

The Legend of Link

David Antuña Jaime Bas Irene González José Luis Moreno

DVI
Universidad Complutense de Madrid

Curso 2017-18



- 1 Introducción al juego
- 2 Mecánicas
- 3 Arquitectura de las clases
 - Personajes y objetos inanimados
 - Escenario y pantallas
- 4 Personajes
 - Personaje principal
 - Enemigos
- 5 Objetos
- 6 aiTrack

- ¿Qué es?
 - Propiedades
 - Funcionamiento
 - Triggers
- 7 Componente stepControls
 - Problema
 - Solución
 - 8 Puzzles
 - 9 Postmortem
 - ¿Qué nos ha ido bien?
 - ¿Qué nos ha ido mal o se podría mejorar?



1 Introducción al juego

2 Mecánicas

3 Arquitectura de las clases

- Personajes y objetos inanimados
- Escenario y pantallas

4 Personajes

- Personaje principal
- Enemigos

5 Objetos

6 aiTrack

- ¿Qué es?

- Propiedades

- Funcionamiento

- Triggers

7 Componente stepControls

- Problema

- Solución

8 Puzzles

9 Postmortem

- ¿Qué nos ha ido bien?

- ¿Qué nos ha ido mal o se podría mejorar?



Introducción al juego

- Basado en la saga The Legend of Zelda (Nintendo, 1986).
- Assets de Four Swords y Minish Cap.
- Simplificado, centrado en combates en una mazmorra sencilla.
- Victoria: derrotar al jefe final.
- Derrota: superado límite de impactos.



1 Introducción al juego

2 Mecánicas

3 Arquitectura de las clases

- Personajes y objetos inanimados
- Escenario y pantallas

4 Personajes

- Personaje principal
- Enemigos

5 Objetos

6 aiTrack

• ¿Qué es?

• Propiedades

• Funcionamiento

• Triggers

7 Componente stepControls

• Problema

• Solución

8 Puzzles

9 Postmortem

• ¿Qué nos ha ido bien?

• ¿Qué nos ha ido mal o se podría mejorar?



- Libertad de movimiento.
- Combate.
- Salud.
- Interfaz.
- Tracking.
- Puzzles.
- Interacción con el escenario.



1 Introducción al juego

2 Mecánicas

3 Arquitectura de las clases

- Personajes y objetos inanimados
- Escenario y pantallas

4 Personajes

- Personaje principal
- Enemigos

5 Objetos

6 aiTrack

• ¿Qué es?

• Propiedades

• Funcionamiento

• Triggers

7 Componente stepControls

• Problema

• Solución

8 Puzzles

9 Postmortem

• ¿Qué nos ha ido bien?

• ¿Qué nos ha ido mal o se podría mejorar?



Personajes y objetos inanimados

- Link y enemigos.
- Basados en Sprite.
- Múltiples parámetros propios de cada personaje.
- Cofres y rupias.



Escenario y pantallas

- Mapa único con zonas de transición.
- Varias habitaciones.
- Menú principal.
- Pantalla de créditos.



Contenidos

- 1 Introducción al juego
- 2 Mecánicas
- 3 Arquitectura de las clases
 - Personajes y objetos inanimados
 - Escenario y pantallas
- 4 Personajes
 - Personaje principal
 - Enemigos
- 5 Objetos
- 6 aiTrack

- ¿Qué es?
 - Propiedades
 - Funcionamiento
 - Triggers
- 7 Componente stepControls
 - Problema
 - Solución
 - 8 Puzzles
 - 9 Postmortem
 - ¿Qué nos ha ido bien?
 - ¿Qué nos ha ido mal o se podría mejorar?



- Mecánicas.
 - Interactuar con objetos.
 - Atacar enemigos.
 - Recibir daño.
 - Movimiento.



- Darknut.
- Shadow Link.
- Varias habitaciones.
 - Persigue al jugador.
 - Ataca.
 - Recibe daño.



