# **PASOS A SEGUIR, POSIBLES APLICACIONES Y ESTRATEGIAS**

Pasos generales para el diseño de la IA:

1. Lista de funcionalidades.
2. Identificar componentes.
3. Identificar conexiones.
4. Identificar dependencias.
5. Herramientas de modificación de comportamientos.
6. Prueba.

Información perfecta o imperfecta.

Tipos de sensores: visión, audición, tacto, olfato...

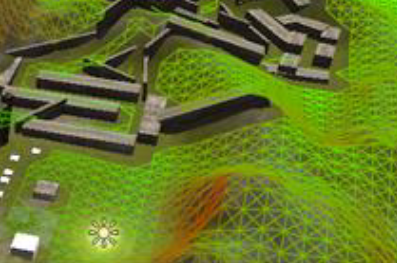
Simplificación del entorno (información estática vs dinámica)

Smart objects

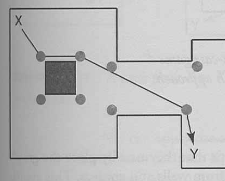
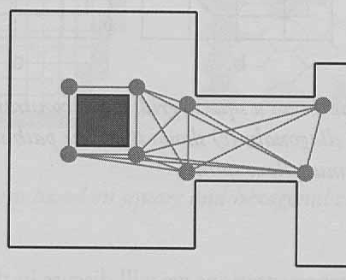
Navegación: Waypoints, regular grids, sense and avoid, corner reduction, path smoothing...

## Navegación

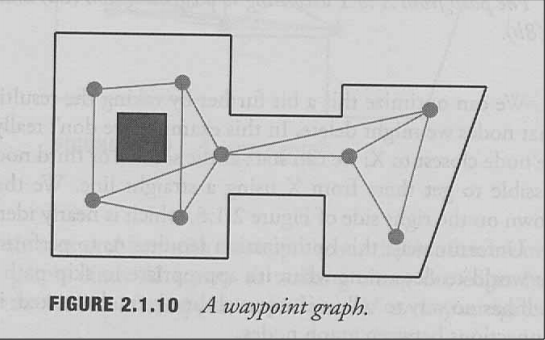
**Regular grids:**

* Búsqueda por acceso aleatorio O(1) para determinar si una baldosa está en una determinada coordenada.
* Requiere un número grande para representar el mundo.
* Calidad del camino.
* El personaje solo puede moverse en 4, 6 direcciones.
* 
* Los grids son también grafos, cada celda es un nodo.

**Corner graphs:**

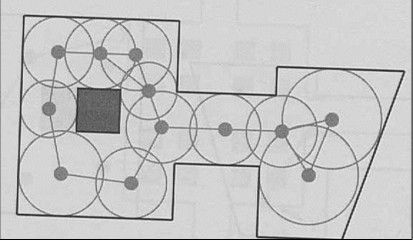
* Puntos de navegación en puntos conflictivos, alrededor de obstáculos...
* 
* Identificar esquinas convexas
* ¿Puede transitarse de un punto a otro?
* Menos memoria, se genera rápido
* Los personajes tienden a moverse pegados a los obstáculos, no emplean espacios abiertos.

**Waypoint graphs:**



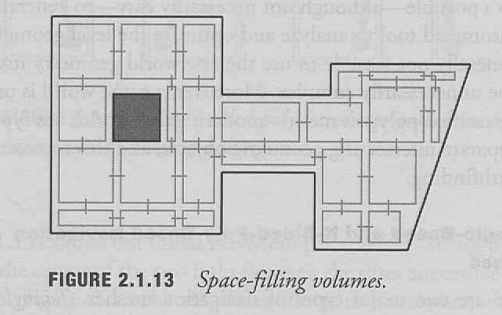
* Uso de nodos
* Bueno para juegos 3d
* Nodos lejos de paredes para evitar problemas
* Fácil encontrar camino óptimo.
* Se colocan los puntos donde queramos, a mano.
* Reduce bastante memoria respecto regular grids.
* O(N^2)
* Malo para espacios abiertos y la calidad de la ruta depende de la complejidad del grafo.

