**EvilVeggie UML**

*<<Abstract>>*

**EvilVeggie**

# [SerializeField] evilness: enum (mild/tough)

# healthPoints : float

# damage : float

# abstract SetHealth() : void

# abstract SetDamage() : void

+ abstract Move() : void

+ DestroyIfOutOfSight() : void

+ RenderEvilnessEffectIfTough () : void *(if tough then red point light, else return;)*

+ TakeDamage(GameObject, damage) : void

Events:

+ static onEvilVeggieDeath: se suscribe onAwardPointsHandler de GameManager (le da 5 o 10 puntos al player dependiendo de si es mild o tough)

*<<Singleton>>*

**EvilVeggieStats**

- [SerializeField] healthPointsMild : Vector3<float>

- [SerializeField] damagePointsMild : Vector3<float>

MILD

|  |  |
| --- | --- |
| healthPoints | 1|1|1 |
| damage | 3|4|5 |

m|t|c

TOUGH values = twice MILD values

m = mushroom

t = tomato

c = carrot

**EvilMushroom**

+ override Move() : void *(empty method)*

+ override SetInitialHealth() : void

+ override SetDamage() : void

**EvilCarrot**

+ override Move() : void

+ override SetInitialHealth() : void

+ override SetDamage() : void

**EvilTomato**

+ override Move() : void

+ override SetInitialHealth() : void

+ override SetDamage() : void