**Import**

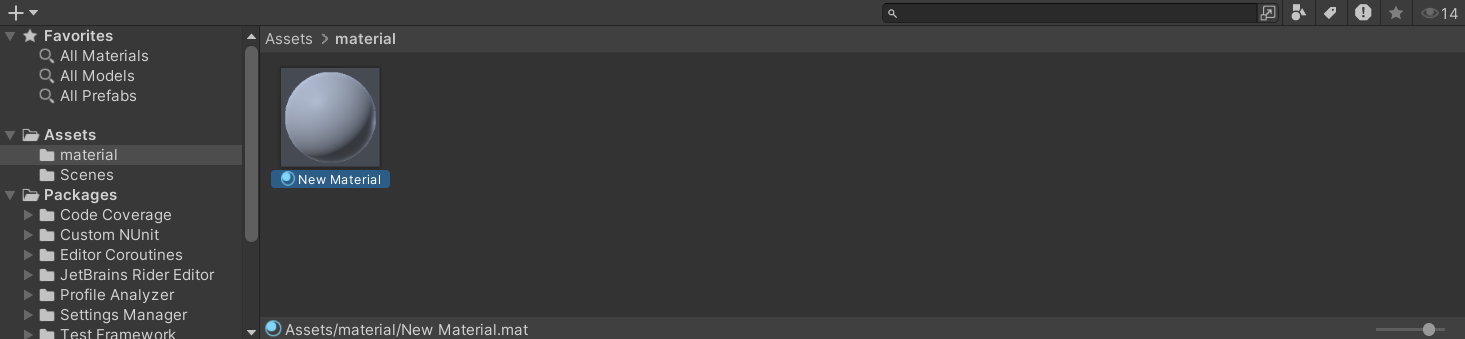
Что бы импортировать наши объекты/текстуры/анимации/аудио мы можем либо скопировать и вставить в папку проекта или drug and drop в открытый Unity.

Создать папку можно с помощь правого клика мыши.

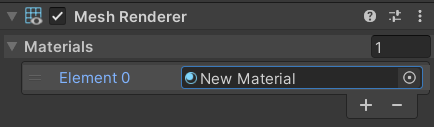


**Creating a Material Asset**

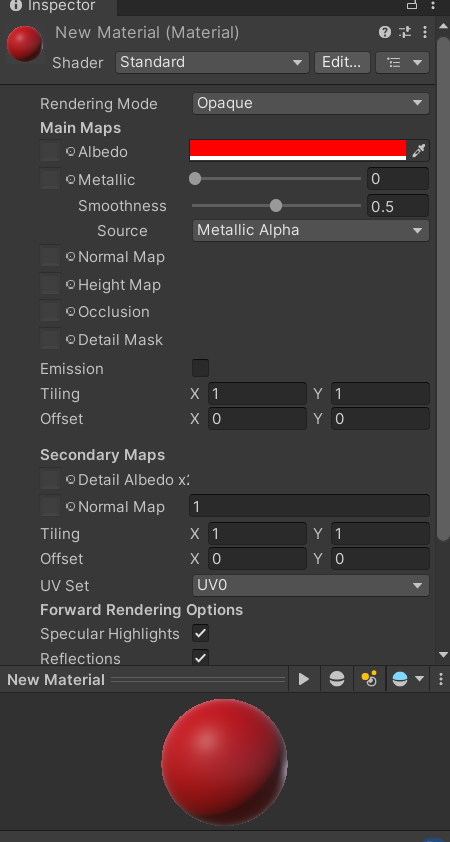
Для начала создаем папку с название Material. И далее создаем материал, либо нажимаем на плюс, либо на правую кнопку мыши. Ползунок в нижнем углу делает наш контент менбше или больш.



Что бы накинуть материал на объект, нужно выбрать объект и окне Inspector -> mesh render -> material перетаскиваем наш материал. Или же мы можем просто накинуть материал на мешь в сцене.

