это удобно, когда вам необходимо ограничить нажатия пользователей на некоторые видимые объекты или наоборот дать возможность пользователям взаимодействовать со скрытыми объектами. Однако наличие таких свойств требует более внимательной настройки сценария при создании проекта. Дело в том, что по умолчанию, при использовании действия “Объект – показать” для объекта будет автоматически включена его видимость, однако по умолчанию нажатия на объект не включены.



Если в свойствах события “Объект – показать” не установить флаг в свойстве Нажатия, то после того, как объект будет показан, нажатия на него обрабатываться не будут, соответственно, мы не сможем потом нажать на кнопку Close_1 для остановки и скрытия видео. Поэтому чтобы проект и дальше корректно работал, и мы могли нажимать на кнопку, необходимо установить флаг в параметре Нажатие для действия “Объект – показать”.

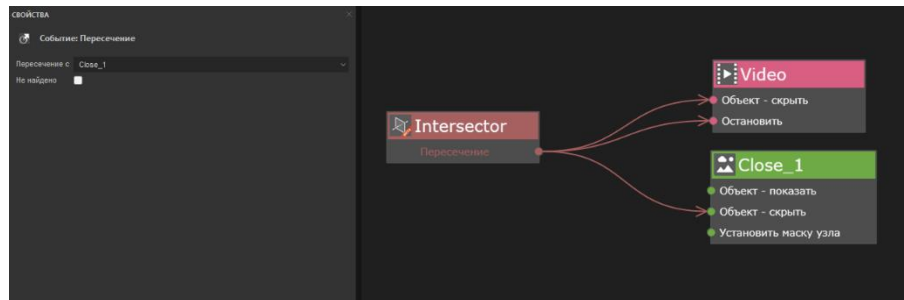


Настроим логику работы для кнопки Close_1. Еще раз вынесем объект Intersector в сценарий. Снова определяем объект, с которым будет осуществляться Поиск пересечений. В данном случае это объект Close_1. Найдем его в выпадающем списке параметра “Пересечение с” действия “Пересечение”.



По нажатию на эту кнопку видео должно быть остановлено и скрыто. Вместе с видео также должна быть скрыта и сама кнопка. При этом анимация у модели Солнца снова запускаться не будет.

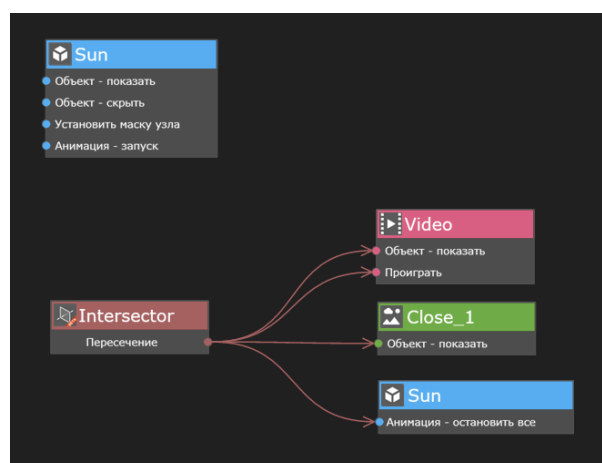
Еще раз перенесем в область сценария объекты Video и Close_1. У объекта Video нам нужны действия “Остановить” и “Объект - скрыть”. Добавляем действия “Остановить” и соединяем между собой событие “Пересечение” у Intersector с действиями “Объект-скрыть” и “Остановить” у объекта Video.



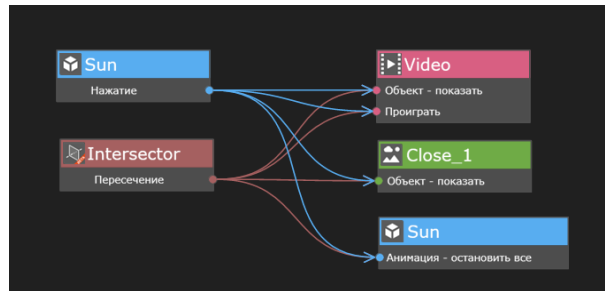
Запустим режим предпросмотра, чтобы проверить, все ли корректно работает. Первая часть сценария с анимацией моделей отлично работает, а вот нажатия не осуществляются.

Дело в том, что мы сейчас настроили логику работы приложения для автономного беспроводного шлема виртуальной реальности, который для работы использует свою собственную аппаратную часть, а его работа не зависит от ПК. В связи с этим имитировать нажатия кнопок контроллера для таких шлемов в предпросмотре возможности нет.

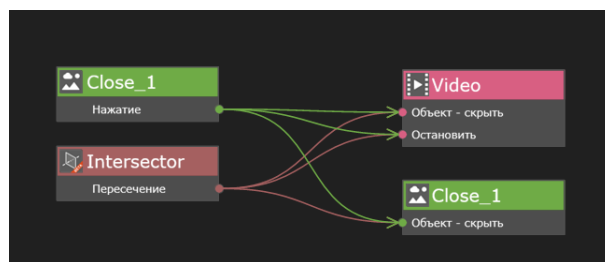
Чтобы проверить созданную логику работы на ПК при помощи компьютерной мышки, нам необходимо продублировать те же самые объекты, только параллельно с объектом Intersector добавить событие “Нажатие” на определенный объект внутри виртуального окружения. Первый объект, с которым пользователь осуществляет взаимодействие – это модель Солнца. Перенесем эту модель в рабочую область окна сценария еще раз.



