Arquitectura de videojuegos

Máster en Programación de videojuegos

Javier Arévalo

2º Trimestre
Curso 2013-2014

Arquitectura de videojuegos

Escenarios y Objetos

Escenarios y Objetos

- Muchos juegos tienen su desarrollo dividido en **fases, misiones, niveles o mundos**
- En muchos juegos, la accion situa al jugador dentro de un **contexto espacial y visual**
- Ese contexto suele incluir tanto un gran contenido estatico como una **descripcion de entidades presentes** en el

Ejemplo sencillo de niveles

- Tetris
  - El jugador siempre empieza en el nivel 1
  - Cada nivel exige un determinado numero de lineas para completarse
  - En cada nuevo nivel, las fichas caen mas rapido
  - No hay limite de niveles
  - El cambio de nivel apenas da un respiro al jugador

Generacion Procedural

```
Level = 1

while !dead:
   PendingLines = 20
   PieceDelay = min(5, 60 - 2*Level)
   ScoreMultiplier = Level
   RandomBlocks = (Level > 5)
   RandomBlockDelay = min(10, 21-Level/5)
   PlayLevel()
```

Generacion Procedural

- La capacidad de generar parametros es muy util sobre todo al principio
  - Igual que poder hacer una IA que sea esencialmente rand().
  - Gameplay y curva de dificultad instantaneos
- En la practica, la mayoria de los juegos tienen una progresion de niveles definida por datos
- Algunos datos pueden ser formulas

Definicion de niveles

- Por ejemplo, un tower defense
- Missions.json:
  - Identificador
  - Tipo de mission
  - Parametros por tipo
  - Objetivos y premios
  - Tipos de enemigos
  - Daño y dificultad, etc

```
[
    {
        "id": 1,
        "type": "survival",
        "goal": { "time": 180 },
        "stars": [ 10000, 15000, 2000 ],
        “reward”: [20, 50, 100]
    },
    {
        "id": 2,
        "type": "conquest",
        "goal": { "pctmap": 80 },
        "stars": [ 30000, 50000, 100000 ],
        “reward”: [20, 50, 100]
    },
    ….
]
```

Definicion de niveles

- Otros parametros tipicos:
  - Contiene tutorial
  - Opcional
  - Descripcion textual y *briefing*
  - Niveles posteriores
  - Posicion y aspecto visual en mapa de campaña
  - Bloqueos o pre-condiciones
- Frecuentemente guardamos los niveles completados y el grado de éxito en ellos (high score, estrellas...)

Definicion de niveles

- Identificador
  - Importantisimo
  - No depender del orden fisico
  - Cualquier referencia persistente es por *id*
  - Es un seguro de vida para cambios futuros

```
[
    {
        "id": 1,
        "type": "survival",
    },
    {
        "id": 105,
        "type": "escort",
    },
    {
        "id": 110,
        "type": "survival",
    },
    {
        "id": 2,
        "type": "conquest",
    },
    ....
]
```

Misiones

- Muchos juegos recientes incluyen multiples misiones simultaneas
  - Concepto de Quests en los juegos de rol
- El concepto es similar a los niveles, y podemos querer ambos
- Las misiones pueden exigir objetivos en una partida, o a lo largo de multiples partidas
  - Estado / progreso de la mision tambien persistente