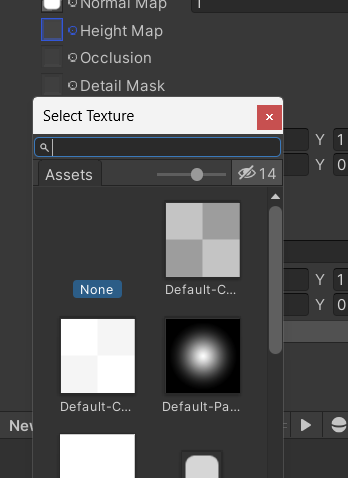
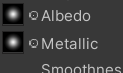


Когда мы выбираем наш материал в Inspector появляются все свойства материал. Тут мы можем модифицировать наш материал.

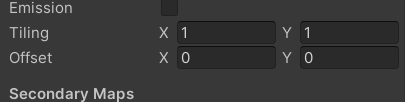


Мы так же можем сделать дубликат материала Ctrl +D.

крадратик возле названия в материале, позволяет нам выбирать шаблоны, как имеено будет накладываться цвет, отрадение, металичность и так же с картами нормалей и высот, это текстуры.

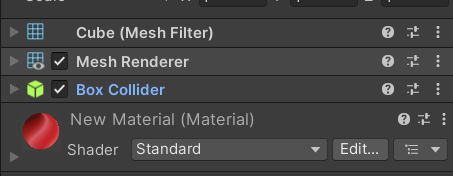


В материале я могу менять значения Tiling.

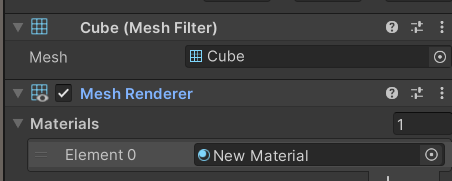


**Component**

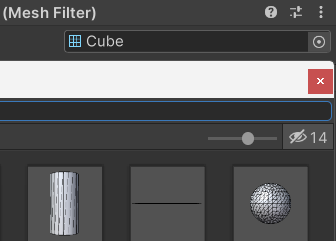
Каждый GameObjects имеет свои компоненты



Одним из компонентов всех видимых объектов является *Renderer*, он определяет материал для объекта, так же есть (*mesh filter*), который визуализирует объект, определяет его фактическую форму.



*Mesh filter* - мы можем объекту задать другую форму, нажав на круг сбоку и нам появляется список форм.

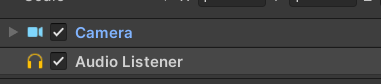


*Collider* - этот компонент указывает на столкновение объекта. С помощью Size мы можем регулировать размер колайдера.





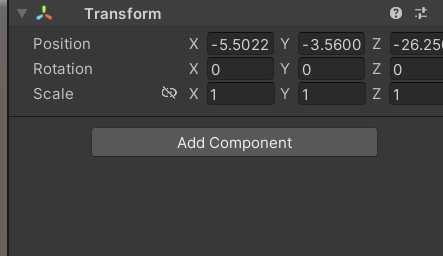
К примеру компоненты камеры



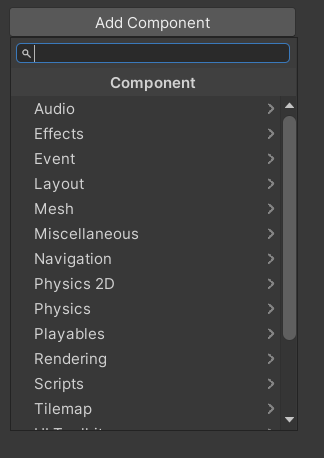
Компонент *Camera -* имет в себе много настроек, которые мы можем изменять под наши потребности.



Игровым объектам мы можем добавлять новые компоненты, к примеру, к объекту Empty -> Add component. Тут я хочу показать что любой объект может быть чем угодно в зависимости от компонента.



