Пример готовой сцены лежит в Assets/Scenes/SampleScene

Оглавление.

Управление:

Выбор и постройка башни- ЛКМ
Отмена выбора- ПКМ/выбор другой башни
Выбор башни для улучшения-ЛКМ+G

Ресурсы.

Все ресурсы и важные переменные(количество здоровья, золота, убийств, уровень) лежат в статичном классе GameStats.

Также этот класс сам подгружает данные из конфиг файла, который должен находиться в “Resources/Config” и иметь название “Main”. Создать его можно через CreateAssetMenu/Config/GameSettings.

Он имеет события изменения переменных, на которые подписываются скрипты из других модулей(отрисовка количества, золота, жизней, условие поражения).

Характеристики врагов.

Создать конфиг файл врага(одного типа) можно через CreateAssetMenu/Config/EnemySpawnInfo(уже созданные и настроенные лежат в папке Assets/Configs).

Конфиг файл задающий настройки спавна врагов можно создать через CreateAssetMenu/Config/EnemySpawnControllerInfo.

Характеристики башен.

Создать конфиг файл башни можно через CreateAssetMenu/Config/BuildInfo(уже созданные лежат в Assets/Configs/Build).

Система.

Все модули разбиты на тематические папки, содержащие в себе функционал для игры, редактирования конфигов, и расширения редактора. Модули находятся в Assets/Scripts.

Общие концепции.

Так, как игра игра считается сессионной(есть четкие циклы победы-поражения), то был применен паттерн FSM для управления игровым состоянием.

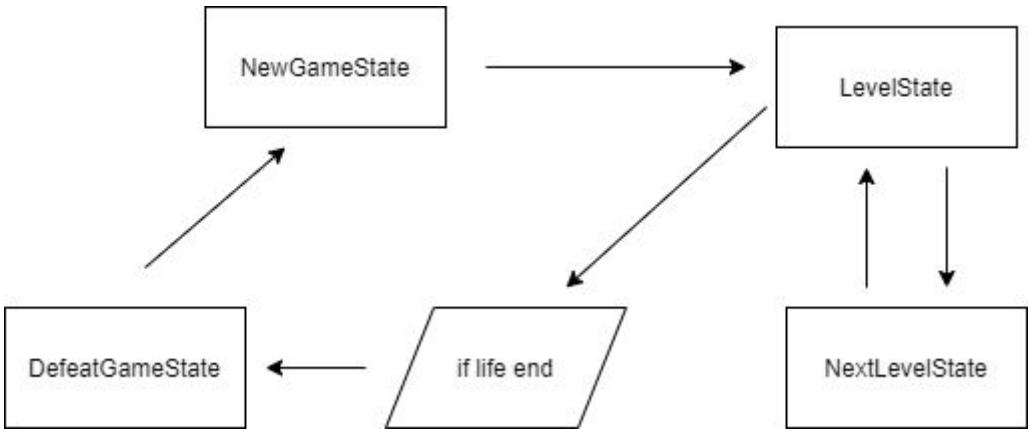
Поскольку враги могут повторяться, и требуется лишь изменять их характеристики, то для экономии на “сборщике мусора” был применен “пулинг объектов”, масштабируемый по необходимости.

Так как количество модулей достаточно обширно, то был использован паттерн “Наблюдатель” через систему событий.

FSM.

Модуль расположен в Assets/Scripts/GameState.

Схема взаимосвязи состояний.

