Script architecture

# Managers:

**GameController**

+GameStates : Begin, Combat, Win, Lose

+SwitchState(GameState)

**PlayerInput:** post events by delegate, ShipMove, PlayerLoot, ShipShot will receive it

# Entity : base class for others

\*Basic properties and behaviours that Enemies can have

-IMove : Move(Vector3)

-Properties: Health, Armor, team

-Behaviour: GetDamage(amount), GetHealth(amount)

Player -> Entity, Building system, SupportShip

+Move : mouse, joystick

+Shot : left mouse, finger tapping

+Ulti : key U, mobile button

**ShipMove** : convert Screen position to WorldSpace position and pass it to Transform.

**PlayerLoot(old : ItemLooter) :** pick the Item near Player up and ship it to Inventory

**ShipShot:** command all SupportShips and ShipComposite to shot

Building system -> Weapon:

**ShipComposite**: m\*n array containing ShipModules

**ShipModule**

ShipComponent:

# SupportShip -> Weapon:

**Move** : follow Player and Enemies

**Shot** : SupportShips shot if the bullet path intersects with Enemies

# Enemy -> Entity, Weapon

Move : follow the Orbit

Shot :

# Item:

ItemCode, ItemType

ItemProfileSO

ItemAbstract

ItemControl

# Inventory

InventoryManager -> Inventory, InventorySlot, ItemInventory

Inventory -> ItemInventory

ItemInventory: Contain info about item

InventorySlot: Use to render item information on Slot

# Weapon -> Bullet

WeaponSO : texture, damage, firerate, ammor penetration

GunShot: create Bullet

LaserShot: launch a Laser

# Bullet -> Entity

**Movement**: Move -> IMove, WormShapeMove -> IMove

**Explode** :

+Deal damage and self-destruct if they collide with targets

+Give the Bullet ability to deal AOE damage

# SupportShip (Pet)

bool check : check if Pets detect entity name enemy ( default false )

**RayCast :** 2 type -> arc type, line type

+ line type ( help spShip adjacent MainShip ) : TargetDistance, AllowedDistance, GameObject followPoint1, followPoint2, RayOb, SupportShip, float followSpeed, obstacleRayDistance

+ arc type (help spShip detect enemy )

# FrameWork : good tools for UwU

**LoadScene** : get string as parameter and load a new scene with the given string

**Timer** : handle timing tasks

**ObjectPool** : Destroying and Creating objects make the game laggy, A pool of pre-created Object will solve this problem

**StrobeEffect**