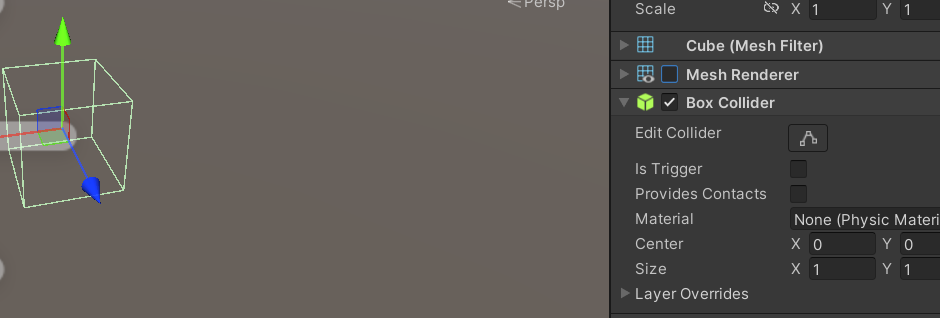
Сначала ищем категорию



А потом выбираем, то чем бы он был



Так же мы может отключать компоненты, сняв галочку



**Light**

Есть два типа освещения

* Освещение в реальном времени(directional light)
* Запеченный свет

Для начла разберем основные источники света и как их настраивать

GameObjects -> lights

*Directional Light*

Нет ограниченого расстояния для влияния

*В Ispector в* ***Type*** *стоит значение Directional, но мы можем поставить point/area/spot*

***Mode*** *- realtime, служит нам источником света в реальном времени и мы не используем запекание, для запекания мы* используем *mode - baked.*

*Intensity -* с какой силой наш источник света будет действоать на сцену.

*Indirect Multiplier* - контролирует интенсивность не прямого освящения (влияние отраженного света от другой поверхности)

*Point light*

Есть ограниченное расстояние на которое он может влиять, освещение подходит для освещения небольших объектов или объектов.

*Range* - помогает увеличить яркость источник cвета.

*Spot light*

Похож на спет фонаря

*Spot Angle* - Можно котрролировать угол открытия чтоб свет был больше

**Shades**

В источниках света у нас есть возможность задать им что б они не отбрасывали тени

*Shadows type* - **No shadows** (объекты, которые затрагивает этот свет, не будут отбрасывать тени), тут мы можем выбрать и **Harg shadows** (тени будут жесткие) и даже такой параметр потребляем намного меньше ресурсов чем **Soft shadows** (более мягкие и реалистичные тени). Тени играют важную часть в производителтности игры.

*Resolution* - качество теней (чем выше значение тем они реалистичнее и будут верно просчитаны и тем больше влияет на производительность игры)

*Strenght* - прозрачность теней

Так же тени мы может регулировать через параметры объекта, который отбрасывает эти тени

Inspector -> *cast shadows* (off/on) (знасит что объект либо булет отбрасывать тени, либо нет)

*Receive shadow* - должен получать тени от других объектов, то есть, будут ли на объекте отображены тени других объектов.

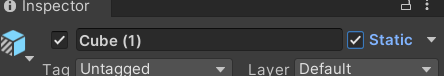
Project Settings => Quality -> shadows - и настраиваем (distance shadows - на каком расстоянии будут рисоваться тени)

**Backed light (ложное освещение)**

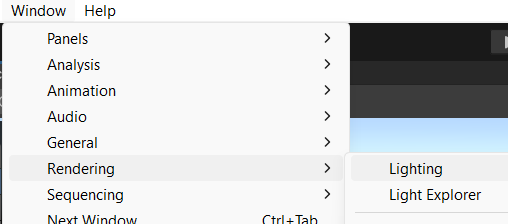
Все источники света в сцене должны быть в Mode (backed)