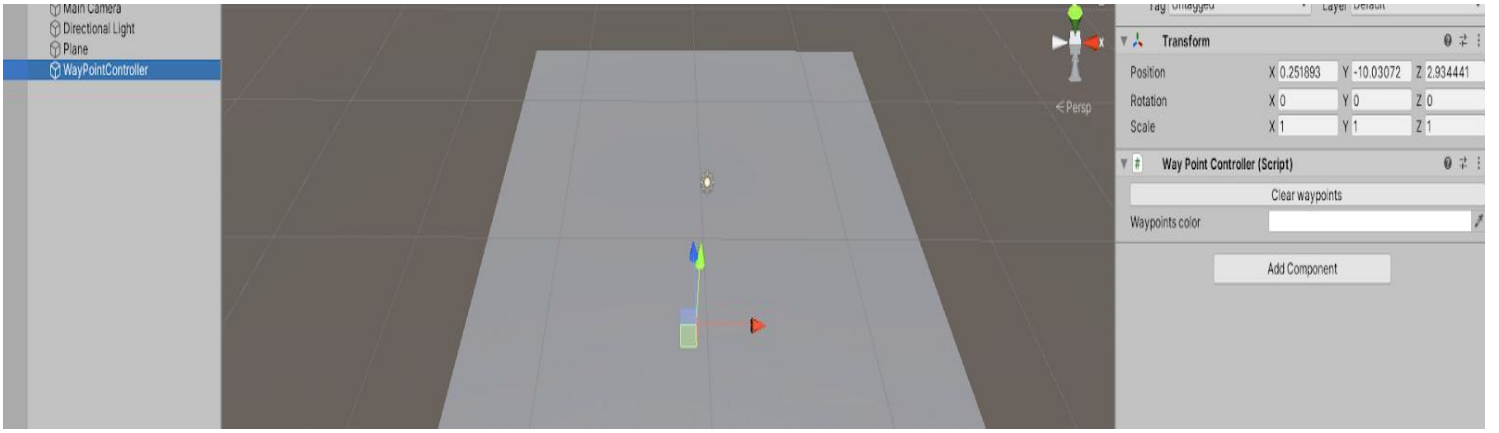


NewGameState загружает стартовые данные из main конфига.
LevelState запускает уровень и ждет окончания спавна волны.
NextLevelState запускает таймер перерыва между уровнями, давая игроку время отдохнуть и подумать.
DefeatGameState останавливает игру, показывает статистику и предлагает начать новую игру.

Построение маршрута мобов.

Для удобного построение маршрутов, по которым пойдут мобы, был написан класс WayPointController из модуля ...Scripts/WayPoint.

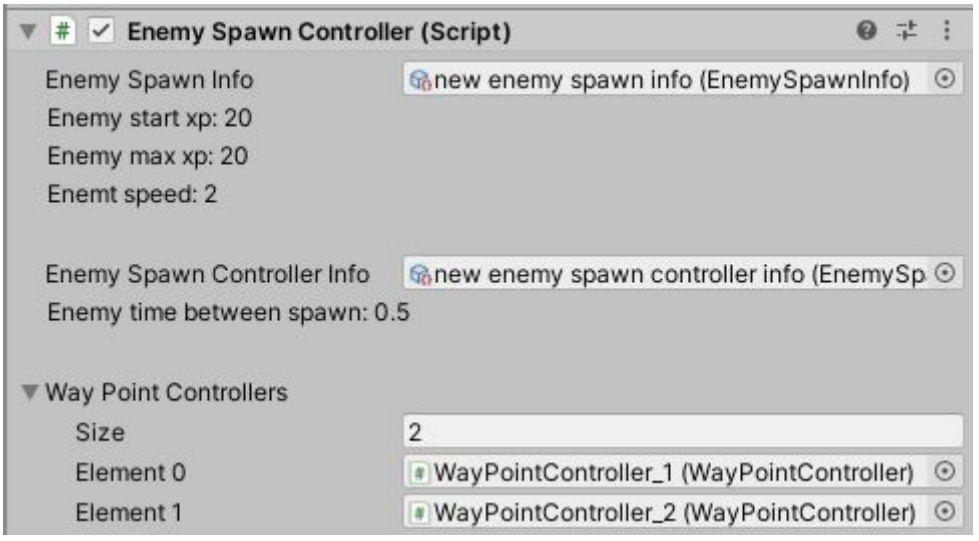
Чтобы начать редактировать маршрут нужно выбрать объект, на который прикреплен скрипт.



Теперь можно создавать WayPoint объект по нажатию ЛКМ.
Если требуется выделить точки данного маршрута, то можно использовать поле WayPoint color.
Так, же для быстрого очищения массива WayPoint”ов есть кнопка “Clear Objects”, которая очистит массив.

Спавн мобов.