**Circle-based Waypoint graphs:**



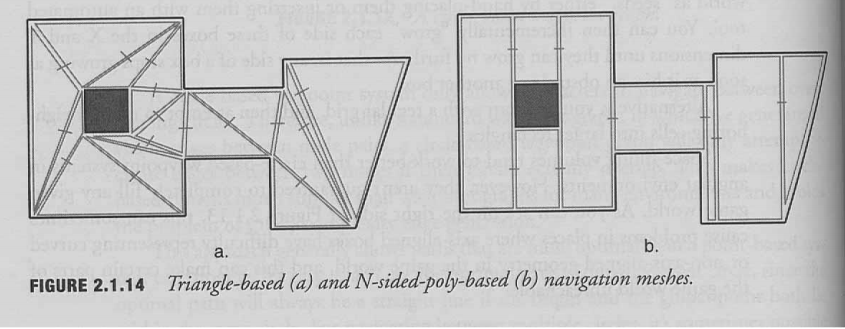
* Se añade un radio para indicar el espacio abierto disponible
* Se suele combinar con Waypoints normales para las zonas cerradas y Waypoints circulares para zonas abiertas.

**Space-Filling Volumes**



* Emplear rectángulos o cubos 3D en lugar de círculos
* Muy parecido al mecanismo basado en círculos, pero maneja mejor los ángulos
* Deja zonas sin cubrir.

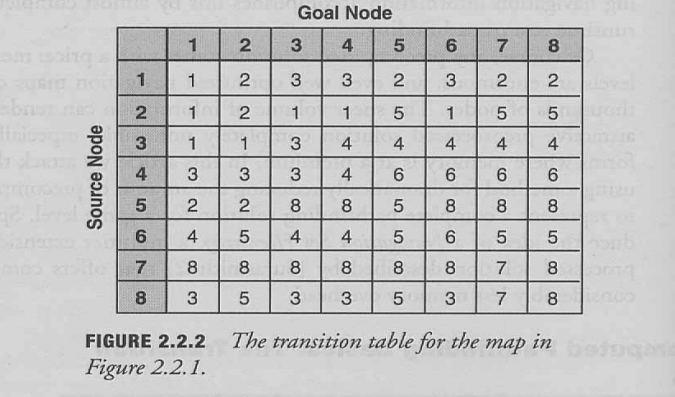
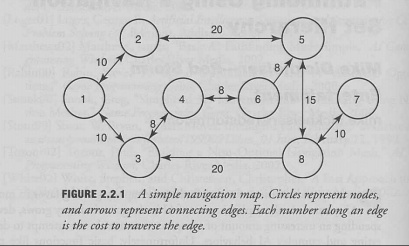
**Mallas de navegación**



* Intentamos cubrir la superficie a transitar con polígonos convexos.
* Los personajes pueden transitar entre polígonos adyacentes.
* Generarlo a mano consume mucho tiempo.
* Existen herramientas automáticas para analizar y optimizar la geometría del mundo
* Demasiado complejo y difícil de representar con una sola malla, suele necesitar de solapamientos, etc.
* Encuentra caminos óptimos.
* Bueno para terrenos interiores y exteriores.
* Puede crecer bastante en complejidad y en uso de memoria.
* Difíciles de generar.

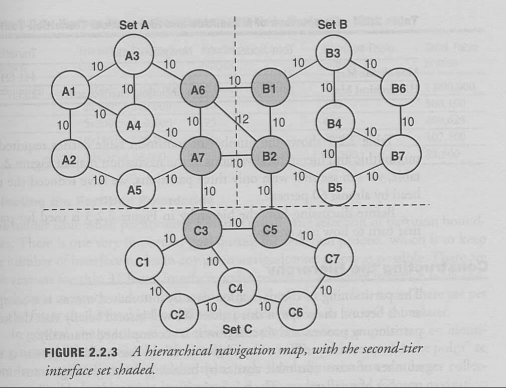
## Optimización de espacios

**Tablas de transición**



* Almacena la información de la transición en función del nodo actual y el nodo objetivo.

**Conjuntos de navegación**



* Se separan los nodos en varios conjuntos.
* El coste de la búsqueda escala con el número de nodos de enlace.

## Comportamientos de direccionamiento

**Comportamiento Seek**

* Se añaden 2 fuerzas de direccionamiento en cada frame en función de: La dirección actual y la dirección hacia el objetivo.

## Sistema de toma de decisiones

Propiedades de un agente: reactividad, autonomía, pro-actividad, habilidad social, encarnado... extra: movimiento, aprendizaje...