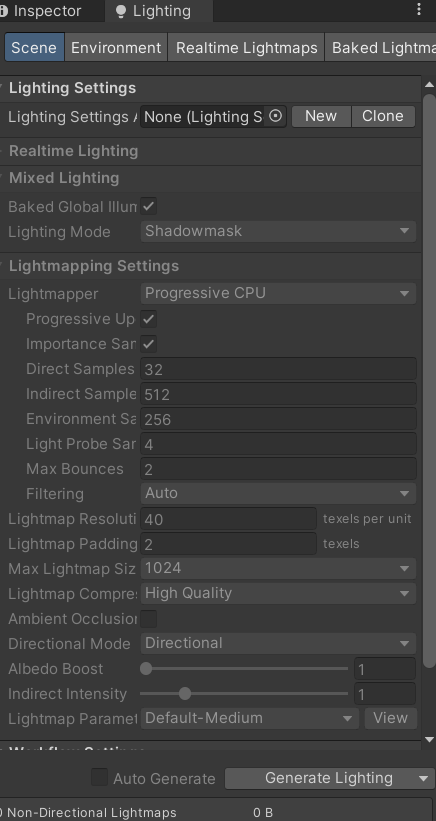
Все объекты должны быть Static



Далее открываем окно для запекания

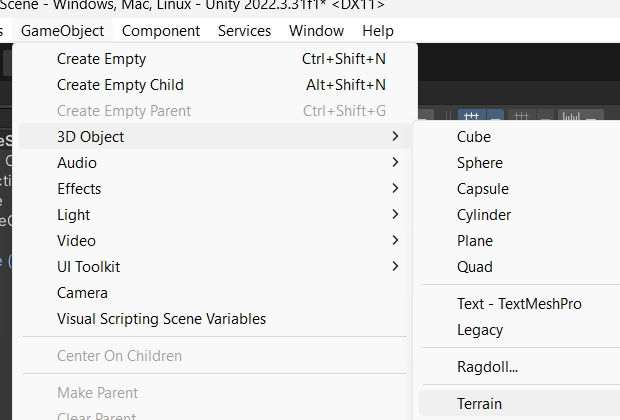


После мы мжем сделать насстройки

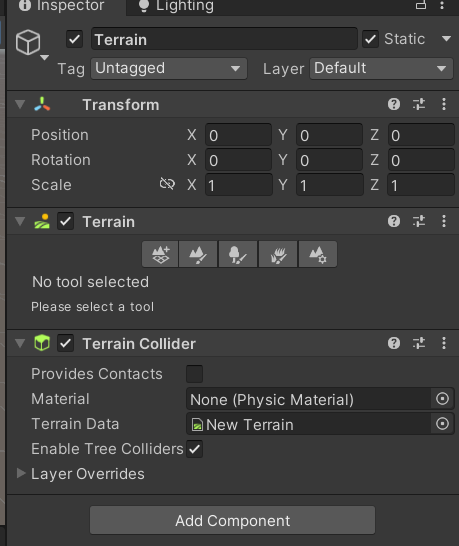


И после запустить запекание, оно может занять время

**Landscape**

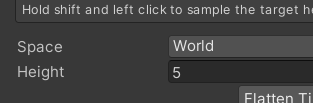


Terrain имеет свои свойства



С помощью мы можем придать нашей земле форму

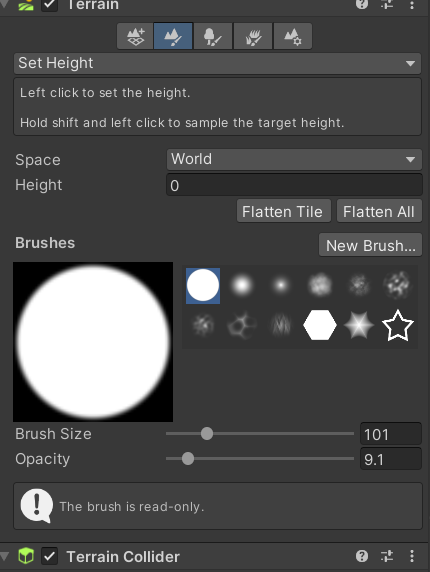
Для начала необходимо задать высоту и потом начать лепить материк



