**Inversion of Control**

Есть модули. У каждого модуля есть адаптер.

Есть медиатор (диспетчер), через которого общаются адаптеры.

Либо иерархическая структура.

На поддержание красивой структуры уходит много времени, выгодно в долгосрочной перспективой.

Service Locator

или

System.Reflection

class Interface

{

[Bindable]

ICursor cursor;

}

Сервисы – живут «вечно» - ServiceController

SoundManager

Object Pool

UnitCreator

Input

UserData

Statistic

Объекты на сцене – RoomController

RoomDispatcher и ServiceDispatcher

CustomMonoBehavior

Вызов метода в SO

AI

Input -> CharacterController.cs->Move.cs, Attack.cs

AIController.cs -> Move.cs, Attack.cs

Хитбокс, радиус обзора, радиус атаки

Сетка карты

Свои теги

State machine

Если есть несколько If с булами – наверно надо перейти к Enum

Dictionary <Enum1, Action>

Dictionary<Enum2, Edge>

Action – вершины графа состояний, Edge – ребра графа/переходы между состояниями

Placeholder

Inventory vs Fight Mode (Коллизии управления)

Input Manager

MVC

Контроллер (Input.W) передает что то (персонаж сдвинулся) Модели. Модель указывает Виду (проиграй анимацию) и двигает трансформом.

MVP – model view presenter

MVVM - model view view model

Слои

enum – всегда нумеровать. Нужны для дублирования сцен, слоев, тэгов.

Hero – тело чувака с подвязками к скриптам.

В инспекторе есть скрипт Attack.

Health

Death

RayCastAll – возвращает все задетые объекты

Bullet b = Instantiate(bullet) as Bullet;

b.init(direction, speed, ()=>{MyBulletHit} );

Пулю передавать не как GO, а как AbstractBullet, чтоб не падало приведение типов.

Пистолет не знает о том, попала ли пуля. Но она может его уведомить через MyBulletHit.

Триггерная пуля не даст координаты попадания.

Collision дает точку попадания и нормаль.

Пулей можно попасть в себя

Объект Settings на сцене. Все типы атак – в список и выбирать их по enum.

**Scriptable Object**

TODO

RayCastHit Attack

Projectile Attack

Sword Attack

Forward Move

Parable

Homing

Bouncing

Добавить на карту небо, облака, закат.

Обсчет камеры – в LateUpdate

Класс Debug (Log, LogWarning, LogError)

Break();

Debug как мод Инспектора (справа вверху) показывает даже приватные переменные.

[SerilizedField] показывает приватную переменную в Инспекторе.

SOLID

КОП

Структура папок

Assets

Data

Models

Textures

Pictures

Icons

Materials

Resources

Scenes

Scripts

Shaders

Plugins