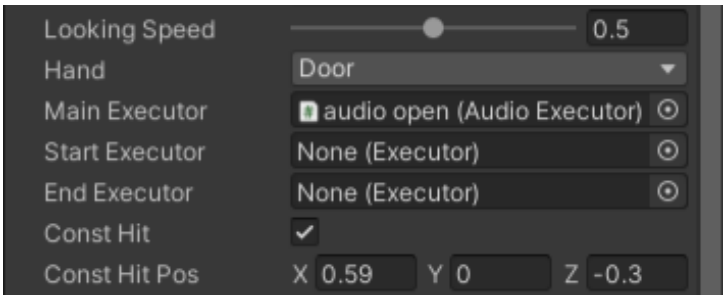
Содержание

1. Общее описание

Ассет предлагает решение по физическому взаимодействию с предметами из окружения сцены в Unity.

Особенностью является готовность работы «из коробки», не нужно создавать новый слой объектов, тэг или параметр в Input System, все работает сразу! Обязательно читайте документацию и вам не составит труда сделать собственные префабы с применением ассета. Все поля скриптов имеют описание и подробно изложены в настоящем документе.



Общие параметры

Looking Speed – коэффициент вращения камеры (Player Looking) при взаимодействии с предметом. При значении 0 камера будет неподвижна при перемещении мышки. Значение 1 оставит вращение камеры неизменным.

Hand – параметр, определяющий какой именно спрайт курсора будет отображаться в месте соприкосновения с предметом: Box, Button, Door, Grab. Смотри раздел Курсор для получения большей информации.

Main, **Start** и **End Executor** – исполнители-посредники, которые воспринимают на себя сигнал от предметов взаимодействия: рычаг, слайдер, кнопка и т.д. – для дальнейшего воздействия на объекты сцены. Соответственно Main воспринимает сигнал в порядке, определяемом индивидуально в дочернем классе от класса Interactable Object, Start при начале взаимодействия, End Executor при завершении взаимодействия.

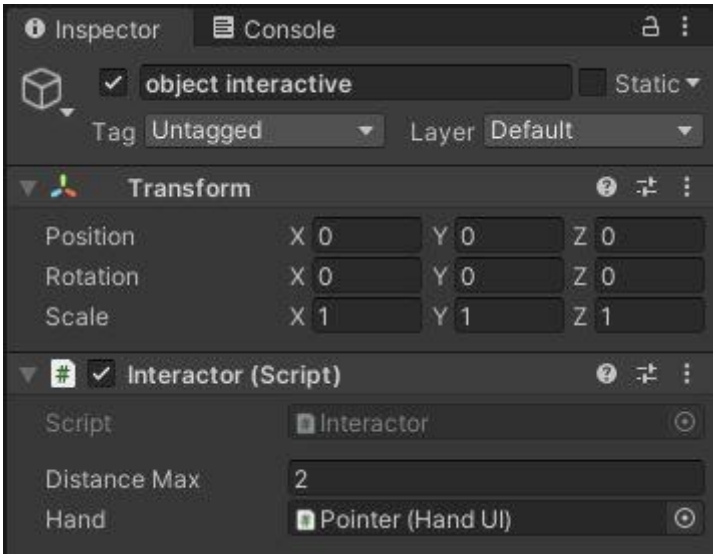
Const Hit – параметр, предопределяющий место взаимодействия с предметом. Если true, то используется предопределенное значение Const Hit Pos (смотри Gizmos для визуального отображения). Если false, то взаимодействие с предметом будет в фактической точке соприкосновения.

Важно!

Включи отображение Gizmos для гибкой настройки и полного понимания происходящего на сцене. С помощью Gizmos отображаются все связи с привязанными исполнителями, а также ключевые состояния предметов взаимодействия в индивидуальном порядке.

2. Interactor

Основной элемент управления интерактвных предметов. Как правило устанавливается в составе префаба персонажа. Подтягивает в Start() основную камеру.