Управляет спавно класс EnemySpawnController, расположенный в модуле ...Scripts/Enemies

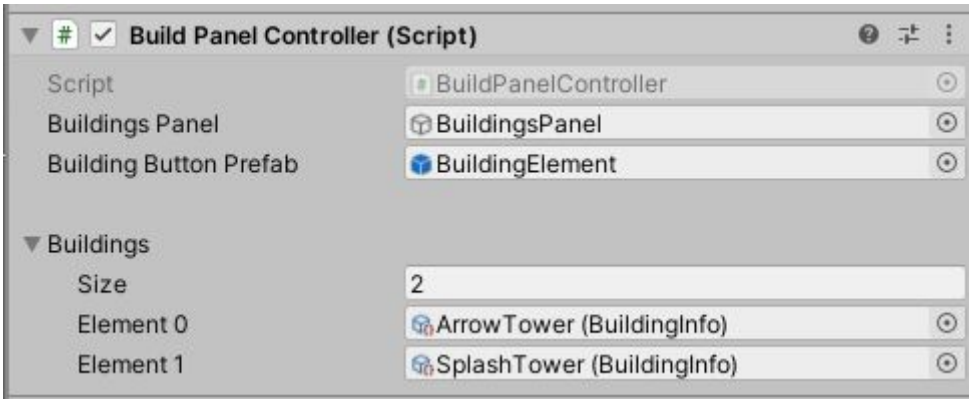


Принимает в себя два конфиг файла(один для регулировки скорости спавна, второй для того, что спавнить(имеет возможность дальнейшей масштабируемости за счет вынесения логики врага в интерфейс)).

Так, же нужно указать массив маршрутов, по которым будут следовать заспавненные враги.

Построение башен.

BuildPanelController- класс реализующий настройку башен, доступных для постройки. Из модуля ...Scripts/UI/Build.

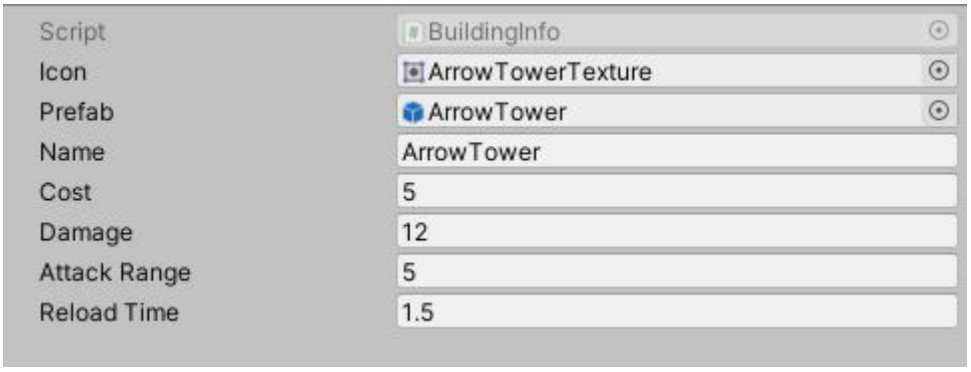


В нем требуется явно указать панель постройки и префаб кнопки интерфейса, по нажатию, на которую, будет запускаться логика постройки.

В массив “Buildings” требуется занести конфиг файлы башен.

Конфиг файл башни.

Создать его можно через путь указанный в характеристики башен.
Конфиг имеет следующий интерфейс:



Все поля интуитивно понятны, однако рассмотрим поле “Prefab” подробнее.

Магия создания префаба.

После создания объекта и вынесения его в префаб, требуется сделать следующие действия:

1)Добавить на него класс, которые наследует ITower(обязательное условие(см модуль Scripts/Buildings, папка “Intefaces”)).

2)Добавить класс GradeInfo, отвечающий за улучшение башни.

3)Добавить дочерний объект, на котором будет сферический коллидер и класс TowerCollisionController.

Улучшения башен.

После добавления класса GradeInfo появится интерфейс для настройки прокачки башни.

В нем нужно указать всю необходимую информацию, которая будет выведена в диалоговое окно улучшения и использована для грейда башни.