



Índice:

1. Introducción a la IA para Videojuegos

2. Modelos de IA en Videojuegos

2.1 Juegos con adversario

2.2 Búsqueda de caminos

2.3 Mecanismos de toma de decisión

2.3.1 Máquinas de estados finitos

2.3.2 Árboles de decisión

2.4 Árboles de comportamiento

3. Bibliografía y Lecturas Complementarias



2.3 Mecanismos de toma de decisión

- Genéricamente, serán aquellas técnicas que nos asistan en la tarea de implementar toma de decisiones de los *Non Player Characters* (NPCs).
- En una primera aproximación se nos ocurriría hacer algo así:

```
if (health < 30)
    Go find health
if (health < 50) && (health nearby) && (no enemies)
    Go find health
if (better weapon nearby)
    Get better weapon
...
```



- Mecanismos de toma de decisión (continuación)
 - Problemas:
 - ... obviamente, trabajar así, conduce a un nivel de complejidad muy elevado y creciente...
 - Poco parametrizable y ajustable
 - Difícil de crear
 - Difícil de manejar
 - Difícil de leer tras algún tiempo (mantener)
 - Difícil de depurar
 - Por ello, se recurre a mecanismos concretos... como los que se verán a continuación.



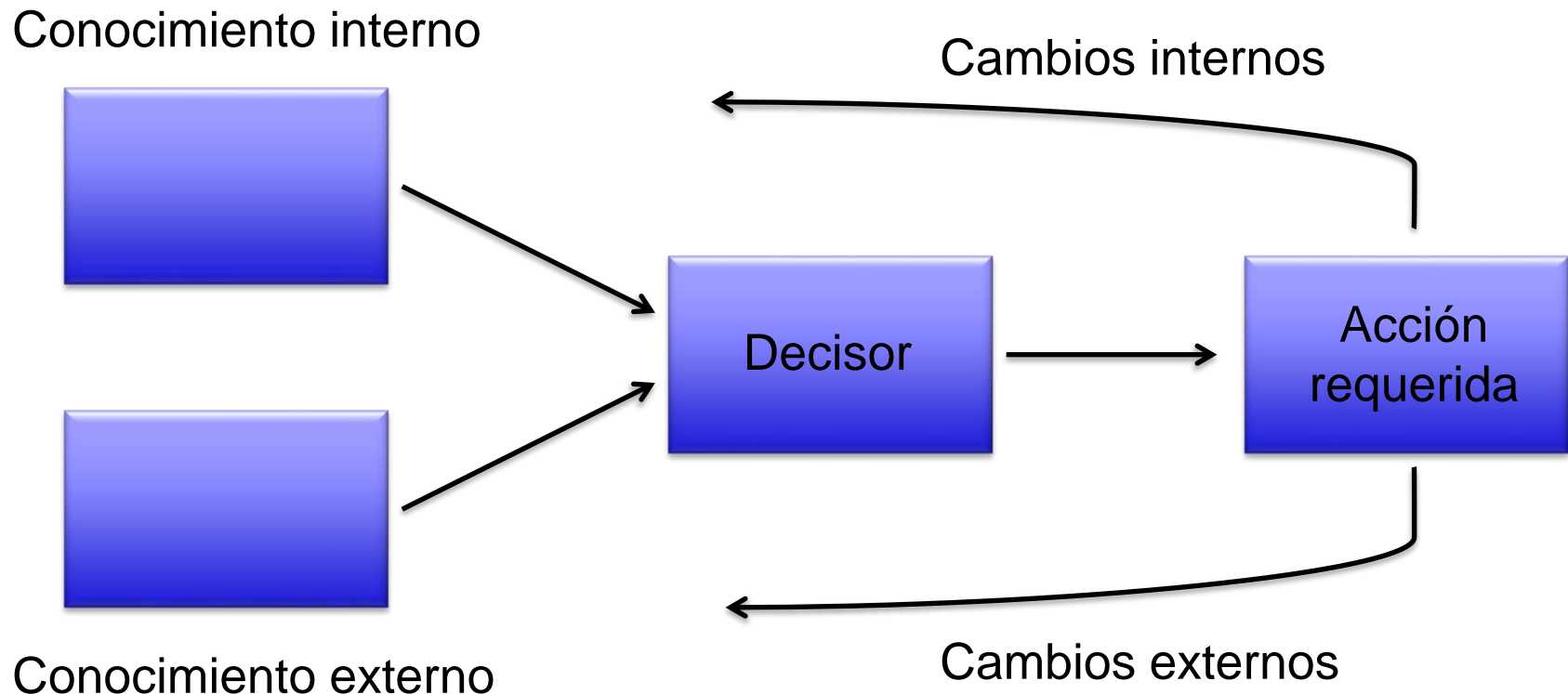
- Mecanismos de toma de decisión (continuación)
 - Los mecanismos de toma de decisiones pueden actuar tanto entre personajes como dentro de un personaje. Los más básicos, y los más empleados, son:
 - Máquinas de estado finito
 - Árboles de decisión
 - Los **sistemas basados en reglas** son menos habituales, pero importantes.
 - Recientemente se utilizan mecanismos más complejos, como es **la lógica difusa**, o **las redes neuronales artificiales**. – Si bien, los programadores de IA en videojuegos por regla general, no los dominan y les cuesta hacerlos funcionar correctamente.