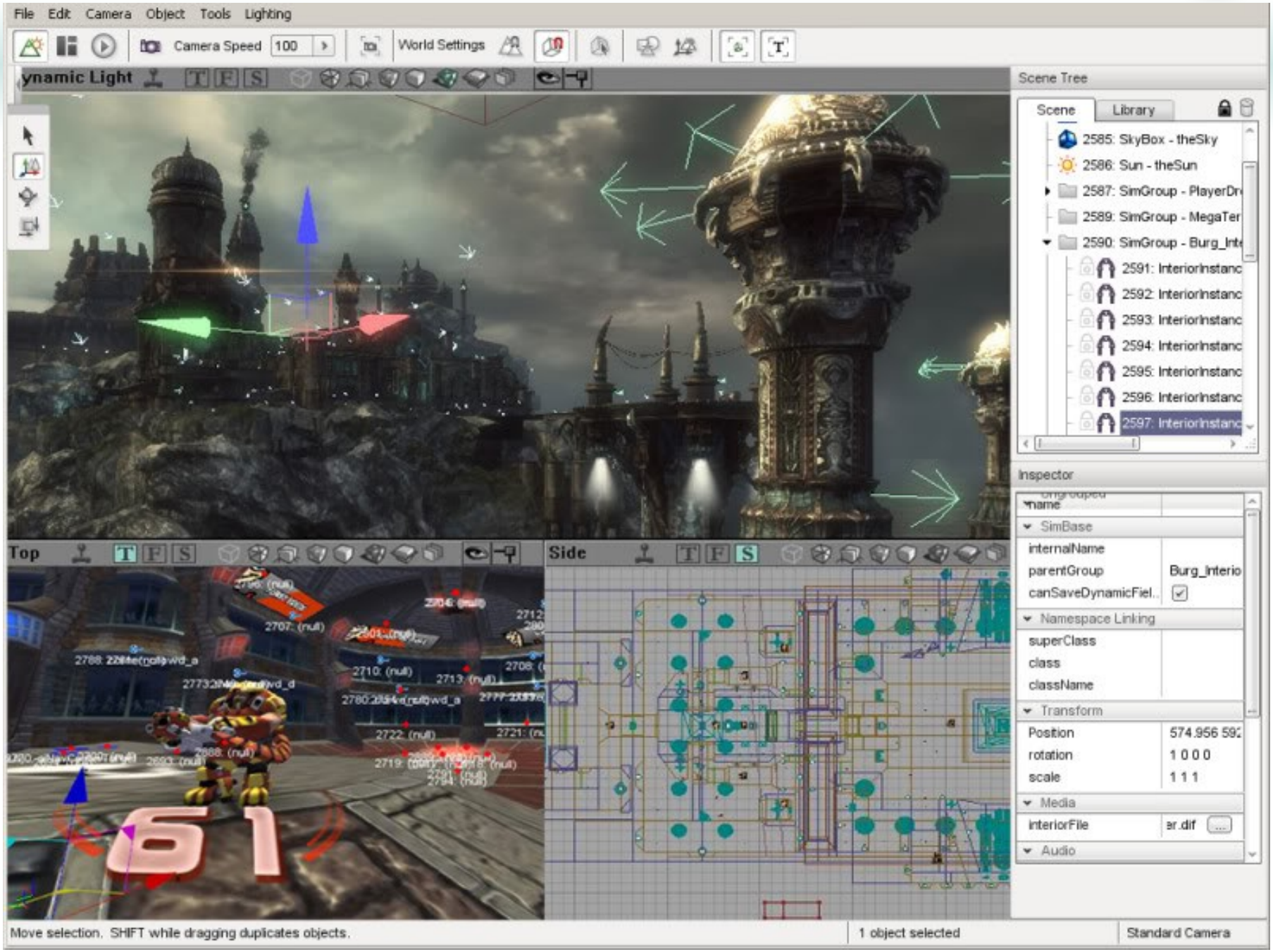
Escenarios

- Entorno visual
- Informacion de navegacion
- Zonas logicas y triggers
- Objetos moviles o props
- Rutas y hints de IA


Praetorians
Misión ZZZ_Puerto.MIS
Ficheros Edición Ver Selección Incluir
Acción del ratón
1-Elem+
2-Mover
3-Angulo
4-Pega(p)
5-Pega(m)
6-Posini++
Movimiento
Alt F5-Uno
Alt F6-Selecci
Respeta Z es
Tamaño del mapa: 170 x 170
MAL (Escenario importado)
PVE MisionIncendiarUnaFlota.PVE
Simple
Multijugador
Bando NEUTRAL (neutral)
Sel
Pos. ini. bando
Ver 1 2 3 4 5 6
Sel
Asigna
Sel 2/206
Borra(Supr)
Centra
Nombre
Tropas
Estructuras
Critters
BARBAROS
GALO_CENTURION
GALO_CENTURION 1-1
Sel x tipo
Sel x form.
Sel todas
Tama 8
Exp. 0
Asigna dato
Propied.
Grupos G_RESISTANCE
Portapapeles
Salva
Visualización
Pantalla: 880, 664
Mundo: 28.7,0,89.2
Cámara: -27, 51, 88
y FOV: 75°, 35° 50
Libre
Front
Orto
48
Undo (Ctrl+Z)
Firma Internal Version (c) Pyro Studios 2002

Entorno visual

- Al igual que describimos la separacion entre logica y rendering
- Separamos la ejecucion visual y la informacion logica de los escenarios
- La parte logica suele resultar en una version “simplificada” de la parte visual
- La parte visual puede incluir animaciones y elementos de fondo sin logica asociada


