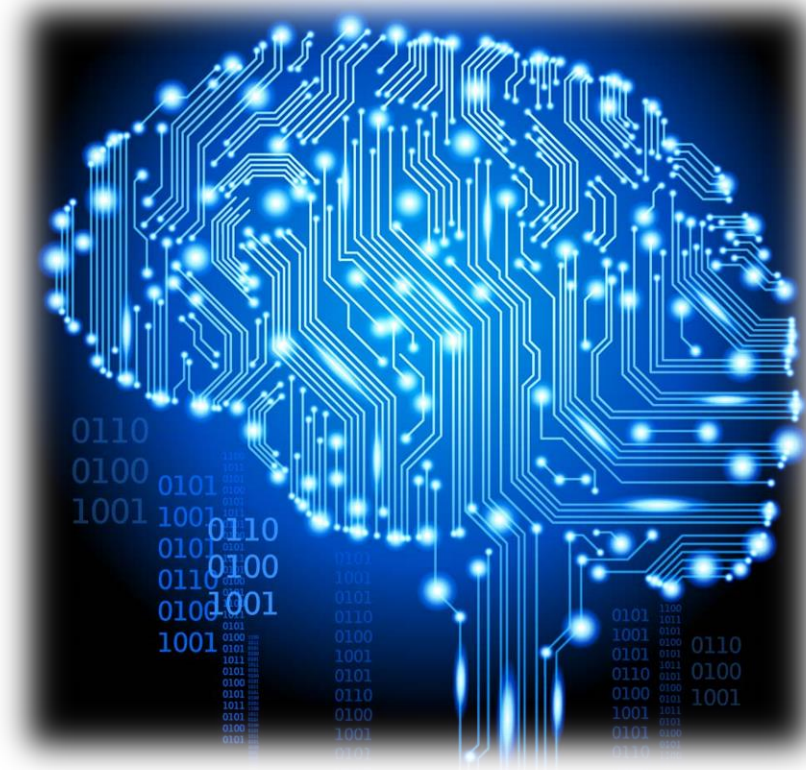


Introducción comportamientos



Javier Lucas Gómez

Programador de Inteligencia Artificial
Lead de Automatización
MercurySteam

Inteligencia artificial

IA en videojuegos:

Qué, Quién, Cómo, Cuándo, Dónde

Scripting

FSMs

Planificadores

Behavior Trees

Inteligencia artificial

IA en videojuegos:

Qué, Quién, Cómo, Cuándo, Dónde

Scripting

FSMs

Planificadores

Behavior Trees

Inteligencia artificial



Inteligencia Artificial

Rama de las ciencias computacionales encargada de estudiar modelos de cómputo capaces de realizar actividades propias de los seres humanos.

Singularidad

Advenimiento hipotético de la Inteligencia Artificial General (o IA fuerte) cuya capacidad intelectual superará al ser humano.

IA fuerte vs IA débil



IA fuerte

También llamada **Inteligencia Artificial General**: Potencial teórico de la IA para simular la cognición humana y tomar decisiones complejas en distintos contextos.



IA débil

También llamada **Inteligencia Artificial Estrecha**: IA aplicada a una tarea única y bien definida, como una traducción o el reconocimiento facial.

IA débil



IA en videojuegos: ¿Qué?

¿Qué debe tener una Inteligencia Artificial (IA) en un videojuego?