У модели Солнца нам нужно событие "Нажатие". Добавим это событие. Логика остается той же, поэтому соединяем между собой событие "Нажатие" у модели Sun и действия "Анимация – остановить все" у Модели Sun, "Объект-показать" и "Проиграть" у объекта Видео и "Объект-показать" у объекта Close_1.



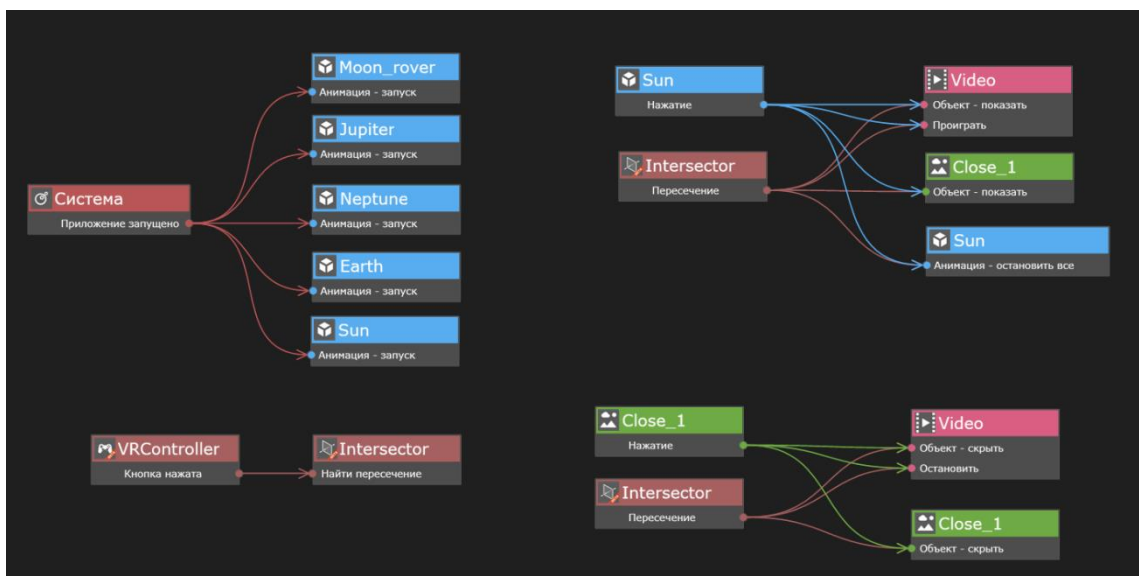
Для пересечения с кнопкой Close_1 дополним сценарий по аналогии.



Не забывайте удалять неиспользованные события и действия.

Еще раз запустим режим предпросмотра и проверим работу проекта, все должно работать отлично.

Не забудьте корректно сохранить ваш проект без потери ресурсов.



Этап 7. Работаем с объектом Аудио в VR проекте. Завершаем работу над VR проектом.

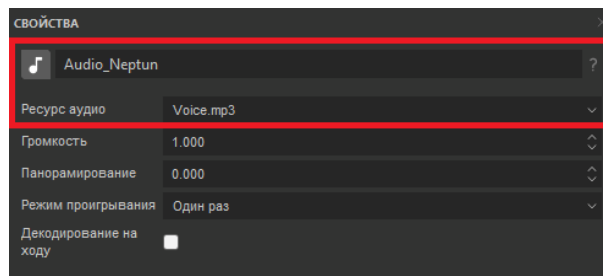
(время работы: 20 минут)

На этом этапе нам предстоит завершить разработку проекта с использованием технологии виртуальной реальности “VR МУЗЕЙ: АСТРОНОМИЯ”.

Создадим в проекте новую часть сценария в соответствии с которой после нажатия кнопки контроллера на модель Нептуна анимация модели останавливается, а на экране проектора появляется изображение этой планеты и проигрывается небольшая аудио справка о ней. После прослушивания справочной информации (т.е. завершения аудиодорожки) Изображение будет автоматически скрыто. При этом анимация у модели Нептуна не запускается.

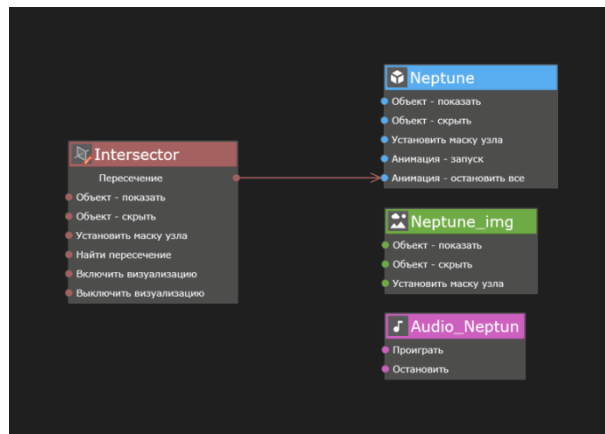
Для реализации такой идеи необходимо добавить еще один новый объект в проект – это объект **Аудио**. Стоит отметить, что в дереве объектов этот объект был добавлен на один уровень со Сценой. Это говорит о том, что данный объект не имеет визуального отображения в сцене.

