

Bug War

Coding Document

Versión: 1.1

Escrito por Edgar Uriel Herrera Franco (HappierGore)

ING. Mecatrónico con diplomado en desarrollo de videojuegos

Scripts: Mobs

Obligatorios

MobAnimManager:

Permite controlar las animaciones de ataque, movimiento y parado. Estas animaciones trabajan con eventos por lo que en cada frame de la animación debe tener las siguientes funciones:

- Idle, Frame 0: SetAttackState(false)
- Attack, Frame exacto donde atacará: SetDealDamage(true)
- Attack Frame final: SetAttackState(false)
- Attack Frame final: SetEndAttackFrame(true)

MobAttack:

Define diversos valores para el ataque

- attackSpeed: Define la velocidad de ataque en segundos, si 1.0 es definido, cada segundo, después de terminada la animación de ataque, volverá a atacar.
- damage: Definirá la cantidad de daño que ocasionará a su objetivo

MovEvents:

Aquí se almacenan todos los eventos y banderas del mob en cuestión. Estos nos ayudarán a la hora de programar nuevos comportamientos y corregir posibles bugs.

Variables:

- MobFlags: Define las banderas comúnmente usadas
 - moving: Define si el mob se está moviendo (Normalmente, se activa si no tiene objetivos a los que atacar)
 - attackedTarget: Lanza una pequeña señal al momento de que se ataca, es utilizado para los eventos del script *HabilityManager* en la parte "ExecuteHabilities"
- Targets Events: Eventos simples relacionados con el objetivo
 - reachedTarget: Lanza una pequeña señal al alcanzar a su objetivo
 - killTarget: Lanza una pequeña señal al asesinar un objetivo
- HealthEvents: Eventos relacionados con la vida del propio mob
 - damaged: Lanza una pequeña señal al ser dañado por algún enemigo
 - died: Lanza una pequeña señal al morir
- Anim Events: Eventos de animación, recomendable no cambiar esos valores
 - endAttackFrame: Se activa cuando la animación de ataque del mob ha terminado
 - attack: Se activa durante el tiempo que dura la animación de ataque
 - dealDamage: Se activa en el frame exacto en donde el mob hará el daño a su objetivo

MobMovement:

Este script controla el movimiento del mob en cuestión

Variables:

- *enemyCastle & ownCastle: Obtiene los castillos correspondientes*
- *started: bandera que nos permitirá obtener los componentes anteriores una sola vez*

Funciones:

- *Move(MobStats stats): Esta función es la encargada de mover al mob por el escenario, es recomendable no realizar cambios a menos que sea necesario, ya que los eventos que se utilizan aquí tienen un impacto en todos los eventos restantes.*
- *Range(MobStats stats): Define el rango de ataque del mob, esto tiene un impacto directo en el movimiento, recomendable no cambiar nada más que valores de rango.*

MobStats:

Establece las estadísticas y variables que influyen en el comportamiento del mob con el entorno y sus enemigos, es el corazón del mob.

Variables:

- *Mode: Define el tipo de ataque, a distancia o a cuerpo a cuerpo*
- *Range: Rango de ataque, aplica para cada modo (Véase función "Range" de *MobMovement*)*
- *health: Define la cantidad de salud que el mob tiene*
- *speed: Define la velocidad de movimiento, cuanto más grande el valor más rápido se moverá*
- *isAlly: Variable de ayuda, definirá si el mob en cuestión es aliado o no*
- *foodCost: Definirá el costo de el mob en cuestión.*
- *target: Variable de ayuda, almacenará el objetivo del mob*
- *mobEvents: Instancia y crea una variable de tipo MobEvents*

Funciones:

Las variables Mode, Range, Health y Speed tienen funciones de tipo Get

Corutinas:

- *TakeDamage(float damageTaken): Esta corutina reducirá la cantidad de vida en "damageTaken". Dentro de esta corutina se manipulan eventos como: Died y Damaged.*

MobTargetSelection

Obtendrá mediante ciertos cálculos el objetivo más cercano al mob en cuestión, sea mob enemigo o el castillo enemigo. Este script parece ya estar listo y no requiere modificaciones.

Funciones:

- *DetectTarget(MobStats stats):* Obtiene el objetivo más cercano.
- *DistanceP1P2(float P1, float P2):* Obtiene la distancia que hay entre dos puntos

Opcionales pero muy recomendados:

NormalAttack:

Establecerá que el mob en cuestión tendrá un ataque de tipo unitario, es decir, sólo podrá atacar un mob a la vez. Esta script puede presentar algunos bugs, se está constantemente actualizando, pero por ahora, es estable.

Corutinas:

- *Attack(MobStats stats, MobEvents mobEvents, mobAttack mobAttack):* Este algoritmo controla la forma en la que el ataque funciona, aquí se manipulan variados eventos y se trabajan con ellos como condicionales, entre estos están *dealDamage*, *reachedTarget*, *endAttackFrame*, *attack*, *attackedTarget*.

Funciones:

- *CheckHabilitiesManager():* Ejecuta desde el script **HabilityManager** esta función, la cual se explica en el script en cuestión.
- *ChooseRoutineDamage(MobStats stats, MobAttack mobAttack):* Seleccionará el tipo de objetivo al que el mob ataca.

HabilityManager:

Administra las habilidades que puede tener un mob en cuestión, este script es NECESARIO si es que se añaden habilidades como KnockBack, MoreDamage, y otros.

Variables:

- *knockBack:* Creamos la variable que almacenará la habilidad si es que existe
- *executingHabilities:* Variable de ayuda, permitirá ejecutar una sola vez la corutina siguiente.

Corutinas:

- *ExecuteHabilities():* Obtendrá (Si es que existen) todas las habilidades que un mob podría tener, para luego, ser ejecutadas en el instante en el que el evento *dealDamage* es activado, es decir, el ataque y las habilidades están en sincronía gracias a esta corutina.

Funciones:

- *KnockBackExecute(): Básicamente, ejecuta la rutina "KnockBackHability" desde el script *KnockBack*.*

KnockBack:

Este script básicamente hará que el mob en cuestión ejecute dicha habilidad.

Variables:

- *level: Definirá la potencia de retroceso de la habilidad*
- *chancePercent: Definirá el porcentaje de probabilidad (0-100) en la que esta habilidad podrá ser activada*
- *alreadyAttaking: Bandera de apoyo para ejecutar una sola vez la corutina*
- *targetType: Si el mob atacado es de un tipo en específico se le ejecutará la habilidad*
- *mobEvents: Eventos del mob en cuestión*

Corutinas:

- *KnockBackHability(): Tras evaluar el porcentaje y evaluar el evento attackedTarget del mob en cuestión, se ejecutará la habilidad hacia el objetivo*

SimpleBehaviour:

Este script define el comportamiento básico de alguna entidad, estableciendo dentro el movimiento, ataque y el tipo de ataque. El resto de scripts harán el trabajo por medios externos. El evento died hará que el objeto del mob en cuestión sea destruido, y, el evento reachedTarget realizará el ataque correspondiente del mob (Normal o de área).