1. Ser **divertida**: que suponga un reto
2. Ser **predecible** (o por lo menos un poco)
3. **No** parecer **estúpida**

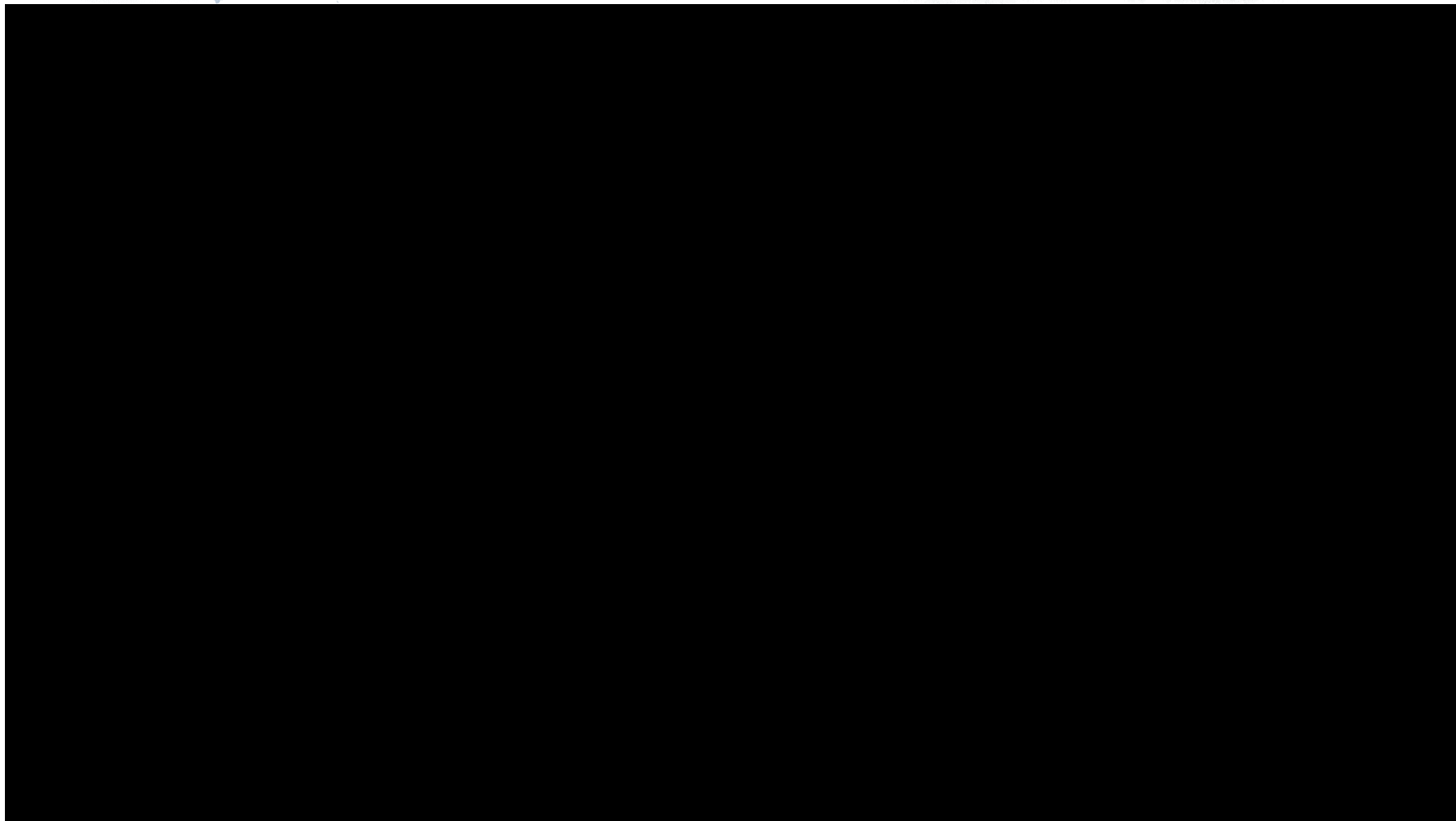
“If the player can't read the AI's behaviour, it just seems random”

Christiaan Moleman - Animator at Ubisoft

“Game AI is seldom about any deep intelligence but about the illusion of intelligence”

Steve Ravin- Nintendo, GDC AI Summit founder

IA en videojuegos: ¿Qué?



IA en videojuegos: ¿Quién?

¿Quién participa en la creación de una IA en un videojuego?

Programadores: Todo super genérico y eficiente

Diseñadores: Todo super versátil y configurable

Artistas: Todo super bonito y complejo

En el **equilibrio** está el acierto...

IA en videojuegos: ¿Cómo?

¿Cómo implementar una inteligencia artificial en un videojuego?

Scripting: Programación pura y dura

FSM: Finite State machines: Máquinas de estados finitos

HTN: Hierarchical Task Networks - Red jerárquica de tareas

BTs: Behavior Trees - Árboles de comportamiento

IA en videojuegos: ¿Cuándo?

¿Cuándo implementar IA en un videojuego?