В окне переименуем объект и назовем его “Audio_Neptun”. Определим для него ресурс. В папке с ресурсами проекта находим папку “Audio” и выбираем в ней дорожку с названием Voice. Остальные свойства оставляем без изменений.

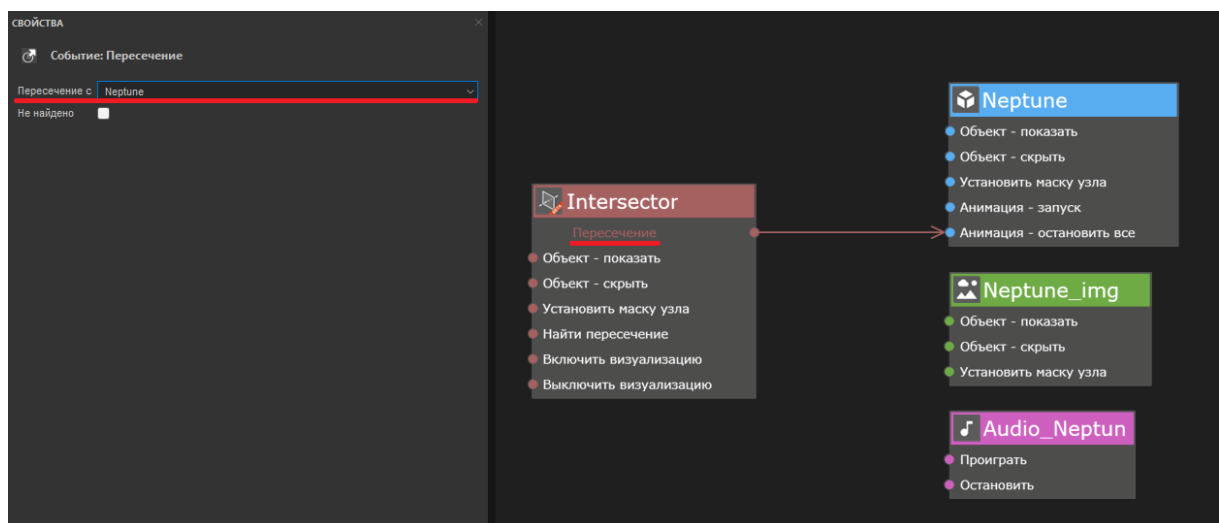


Возвращаемся к созданию логики работы проекта. Перенесем в рабочую область окна сценария новый объект Audio_Neptun, а также еще один блок объекта Intersector, модель Neptun и изображение Neptun_img.

У объекта Intersector нам нужно событие “Пересечение”, оно добавлено в список событий по умолчанию. У модели Neptun – действие “Анимация – остановить все” отсутствует. Добавим это действие и соединим между собой событие “Пересечение” у объекта Intersector и действие “Анимация – остановить все” у объекта Neptune.



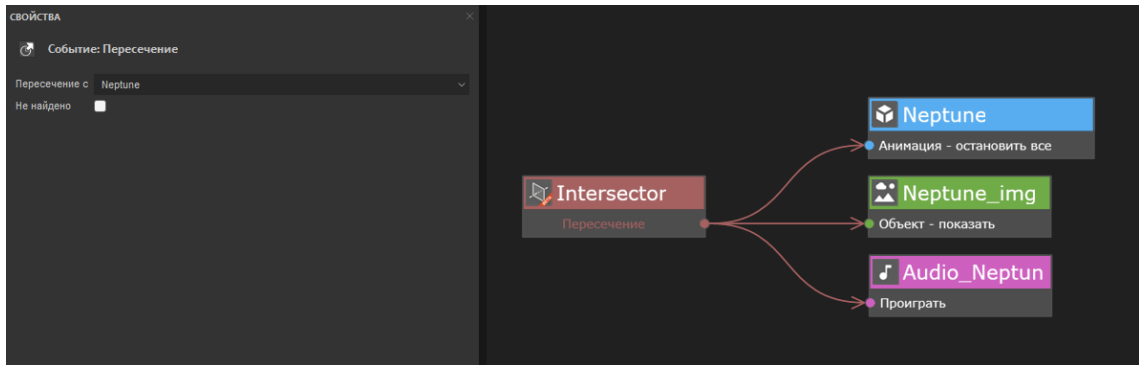
Для события “Пересечение” важно выбрать с каким объектом должно быть найдено пересечение для остановки анимации модели Нептуна – это сама модель Нептуна. Выделим событие “Пересечение” и в панели свойств выберем необходимый объект.