File Edit Camera Object Tools Lighting
Camera Speed 100
World Settings
ynamic Light
Scene Tree
Scene
Library
2585: SkyBox - theSky
2586: Sun - theSun
2587: SimGroup - PlayerDr
2589: SimGroup - MegaTer
2590: SimGroup - Burg_Int
2591: InteriorInstanc
2592: InteriorInstanc
2593: InteriorInstanc
2594: InteriorInstanc
2595: InteriorInstanc
2596: InteriorInstanc
2597: InteriorInstanc
Inspector
SimBase
internalName
parentGroup
Burg_Interio
canSaveDynamicFiel..
Namespace Linking
superClass
class
className
Transform
Postion
574.956 59
rotation
1 0 0 0
scale
1 1 1
Media
interiorFile
ar.dif
Audio
Top
Side
Move selection. SHIFT while dragging duplicates objects.
1 object selected
Standard Camera

Separacion visual/logica

- Los diseñadores se centran en editar zonas logicas, usando editores especializados
- Los artistas crean contenido grafico usando herramientas de creacion generales
  - Photoshop, Max, Maya, etc.
- La parte visual puede ser imprecisa, la parte logica no
  - Agujeros en el suelo

Navegacion

- Describir areas del escenario con diferentes propiedades de movimiento
  - Espacio abierto
  - Agua
  - Pasadizo estrecho
  - Matorrales y bosques
  - Nieve
  - Tunel
  - ...

Navegacion

- Representan diferentes aspectos reales
  - Material del terreno
  - Volumen de la zona
  - Densidad de obstaculos
  - Inclinacion del suelo
  - Puntos de agarre
- Y propiedades abstractas de gameplay
  - Camino a los *boxes*

Navegacion

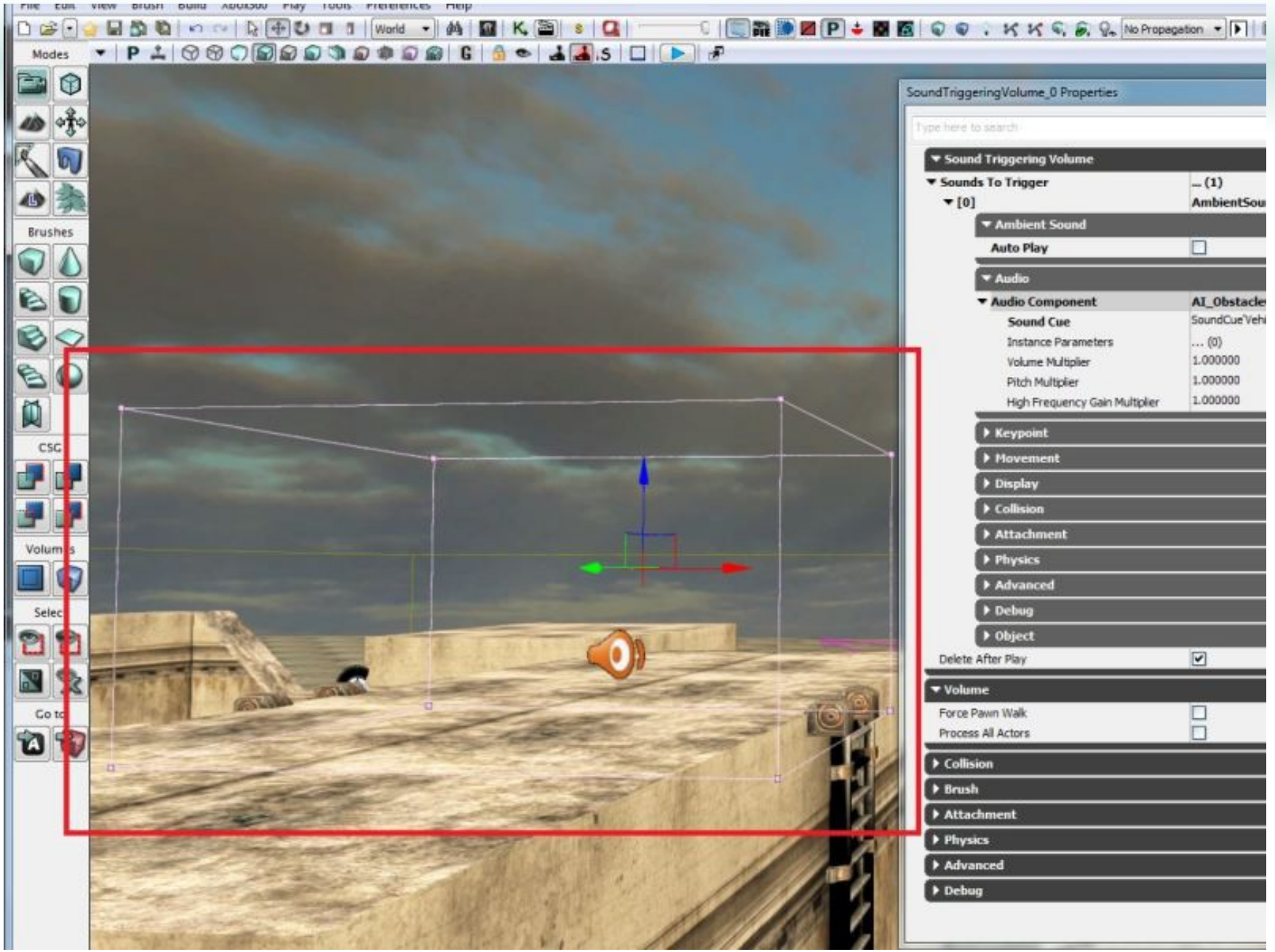
- Pueden afectar a la logica de movimiento
  - Velocidad, consumo de turnos, etc
- Visibilidad y percepcion
  - Tropas escondidas en un bosque
  - Rango de vision en general
  - Sonidos emitidos al atravesarlas
- Efectos visuales
  - Huellas en la nieve

