

¿Qué sería una Ciudad Zombie sin zombies?

Preparación

Ya sabés cómo prepararte para programar con objetos. Lápiz y papel para diagramar, trabajar en el código de forma ordenada, etcétera, etcétera. Ya que estás, preparate psicológicamente: cuando termines esta guía, ¡terminás el juego!

Paso 1: Diagramar la herencia de los enemigos

En este paso vas a tener que diagramar a los enemigos. Entre ellos están los *zombies caminantes* y los *zombies conductores*, que manejan trenes descontrolados a toda velocidad.

Ambos toman el comportamiento de un Enemigo modificando solamente algunas cosas. Los “zombies caminantes” caminan por toda la pantalla de manera zigzagueante buscando carne fresca para alimentarse. Los “zombies conductores” tomaron el control de los trenes. Se mueven por las vías del tren de forma vertical y horizontal y atacan de manera letal al jugador. Eran trabajadores del ferrocarril que manejan muy bien los trenes, ¡así que cuidado!

Recomiendan los/as pro: *Es una buena práctica identificar cómo se relacionan los objetos que comparten responsabilidades. ¿Qué tienen en común los enemigos? ¿Qué diferencia a la forma de atacar de un zombie a la de un tren controlado por zombies? ¿Tendrán un prototipo en común?*

Paso 2: Creá a los enemigos

A partir del objeto *Enemigo* dado, tenés implementado a los “zombies caminantes”.

Deberás implementar a los “zombies conductores”.

Pista: Tené en cuenta que el zombie conductor recibe un parámetro extra al momento de ser creado: la dirección. Esto es un carácter ‘v’ o ‘h’ que indica si su movimiento es vertical u horizontal.

Si te animás, vas a poder crear infinitos tipos de Enemigos, pueden ser zombies más veloces o tal vez zombies motoqueros. ¡Que vuele tu imaginación!

Paso 3: Agregá a los enemigos al Juego

Una vez que tengas implementados a los zombies será hora de instanciarlos y mostrarlos en la pantalla. Para eso existe en el objeto Juego un arreglo de enemigos donde deberás agregar las diferentes instancias de los zombies.

Deberás:

- Agregar cinco (o más) zombies caminadores al arreglo de enemigos, con una posición inicial x,y a elección, un ancho y alto de 10 píxeles, y una velocidad y un rango de movimiento a elección (recomendamos usar el tamaño del canvas: 577x961)
- Agregar tres (o más) zombies conductores: al menos uno para la vía horizontal y otros dos para las vías verticales al arreglo enemigos.

Las posiciones iniciales de las vías son las siguiente:

Trenes	x	y	ancho	alto
vertical 1	644	0	30	90
vertical 2	678	0	30	90
horizontal	400	322	90	30

La velocidad será a elección y el rango de movimiento también. ¡No olvidés el parámetro para indicar si su movimiento es vertical u horizontal!

Dibujar los enemigos en el mapa

Una vez que agregaste a los enemigos, será hora de dibujarlos en el mapa con ayuda del objeto Dibujante. **Desde el método .dibujar() del objeto Juego tendrás que poder pintar a cada uno de los enemigos.**

Pista: Recordá que el dibujante sabe dibujar todo tipo de Entidades.

Paso 4: Moviendo a los a enemigos

Movimiento del zombie conductor

Implementá el movimiento del “zombie conductor”. Sus movimientos deben ser siempre en línea recta, dependiendo de la dirección en la que fue creado, y **deberá ir y volver** de un lado al otro del mapa todo el tiempo o bien **reaparecer en su posición original** para volver a comenzar su recorrido.

Recordá que el “zombie caminante” ya tiene implementados sus métodos y te sirve para hacer pruebas. Este zombie tiene un movimiento impredecible y depende de un número aleatorio generado por la computadora. Para esto, en su método mover, se utiliza Math.random(), que es un método para obtener un número aleatorio entre 0 (incluido) y 1 (excluido).

A mover a todos los zombies

Cada vez que se actualice el juego desde el .buclePrincipal() (ya implementada), se moverán todos los zombies. Para ello, deberás completar el método del Juego, moverEnemigos().

Pista: recordá el uso del .forEach().

¡Ahora sí que comenzó el Apocalipsis!

Recomiendan los/as pro: Antes de continuar con el siguiente paso, abrí el HTML y chequeá que todo esté en orden. Es decir, que los nuevos zombies agregados ya estén en movimiento y cumpliendo con todas las instrucciones que les diste.

Paso 5: Ataque de los enemigos

Ahora vas a implementar el ataque de los enemigos al jugador. Recordemos que el objeto enemigo ya tenía un método implementado de ataque que le saca una vida al jugador:

// Por defecto, un enemigo ataca sacando una vida al jugador.

```
Enemigo.prototype.atacar = function (jugador) {
```

```
jugador.perderVidas(1);  
}
```

Este puede sobre-escribirse en los objetos que heredan de él de ser necesario.

- Ataque zombie caminante El zombie caminante usará el ataque por defecto de los enemigos, por eso no necesita una implementación adicional, pero... ¿te animás a modificarlos para que tenga un ataque diferente al de su prototipo?
- Ataque zombies conductores Los zombies que tomaron el control de los trenes atacan de manera letal al jugador, es decir, al atacarlo le quitan todas sus vidas. Por lo tanto, no mantiene la funcionalidad del prototipo Enemigo y deberá modificarse.

Una vez implementado el método, al entrar en contacto un tren con el jugador, el juego deberá terminar apareciendo una pantalla de *Game Over*.

Paso 6: Ejecutar el ataque de los zombies

Ya tenés implementados los métodos para realizar ataques al jugador, pero estos no se están ejecutando. Se necesita, como siempre, enviarles un mensaje a los zombies para que cuando choquen con el jugador lo ataquen.

Fijate que en el objeto “enemigo” podemos encontrar 3 métodos diferentes que hacen referencia al ataque:

- atacar(jugador)
- comenzarAtaque(jugador)
- dejarDeAtacar()

No queremos que el zombie siga atacando repetidas veces una vez que chocó a un jugador. Si no, cada vez que se actualice el juego (que es muy rápido) se le sacarían vidas al jugador, perdiendo instantáneamente.

Para evitar esto se usa comenzarAtaque que chequea si ya se está atacando, y si no lo está, ataca. dejarDeAtacar indica cuándo se dejó de atacar.

Sabiendo esto, **tenés que implementar el método calcularAtaques del objeto juego.js.**

Allí vas a recorrer a todos los enemigos enviándole el mensaje de iniciar un atacar al jugador o dejar de atacarlo según corresponda.