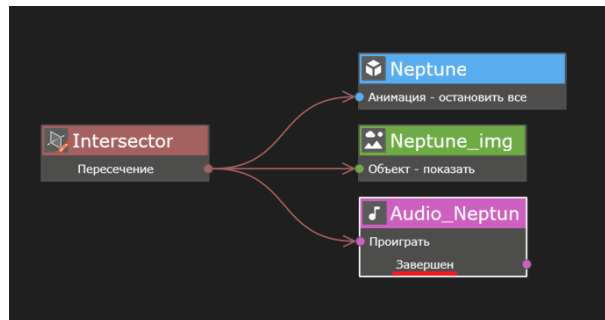
У объекта Neptune_img нам нужно действие “Объект-показать”. Соединяем между собой событие “Пересечение” у объекта Intersector и действие “Объект-показать” у объекта Изображение.

У объекта Audio_Neptun нам нужно действие “Проиграть”. Соединяем между собой событие “Пересечение” у объекта Intersector и действие “Проиграть” у объекта Audio_Neptun.

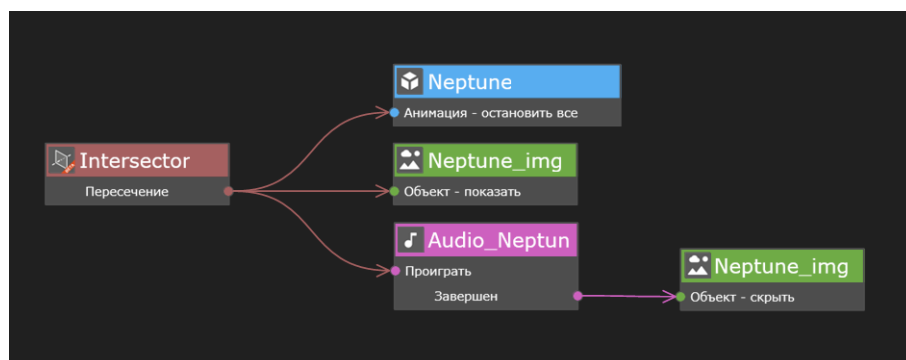
Не забудем удалить неиспользуемые события и действия, где это необходимо.



Далее необходимо скрыть Изображение на экране проектора после завершения проигрывания аудиодорожки. Для этого нам понадобится событие “Завершен” у объекта Аудио. Добавляем его.

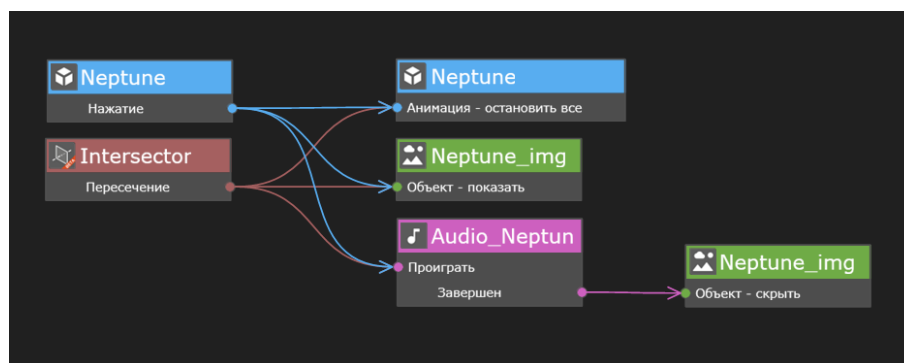


Также еще раз вынесем Изображение Neptune_img в область сценария. У этого объекта нам нужно действие “Объект-скрыть”. Оно добавлено по умолчанию. Соединяем между собой событие “Завершен” у объекта Аудио и действие “Объект-скрыть” у объекта Изображение.



Вспомним о том, что мы только что создали логику работы приложения для работы в автономных шлемах виртуальной реальности. Сейчас нужно будет продублировать ту же логику работы, только для проверки работы проекта в режиме Предпросмотра. Для этого нам необходимо еще раз вытащить в пространство сценария блок модели Нептуна. Нам нужно событие "Нажатие", находим и добавляем его.

Логика остается той же, поэтому соединяем между собой событие "Нажатие" у модели Neptune и действия "Анимация-стоп" у Модели, "Проиграть" у объекта Audio и "Объект-показать" у объекта Neptune_img.