La IA: Desde el principio de los videojuegos

Yo: De 8 a 16 de Lunes a Viernes



IA en videojuegos: ¿Dónde?

MERCURYSTEAM



Inteligencia artificial

IA en videojuegos:

Qué, Quién, Cómo, Cuándo, Dónde

Scripting

FSMs

Planificadores

Behavior Trees

Scripting: Programación pura y dura

Pseudocódigo

```
if(HaveAmmo())  
{  
    Shoot();  
}  
else  
{  
    Reload();  
}
```


Scripting: Programación pura y dura

PROS

- Flexibilidad
- Computacionalmente muy barato

CONTRAS

- Conocimientos de programación necesarios
- Mantenimiento arduo y difícil escalabilidad
- Comprobar si las acciones han funcionado o han fallado.
- No comprobarlo genera bugs.

Scripting: Programación pura y dura

Pseudocódigo

```
void AttackWhileHidding()  
{  
    Hide();  
    while(!IsHidden())  
    {  
        if(HaveAmmo())  
        {  
            Shoot();  
        }  
        else  
        {  
            Reload();  
        }  
        KeepHidding();  
    }  
}
```

Inteligencia artificial

IA en videojuegos:

Qué, Quién, Cómo, Cuándo, Dónde

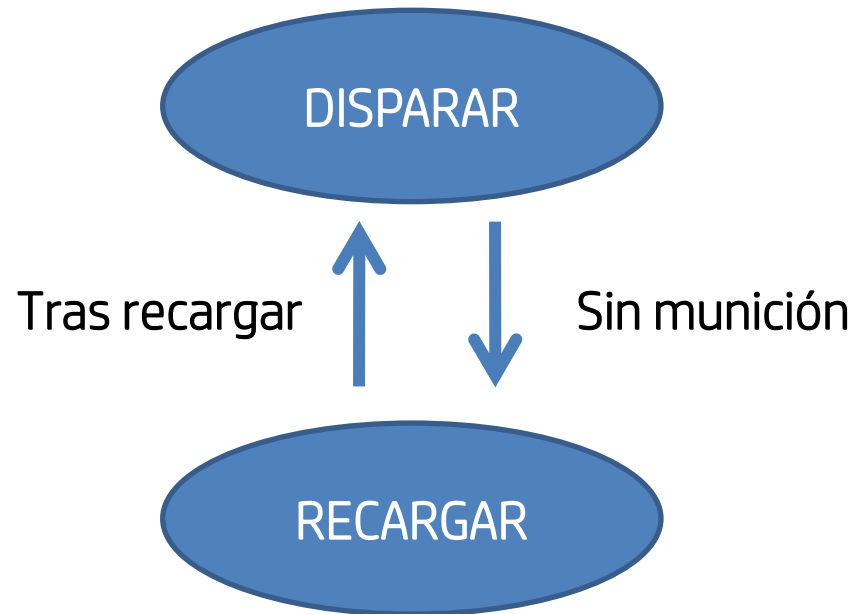
Scripting

FSMs

Planificadores

Behavior Trees

FSM: Finite State machines



FSM: Finite State machines

Históricamente más utilizadas

Conjunto de estados y transiciones

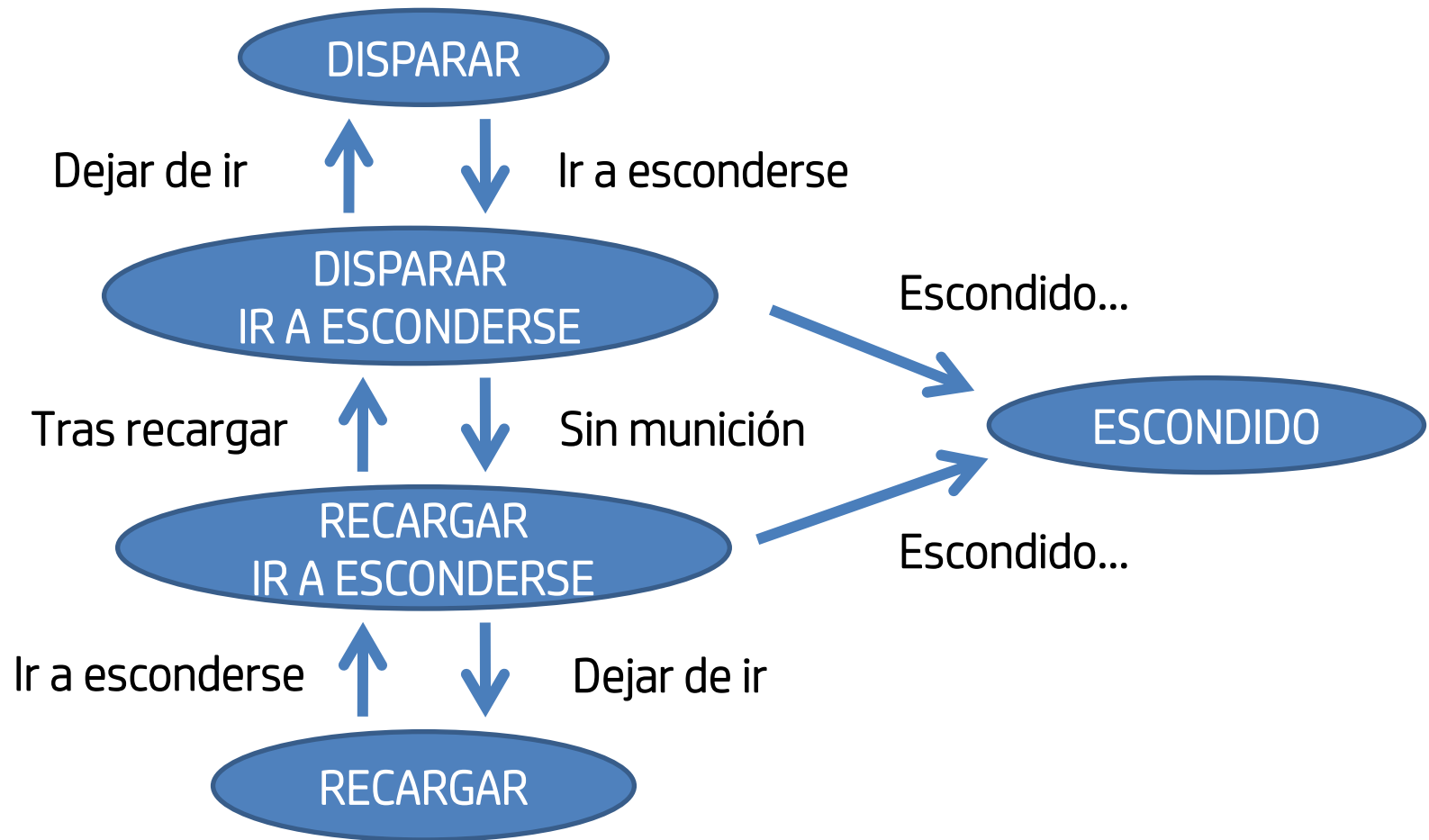
PROS

- Representación visual simple e intuitiva
- No requieren conocimientos de programación
- Computacionalmente barato
- Gran control del diseñador

CONTRAS

- Problemas cuando el número de transiciones es muy grande, crece muy rápido si el problema a resolver se hace más complejo.

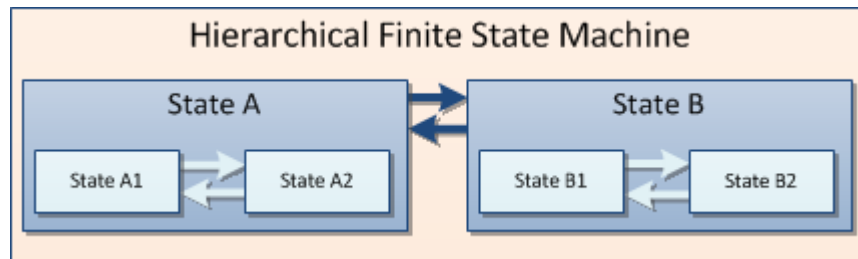
FSM: Finite State machines



FSM: Finite State machines

HFSMs (Hierarchical FSMs)

- Máquinas de estados jerárquicas para **mayores complejidades**
- **Reusabilidad** de transiciones, habiendo agrupado estados con mismas transiciones.
- Crear una secuencia es difícil porque hay que poner transiciones “acabado” de los estados, y transiciones hacia fuera para los posibles fallos.



