

ELIOT INTEGRATION



The following document describes the additional steps that have to be done manually by the User to achieve an integration between Eliot and one of the listed third party assets. Note that some of the integrated third party assets require no additional actions from the User. This document only features those that do require any action.

Navigation

1. Opsive UCC
2. Invector Character Controllers
3. Two Cubes Game Kit Controller
4. UFPS
5. Ootii Third Person Motion Controller
6. General Notes

Opsive UCC

To make sure Opsive system does not crush when an Eliot Agent's gameObject is removed upon "death", open the RecoilAnimatorAudioStateSelector.cs script. Find the IsBlocked() method and add the following line, as it is displayed on the following screenshot:

```
if(!m_HitColliders[i]) continue;
```

```
/// <summary>
/// Is the item currently being blocked by an object that should cause recoil?
/// </summary>
/// <returns>True if the item is currently being blocked by an object that should cause recoil.</returns>
private bool IsBlocked()
{
    for (int i = 0; i < m_HitColliderCount; ++i) {
        ShieldCollider shieldCollider;
        if (!m_HitColliders[i]) continue;
        var hitGameObject = m_HitColliders[i].gameObject;
        if ((shieldCollider = hitGameObject.GetCachedComponent<ShieldCollider>()) != null) {
            if (shieldCollider.Shield.CurrentDurability > 0) {
                return true;
            }
        } else if (hitGameObject.GetCachedComponent<RecoilObject>() != null) {
            return true;
        }
    }
    return false;
}
```

Investor Character Controllers

To make an Investor character damage an Eliot Agent open the vMeleeAttackObject.cs script. Find the ApplyDamage() method and add the following lines, as it is displayed on the following screenshot:

```
var eliotAgent = other.GetComponent<Eliot.AgentComponents.Agent>();  
if (eliotAgent)  
    eliotAgent.Damage(damage.damageValue);
```



```
/// <summary>  
/// Apply damage to target collider (TakeDamage, damage)  
/// </summary>  
/// <param name="hitBox">vHitBox object</param>  
/// <param name="other">collider target</param>  
/// <param name="damage"> damage</param>  
public void ApplyDamage(vHitBox hitBox, Collider other, vDamage damage)  
{  
    vDamage _damage = new vDamage(damage);  
    _damage.receiver = other.transform;  
    _damage.damageValue = (int)Mathf.RoundToInt((((float)(damage.damageValue + damageModifier) * (((float)hitBox.damagePercentage) * 0.01f))));  
    _damage.hitPosition = hitBox.transform.position;  
    other.gameObject.ApplyDamage(_damage, meleeManager.fighter);  
  
    var eliotAgent = other.GetComponent<Eliot.AgentComponents.Agent>();  
    if (eliotAgent)  
        eliotAgent.Damage(damage.damageValue);  
}
```

Two Cubes Game Kit Controller

To make a GKC character damage an Eliot Agent open the applyDamage.cs script. Find the checkHealth() method and add the following lines, as it is displayed on the following screenshot:

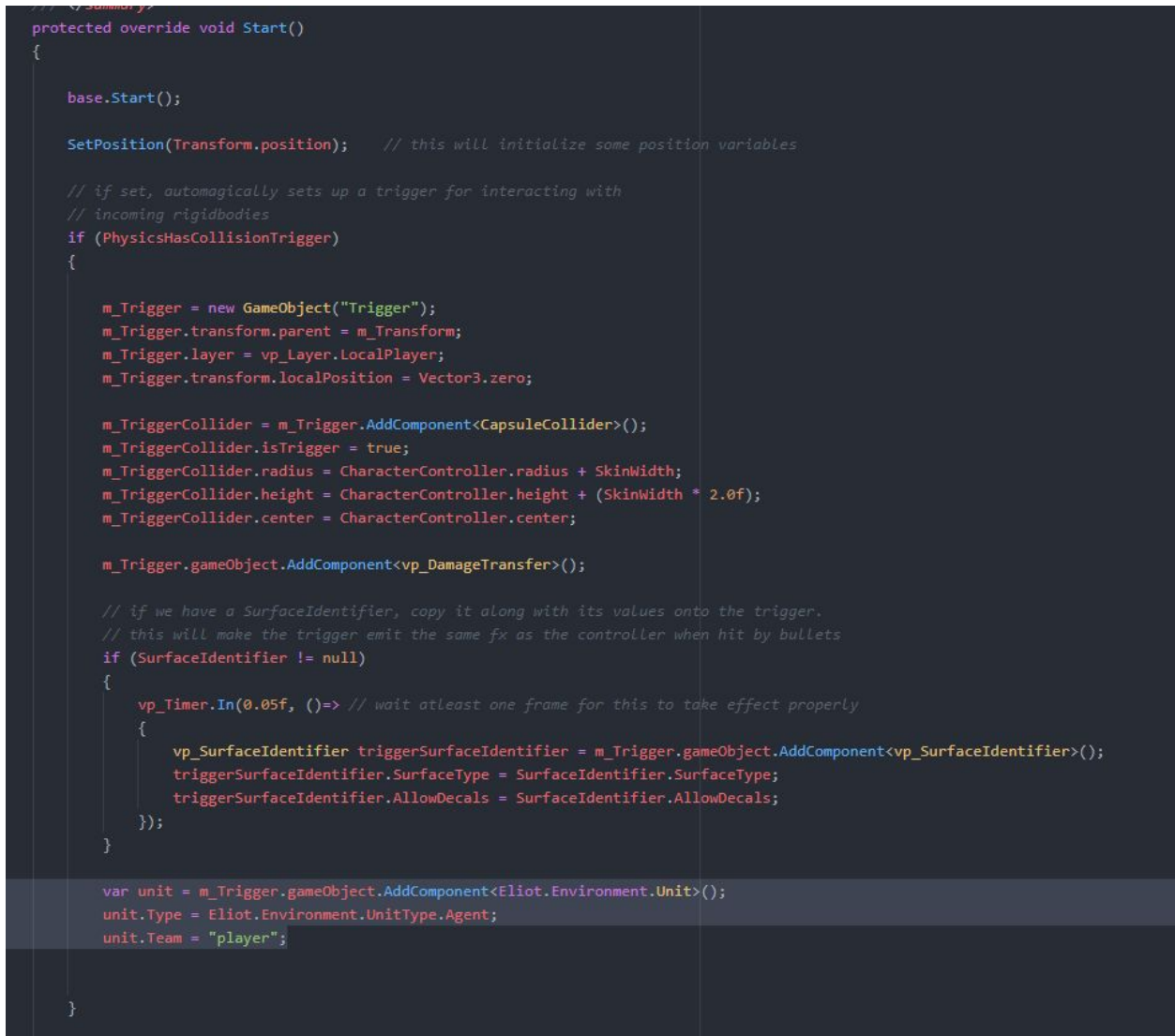
```
var eliotAgent =  
objectToDamage.GetComponent<Eliot.AgentComponents.Agent>();  
if (eliotAgent)  
{  
    eliotAgent.Damage(damageAmount);  
    eliotAgent.Motion.Push(position, damageAmount*0.1f);  
}
```

```
//also, check if the object is a vehicle to apply damage too  
public static void checkHealth (GameObject projectile, GameObject objectToDamage, float damageAmount, Vector3 direction, Vector3 position, GameObject projectileOwner,  
                                bool damageConstant, bool searchClosestWeakSpot)  
{  
    characterDamageReceiver characterDamageReceiverToCheck = objectToDamage.GetComponent<characterDamageReceiver> ();  
    if (characterDamageReceiverToCheck) {  
        characterDamageReceiverToCheck.setDamage (damageAmount, direction, position, projectileOwner, projectile, damageConstant, searchClosestWeakSpot);  
        return;  
    }  
  
    vehicleDamageReceiver vehicleDamageReceiverToCheck = objectToDamage.GetComponent<vehicleDamageReceiver> ();  
    if (vehicleDamageReceiverToCheck) {  
        vehicleDamageReceiverToCheck.setDamage (damageAmount, direction, position, projectileOwner, projectile, damageConstant, searchClosestWeakSpot);  
        return;  
    }  
  
    vehicleHUDManager vehicleHUDManagerToCheck = objectToDamage.GetComponent<vehicleHUDManager> ();  
    if (vehicleHUDManagerToCheck) {  
        vehicleHUDManagerToCheck.setDamage (damageAmount, direction, position, projectileOwner, projectile, damageConstant, searchClosestWeakSpot);  
        return;  
    }  
  
    health healthToCheck = objectToDamage.GetComponent<health> ();  
    if (healthToCheck) {  
        healthToCheck.setDamage (damageAmount, direction, position, projectileOwner, projectile, damageConstant, searchClosestWeakSpot);  
        return;  
    }  
  
    // Emerald_AI Emerald_AIToCheck = objectToDamage.GetComponent<Emerald_AI> ();  
    // if (Emerald_AIToCheck) {  
    //     Emerald_AIToCheck.Damage ((int)damageAmount, Emerald_AI.TargetType.Player);  
    // }  
    //  
    // if ELIOT  
    var eliotAgent = objectToDamage.GetComponent<Eliot.AgentComponents.Agent>();  
    if (eliotAgent)  
    {  
        eliotAgent.Damage(damageAmount);  
        eliotAgent.Motion.Push(position, damageAmount*0.1f);  
    }  
}
```