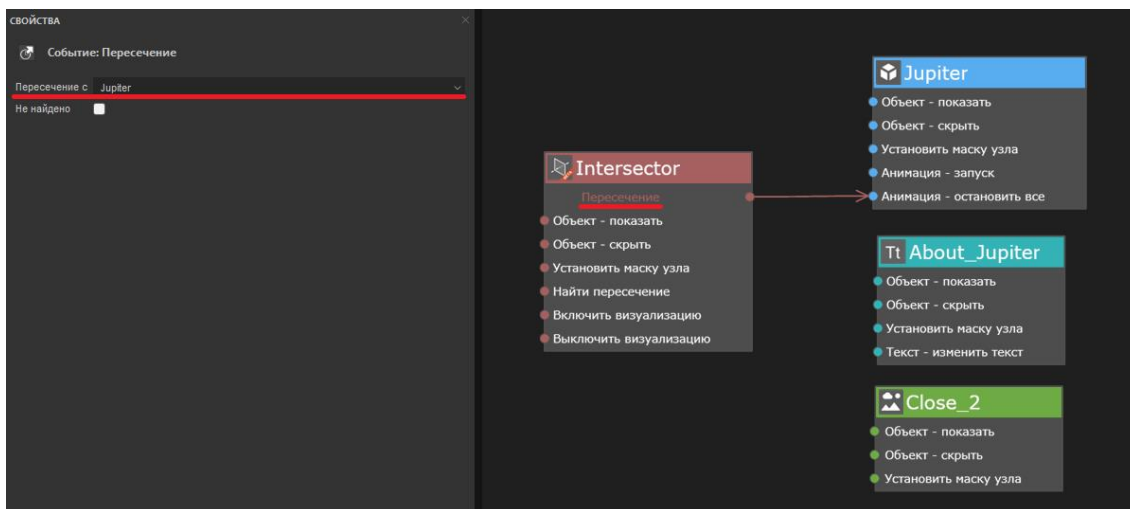


Проверьте корректность работы созданной части сценария в режиме Предпросмотра.

Добавим еще одну часть сценария по работе с моделью Юпитера: после нажатия кнопки контроллером на модель Юпитера анимация модели останавливается, на экране проектора появляется небольшая информационная справка об этой планете и кнопка для того, чтобы эту справку можно было закрыть. После нажатия на эту кнопку и справка, и сама кнопка будут скрыты.

Для реализации такой логики переносим в пространство сценария текстовый блок About_Jupiter, а также еще один блок объекта Intersector, модель Jupiter и изображение Close_2. У объекта Intersector нам нужно событие "Пересечение", оно добавлено в список событий по умолчанию. У модели Jupiter – действие "Анимация – остановить все", добавим его. Соединяем между собой событие "Пересечение" у объекта Intersector и действие "Анимация – остановить все" у модели Jupiter.

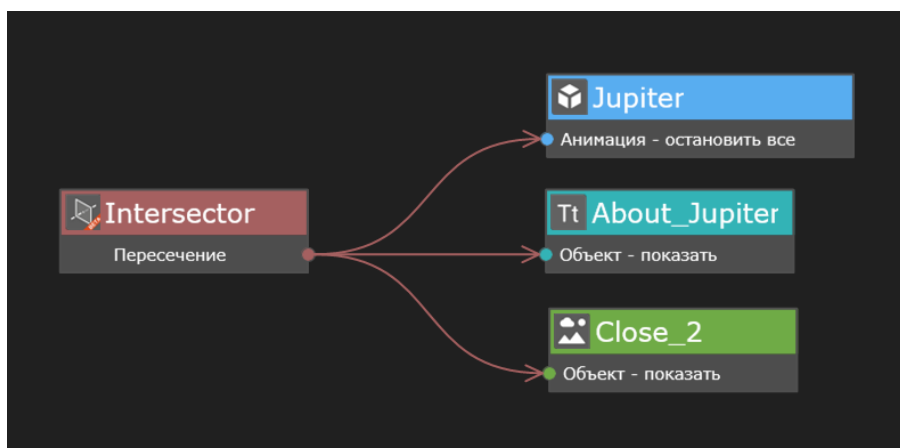
Для события "Пересечение" важно выбрать с каким объектом должно быть найдено пересечение для остановки анимации модели Юпитера – это сама модель Юпитера. Выделим событие "Пересечение" и в панели свойств выберем необходимый объект.



Для объекта Close_2 нам нужно действие “Объект-показать”. Соединяем между собой событие “Пересечение” у объекта Intersector и действие “Объект-показать” у объекта Изображение Close_2.

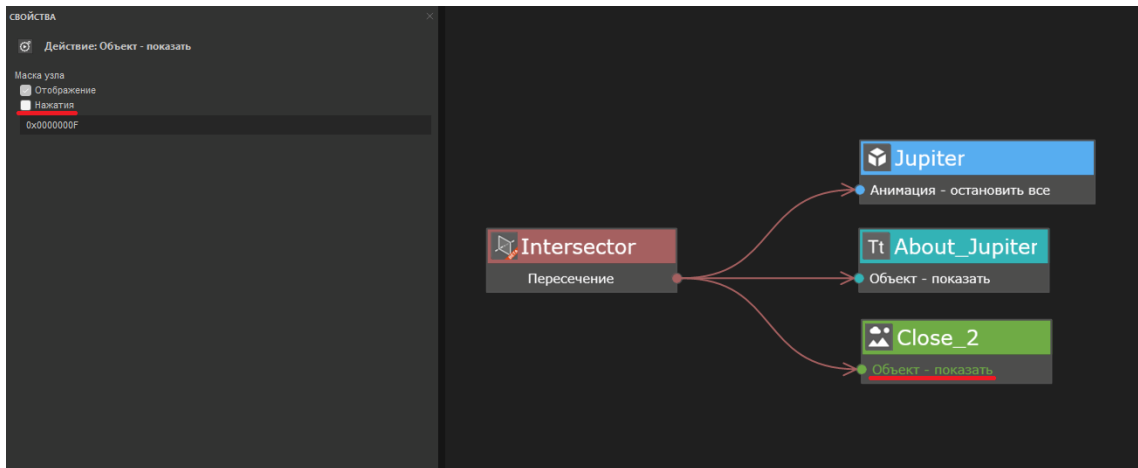
У текстового блока About нам также необходимо действие “Объект-показать”. Соединяем также событие “Пересечение” у объекта Intersector и действие “Объект-показать” у объекта Текст.