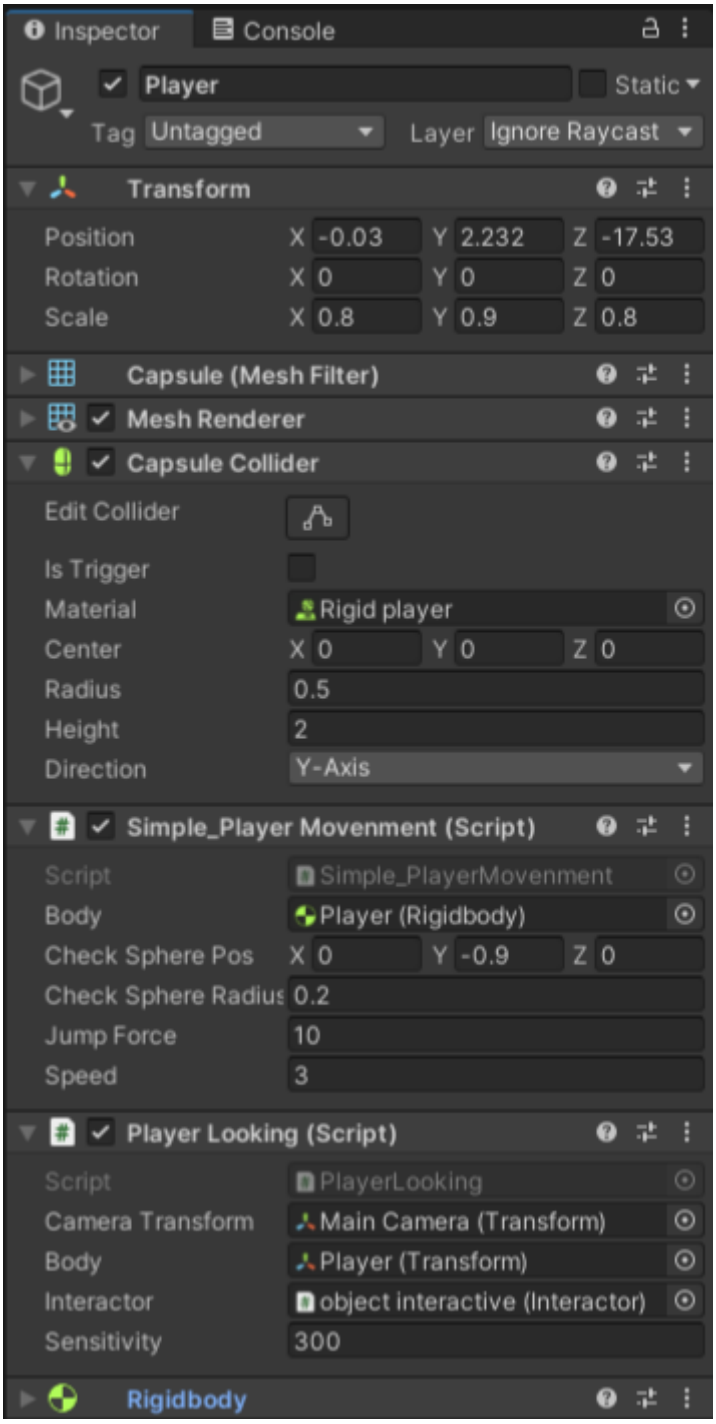
Параметры

Max Distance – расстояние взаимодействия с предметами класса Interactable Object.

Hand – курсор, который меняется в зависимости от настройки параметра Hand класса Interactable Object.

3. Player Looking

На ряду с компонентом Simple Player Movenment позволяет управлять персонажем. Скорость вращения камеры по оси X и корневого объекта персонажа по оси Y зависит от настройки Looking Speed предметов взаимодействия класса Interactable Object.



Параметры

Camera Transform – ссылка на камеру (основную камеру), которая будет подвергаться вращению по оси X.

Body – ссылка на родительский объект персонажа, который будет вращаться по оси Y.

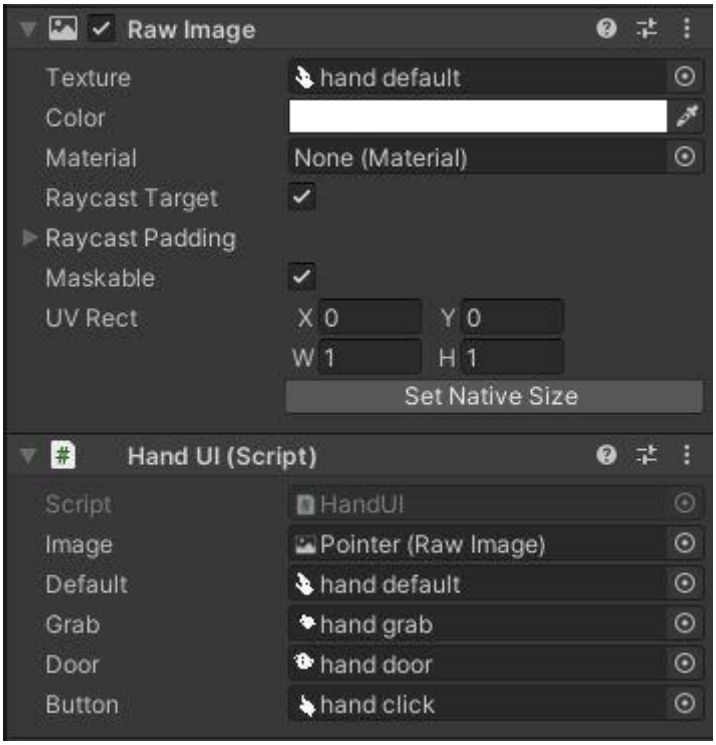
Interactor – ссылка на компонент Interactor.

Sensitivity – чувствительность управления мышкой.

4. Hand UI

Hand UI имеет несколько текстур отображения в зависимости от значения параметра Cursor Mode предмета взаимодействия. Обязательный компонент React Transform нужен для перемещения

курсора по двум осям экрана в зависимости от ракурса камеры относительно предмета взаимодействия. Компонент курсора Image включается при наличии предмета взаимодействия и наоборот.



