

UBOA

UBOA

UBOA VMF Obfuscator

-

Ofuscador de ficheros VMF
(Valve Map File)

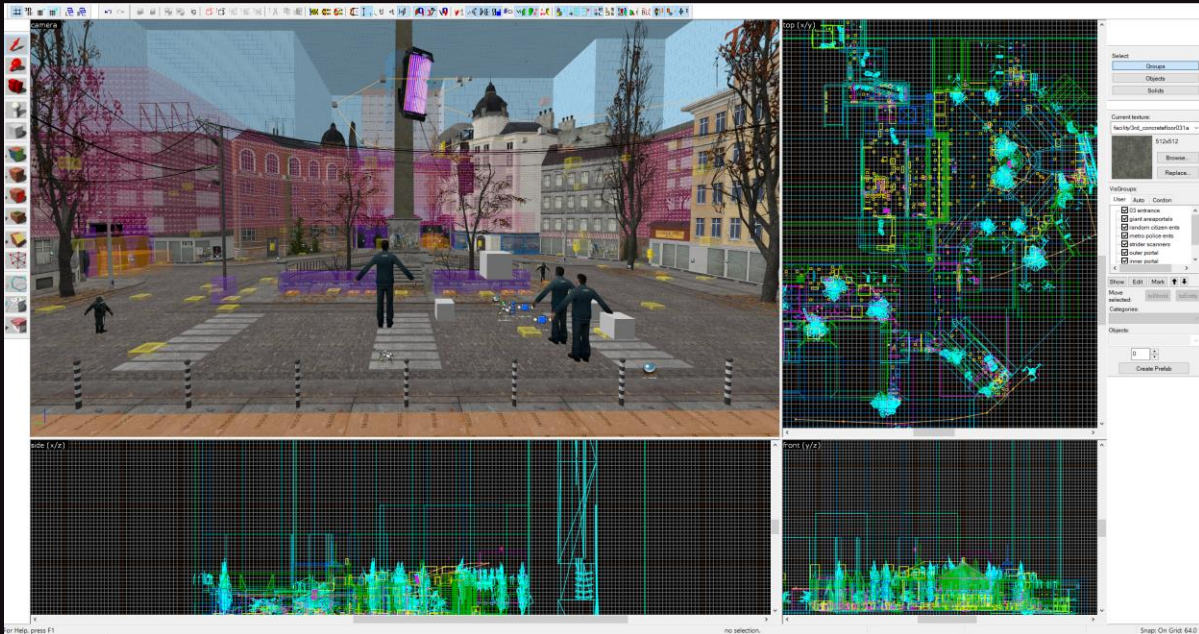
Valve Map File (VMF) y el proceso de compilación

Paso 1: El fichero VMF

El código fuente de un mapa (nivel) diseñado para videojuegos operando bajo el motor Source (ej. Half-Life 2, Counter-Strike: Source)

- Texto plano (iiiimportante!!!)
- Define la arquitectura y el funcionamiento (lógica) de un nivel
- Puede ser descrito como un lenguaje de marcas
- Estructura organizada y claramente definida
(metadatos > solidos > entidades)

Paso 2: Valve Hammer Editor



(Anticuada) Herramienta de diseño de niveles para juegos operando bajo el motor Source

Interpreta y visualiza ficheros VMF

Permite la fácil edición y compilación de los mapas.

Paso 3: Compilar y jugar



Al someterse a una serie de operaciones

UMF \rightarrow BSP

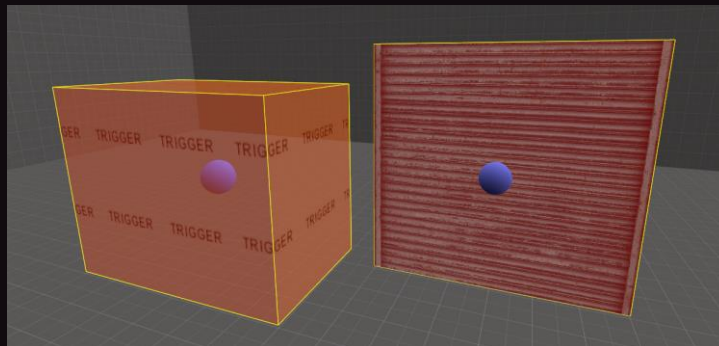
(binary space partition)