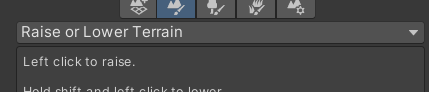
Первая опция, означает что мы можем исправить размер нашего ландшафта



Вторая опция деформирует рельеф

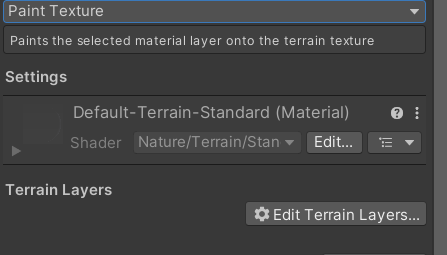


Список помогает нам переключаться межлу способами деформации ландшафта

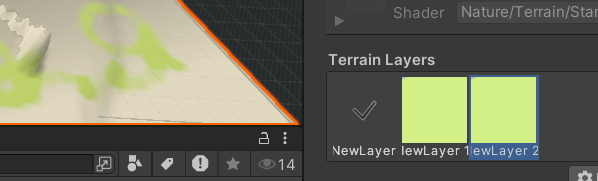


Что бы покрыть текстурой

Режим Paint texture, Edit terrrain Layers

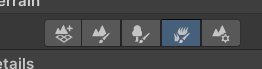
’

Что бы рисовать двемя текстурами, нужно добавить несколько, первый основной и второй тот которым будем рисовать

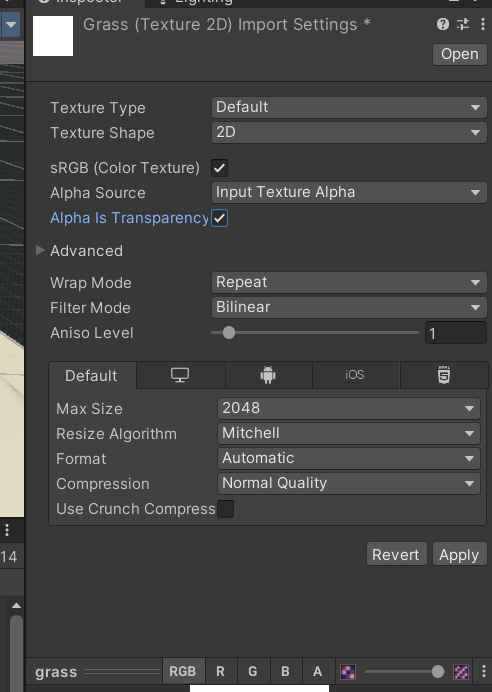


**Placing details**

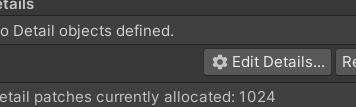
Paint details

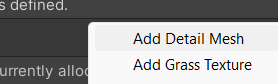


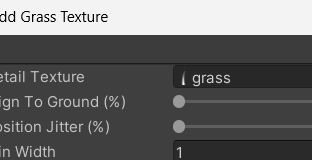
Будем распределять траву, для этого есть картика с alpha каналом и на свойствах картинка ствами прозрачность -> Apply



Идем в свойста ландшафта







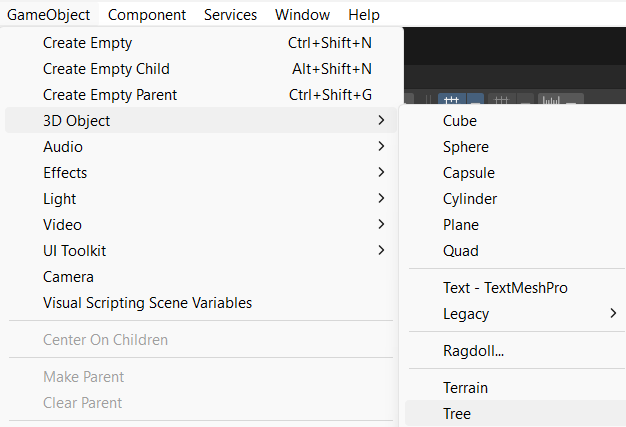
Shift - позволяет нам стерать не нужную траву