Navegacion

- Su estructura debe permitir
  - Identificar la zona de cada entidad
  - Conectividad y navegabilidad entre zonas
  - Trazado de rayos y consultas especulativas
- Busqueda de caminos
  - Eficiente
  - Parametrizada
  - Optimizacion y embellecimiento del resultado

Zonas logicas y *triggers*

- Objetivo principal
  - Dar soporte a la narrativa del gameplay y de la IA
- Usos
  - Detectar eventos de gameplay: entrada y salida de zonas
  - Disparar scripts y actuaciones de gameplay y de IA
  - Facilitar decisiones de comportamiento a mas alto nivel que la navegacion
  - Alterar parametros de gameplay: gravedad, daño


SoundTriggeringVolume_0 Properties
Sound Triggering Volume
Sounds To Trigger
[0]
Ambient Sound
Auto Play
Audio
Audio Component
Sound Cue
Instance Parameters
Volume Multiplier
Pitch Multiplier
High Frequency Gain Multiplier
Keypoint
Movement
Display
Collision
Attachment
Physics
Advanced
Debug
Object
Delete After Play
Volume
Force Pawn Walk
Process All Actors
Collision
Brush
Attachment
Physics
Advanced
Debug
Modes
Brushes
CSG
World
No Propagation

Zonas logicas y *triggers*

- Conceptualmente similar a la navegacion
  - Se pueden mezclar en la misma estructura
  - Encajan con la visualizacion a muy alto nivel, y de manera mas arbitraria.
  - Colocacion totalmente arbitraria por diseño
- P.ej. Un paso de cebra
  - 'cruzar la calle' (alto nivel) vs 'andar' (bajo nivel)
  - No marcarlo si la calle esta bloqueada

Zonas logicas y *triggers*

- Requisitos
  - Identificables por nombres 'legibles'
  - Consultas rapidas, reales y especulativas, con *flags*
  - Deteccion de entrada, salida, presencia y movimiento a traves
  - Habilitar y deshabilitar por codigo/script
- No solo zonas estaticas
  - Creables y destruibles tambien dinamicamente
  - Enganchables a objetos moviles

Objetos y *props*

- Descripcion de entidades que deben aparecer de forma fija como parte del escenario
  - Enemigos
  - Puertas
  - Powerups
  - Arboles
  - Cajas
  - Generadores


