

# The Legend of Link

David Antuña   Jaime Bas   Irene González   José Luis Moreno

DVI  
Universidad Complutense de Madrid

Curso 2017-18



## 1 aiTrack

- ¿Qué es?
- Propiedades
- Funcionamiento
- Triggers
- Demostración

## 2 Componente stepControls

- Problema
- Solución



## 1 aiTrack

- ¿Qué es?
- Propiedades
- Funcionamiento
- Triggers
- Demostración

## 2 Componente stepControls

- Problema
- Solución



# ¿Qué es?

Es un nuevo componente para la librería 2d que otorga a un sprite la capacidad de localizar, seguir y atacar a otro(s).

Está basado en cuadrícula, por lo que solo sirve para 2d con vista cenital.



## 1 aiTrack

- ¿Qué es?
- **Propiedades**
- Funcionamiento
- Triggers
- Demostración

## 2 Componente stepControls

- Problema
- Solución



- *view\_range* y *attack\_range*: Rango al cual se activará el movimiento o ataque
- *tile\_size*: Tamaño de un tile de la cuadrícula
- *vfactor*: Multiplicador de velocidad  
 $v = \text{tile\_size} * \text{vfactor}$
- *attacking* y *tracking*: Indica si el sprite está atacando o persiguiendo a un enemigo
- *reloadSpeed*: Tiempo de enfriamiento de un ataque
- *reload*: Enfriamiento actual
- *track\_class*: Lista de clases con las que el sprite interactuará



## 1 aiTrack

- ¿Qué es?
- Propiedades
- **Funcionamiento**
- Triggers
- Demostración

## 2 Componente stepControls

- Problema
- Solución



En cada iteración del bucle de juego el modulo Scenes activará la señal **range.check** para todos los componentes de la escena.

El componente escucha esta señal y comprueba la distancia entre la entidad y los elementos de la escena cuya clase esté en la lista.

Hay tres posibilidades:

- No está en rango → No ocurre nada
- Está en rango de ataque → Activa la señal **range.attack**
- Está en rango de movimiento → Activa la señal **range.view**





## 1 aiTrack

- ¿Qué es?
- Propiedades
- Funcionamiento
- **Triggers**
- Demostración

## 2 Componente stepControls

- Problema
- Solución



Esta señal activa la función attack del componente.

Si reload es menor o igual que 0.

- Activa el atributo attacking
- Resetea el tiempo de enfriamiento
- Invoca la función hit y le pasa el atributo damage de la entidad



Esta señal activa la función move del componente.

Esta función trata de equiparar el componente x e y de la entidad a los del elemento que esta siguiendo.

$$vx = \pm \text{tile\_size} * \text{vfactor}$$

$$vy = \pm \text{tile\_size} * \text{vfactor}$$



## 1 aiTrack

- ¿Qué es?
- Propiedades
- Funcionamiento
- Triggers
- Demostración

## 2 Componente stepControls

- Problema
- Solución



# Demostración



## 1 aiTrack

- ¿Qué es?
- Propiedades
- Funcionamiento
- Triggers
- Demostración

## 2 Componente stepControls

- Problema
- Solución



Para que el sprite se pueda mover en las 4 direcciones usamos el componente de Quintus stepControls pero genera clipping al combinarse con nuestro aiTrack.

El componente comprueba si la entidad colisiona y de hacerlo la devuelve al origen, el problema es que los enemigos te persiguen y este componente impide que el sprite escape.



```
collision: function(col) {  
    var p = this.entity.p;  
  
    if (p.stepping) {  
        p.stepping = false;  
        p.x = p.origX;  
        p.y = p.origY;  
    }  
},
```





## 1 aiTrack

- ¿Qué es?
- Propiedades
- Funcionamiento
- Triggers
- Demostración

## 2 Componente stepControls

- Problema
- Solución



# Problema

Para solucionar este problema nos aprovechamos de un atributo de los sprites en los que usamos este componente, todos tienen el atributo `direction`.

Gracias a dicho atributo sabemos en que dirección se esta moviendo por lo que podemos modificar el código para que se resetee la posicion solo si se mueve hacia el objeto.



```
collision: function(col) {  
    var p = this.entity.p;  
  
    if (  
        p.stepping &&  
        ((p.direction === 'up' && col.normalY === 1) ||  
         (p.direction === 'down' && col.normalY === -1) ||  
         (p.direction === 'left' && col.normalX === 1) ||  
         (p.direction === 'right' && col.normalX === -1))  
    ) {  
        p.stepping = false;  
        p.x = p.origX;  
        p.y = p.origY;  
    }  
},
```