Не забудем удалить неиспользуемые события и действия для объектов.

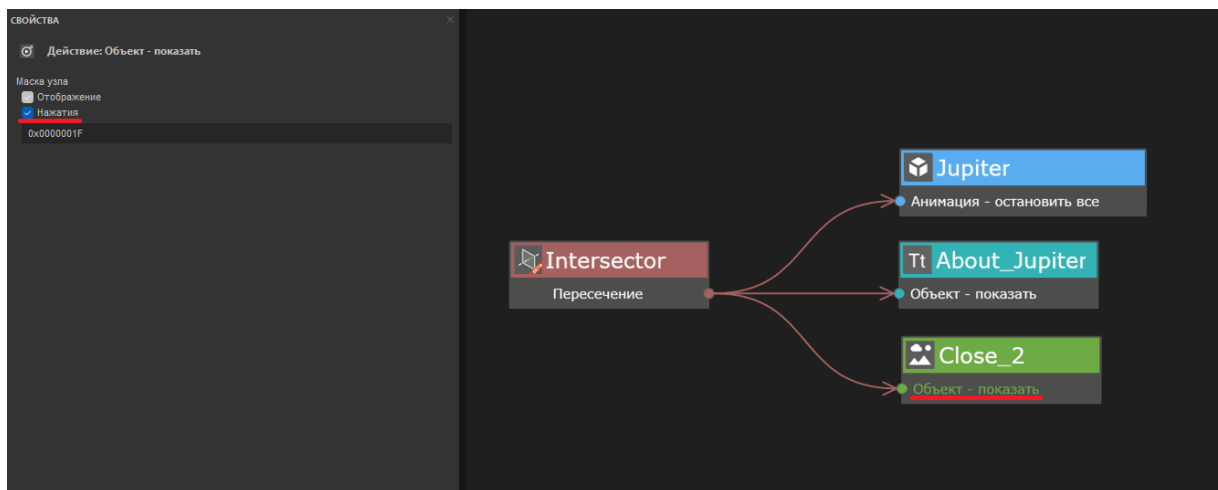


ВАЖНО: обратите, пожалуйста, внимание, что при работе в версиях программы 3.4.x для действий “Объект – показать” и “Объект – скрыть” появились новые свойства, благодаря которым можно отдельно управлять видимостью и обработкой нажатий для объектов, имеющих визуальное представление в сцене, таких как текст, изображение, видео и др. Это удобно, когда вам необходимо ограничить нажатия пользователей на некоторые видимые объекты или наоборот дать возможность пользователям взаимодействовать со скрытыми объектами. Однако наличие таких свойств требует более внимательной настройки сценария

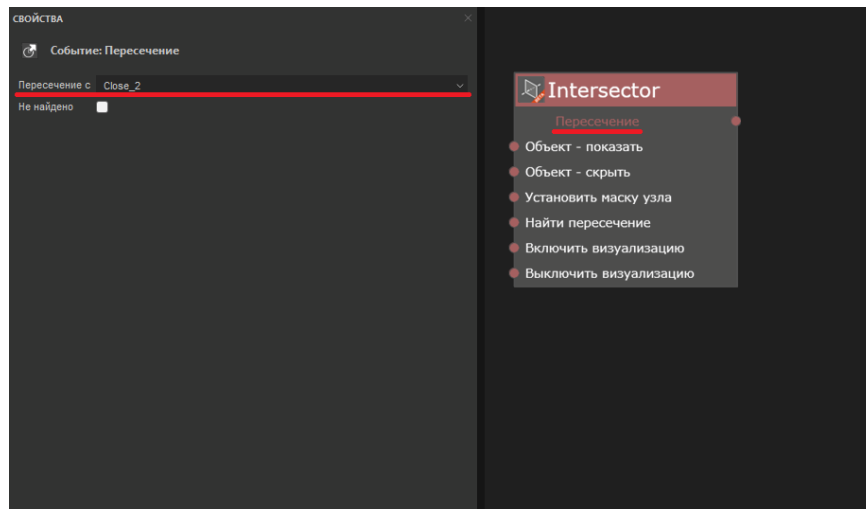
при создании проекта. Дело в том, что по умолчанию, при использовании действия “Объект – показать” для объекта будет автоматически включена его видимость, однако по умолчанию нажатия на объект не включены.



Если в свойствах события “Объект – показать” не установить флаг в свойстве “Нажатия”, то после того, как объект будет показан, нажатия на него обрабатываться не будут, соответственно, мы не сможем потом нажать на кнопку Close_1 для остановки и скрытия видео. Поэтому чтобы проект и дальше корректно работал, и мы могли нажимать на кнопку, необходимо установить флаг в параметре Нажатие для действия “Объект – показать”.