Contenidos

1 Introducción al juego

2 Mecánicas

3 Arquitectura de las clases

- Personajes y objetos inanimados
- Escenario y pantallas

4 Personajes

- Personaje principal
- Enemigos

5 Objetos

6 aiTrack

• ¿Qué es?

• Propiedades

• Funcionamiento

• Triggers

7 Componente stepControls

• Problema

• Solución

8 Puzzles

9 Postmortem

• ¿Qué nos ha ido bien?

• ¿Qué nos ha ido mal o se podría mejorar?



- Cofres.
 - Contienen una rupia.
 - Se abren cuando el jugador interactúa con ellos estando al lado.
- Rupias.
 - Aparecen al abrir un cofre.
 - Están animadas con tween.



Contenidos

- 1 Introducción al juego
- 2 Mecánicas
- 3 Arquitectura de las clases
 - Personajes y objetos inanimados
 - Escenario y pantallas
- 4 Personajes
 - Personaje principal
 - Enemigos
- 5 Objetos
- 6 aiTrack

- ¿Qué es?
 - Propiedades
 - Funcionamiento
 - Triggers
- 7 Componente stepControls
 - Problema
 - Solución
 - 8 Puzzles
 - 9 Postmortem
 - ¿Qué nos ha ido bien?
 - ¿Qué nos ha ido mal o se podría mejorar?



¿Qué es?

- Nuevo componente 2d.
- Otorga a un sprite capacidad para:
 - Localizar.
 - Seguir.
 - Atacar.
- Basado en cuadrícula → Solo 2d con vista cenital.



- *view_range* y *attack_range*: Rango de activación.
- *tile_size*: Tamaño de un tile de la cuadrícula.
- *vfactor*: Multiplicador de velocidad
$$v = \text{tile_size} * \text{vfactor}$$
- *attacking* y *tracking*: Indicadores de acción en curso
- *reloadSpeed*: Tiempo de enfriamiento de un ataque
- *reload*: Enfriamiento actual
- *track_class*: Lista de clases con las que el sprite interactuará



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**



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 Introducción al juego
- 2 Mecánicas
- 3 Arquitectura de las clases
 - Personajes y objetos inanimados
 - Escenario y pantallas
- 4 Personajes
 - Personaje principal
 - Enemigos
- 5 Objetos
- 6 aiTrack

- ¿Qué es?
- Propiedades
- Funcionamiento
- Triggers

7 Componente stepControls

- Problema
- Solución

8 Puzzles

9 Postmortem

- ¿Qué nos ha ido bien?
- ¿Qué nos ha ido mal o se podría mejorar?



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;  
    }  
},
```



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;  
    }  
},
```



Contenidos

1 Introducción al juego

2 Mecánicas

3 Arquitectura de las clases

- Personajes y objetos inanimados
- Escenario y pantallas

4 Personajes

- Personaje principal
- Enemigos

5 Objetos

6 aiTrack

• ¿Qué es?

• Propiedades

• Funcionamiento

• Triggers

7 Componente stepControls

• Problema

• Solución

8 Puzzles

9 Postmortem

• ¿Qué nos ha ido bien?

• ¿Qué nos ha ido mal o se podría mejorar?



- Propiedades.
 - Sprite → *activated*.
 - Stage → *label*.
- Funcionamiento.



1 Introducción al juego

2 Mecánicas

3 Arquitectura de las clases

- Personajes y objetos inanimados
- Escenario y pantallas

4 Personajes

- Personaje principal
- Enemigos

5 Objetos

6 aiTrack

- ¿Qué es?

- Propiedades

- Funcionamiento

- Triggers

7 Componente stepControls

- Problema

- Solución

8 Puzzles

9 Postmortem

- ¿Qué nos ha ido bien?

- ¿Qué nos ha ido mal o se podría mejorar?



¿Qué nos ha ido bien?

- Idea muy clara desde el principio.
- Tareas bien repartidas y trabajo simultáneo



¿Qué nos ha ido mal o se podría mejorar?

- Complicaciones con Tiled.
- Implementación del cambio de habitación.
- Proyecto demasiado ambicioso.

