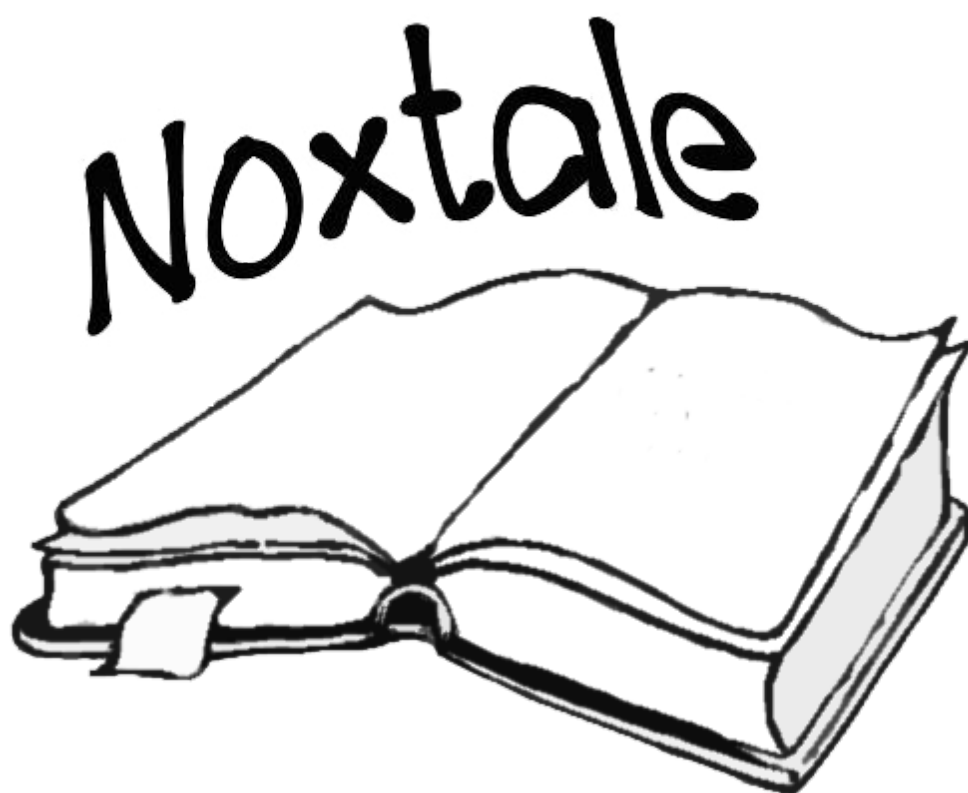


# DOCUMENTACIÓN HITO 4



Grupo Noxtale

Dorado Vide, Guillermo

Sirvent Jerez, Joan Albert

Valls Luna, Ricardo

Salieta Gómez, Alberto

Oliver Rull, Carlos

# ÍNDICE

DEPURACIÓN DE RED E IA

RED

IA

SENSORES

VISTA

IA DEL PROYECTO

IAS ENEMIGAS CON SENSORES

PATHFINDING

A\*

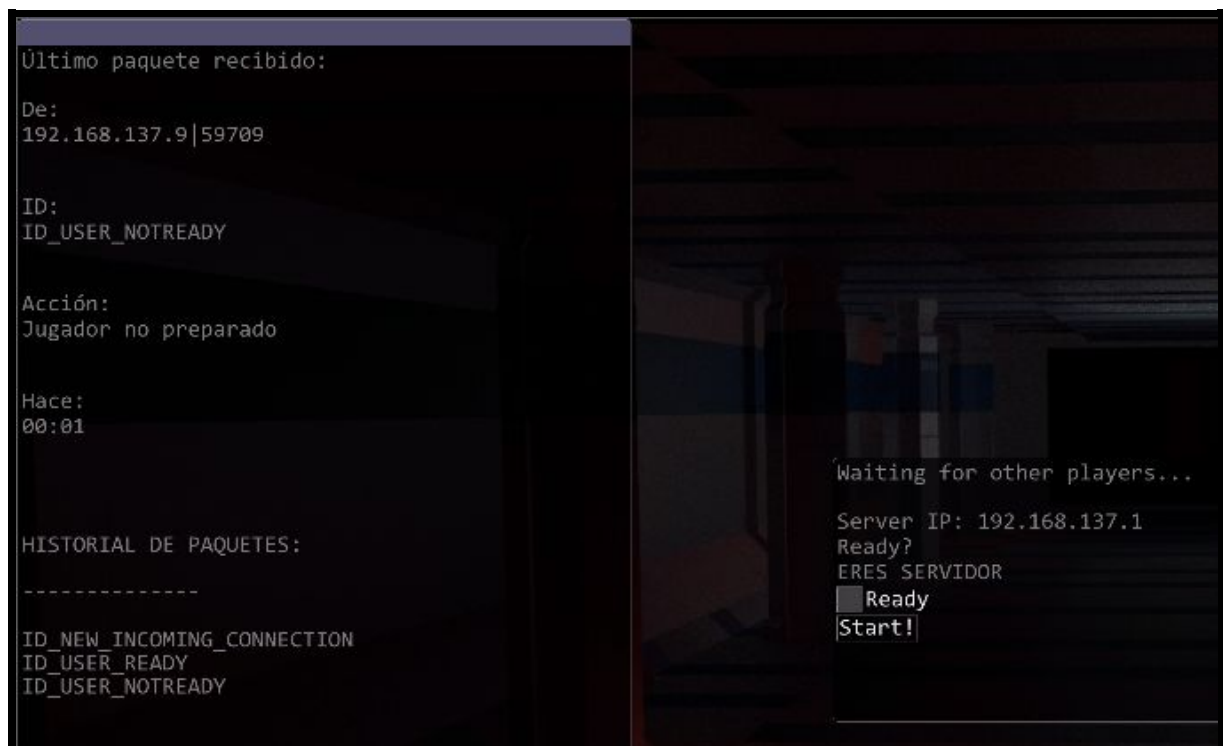
TÁCTICO

# DEPURACIÓN DE RED E IA

## RED

Para la depuración y debug de red, hemos seguido empleando la interfaz proporcionada por la librería ImGui. En esencia, hemos añadido una ventana que muestra los siguientes valores sobre el último paquete recibido:

- Emisor del paquete: IP y puerto del cliente
- ID: ID del paquete.
- Acción: Acción desencadenada por la llegada del paquete (más entendible que mirando la ID)
- Hace: Hace cuanto que ha llegado el paquete. Se reinicia al llegar un nuevo paquete.
- Historial de paquetes: Últimos paquetes recibidos, ordenados de antiguos a recientes.



## IA

En cuanto al proyecto de IA, hemos utilizado AntTweakBar para gestionar los valores y depurar de una forma más sencilla.



Los parámetros básicos del jugador que pueden editarse y que se definen por sí mismos son: velocidad, velocidad de giro, posición y rotación. El segundo grupo corresponde al rango de escucha y de visión, así como el ángulo de visión, que se corresponde al “cono” de visión frontal del personaje.