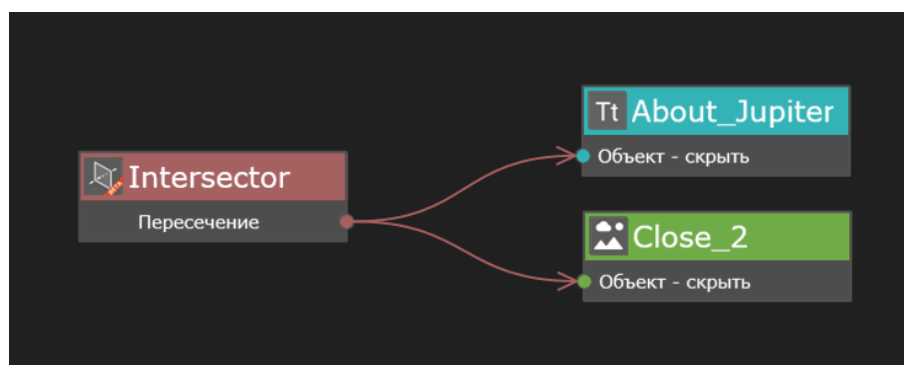
Теперь настроим логику работы для кнопки закрытия текстового блока Close_2. Еще раз вынесем объект Intersector в сценарий. В свойствах события “Пересечение” снова определяем объект, с которым будет осуществляться поиск пересечений. В данном случае это объект Close_2.



По нажатию на кнопку Close_2 информация о Юпитере должна быть скрыта. Вместе с текстом с экрана проектора также должна быть скрыта и кнопка Close_2. При этом анимация у модели Юпитера снова запускаться не будет. Перенесем в область сценария два объекта: About и Close_2. У объекта Текст нам нужно действие "Объект-скрыть". Соединяем между собой событие "Пересечение" у Intersector с действием "Объект-скрыть" у объекта About.

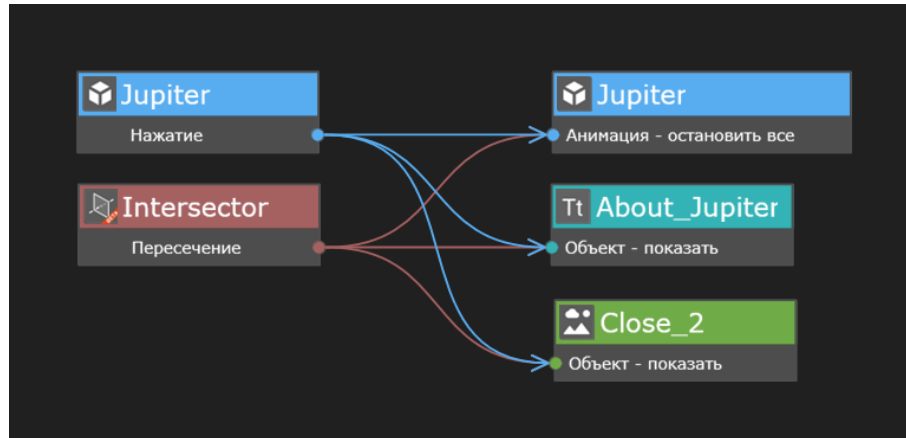
У объекта Close_2 нам нужно действие "Объект-скрыть". Соединяем между собой событие "Пересечение" у объекта Intersector и действия "Объект-скрыть" у объекта Close_2.

Не забудем удалить неиспользованные события и действия.

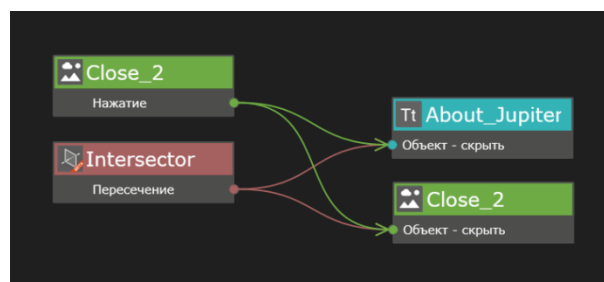


Чтобы проверить логику работы на ПК в режиме предпросмотра, нам необходимо продублировать те же самые объекты, только параллельно с объектом Intersector добавить событие "Нажатие" для модели Юпитера. Перенесем эту модель в область сценария еще раз.