UFPS

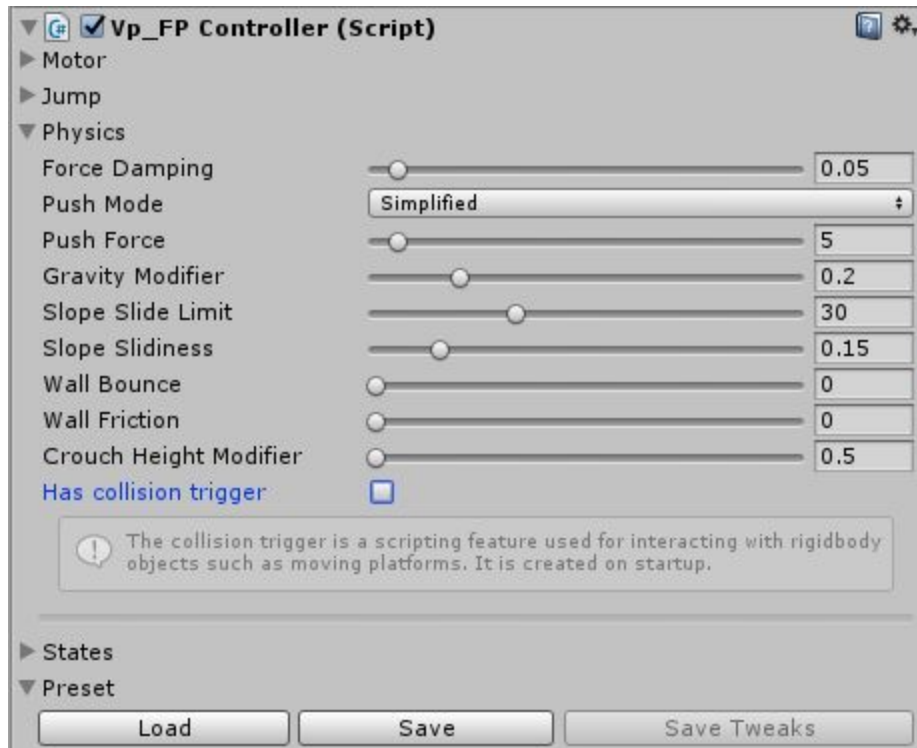
To make sure that UFPS character is detectable by an Eliot Agent open the vp_FPController.cs script. Find the Start() method and add the following lines, as it is displayed on the following screenshot:

```
var unit =  
m_Trigger.gameObject.AddComponent<Eliot.Environment.Unit>();  
unit.Type = Eliot.Environment.UnitType.Agent;  
unit.Team = "player";
```