Estos cambios pueden verse de forma visual activando el modo debug. El círculo morado representa la zona en la que el personaje puede escuchar algo y el círculo amarillo

la zona de visión. Dentro de esta zona, vemos un cono verde que indica la zona de visión frontal.

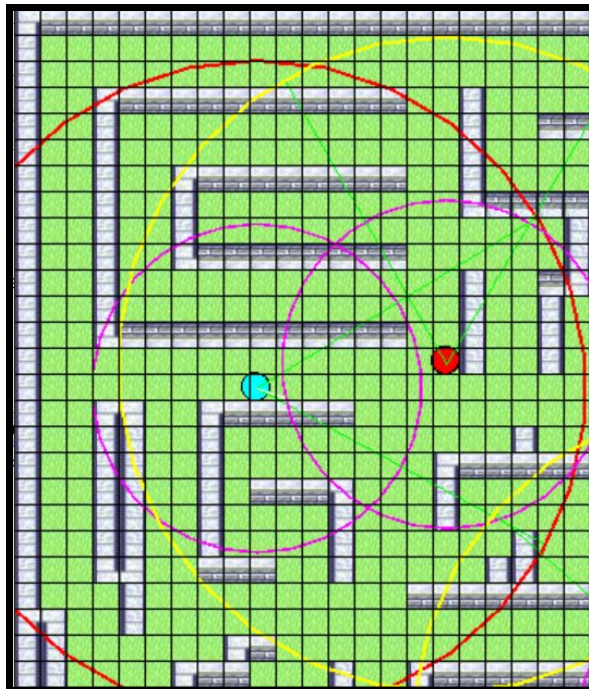
Además, si activamos el modo táctico, y mostramos los pesos, se mostrarán coloreadas de colores azul//verde/blanco las casillas en las que nos encontremos con los enemigos, siendo las azules aquellas con más peso (casillas que nuestro personaje debería evitar). Por último, tenemos el modo sigilo, que indica que nuestro personaje camina “a hurtadillas”, por lo que no emite ruido, así como unos indicadores que revelan si el personaje o enemigo en cuestión está viendo y/o escuchando algo.

## SENSORES

Las IAs, la que controla al jugador y a los enemigos, poseen un sensor de tipo vista y otro de tipo oído. Ambos sensores captan la posición o posible posición del resto de las IAs y actualiza la información para las matrices que se usan en el pathfinding táctico. También se usan para el comportamiento básico de las IAs enemigas.

### VISTA

El sensor “vista” recoge la información en un cono delante de la IA, representado por dos líneas formando dicho cono y un círculo que rodea al “personaje” que indica la distancia máxima en la que puede detectar otros personajes. La circunferencia por defecto es de color amarillo, pero al detectar a otro personaje cambia a un color rojo mientras mantenga línea de visión.



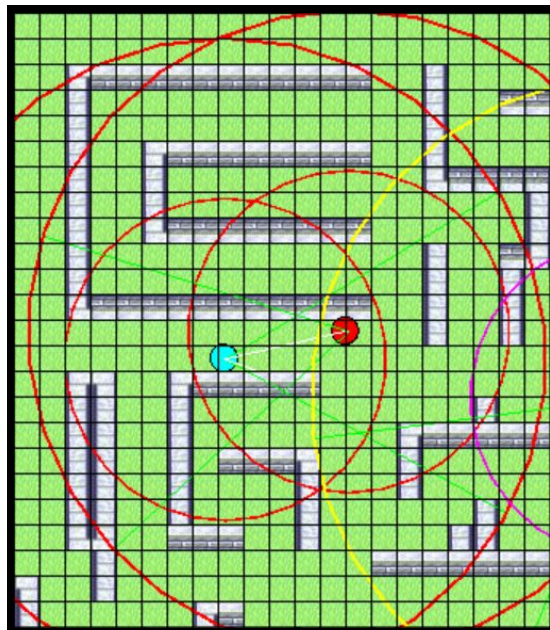
Jugador (cyan) ha visto a un enemigo (rojo)



Primero se comprueba si está a la distancia adecuada, luego si se encuentra dentro del cono de visión del persona y por último, se comprueba si hay algún objeto bloqueando la visión. Si todo se cumple la IA detecta al objetivo. El requisito de distancia se usa para reducir el coste computacional, ya que automáticamente se deja de calcular el resto de casos.

### OÍDO

El sensor “oído” detecta la posición de una fuente de sonido si dicha fuente de sonido entra dentro del radio de acción del sensor. Si el emisor del sonido deja de hacer ruido (activando el modo sigiloso por ejemplo) dejaría de detectar el sonido y pasaría a estar inactivo (aunque podría haber guardado la posición de una anterior fuente de sonido y tenerla en cuenta para investigar). El sensor oído tiene un color morado si no detecta ningún sonido, pero cambia a rojo al detectar ruido. El sensor oído también tiene el requisito de la distancia para realizar los cálculos y cambios en los pesos del pathfinding táctico.



Jugador (cyan) y enemigo (rojo) se están viendo y escuchando, ambos círculos están marcados en rojo.

## IA DEL PROYECTO

### IAS ENEMIGAS CON SENSORES

El comportamiento de las IAs enemigas es el siguiente: Empiezan en un modo patrulla entre dos puntos. Si te ven te persiguen hasta que te pierden de vista. Si no te pueden ver pero te escuchan irán a la zona en la que te escucharon por última vez. Si no te ven ni te escuchan volverán al modo original de patrulla.

# PATHFINDING

## A\*

Este es el pathfinding estándar, el cual dado un inicio y un final obtiene el camino más corto entre ambas posiciones. Para ello el algoritmo recibe una matriz de char, donde dentro de cada posición contiene unos flag a nivel de bit, que indica hacia donde puede expandirse dicha posición

(UP,DOWN,RIGHT,LEFT,UPRIGHT,UPLEFT,DOWNRIGHT,DOWNLEFT y NONE, este último significa que no puede ser accedido desde ninguna posición ni puede acceder a ninguna otra).

Para obtener el camino se usa un sistema de nodos en el cual se van obteniendo sus posiciones (de la matriz) adyacentes (a las que puede acceder) y se van mirando en función de la distancia y del coste total del camino hasta el momento. Y una vez hemos detectado que se ha llegado al final recorreremos desde la última posición/nodo hasta el primero, obteniendo los padres de cada nodo de forma consecutiva y más tarde se ordena el camino obtenido invirtiendo la lista en la que se guardan las posiciones de la matriz que contiene el camino.

## TÁCTICO

El pathfinding táctico usado es una modificación de la implementación del pathfinding A\*. A diferencia del A\*, se usan varias matrices de datos para calcular el camino. Cada matriz representa distintos datos del entorno del personaje. En nuestro caso, hemos usado 3 matrices distintas para calcular el camino del protagonista.

- La primera matriz es la misma que se usa en el A\*, se trata de la información estática del mapa, las paredes por donde no se puede pasar.
- La segunda matriz contiene la información recibida por el sensor oído del personaje. Modifica sus valores en función de lo que oye el personaje y le permite esquivar enemigos usando su oído.
- La tercera matriz trata de la vista, y permite esquivar enemigos que el personaje tenga en su rango de visión.

Además, debido a que los enemigos son dinámicos, mientras el personaje se mueve es necesario recalculer el camino por si un enemigo se hubiera cruzado en el camino calculado originalmente, para ello se recalcula sólo si el personaje oye o ve a un enemigo.

El algoritmo calcula una matriz de pesos a partir de las matrices pasadas por parámetro y elige el camino de menor peso y distancia, de manera que siempre buscará el camino por el que el personaje corra el menor riesgo pero sin hacer el camino excesivamente largo.