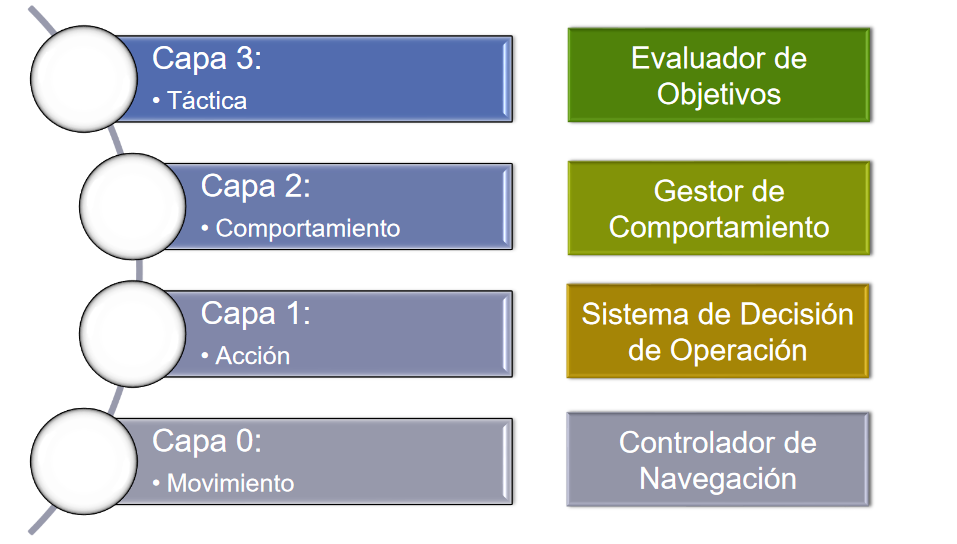
Problemática a resolver: corto, largo u objetivo complejo...

* ARQUITECTURAS REACTIVAS:
  + Difícil de fallar
  + Agentes sencillos
  + Computacionalmente sencillo
  + Incapacidad de pensar a largo plazo
  + Incapacidad de aprender
  + SUBSUMPTION ARCHITECTURE
* ARQUITECTURAS DELIBERATIVAS:
  + Razonamiento lógico,
  + Belief + Desire + Intention
  + limitación si el entorno es dinámico hay que actualizar el modelo simbólico, el mundo puede cambiar mientras planeamos qué hacer.
* ARQUITECTURAS EN CAPAS:
  + Forma de estructurar un ejemplo de ia (capas): Reactiva → Conocimiento (deliberativa) → Social (BDI).
* ARQUITECTURAS COGNITIVAS:
  + Procesos computacionales actúan como sistemas cognitivos
  + Alcanzan juntos un objetivo controlados por algún mecanismo de auto sincronización

## Acciones

Nivel de decisiones:

* División Temporal: Corto, Medio y largo Plazo
* División Orgánica: Gestión, Combate, Investigación
* División cognitiva: reactivo, deliberativo, emotivo



# **IDEAS**

**Idea 1**

Eres el defensor de villa “Opal”. Un malvado ejército de “jaja” quiere acabar con tu preciado poblado. Frena el ataque de los “jajas” antes de que acaben con las estructuras más importantes de la villa.

* Vista isométrica.
* Deberás girar la cámara para ver si intentan colarse por un punto ciego.
* Primero debes colocar cañones alrededor de la villa para evitar que entren al interior.
* Si consiguen penetrar las barreras la villa pasará a ser tu campo de batalla y deberás redistribuir tus recursos para frenar su avance.
* Otro elemento que entra en juego cuando caen tus barreras es que puedes elegir héroes y colocarlos en las calles para que ataquen y aguanten.
* Una vez dentro de la villa irán a atacar los edificios más importantes y tu misión será acabar con ellos antes de que los alcancen.