105753,600
0:02:56.42

Objetos y *props*

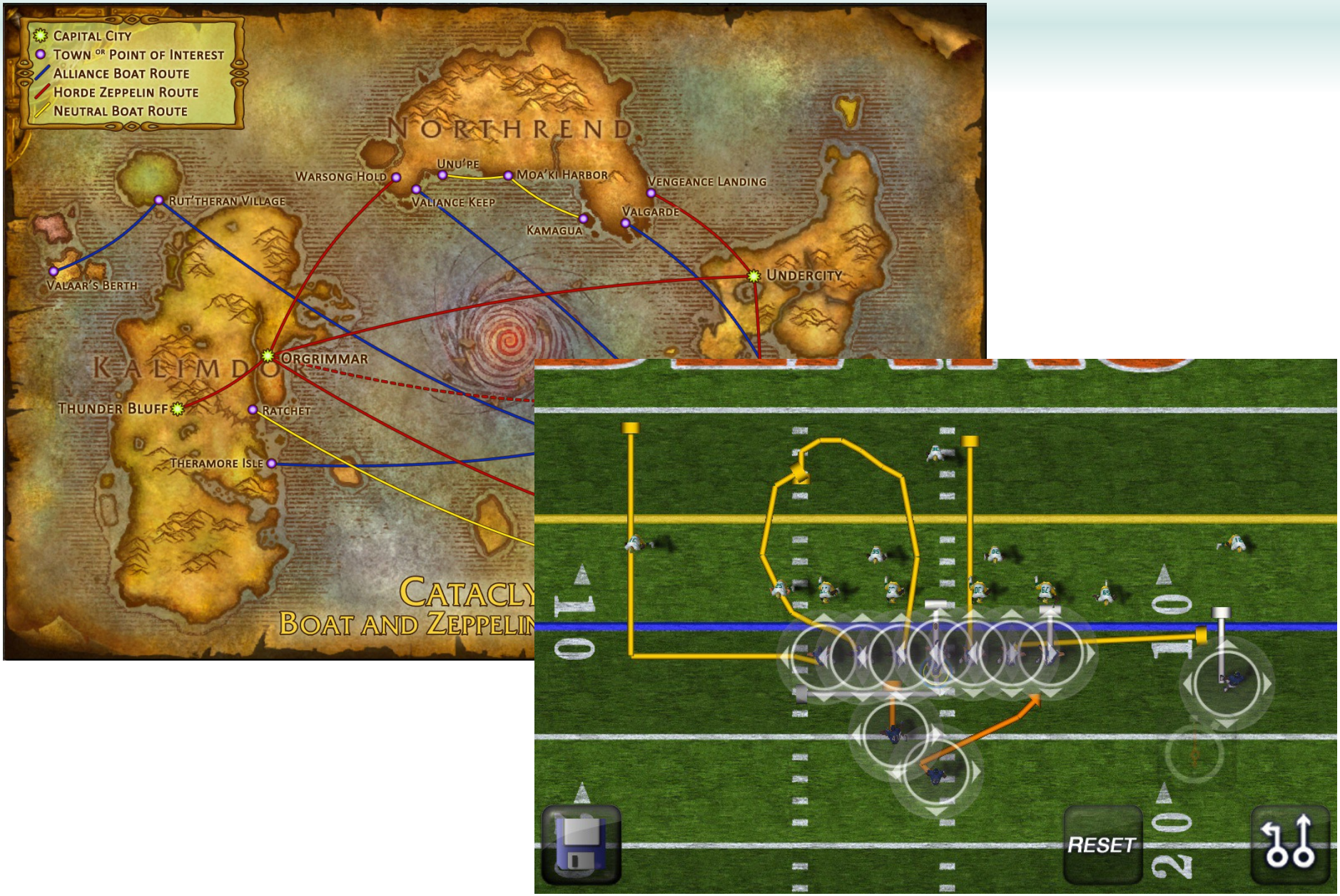
- En esencia, estan compuestos de:
  - Posicion, orientacion y bounding box
  - Tipo de objeto a crear
  - Parametros asociados
    - Color, tamaño, nivel
    - Tabla de variantes
    - Regeneracion
    - Parametros de comportamiento, ruta, etc

Objetos y *props*

- Creacion de la entidad
  - En funcion de la bounding box, distancia, visibilidad
  - Inicialmente en estado 'dormido' o estable
  - Potencialmente se eliminan por criterios similares
- Problema clasico: stacking
  - Entidades encima de otras
  - Dependencia fisica

Rutas y *hints* de IA

- Lugares logicos usados para generar y calcular comportamientos
- Ruta
  - Trayectoria generalmente cerrada
  - Indica puntos, velocidades, orientaciones, paradas
- Hints
  - Usadas para dar sentido semantico a zonas del mapa