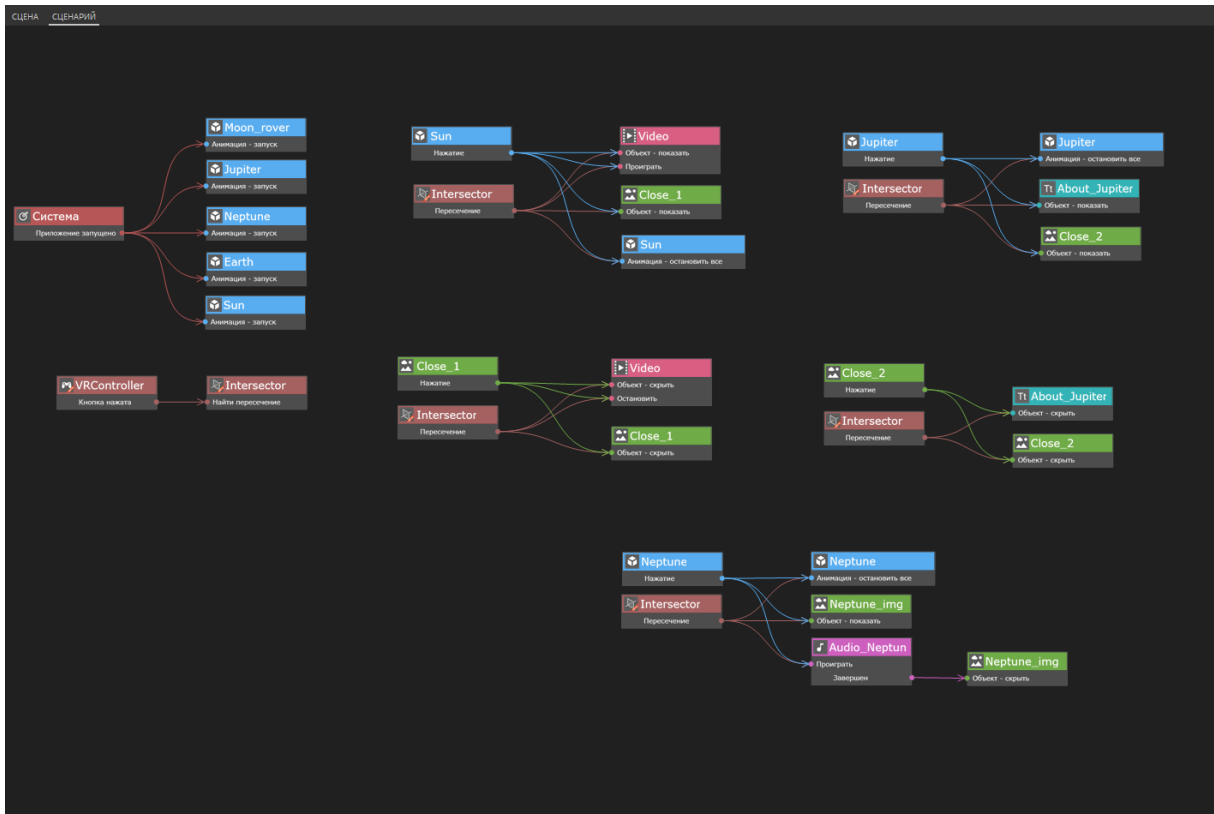
Добавим для него событие "Нажатие". Логика остается той же, поэтому соединяем между собой событие "Нажатие" у модели Jupiter и действия "Анимация-стоп" у Модели, "Объект-показать" у объекта Текст и "Объект-показать" у объекта Close_2.



Для объекта Close_2 можно дополнить логику работы по аналогии. Переносим блок объекта Close_2 в сценарий проекта. Добавляем событие "Нажатие". Соединяем между собой событие "Нажатие" у объекта Close_2 и действия "Объект-скрыть" у объекта Текст и действия "Объект-скрыть" у объекта Изображение Close_2.

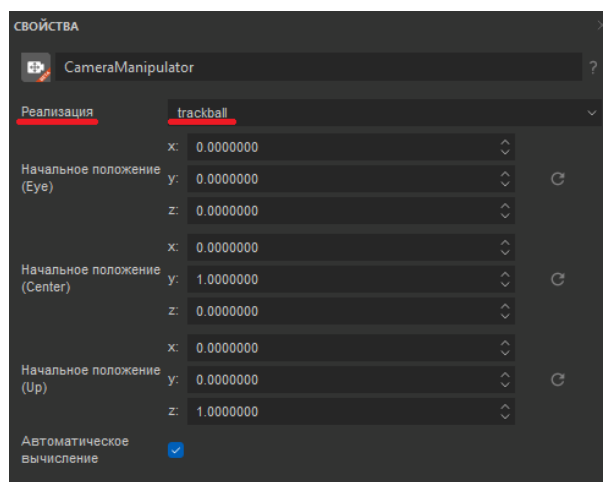


Не забудем удалить неиспользованные события и действия.



Процесс разработки нашего VR проекта подошел к концу. Напомним, что если вы работаете над проектом в триальной (пробной) версии конструктора, то вы можете протестировать работу вашего проекта только в режиме Предпросмотра, так как экспорт (сборка) проектов доступны только в лицензионных версиях конструктора EV Toolbox.

Перед тестированием проекта в режиме предпросмотра проверьте выбранный тип реализации манипулятора камеры в свойствах объекта **CameraManipulator**. Для тестирования проекта в режиме предпросмотра должен быть выбран тип манипулятора **trackball**.



Реализация VR предназначена для работы приложения уже непосредственно в шлеме виртуальной реальности, поэтому если установить такой тип манипулятора, проверить работу проекта в режиме Предпросмотра будет невозможно.

Проверьте работу вашего проекта в режиме Предпросмотра – все должно работать корректно. Обратите внимание на то, что при параллельном нажатии на различные объекты не все из них работают корректно, например, можно поочередно нажать на модели планеты Юпитер и Солнца, и тогда дополнительный контент о них на экране проектора будет накладываться друг на друга.

При желании вы можете доработать и усложнить сценарий своего проекта таким образом, чтобы при параллельном нажатии обрабатывалось только первое из них, а нажатия на другие объекты не приводили к сбою работы вашего приложения. Также по желанию вы можете добавить дополнительный контент для планеты Земля и доработать сценарий проекта.

Успешной вам работы над проектами!