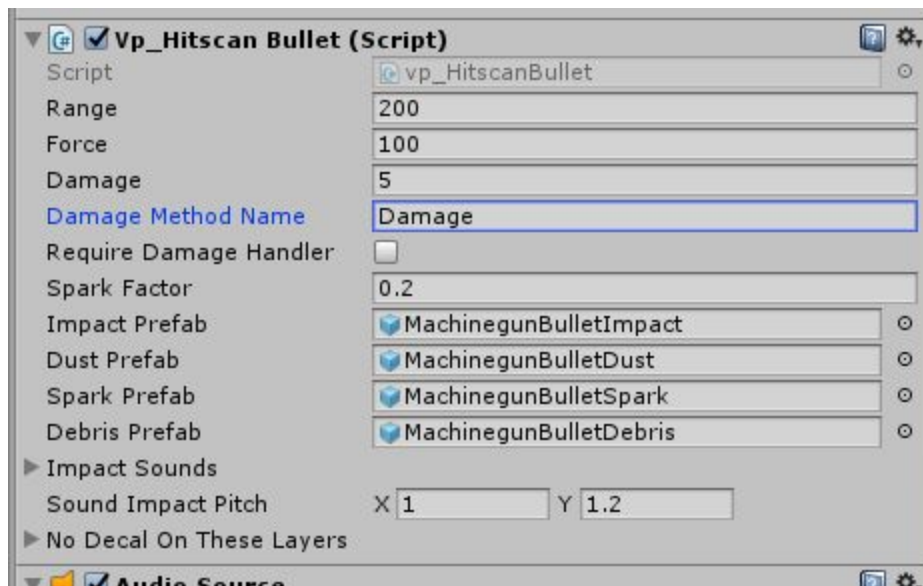


```
protected override void Start()  
{  
    base.Start();  
  
    SetPosition(Transform.position);    // this will initialize some position variables  
  
    // if set, automatically sets up a trigger for interacting with  
    // incoming rigidbodies  
    if (PhysicsHasCollisionTrigger)  
    {  
        m_Trigger = new GameObject("Trigger");  
        m_Trigger.transform.parent = m_Transform;  
        m_Trigger.layer = vp_Layer.LocalPlayer;  
        m_Trigger.transform.localPosition = Vector3.zero;  
  
        m_TriggerCollider = m_Trigger.AddComponent<CapsuleCollider>();  
        m_TriggerCollider.isTrigger = true;  
        m_TriggerCollider.radius = CharacterController.radius + SkinWidth;  
        m_TriggerCollider.height = CharacterController.height + (SkinWidth * 2.0f);  
        m_TriggerCollider.center = CharacterController.center;  
  
        m_Trigger.gameObject.AddComponent<vp_DamageTransfer>();  
  
        // if we have a SurfaceIdentifier, copy it along with its values onto the trigger.  
        // this will make the trigger emit the same fx as the controller when hit by bullets  
        if (SurfaceIdentifier != null)  
        {  
            vp_Timer.In(0.05f, ()=> // wait atleast one frame for this to take effect properly  
            {  
                vp_SurfaceIdentifier triggerSurfaceIdentifier = m_Trigger.gameObject.AddComponent<vp_SurfaceIdentifier>();  
                triggerSurfaceIdentifier.SurfaceType = SurfaceIdentifier.SurfaceType;  
                triggerSurfaceIdentifier.AllowDecals = SurfaceIdentifier.AllowDecals;  
            });  
        }  
  
        var unit = m_Trigger.gameObject.AddComponent<Eliot.Environment.Unit>();  
        unit.Type = Eliot.Environment.UnitType.Agent;  
        unit.Team = "player";  
    }  
}
```

The other way around this is to uncheck the “Has collision trigger” toggle in the Physics section of the player’s vp_FpController component in the Inspector:



To let UFPS character can damage an Eliot Agent, make sure that whatever Projectile the UFPS character uses, its “Damage Method Name” is set to “Damage”:



Ootii Third Person Motion Controller

To make an Ootii TPMC character damage an Eliot Agent open the WeaponCore.cs script (ArrowCore.cs for ranged attack). Find the following line using ctrl+f:

```
lRigidBody.AddForceAtPosition(rHitInfo.Vector * lMessage.ImpactPower,  
rHitInfo.Point, ForceMode.Impulse);
```

This line is embedded inside an if statement. Insert the following code right after that if statement:

```
var eliotAgent =  
rHitInfo.Collider.gameObject.GetComponent<Eliot.AgentComponents.Agent>();  
if (eliotAgent)  
{  
    eliotAgent.Damage(damageAmount);  
}
```

General Notes

Whenever you want to make any object, like player controller, detectable by Eliot agents, simply add the Unit component to them. For more details about the usage of the Unit component, see manual: www.eliot-ai.com/learn