Inteligencia artificial

IA en videojuegos:

Qué, Quién, Cómo, Cuándo, Dónde

Scripting

FSMs

Planificadores

Behavior Trees

HTN: Hierarchical Task Networks

Red jerárquica de tareas

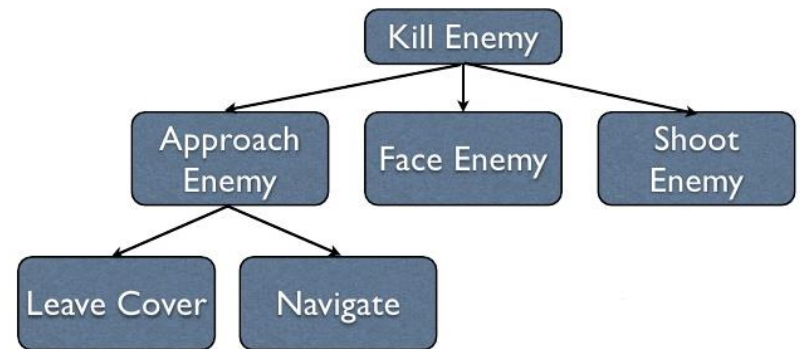
- Algoritmo de **planificación** automática que crea un plan por descomposición de tareas en subtareas hasta lograr primitivas que pueden ser ejecutadas directamente.
- Muy usadas en **robótica**

PROS:

- Autonomía para la IA
- Búsqueda para creación de planes
- Dirigidas por objetivos

CONTRAS:

- La búsqueda puede llegar a ser MUY costosa
- Poco control de los diseñadores



Inteligencia artificial

IA en videojuegos:

Qué, Quién, Cómo, Cuándo, Dónde

Scripting

FSMs

Planificadores

Behavior Trees

Behavior Trees: Árboles de comportamiento

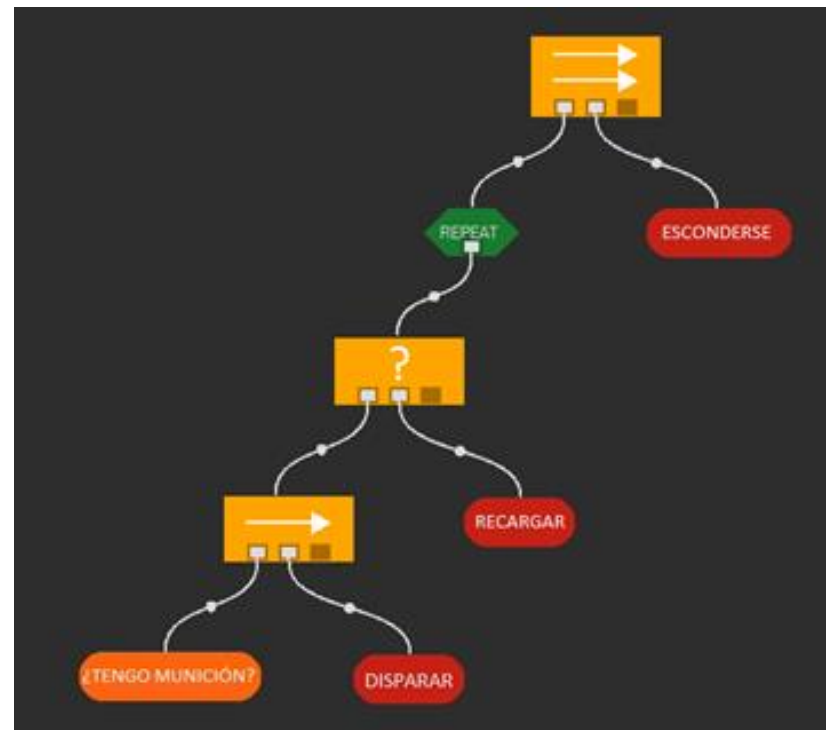
Árbol

Jerárquico

Flujo de decisión

Inteligencia artificial

Un agente



Behavior Trees: ¿por qué molan?