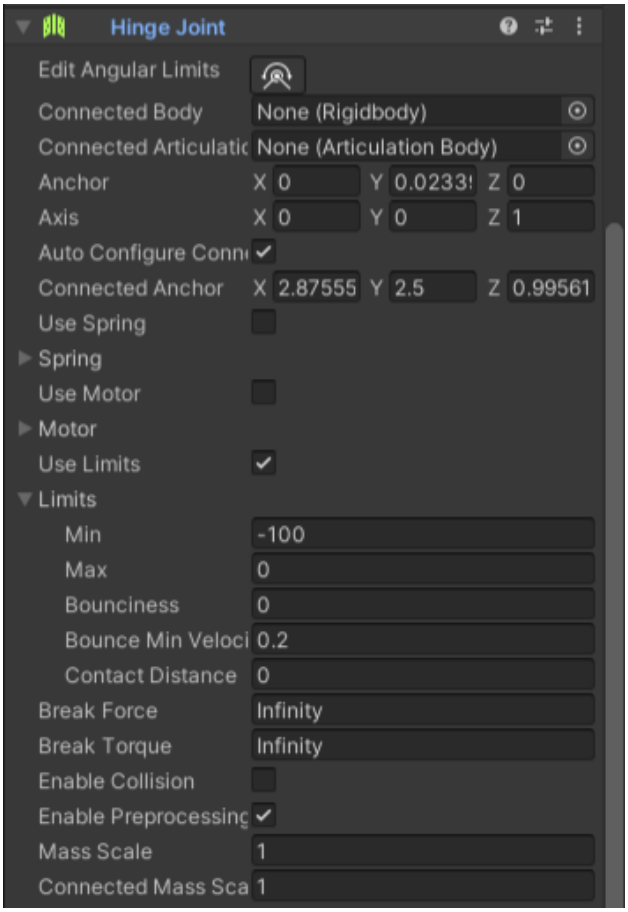
Параметры

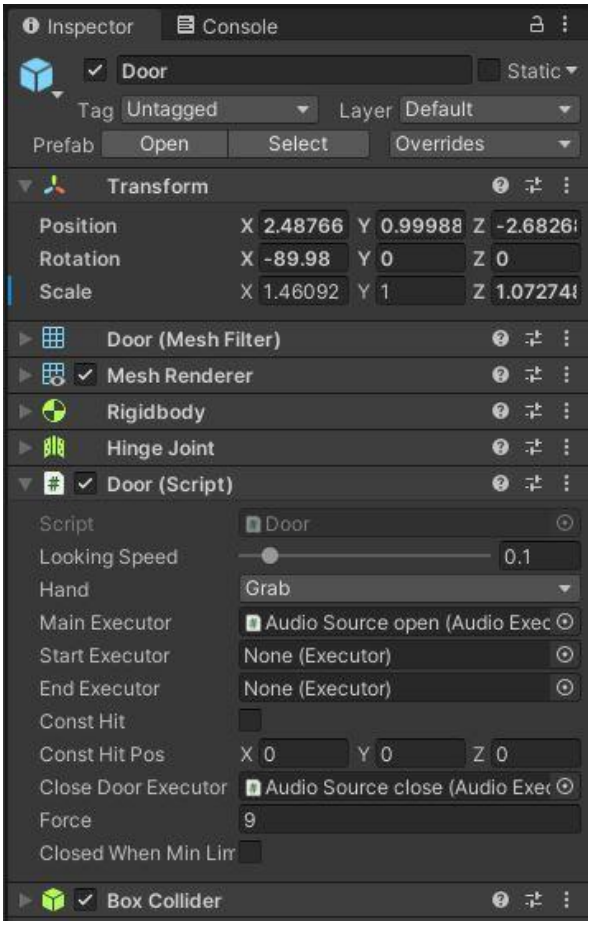
Image – экземпляр класса Raw Image, который будет подвергаться изменению в зависимости от объекта взаимодействия. Выключается при отсутствии предмета взаимодействия. Меняет свою текстуру в зависимости от настроек предмета взаимодействия.

Default, **Grab**, **Door**, **Button** – текстуры, которые будут применяться к Raw Image в зависимости от настроек предмета взаимодействия.

5. Door

Скрипт для работы физических предметов в качестве двери. Она может быть открыта или закрыта с помощью мышки, потянув вперед или назад, или с помощью столкновения. Дверь также может быть захлопнута (не может быть открыта, если не взаимодействовать мышкой – нажать на ручку двери).





Параметры

Close Door Executor – исполнитель, который срабатывает при захлопывании дверцы.

Force – физическая сила, с которой игрок будет воздействовать на дверь. Рекомендуемое значение - 9. Если дверь тяжелая или это люк с горизонтальным расположением, то значение стоит увеличить, например, до 30.

Closed When Min Limits – угол поворота двери, при котором дверь будет считаться захлопнутой. Подтягивает значения из параметров Limits компонента Hinge Joint. Если true, то параметр Min Angle будет восприниматься как захлопнутое состояние двери.

Важно!