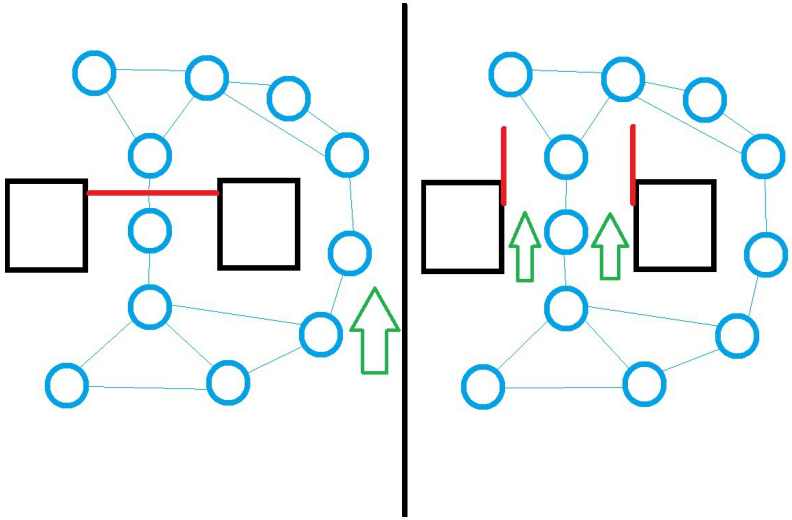

Capital City
Town or Point of Interest
Alliance Boat Route
Horde Zeppelin Route
Neutral Boat Route
NORTHREND
Unu'pe
Warsong Hold
Moa'ki Harbor
Vengeance Landing
Rut'theran Village
Valiance Keep
Valgarde
Kamagua
Undercity
Valaar's Berth
Orgrimmar
KALIMDOR
Thunder Bluff
Ratchet
Theramore Isle
CATACLY
BOAT AND ZEPPELIN
RESET

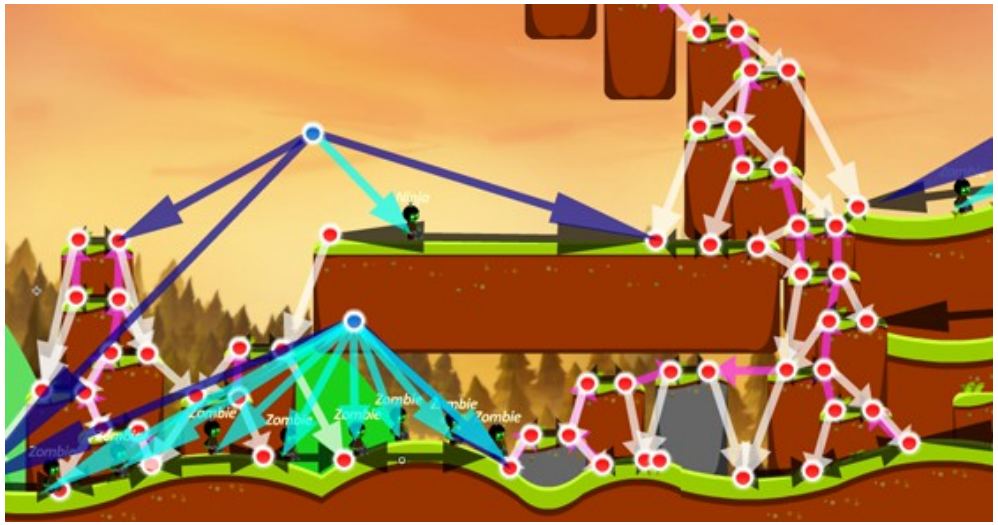
Rutas y *hints* de IA

- Ejemplos de rutas
  - Movimiento prefijados en un juego de plataformas
    - Para enemigos, suelos moviles y transportes
  - Ruta ideal en un circuito de carreras
  - Rutas de un libro de jugadas deportivas
  - Rutas posibles de patrulla y busqueda para IAs
  - Rutas de movimiento forzado del jugador
  - Rutas y railes para la camara

Rutas y *hints* de IA

- Ejemplos de hints
  - Zonas de interes y puntos importantes
  - Puntos de una malla de navegacion
  - Puntos y direccion / fuerza de salto



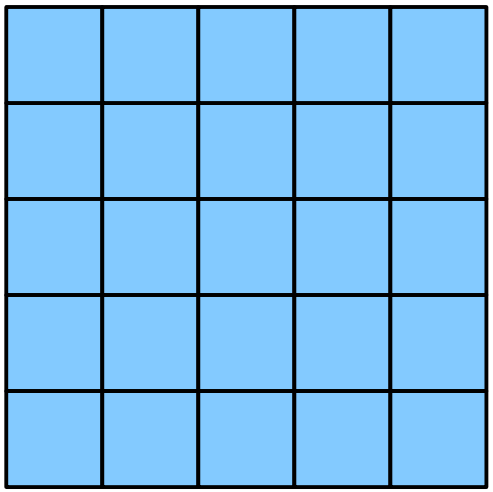


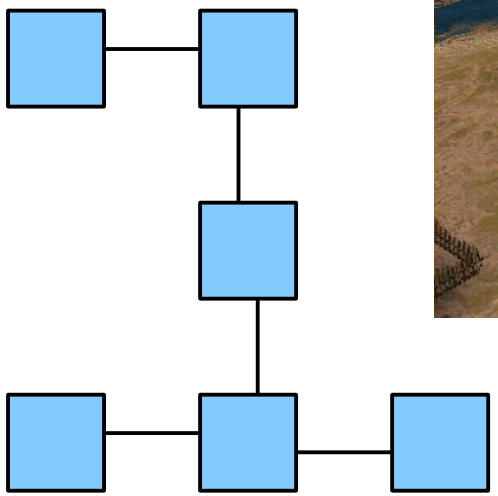
Estructura espacial

- 2D vs 3D
  - 2D cenital o de perfil
  - 3D completo
  - 2.5D
    - 2D pero decorando con un mapa de alturas o con alturas simples
    - Doom, RTS, shooters 2D

Estructura espacial

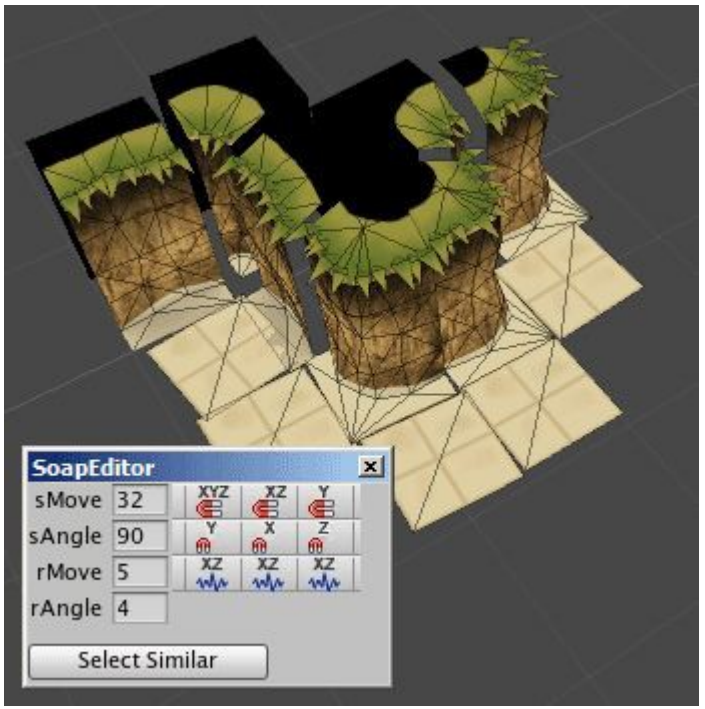
- Discreto frente a continuo
  - Rejillas y celdas fijas: Minecraft, Mario, Sims, RTS
  - Continuos: FPS, 3rd person
  - Geometrias extrañas: Nodos conectados, esfericos...





Estructura espacial

- Construccion
  - Por piezas vs geometria arbitraria
- Piezas
  - Conectables
  - 'Sets' de piezas
  - Editable por jugadores
- Geometria arbitraria
  - Mas cara de construir



Mundos abiertos

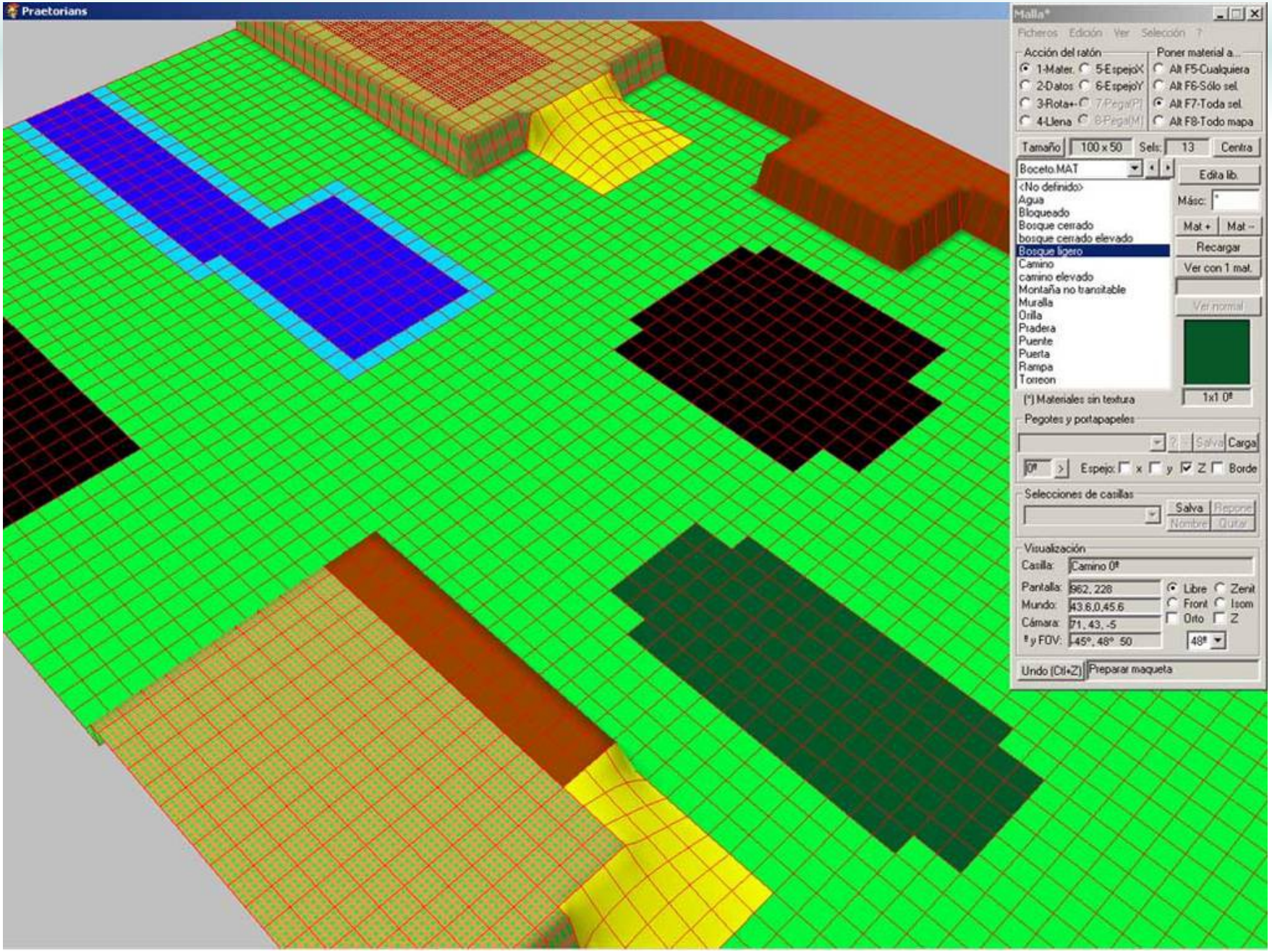
- Mas grandes que la memoria del sistema
- Subdivision en areas o zonas
- Carga y descarga dinamica
  - Zonas por movimiento
  - Niveles de detalle de zonas
  - Informacion de rutas y hints por mision / nivel
- Persistencia limitada de los cambios
- Comparticion de elementos entre zonas
- Elementos supra-zonales: rutas, puntos de interes, etc

Generacion procedural

- Generacion de laberintos
- Mapas de alturas fractales
  - Refinamientos posteriores: erosion, etc
- Creacion de puntos de interes
- Enlazando patrones pre-generados
- Consideraciones sobre la dificultad
  - Y sobre el ritmo

Edicion

- Editores externos
  - WYSIWYG siempre que sea posible!
- Multiples tipos de edicion integrados
- Editores integrados en el juego
  - Duke Nukem
  - Version especial del juego (PC only)
- Operaciones automaticas
  - Verificacion
  - Generacion de niveles jugables para prototipado


Praetorians
Malla*
Ficheros Edición Ver Selección ?
Acción del ratón
1-Mater. 5-EspejoX
2-Datos 6-EspejoY
3-Rota+- 7-Pega(P)
4-Llena 8-Pega(M)
Poner material a...
Alt F5-Cualquiera
Alt F6-Sólo sel.
Alt F7-Toda sel.
Alt F8-Todo mapa
Tamaño 100 x 50 Sels: 13 Centra
Boceto.MAT
Edita lib.
<No definido>
Agua
Bloqueado
Bosque cerrado
bosque cerrado elevado
Bosque ligero
Camino
camino elevado
Montaña no transitable
Muralla
Orilla
Pradera
Puente
Puerta
Rampa
Torreon
Másc:
Mat + Mat --
Recargar
Ver con 1 mat.
Ver normal
(*) Materiales sin textura
1x1 0ª
Pegotes y portapapeles
Salva Carga
0ª Espejo: x y Z Borde
Selecciones de casillas
Salva Repone
Nombre Quitar
Visualización
Casilla: Camino 0ª
Pantalla: 962, 228
Mundo: 43.6,0,45.6
Cámara: 71, 43, -5
ª y FOV: -45°, 48° 50
Libre Zenit
Front Isom
Orto Z
48ª
Undo [Ctl+Z]
Preparar maqueta

Vuestro turno!

Preguntas?