- Son **Intuitivos**
- **Prototipados** e **iteraciones** rápidas.
- Más **escalables, mantenibles y versátiles** que las FSMs
- **Modulares**: Más reutilizables; complejidad por combinación
- **Sub-árboles reutilizables** entre comportamientos y personajes (algo clave en el desarrollo de Raiders).
- **Facilidad de extensión** de nuevos nodos.

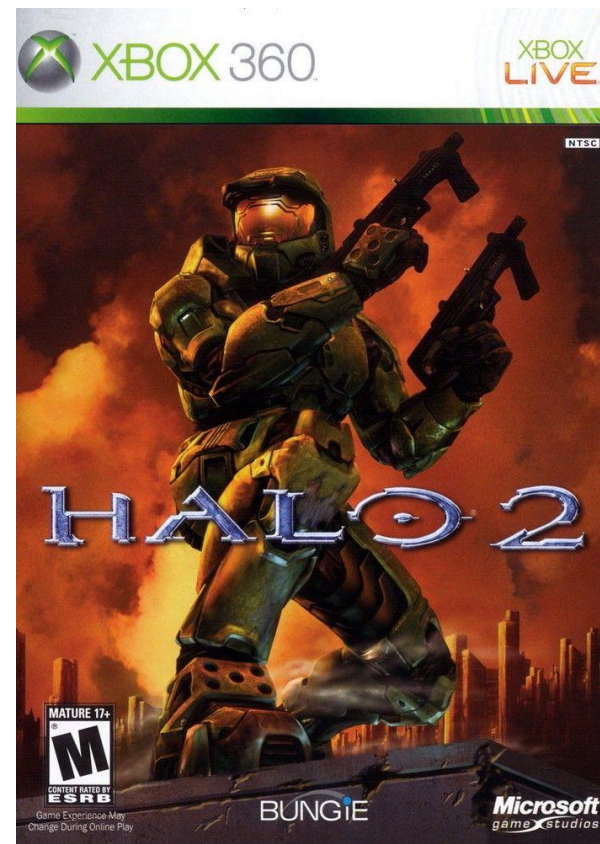
Behavior Trees: ¿contras?

- Ocupan mucha **memoria**
- Pueden ser complicados de **depurar** -> necesidad de depurador gráfico
- Hace falta aprender a generar árboles con un **buen diseño**
- Si se hacen muy complejos pueden llegar a ser **costoso ejecutarlos**