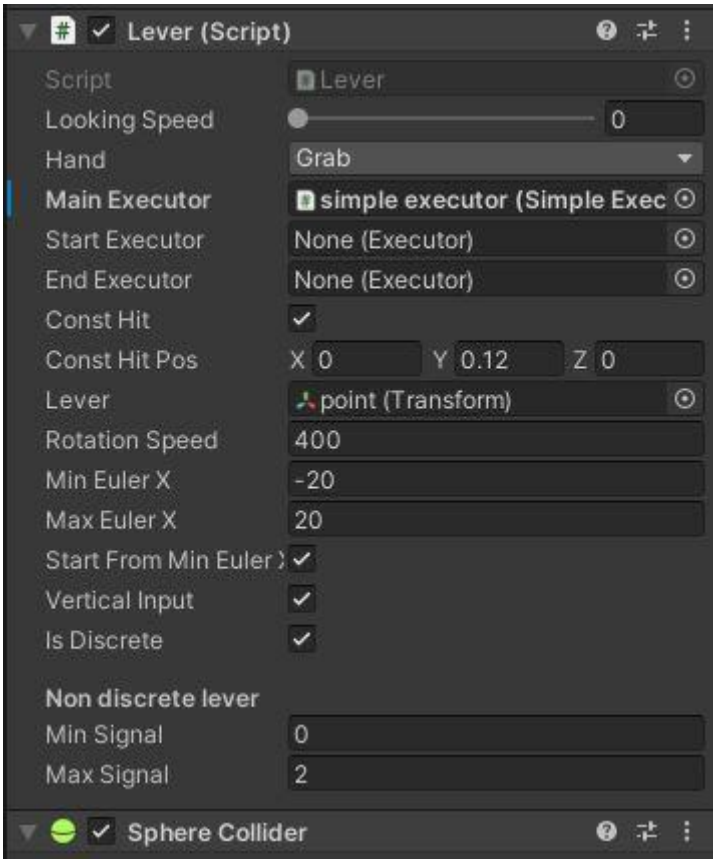
Для работы параметра Closed Is Min Limits необходимо отметить Use Limits в Hinge Joint.

6. Lever

Рычаг вращается относительно своей локальной оси right. В зависимости от настройки может принимать 2 положения, не зависая в среднем состоянии, либо занимать любое из состояний в пределах ограничений.

Важно!

Для вращения относительно необходимой точки используй пустой объект с визуальной частью внутри или учитывай положение pivot point при моделировании своего рычага.



Параметры

Lever – визуальная часть рычага (или пустой объект с визуальной частью внутри), которая будет вращаться в зависимости от перемещения мышки.

Rotation Speed – скорость вращения рычага в зависимости от перемещения мышки.

Min Euler X и **Max Euler X** – углы-ограничители вращения рычага. Рекомендуемые значения -20 и 20.

Start From Min Euler – исходное положение рычага – минимальное выдаваемое значение.

Vertical Input – если true, то для изменения положения рычага будет использовано вертикальная ось перемещения мышки.

Is Discrete - рычаг опускается и поднимается в одно из двух состояний, не зависая в среднем положении.

Min и **Max Signal** – значения выходного сигнала, которые может выдавать рычаг при недискретном функционировании.

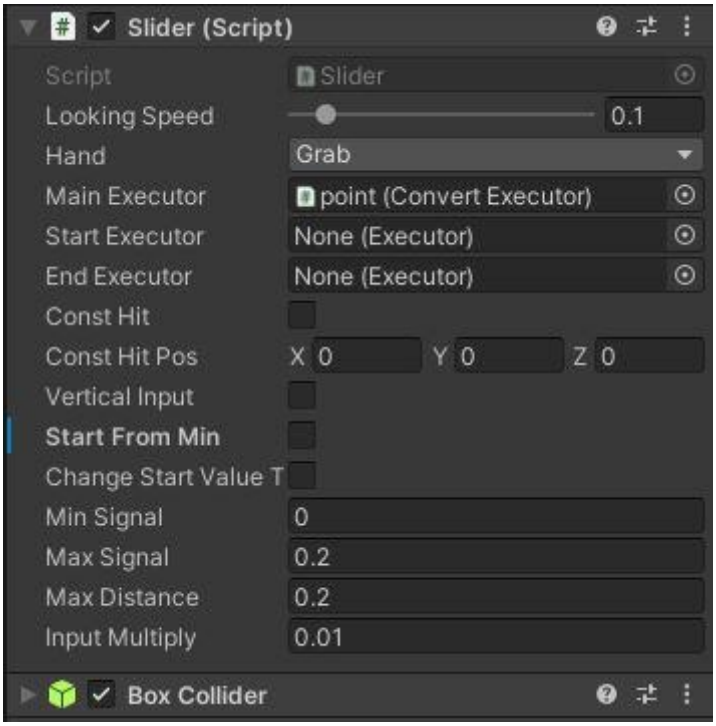
7. Slider

Слайдер – это орган управления, который нужно перемещать для изменения выходного сигнала. Включи отображение Gizmos для понимания работы слайдера.

Важно!

Слайдер движется только по локальной оси forward. Необходимо учитывать это при моделировании объектов в сторонних программах. Либо просто использовать пустой объект с любым

компонентом Collider, внутри которого будет визуальная составляющая слайдера, повернутая на необходимый угол.



Параметры

Vertical Input – при перемещении слайдера будет использоваться вертикальная ось мышки.

Start From Min – если true, то текущее состояние на сцене будет восприниматься как минимальное значение слайдера (слайдер стоит в своей минимальной точке).

Change Start Value To Difference – если Start From Min смещает заданные значения Max Value и Min Value.

Min и **Max Signal** – минимальное и максимальное значение выходного сигнала.