**Trees**

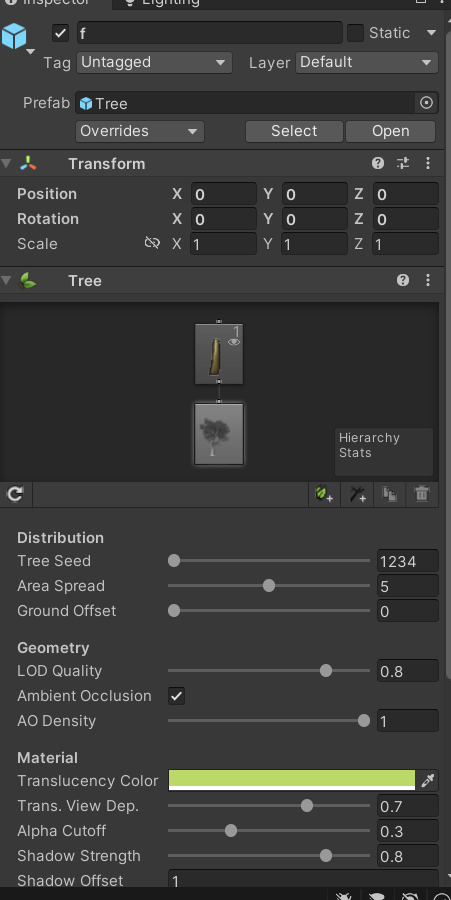
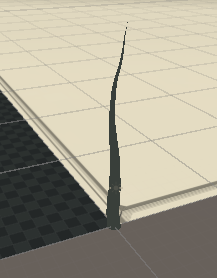
У нас есть текстура дерева и листвы

Тут нам будет нужен следующий инструмент

GameObject -> 3DObject -> tree



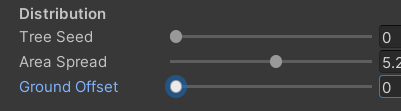
У нас появдяется ствол



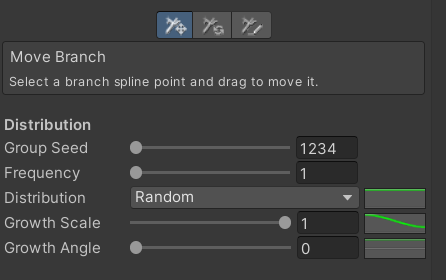
Далее мы можем выбрать редактировать ствол или крону дерева



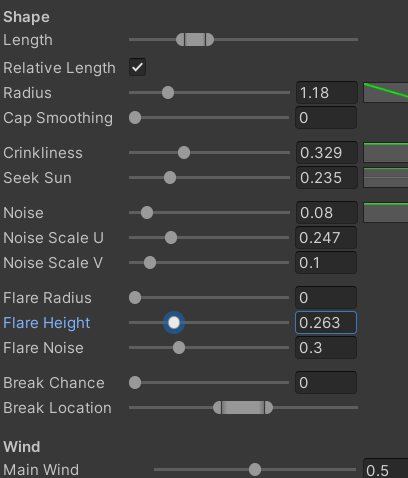
Далее своиства помогут нам разместить на ландщафте сверху или опустить ниже, так же какая толщина ближе к земле



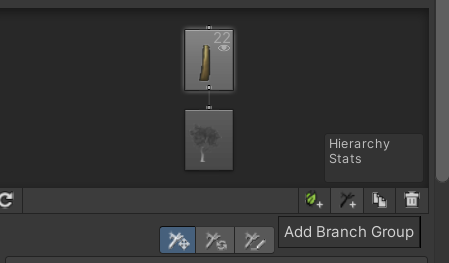
Эти свойства помогут сделать направдение ствола и количество



