**Слои**

Все разделено на тайлы земли и игровые объекты. Создано два слоя groundLevel и objectsLevel.

**Организация слоев отрисовки спрайтов.**

Сделано разделение на:

Sorting Layers – отвечают за «уровень земли», то есть платформу, на которой играем: создаются с именами Ground1, ground2, т.д. Cделано для возможности разместить одну платформу над другой, корректной отрисовки этого.

OrderInLayer их два : уровень 1 для всех игровых объектов, присваивается автоматически при добавлении к объекту скрипта IsoObject; уровень 0 предназначен для тайлов земли.

Позиционирование по оси Z, целочисленное, чем больше значение (т.е. чем выше объект на экране) тем глубже он расположен, т.е. перекрывается более близкими объектами с меньшим значением Z.

**Скрипты**

IsoObject скрипт = отрисовка спрайтов ИГРОВЫХ ОБЪЕКРОВ, не земли то есть, в правильной последовательности. Автоматически присваивает объекту значение 1 в OrderInLayer. Есть три чекбокса:

isPassable- можно ли пройти через; isCoverable-если стоишь в клетке объекта, закрывается ли он тобой; isInterPlayable – предполагается ли взаимодействие, требующее твоей остановки? Если да, игрок доходит в объект, и останавливается в одной с ним клетке.

Если отключаем isPassable, то автоматически отключаются теги isCoverable и isInterPlayable.

Тег isCoverable включает закрытие игроком объекта, на котором он стоит.

EditorPrefs cкрипт = позиционирование спрайтов по 2D сетке в режиме редактирования, привязывется к pivot спрайта, должен быть добавлен ко всем игровым объектам; при добавлении polygonCollider2D объекту придает коллайдеру ромбовидную форму.

**Движение объектов**

Задаем направление движения игрока,

В указанном направлении на длину distance стреляет луч, записываются все столкновения на слое groundLayer,

Distance уменьшается на 1, если не равно количеству столкновений. Итоговый вариант distance записывается, и кубик перемещается на его величину.

Запускаем еще один луч, проверка в слое Objects.

Если есть столкновение с объектом, проверяем, можно ли через него пройти, по тегу IsPassable. Если нет, то запускаем цикл уменьшения distance и стрельбы луча до момента, пока луч не перестанет сталкиваться с объектом.

Если через объект можно пройти, то проверяем стоит ли тег isInterPlayable. Если стоит, то при достижении позиции объекта мы остановимся.

(!!!ПОФИКСИТЬ Отображение этих тегов на модельке игрока!!!)