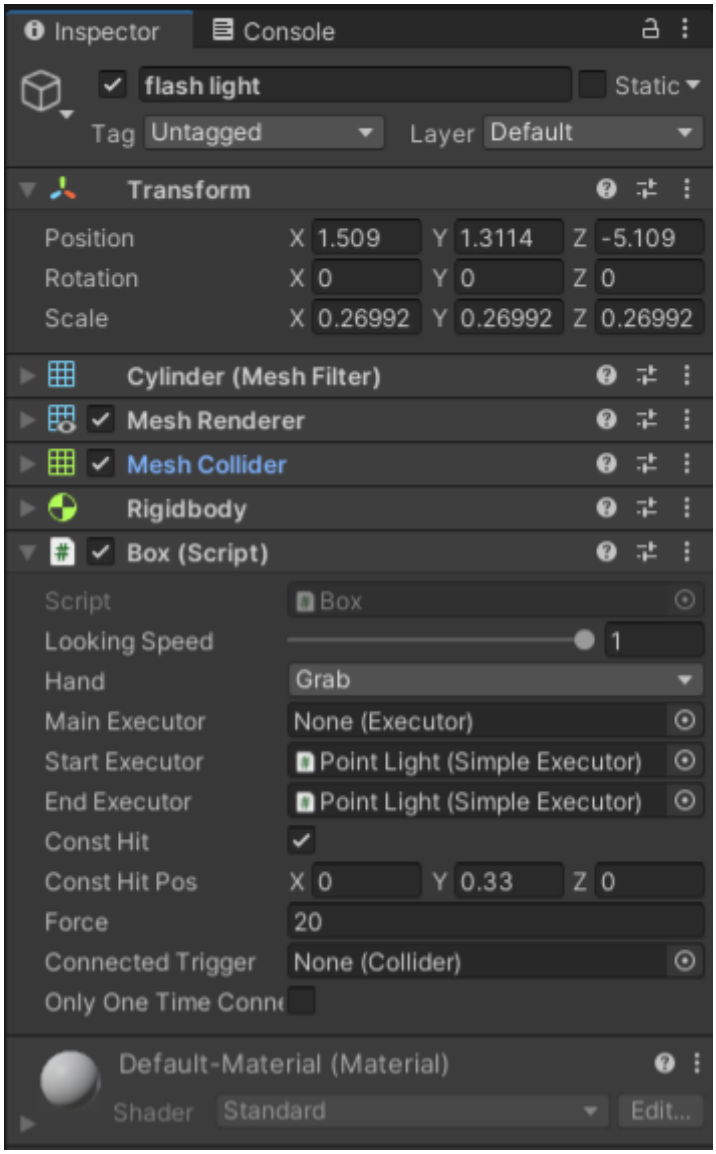
Max Distance – дистанция перемещения слайдера от его начальной точки.

Input Multiply – множитель движения мышки. Параметр, определяющий на сколько плавно и медленно двигается слайдер в зависимости от перемещения мышки.

8. Box

Ящик – это предмет взаимодействия, который можно свободно двигать и вращать. При этом точка соприкосновения вносит во взаимодействие свои коррективы (в зависимости от настроек), что позволит положить предмет на пол именно в том положении, в котором необходимо.

В зависимости от настроек ящик может быть объектом-ключом в создаваемой игре. Занимая определенное положение в сцене (соприкосновение с триггером), он выдает сигнал для связанных с ним исполнителей.



Параметры

Force – сила, с которой предмет притягивается к целевой точке соприкосновения. Рекомендуемое – 30.

Connected Trigger – компонент Collider с отметкой isTrigger, при соприкосновении с которым ящик займет положение триггера и выпустит сигнал для привязанного исполнителя Main Executor. Это можно использовать как ключ для двери, источник питания для механизма и т.п.

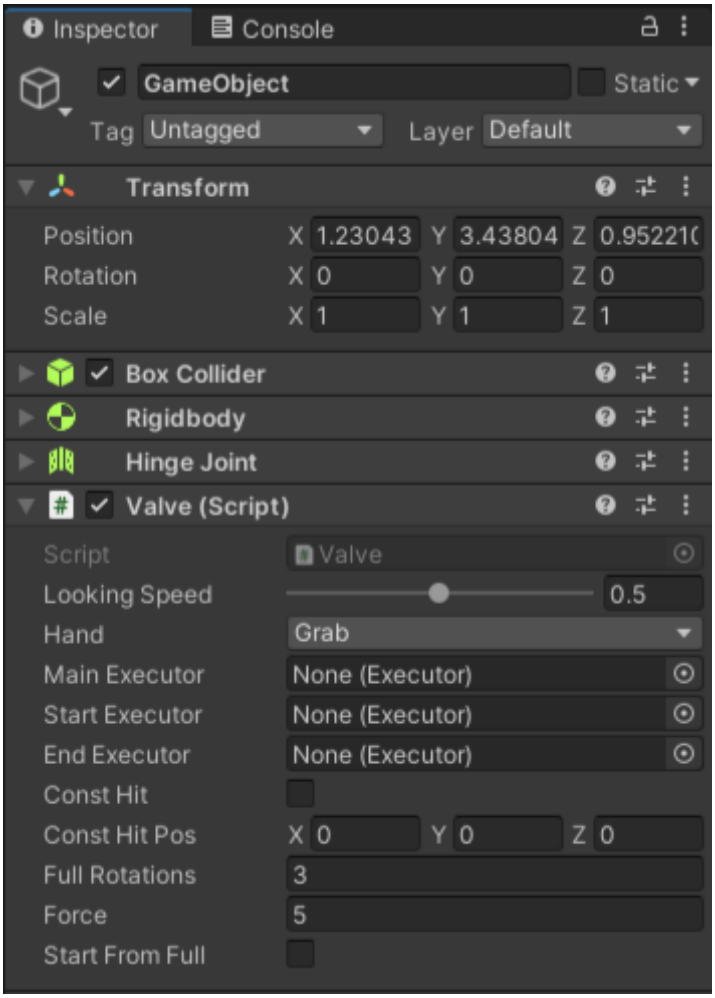
Only One Time Connect – одноразовое применение триггера. Примагнитить единожды и не отпусткать.

