Weapon System 3.2

The Weapons Template with Pooling System

Complete weapon system with features and weapon scripts template

Complete Weapons Included

- Prism Gun
- Homing Missile
- Machine Gun
- Storm Rocket
- Cannon
- Fragment Rocket
- Grenade Launcher
- Shotgun

How to add the weapons

This sample will show you how to install the weapon to your game object

1. create Empy Game Object named **GunHandle**
2. Add WeaponController.cs to **GunHandle**
3. Select Launcher from **Prefabs/Weapons/Launcher** and attached to **GunHandle** as a child
4. The Launcher will added to **WeaponController** automatically

How to access the weapons

you can access all parameters and shooting by follow this step

1. create new scripts named **GunHanddle.cs**
2. declare variable public **WeaponController** weapon;
3. get **WeaponController** variable under it child by finding component

```
Start(){
        weapon = this.transform.GetComponentInChildren
        (typeof(WeaponController)).GetComponent<WeaponController> ();
}
```

4. Shooting test

```
Update(){
        if (Input.GetButton ("Fire1")) {
                if (weapon)
                weapon.LaunchWeapon ();
        }
}
```

5. You can access all weapon parameters by call **GetCurrentWeapon**
   e.g.

```
        weapon.GetCurrentWeapon().Ammo
        weapon.GetCurrentWeapon().AmmoMax
        weapon.GetCurrentWeapon().ReloadingProcess
```

How to Create New Weapon

The weapon included with 3 part

1. Launcher
2. Bullet
3. Combine

Part 1. Launcher

- Create empty object or using 3d mesh such as Gun model , Rocket Launcher etc...
- Add WeaponLauncher.cs Component
- Edit **Parameter**

  - **Target Tag** is tag of target e.g.. Enemy , Player
  - **Riggidbody Projectile** set True the gun will shoot with AddForce()
  - **Torque Speed Axis** Rotation Axis of Torque object
  - **Torque Object** using to rotation when shooting
  - **Missile Outer** position of bullet shoot out
  - **Missile** is Bullet Prefeb
  - **Fire Rate**
  - **Spread** is how the gun accuracy 0 is very stable
  - **Force Shoot** shooting force of this gun
  - **Num Bullet** is num of bullet per shot
  - **Ammo**
  - **Ammo Max**
  - **Infinity Ammo**
  - **Time Reload**
  - **Show HUD** set True will show indicator of target seeker on screen
  - **TargetLockOnTexture**
  - **TargetLockedTexture**
  - **Distance Lock** Distance between weapon and target are posible to lock
  - **Time to Lock** Duration time before locked
  - **Aim Drection** 0 - 1 ,0 can lock every whare around weapon.
  - **Seeker** enable / dusable Target Seeker
  - **Shell** shell prefab
  - **ShellOuter** position shell spawning
  - **ShellOutForce** is sheel out force
  - **Muzzle** muzzle effect object
  - **Muzzle Life Time**
  - **Sound** Gun
  - **Sound Reloading** sound when start relload
  - **Sound Reloaded** sound when reload finish

Part 2. Bullet you have 2 part

1. Mover is type of bullet move e.g.. Missile using MoverMissile.cs , Normal bullet with projectile using MoverBullet.cs
2. Damage using to be warhead or type of bullet damage you can also add any scripts or effect to this bullet

- Create empty object or use 3d mesh such as Bulelt model , Rocket model
- Add MoverBullet.cs or MoverMissile.cs to the object
- Edit Parameter of Mover Component
- Add Damage.cs Component

- **Edit Parameter**
  - **Effect** , ex. Explosion particle
  - **Damage** is Damage point
  - **Target Tag** can be null
  - **Explosive** Enable/Disable Explosion damage
  - **Explosion Radius**
  - **Explosion Force** is force of explosion
  - **Hited Actice**
  - **Time Active** time count down to active damage using to create grenade

Part 3 Combine

- Add bullet prefeb to Launcher at **'Missile'** Parameter
- Call gameObject.GetComponent<**WeaponLauncher**>().Shoot();

How to use Pooling System

- Place **WeaponSystem** prefab to every scene.
- Instantiate object by Use WeaponSystem.Pool.Instantiate instead of GameObject.Instantiate
- Must have **Object Pool** component in all instantiate object or prefab.
- Destroy game object with this.GetComponent<ObjectPool> ().Destroying(); instead of GameObject.Destroy();

Thank you.

if you have any quesion of need an assistance feel free and let me know
Email. hwrstudio@gmail.com
Rachan Neamprasert