

# EMERALD INTEGRATION

Remember to check out this video for the settings in the inspector, once that is done, you can follow the steps above:

[https://www.youtube.com/watch?v=miCpkYpyZ\\_E](https://www.youtube.com/watch?v=miCpkYpyZ_E)

-Add this code in the script **EmeraldAIPlayerDamage.cs**:

```
void DamageGameKitController(int DamageAmount, GameObject attacker)
{
    applyDamage.checkToDamageGKCCharacterExternally (DamageAmount,
    gameObject, attacker);
}
```

-And in that same script, add this code inside **SendPlayerDamage** function:

```
GameObject targetObject = null;
if (Target != null){
    targetObject = Target.gameObject;
}

DamageGameKitController(DamageAmount, targetObject);
```

-Add the following code in the script **EmeraldAISystem.cs**, after the public class { line:

```
public override void setDamageWithHealthManagement(float damageAmount,
Vector3 fromDirection, Vector3 damagePos, GameObject attacker, GameObject projectile,
bool damageConstant,
bool searchClosestWeakSpot, bool ignoreShield, bool
ignoreDamageInScreen, bool damageCanBeBlocked, bool
canActivateReactionSystemTemporally, int damageReactionID, int damageTypeID)
{
    Damage((int)damageAmount, EmeraldAISystem.TargetType.Player);
}
```

-And in that same script, replace **MonoBehaviour** word for **healthManagement** at the beginning of that same script, **EmeraldAISystem.cs**:

```
18 [RequireComponent(typeof(LimitDamageController))]
19 [SelectionBase]
20
21 @ UnityScript | 99+ references
22 public class EmeraldAISystem : MonoBehaviour
23 {
24     0 references
25     public override void setDamageWithHealthManagement(float damageAmount,
26     Vector3 fromDirection, Vector3 damagePos, GameObject attacker, GameObject projectile,
27     bool damageConstant,
28     bool searchClosestWeakSpot, bool ignoreShield, bool
29     ignoreDamageInScreen, bool damageCanBeBlocked, bool
30     canActivateReactionSystemTemporarily, int damageReactionID, int damageTypeID)
31     {
32         Damage((int)damageAmount, EmeraldAISystem.TargetType.Player);
33     }
34     #region Variables
35     public float ForceWalkDistance = 2.5f;
36     public bool LockTurning;
37     //3.0 Variables
38     public bool NotifiedOfNewVersion = false;
39     float TimeSinceStart;
40     public bool ReachedDestination;
41     float DelayTimer;
42     Coroutine SwitchWeaponCoroutine;
43     6 references
44     public enum SwitchWeaponTypes { Distance, Timed};
45     public SwitchWeaponTypes SwitchWeaponType = SwitchWeaponTypes.Timed;
46     public int SwitchWeaponTimeMin = 10;
```