Важно!

Класс Box использует параметр isKinematic компонента Rigidbody для «приколачивания объекта». Если изначально поставить в сцене парметр isKinematic в true, то при взаимодействии параметр переключится, исполнится Main Executor с сигналом 0. При работе с триггером происходит то же – при соприкосновении с тригерром Main Executor выполняет сигнал 1, при отключении от триггера – 0.

9. Valve

Вентиль вращается вокруг заданной оси с помощью компонента Hinge Joint.



Параметры

Full Rotations – количество оборотов, которые может сделать вентиль.

Force – сила, с которой вентиль будет тянуться к целевой точке вращения.

Start From Full – исходное положение – максимальное.

Важно!

Фактическое количество оборотов немного превышает заданные значения на четверть оборота вперед от максимума и четверть оборота назад от минимума.

10.Button

Кнопка выдает сигнал привязанным исполнителям при отпускании нажатия. Анимация нажатия реализована с помощь кода, а не компонента Animator. Кнопка нажимается (опускается по оси) только по локальной оси forward.

Параметры

Visual Button – визуальная часть кнопки, мэш, который будет подвержен перемещению. Используй пустой объект, внутри которого мэш, для точного позиционирования кнопки.

Click Distance – расстояние до нажатого состояния от текущего.

Only One Switch On– одноразовое нажатие кнопки без возможности дальнейшего применения.

Start From On – начальное состояние кнопки – включенное.

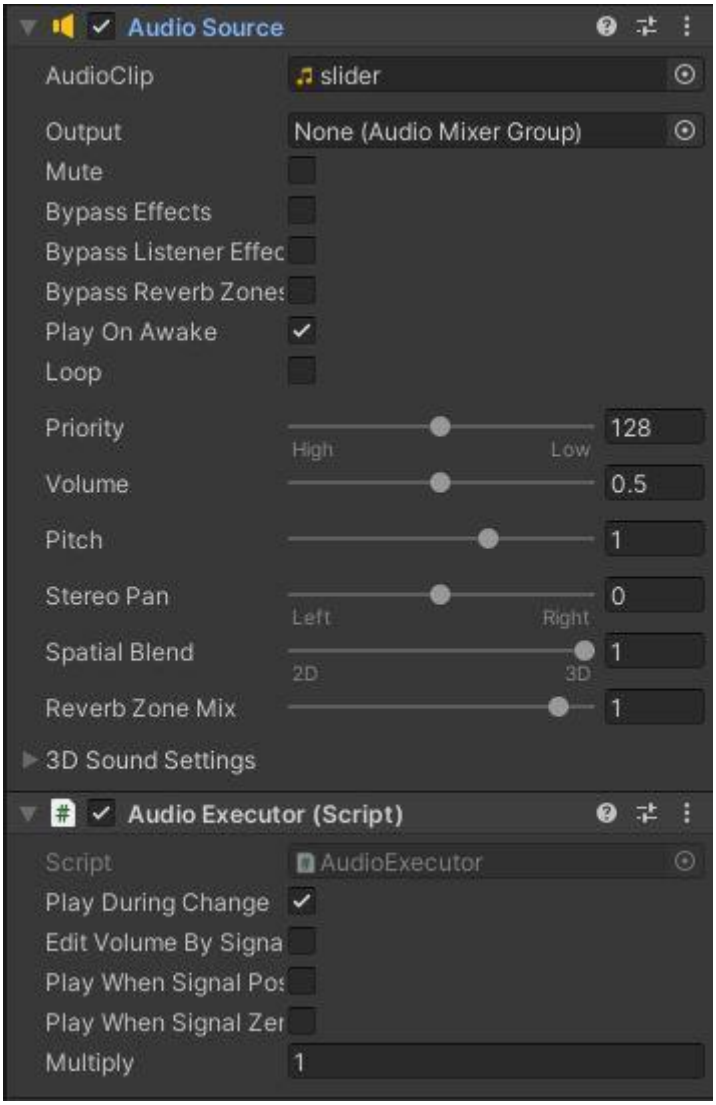
Only Switching On – работа на выдачу положительного сигнала (без переключения на сигнал 0)

11.Исполнители

Ряд исполнителей-посредников между предметами взаимодействия и конечными объектами сцены, подвергающимися изменениям, такие как свет, мэш, звуковые источники и т.д.

Audio Executor

Позволяет включать Play() и выключать Stop() источники звука Audio Source.



Play During Change позволяет воспроизводить звук только при изменении входящего сигнала. Например, звук скрежета будет воспроизводиться только при вращении вентиля, скрип только при открывании двери.

Edit Volume By Signal регулирует громкость источника в зависимости от входящего сигнала.

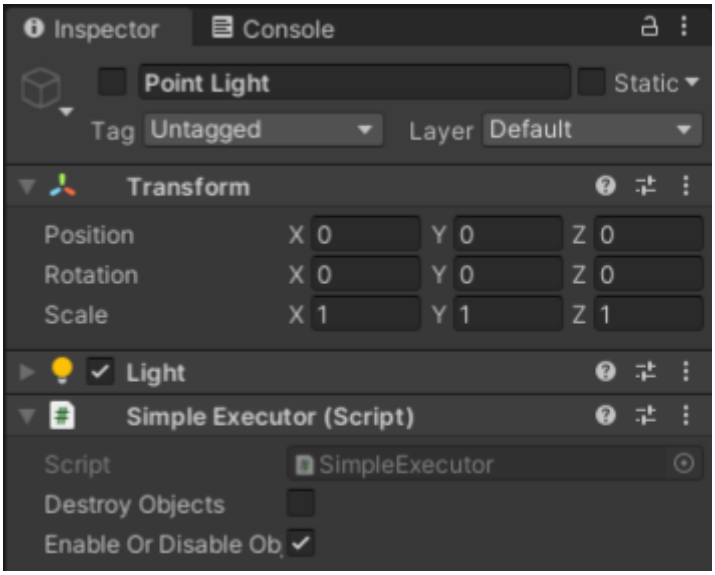
Play When Signal Positive запускает аудио дорожку при сигнале >= 1.

Play When Signal Zero запускает аудио дорожку при сигнале < 1.

Multiply преобразует входящий сигнал.

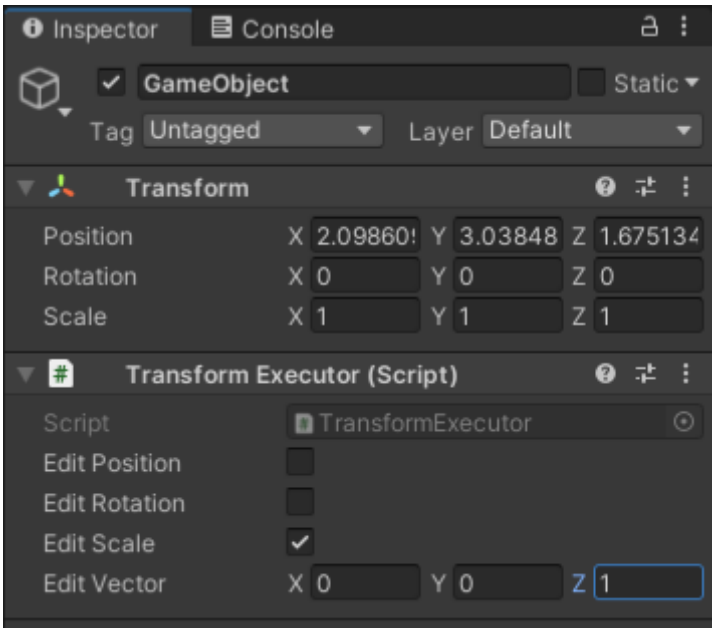
Simple Executor

Включает/выключает или уничтожает объект, на котором находится при получении сигнала >= 1.



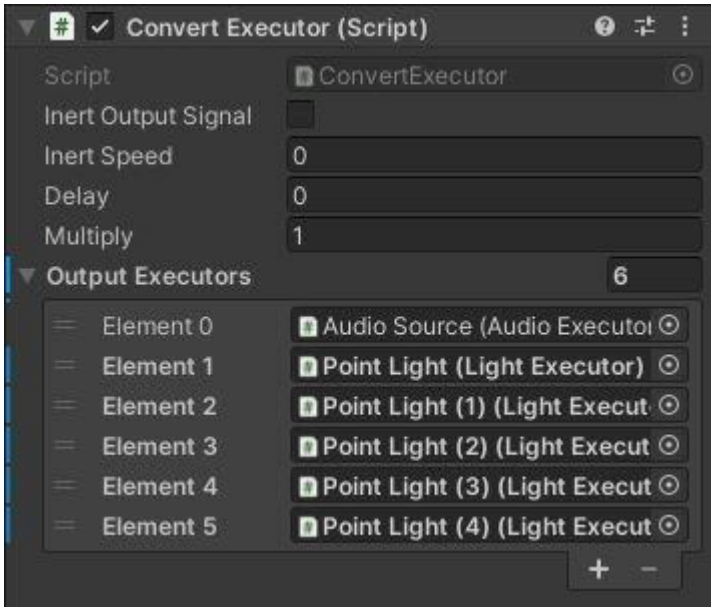
Transform Executor

Изменяет параметры компонента Transform в зависимости от входящего сигнала. **Edit Vector** определяет на сколько будет изменен компонент по трем осям от начального состояния.



Convert Executor

Посредник, преобразующий входящий сигнал. Позволяет сделать инертность конечного исполнителя и медленное нарастание текущего выходного сигнала к целевому входящему.



Inert Speed определяет скорость нарастания сигнала от текущего к целевому входящему.

Delay – задержка выходного сигнала, в секундах.

Multiply преобразует входящий сигнал.

Output Executors выполняют преобразованный сигнал.

Light Executor

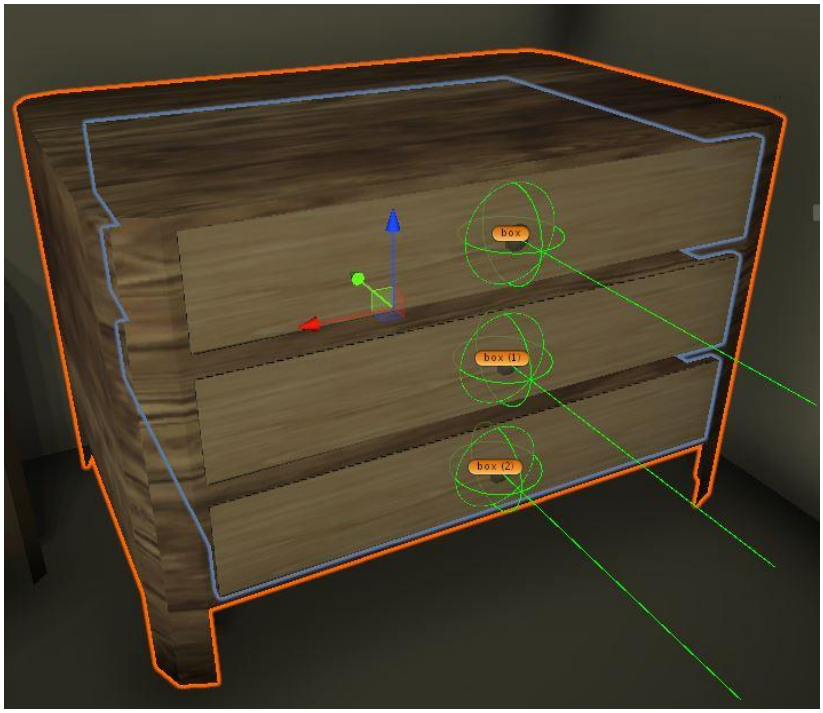
Манипулирует параметром Intensity компонента Light.

12.Неочевидное применение

Некоторые сочетания компонентов ассета могут приятно порадовать дополнительным функционалом!

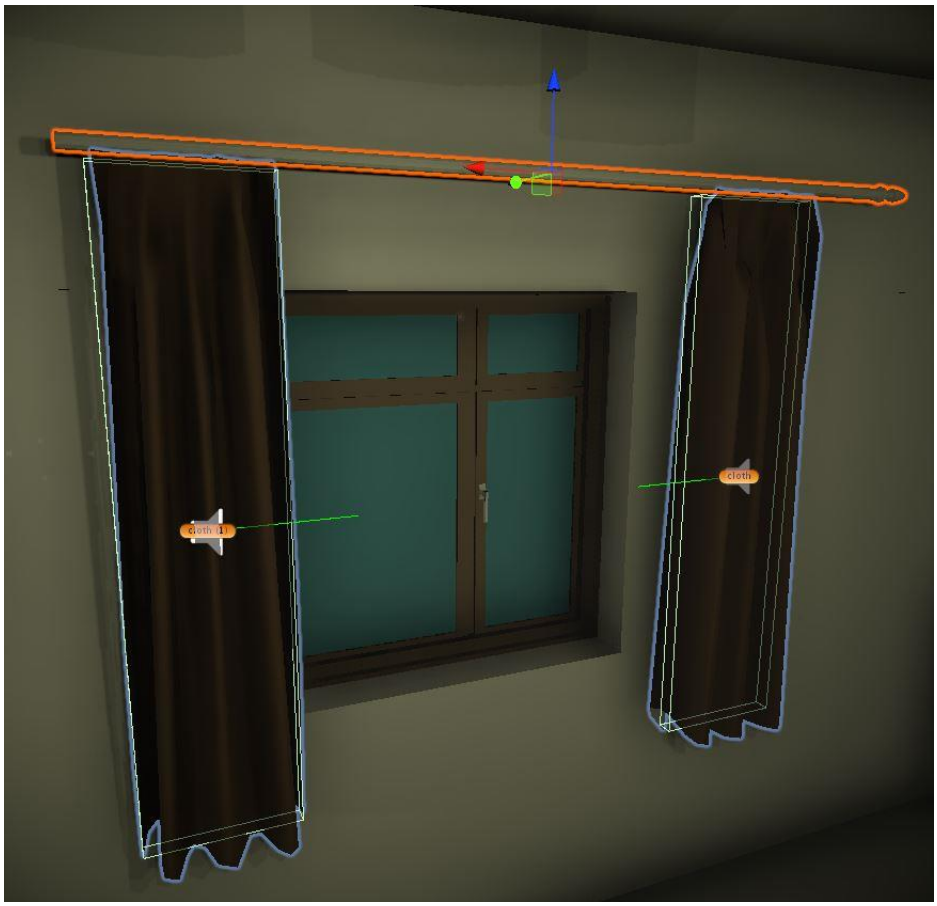
Slider

Слайдер можно использовать в других направлениях без привязанных исполнителей. Компонент может быть использован для реализации выдвижного ящика в столе или шкафу, а так же для сдвигаемой двери или задвижки.



Шторы, жалюзи

Сочетание компонентов Slider и Transform Executor (включить Edit Scale) могут использоваться в качестве штор или жалюзи, которые могут сдвигаться, при этом уменьшаясь в длине.



Фонарь

При подбирании фонарь с компонентом Box может загораться светом с помощью Simple Executor на источнике света (компонент Light) в составе фонаря и выключаться при отпускании.

13.Ссылки

В случае необходимости в помощи, мы всегда рады приветствовать вас в Discord-канале, где постараемся оказать максимальную техническую поддержку!

Discord https://discord.gg/R4JfuVS

Остальные ссылки для связи с нами.

Asset store https://assetstore.unity.com/publishers/46819

YouTube https://www.youtube.com/channel/UCwarNiEVDupG3ape9buKB9Q

Отдельное спасибо людям, которые оставляли комментарии под описаниями предыдущих ассетов. Ваша поддержка и отзывы подталкивают нас заниматься новыми работами.