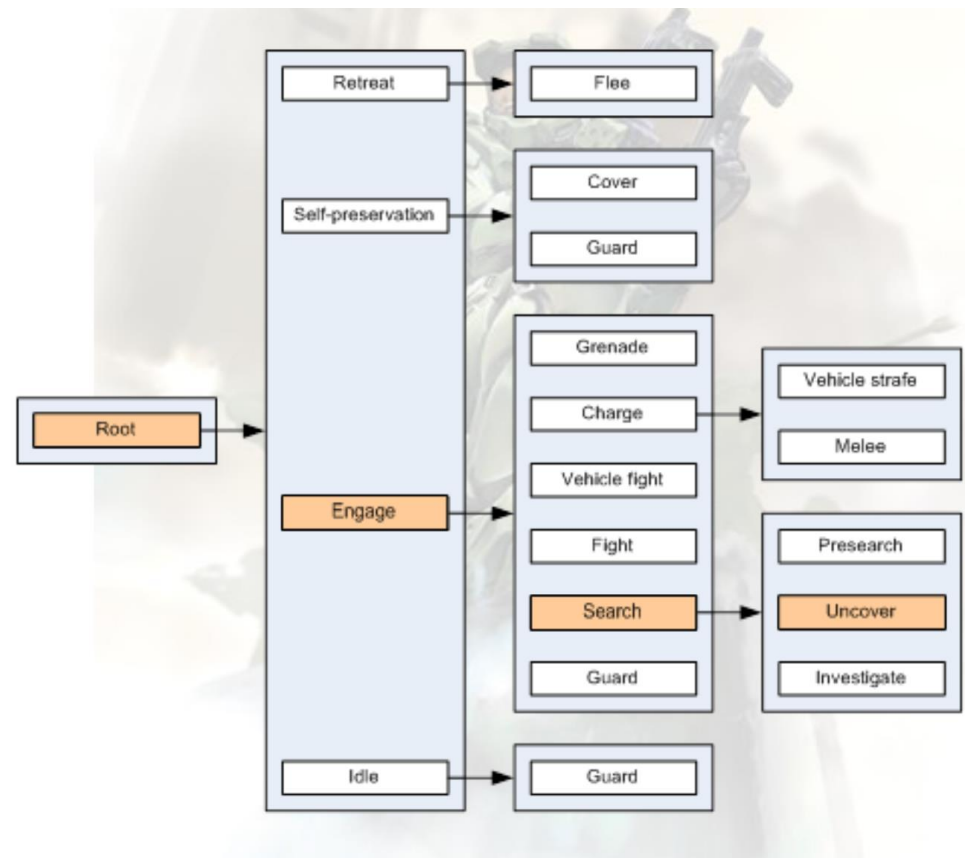
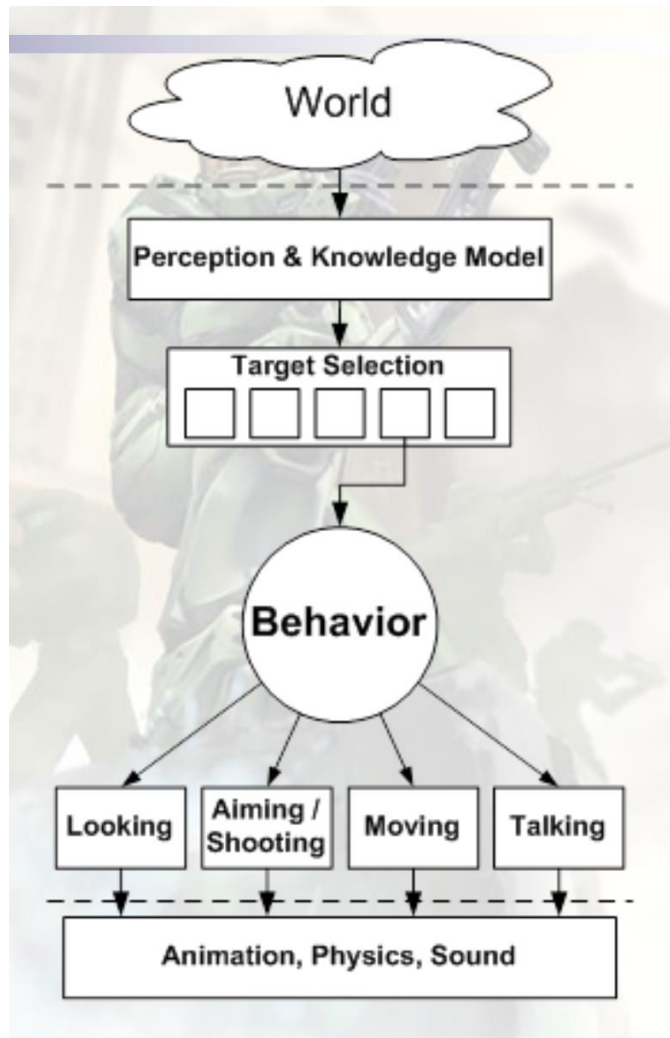
Historia de los Behavior Trees

Managing Complexity in the
Halo 2 AI System
Damian Isla
Bungie Studios
GDC 2005

“The Brute Force Approach to
Common Sense”



Halo2: Behavior Trees



BT's vs FSM's y Planificadores

- ✓ **Estados:** Acciones
- ✓ **Transiciones:** Nodos de control y condiciones
- ✓ **Prioridades:** Estructura jerárquica
- ✓ **BT:** Planificación reactiva
- ✓ **BT:** Planificación compilada

Referencias

Mark Deloura (Ed.) (2000). Game Programming Gems. Massachusetts: Charles River Media
Boston Dynamics reserach and resources

Christiaan Moleman : <https://ninjadodo.wordpress.com/>

AiGameDev.com

Rabin, Steve (Ed.) (2014). Game AI Pro: Collected Wisdom of Game AI Professionals. New
York: A K Peters/CRC Press.

GDC '15 Managing Complexity in the Halo 2 AI System - Damian Isla - Bungie Studios