- Texto plano \rightarrow archivo binario
- Formato entendible por el motor de juego
- Puede ser descompilado

En resumen:

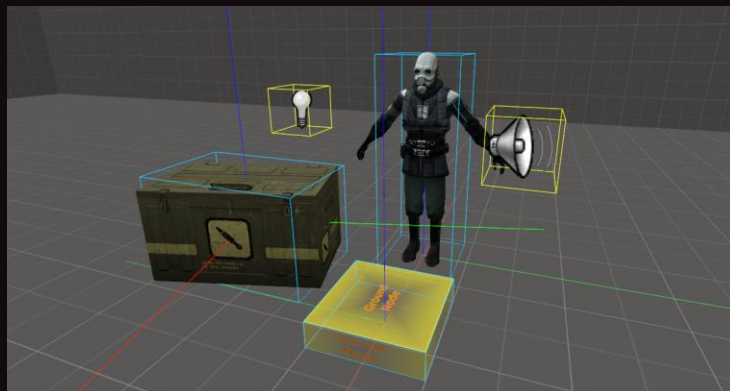
- Dos tipos de entidades:

1. Entidades sólidas

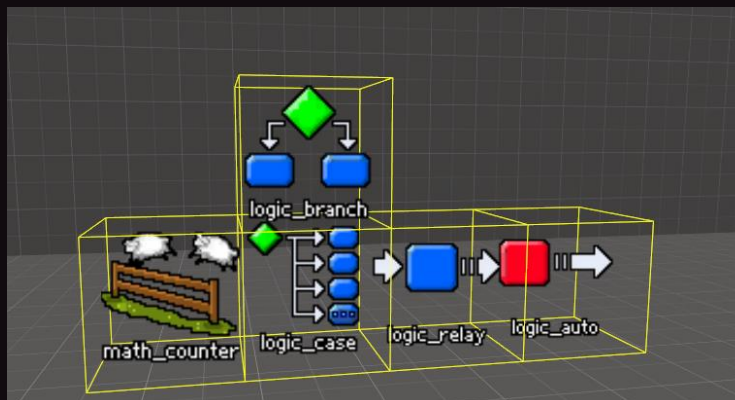


2. Entidades de punto...

2.1. Entidades físicas



2.2. Entidades lógicas



Conociendo esto...

- Abrir fichero UMF, analizar, obtener datos relevantes

Si (entidad lógica) ...

- Ofuscar coordenadas
 1. Mover al primer punto valido
 2. ¿Intercambiar coordenadas?
 3. Infinitas posibilidades
- ¿Ofuscar nombres?
 1. Cadena aleatoria
 2. Intercambiar nombres
 3. Y más...
- Guardar copia
- Asegurar la integridad del archivo

Candidatura para el TFG

Parece simple en la superficie, pero:

- Poner a prueba mis conocimientos
 - BBDD, AD, UI, UX, multihilo
- Investigación de algoritmos
 - ¿Cuál es la manera más eficiente de lograr el objetivo?
- Uso de librerías
 - Audio, APIs, LAF, librerías de terceros
- Nuevos conceptos
 - JAR > EXE (Launch4J)
 - JRE personalizado
 - JRE integrado
- Código libre y finalmente producción

Declaración de la misión:

-Propósito

Buscar y reemplazar dentro del fichero UMF según parámetros del usuario, operaciones en lote

-¿Cómo?