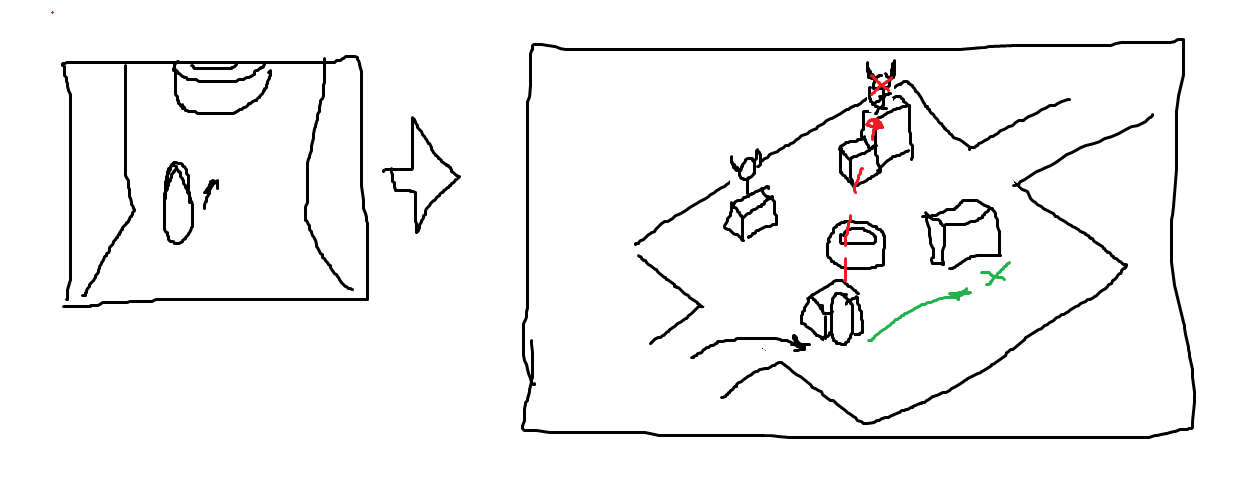
**IA**

* Movimiento por el pueblo: NavMesh
* Enemigo
  + Objetivo principal: derrumbar puntos clave
  + Arquitectura Subsumption (Steel’s Mars Explorer)
* Armas
  + Calcular áreas (qué área coge más enemigos)
  + Máquina de estados (puede disparar a x enemigos, pero si se da otro estado cambia de objetivo)
* Aliados
  + Objetivo principal: acabar con enemigos
  + Percepción visual
  + Moverte hacia donde crees que va el enemigo (predicción)

**Idea 2**

Héroe de la villa “Opal” tú misión es proteger a sus ciudadanos de las malvadas incursiones del grupo de ladrones “jaja”.

* Vista tercera persona o isométrica.
* Como héroe, no eliges moverte por el escenario, vas patrullando por las calles de la villa (se mueve automáticamente hacia los enemigos o hacia eventos que se activen).
* Los enemigos están escondidos en ciertas partes de la villa, con los que tendrás encuentros.
* Cuando se produce un encuentro la cámara para de general a cámara de combate (tú vs los enemigos) y deberás acabar con los enemigos disparándoles (click / tap).
* Al acabar con los enemigos del encuentro, si todavía quedan encuentros en el nivel, se dirigirá al siguiente evento (volviendo a la cámara isométrica o 3a persona).



**IA**

* Movimiento por el pueblo: NavMesh
* Cuando seleccionas un punto, si es un enemigo, disparas y si es una zona, te desplazas
* Enemigos
  + Percepción visual (si te detectan atacan)
  + Decisiones en base a su estado (vida, munición, posición...)
  + Reacciones a tus acciones (si te cubres a la izquierda que busque un punto donde atacarte, si estas atacando que busque cómo cubrirse)
  + Que tengan en cuenta a sus aliados (no dispararse, colaborar, apoyarse, ¿coordinarse entre ellos?)

**Idea 3**

El *marshal* de villa “Opal” está muy ocupado debido a las fiestas del pueblo, y para colmo se comenta entre los habitantes que un ladrón anda suelto. Tu trabajo te obliga a encontrarlo, pero da la casualidad de que se ha mimetizado con la gente del pueblo. Por suerte algunos de ellos le han visto y tienen ciertas indicaciones para ayudarte. Acercándote a ellos, si le han visto, podrán decirte detalles sobre el ladrón, como, por ejemplo: lleva gorro de copa, una pajarita y creo que era de color verde; tiene dos ojos, cuernos y creo que guantes; es de color rosa, lleva capucha y creo que dos piernas... Todo ello será mediante símbolos e iconos en bocadillos, no hay diálogos con texto.

**IA**

* Movimiento por el pueblo: NavMesh + entorno dinámico (el ladrón puede obstaculizarte poniendo obstáculos)
* Comportamiento ladrón:
  + Tiene un objetivo principal: “robar a los aldeanos sin ser descubierto” (¿BT?)
  + Sistema de percepción visual (cono por ej) (si te ve, no roba)
  + Movimiento similar a los aldeanos para despistar
  + Almacena información falsa para despistar
  + Si el ladrón ha sido descubierto huye
* Comportamientos aldeanos:
  + Funcionan como Smart Objects (almacenan información sobre lo que han visto, como, por ejemplo, una prenda del ladrón).
  + Movimiento FLOCKING SEPARACIÓN
  + Objetivos de navegación (que vayan hacia x puntos del mapa y cuando lleguen al objetivo se paren un rato y vayan a otro sitio)
  + ¿Se solapan los comportamientos?

**Idea 4**

Beat’em up con comandos (Major Mayhem, patapon).

**Idea 5**

Juego similar a **You Died But a Necromancer Revive You (Redungeon, sprint RPG).**