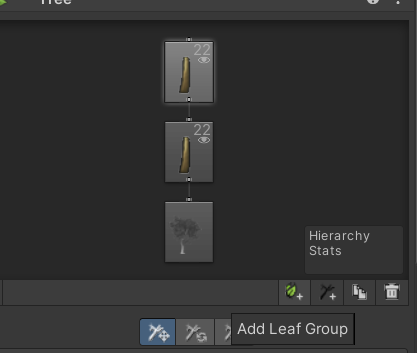
Можем играть с формой дерева



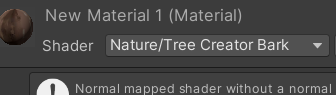
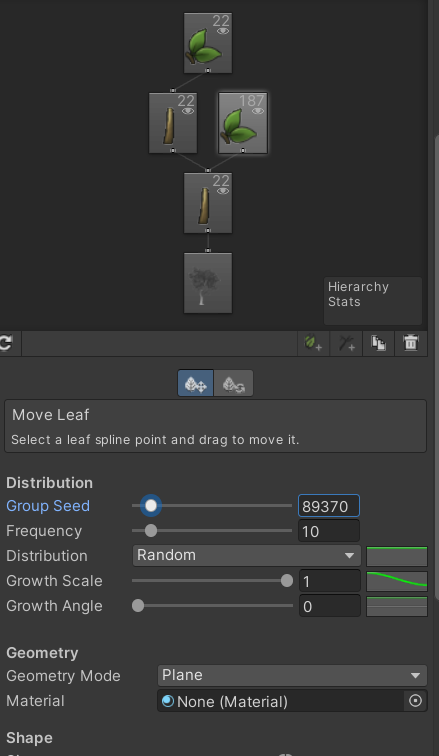
Добавляя ветки, мы их редактируем так же как и ствол



Выбрав нод добавляем листья

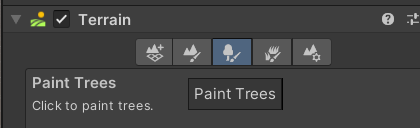


Нод вырастает и появляются настройки листьев, там есть материал, создаем новый материал, где будет картинка наших листье и дропаем