Aplicación de escritorio escrita en Java usando la librería Swing

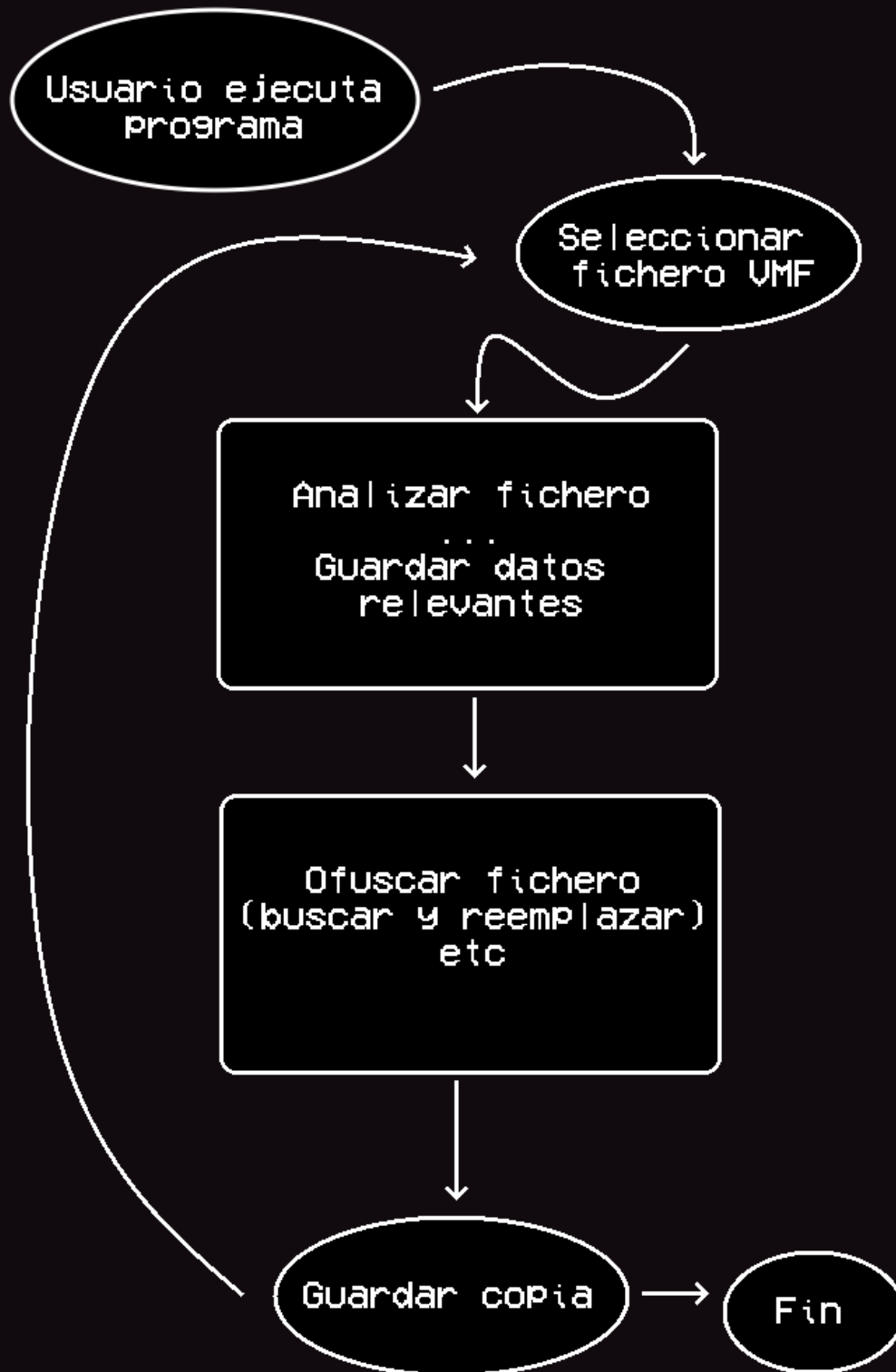
-¿Para quién?

Diseñadores de niveles

-¿Por qué?

Crear una herramienta modernizada y fácil de usar

Funcionamiento



Gráficos

- Temática: alto contraste
- Posibles colores alternativos
- Interpolación por el vecino más cercano (pixelado)

U B D A

U B D A

U B D A

U B D A

Iconos:

