# **Задание**

Собрать тестовую сцену “Зиккураты” для проверки ИИ ботов.  
 “Зиккураты” - это игровая карта, где постоянно создаются новые персонажи под управлением бота, затем персонажи вступают в бой друг с дружкой, погибают и цикл повторяется. Игрок выступает в роли наблюдателя, который может в режиме реального времени балансировать персонажей.

**Обязательные требования**

1. Создать сцену, в которой:

* Расположено три Зиккурата разных цветов;
* Каждый Зиккурат имеет собственные настройки параметров создаваемых персонажей;
* Зиккураты создают юнитов, окрашенных в свой цвет, и отправляют их в центр карты, где те действуют по логике, описываемой далее.

1. Создаваемые юниты должны:

* Находить ближайшего не союзного юнита и двигаться к нему;
* При приближении, юнит атакует быстрой и слабой или медленной и сильной атакой. В случае победы - переключается на следующего противника.
* У юнитов есть следующие параметры: здоровье, скорость перемещения, урон от слабой и от медленной атаки, вероятность промаха, вероятность двукратного урона, процентное соотношение вероятности слабой и сильной атак.

1. У игрока имеется:

* Управление посредством WASD и мышки или аналогично управлению камеры в редакторе;
* Выделение юнита или Зиккурата по щелчку мыши;
* При выделении Зиккурата, игроку открывается панель с параметрами создаваемых юнитов, при изменении которых, следующие создаваемые юниты имеют измененные параметры.

**Ограничения**

1. Вся информация о юнитах и Зиккуратах должна выводиться в игровом UI при нажатии мышкой на соответствующий игровой объект. UI должен позволять изменять баланс во время игры.
2. Все описанные в пункте 2.3 свойства юнитов должны влиять на игру, то есть обязательно реализовать логику, учитывающие перечисленные параметры.
3. Все добавленные панели UI должны быть выдвижными, то есть по нажатию на шапку прятаться/извлекаться. Движение необходимо реализовать через Update() или корутины, во время анимации, кнопки не должны нажиматься.

**Дополнительные требования**

1. Расширить возможности игрока, добавив дополнительную панель управления, на которой:

* Кнопка “Убить всех” - уничтожает всех живых юнитов;
* Кнопка “Отобразить здоровье” - открывает/скрывает шкалы здоровья над каждым юнитом.

1. Добавить логику блуждания для ботов, в случае, если в центре карты нет противников;
2. Добавить панель со статистикой по Зиккуратам:

* Количество живых юнитов;
* Количество убитых юнитов;
* Оставшееся время до создания нового юнита;
* Кнопка “Очистить” - сбрасывающая счетчик убитых юнитов.

**Подсказки**

1. У Зиккуратов уже настроены коллайдеры, осталось только реализовать один из интерфейсов EventTrigger’а (3 модуль 4 урок).
2. В центре карты есть пустой GameObject, его можно использовать как первоначальную цель для создаваемых юнитов, если применить логику из урока по имитации движения.
3. В играх, обычно, вероятности работают по следующим принципам:

* Бросается случайное число (класс Random в библиотеке System или UnityEngine) от 0 до 1, где 0 - событие гарантированно провально, а 1 - событие гарантированно успешно;
* У вероятности есть какое-то значение, например 0.5 (то есть событие будет успешно в половине случаев);
* Если случайное число меньше значения вероятности - событие не срабатывает, если больше или равно - событие срабатывает. Таким образом, во всех играх реализуется неуправляемый рандом.

1. Процентное соотношение слабой и сильной атак можно трактовать следующим образом: чем выше значение (ближе к 1) тем чаще юнит применяет быструю атаку. Иначе говоря, при вероятности равной 0 - юнит всегда делает сильные атаки, при 0.5 - в половине случаев, а при 1 - только быстрые атаки.
2. Управление камерой можно реализовать по аналогии с камерой из урока по имитации естественного движения.
3. Чтобы добиться плавности движения панелей UI, можно заранее определить начальное и конечное положение и покадрово изменять положение через Vector2.Lerp(startPos, endPos, time).
4. Чтобы отображать здоровье, можно воспользоваться канвасом с режимом Render Mode - World Space. Кроме того, будет наглядно и удобно, если виджет здоровья будет выполнен в качестве шкалы с числовыми обозначениями и будет поворачиваться в камеру каждый кадр.