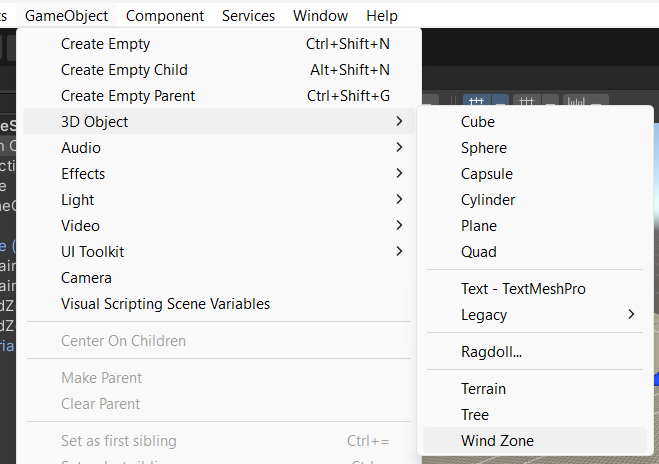


Далее создаем из дерева префаб и используем

И с помощью инструмента можем их раположить





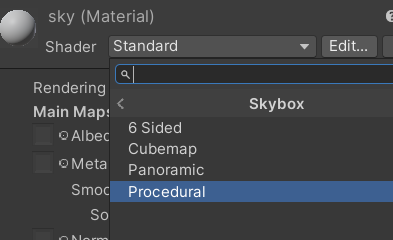


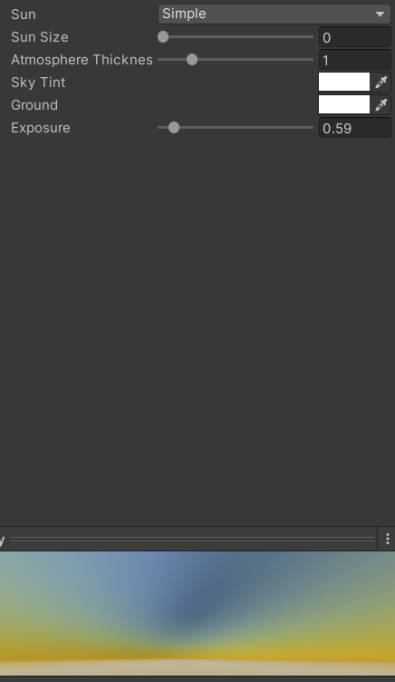
Созданный ветер можно будет увидеть в симуляции

**Sky**

Сделаем skybox

Создаем материал и тип у него будет skybox procedural



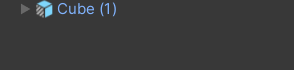


**Create Prefab**

Поскольку объекты имеют множество уникальных компонентов и настроек, мы хотим сохранить игровой объект, что бы мы могли повторно использовать его позже. Мы можем сделать общие объекты, этого мы можем добиться с помощью Prefabs.

Prefabs - это способ упоковать ваши объекты со всеми внесеными изменениями, что бы мы могли использовать их снова и снова. И что бы изменить один объект и тогда изменения распространяться на экземпляры.

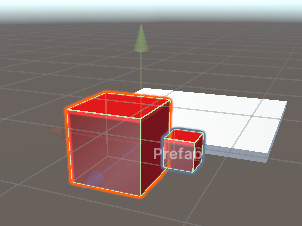
Мы имеем куб с физикой, для примера сдеблируем куб и сделаем его меньше и так же удалим с него физику и сделаем этот маленький куб дочерним. Из окна Иерархи я перетаскиваю эту конструкцию в Project window. Он будет подкрашиваться в синий цвет и обозначать что это Prefab.

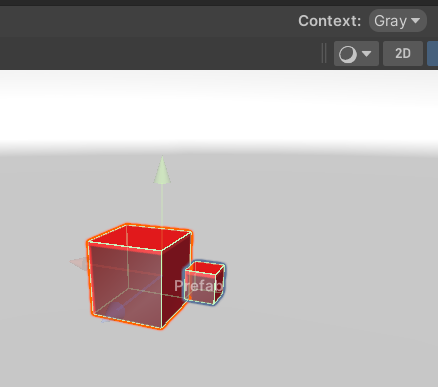


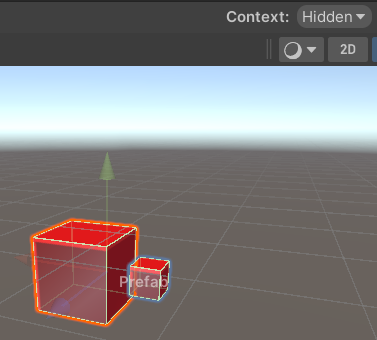
Если мы даже удалим этот Prefab из окна Иерархии, то он у нас есть в Projects window в папке, куда мы его перетащили.

Мы можем закинуть на Prefab в сцене и в инспекторе нажать Open, так мы войдем в режим изоляции и в иерархии остануться только те объекты которые есть в этом Prefabe. И как только в этом режиме мы начнем что то менять (rotayion/scale/move) это будет отображаться на экземплярах. В этом режиме мы можем воспользоваться Context mode и выбрать как лучше нам просматривать.









Создавая префабы, дочернии элементы, мы можем использовать пивот родителей для дочерних элементов, если нам нужно чтоб куб крутился оносительно нижней части, то мы можем подвинуть куб так как нам нужнно оносительно центра (пивота) родителя.