- Mecanismos de toma de decisión (continuación)
 - Si nos centramos en la **toma de decisión de un personaje**, la entrada es el **conocimiento** que él posee, y la salida es la **acción a realizar**. Ese conocimiento que posee es de dos tipos:
 - **Externo**: es el que le llega del entorno exterior, como la posición de otros personajes, estado del terreno, sonidos de otros personajes...
 - **Interno**: es el estado de sus propias variables internas o procesos como su salud, sus últimos logros, de dónde viene, etc.
 - Las acciones pueden suponer:
 - Modificar su **estado externo**: ubicación, cambios en el entorno, etc.
 - Modificar su **estado interno**: variables o estados: actitud, valoración, etc.



- Mecanismos de toma de decisión (continuación)





Índice:

1. Introducción a la IA para Videojuegos

2. Modelos de IA en Videojuegos

2.1 Juegos con adversario

2.2 Búsqueda de caminos

2.3 Mecanismos de toma de decisión

2.3.1 Máquinas de estados finitos

2.3.2 Árboles de decisión

2.4 Árboles de comportamiento

3. Bibliografía y Lecturas Complementarias



- Mecanismos de toma de decisión (continuación)
 - Ejercicio Propuesto. Al final de la explicación de estos dos **mecanismos** (*máquinas de estados finitos y árboles de decisión* (en el contexto en el que nos encontramos), **compare ambos** (ventajas, inconvenientes, etc.).



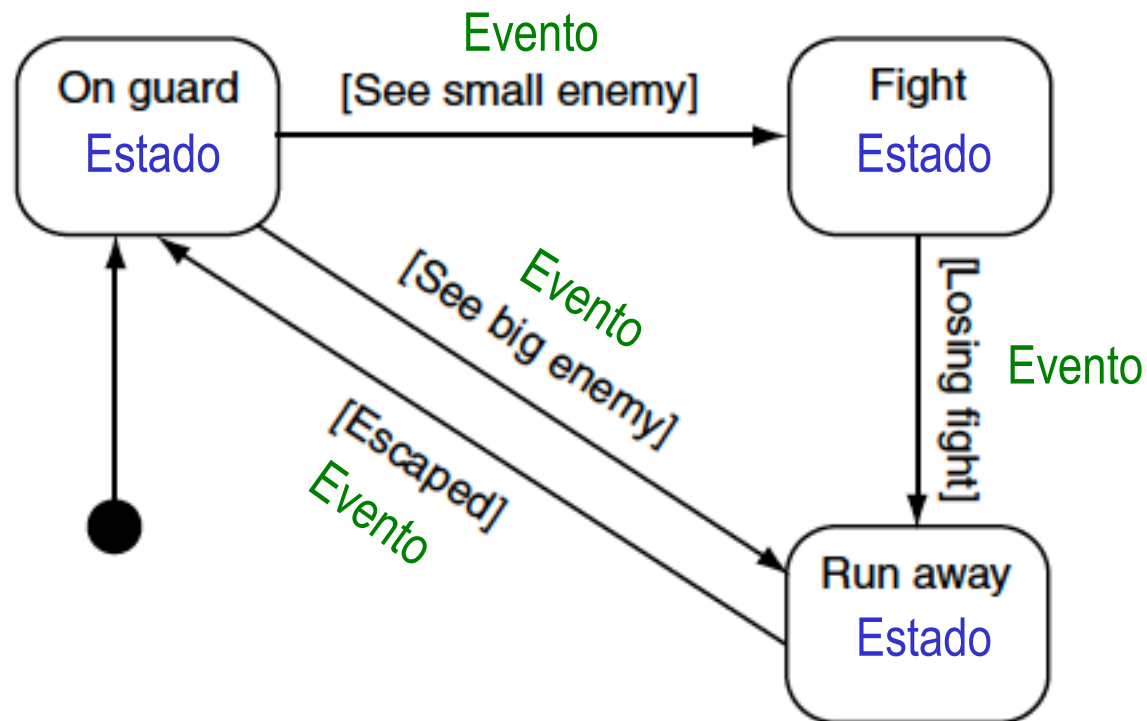


2.3.1 Máquinas de estados finitos

- Son un **mecanismo de decisión** que tiene representaciones abstractas formadas por estados y transiciones entre ellos.
- Cada **estado** supone una serie de **acciones** (Ej: movimiento, lanzar cuchillo, huir, perseguir, esconderse, etc.).
- Las **transiciones** se desencadenan cuando ocurren determinadas precondiciones o **eventos** (Ej: nos disparan, *pacman* se come una *power pill*, transcurre determinado tiempo, etc.).
- Existe toda una teoría tras ellos (que nos permite simplificar, traducir, convertir, verificar, etc.) muy útil cuando alcanzan ciertos niveles de complejidad.

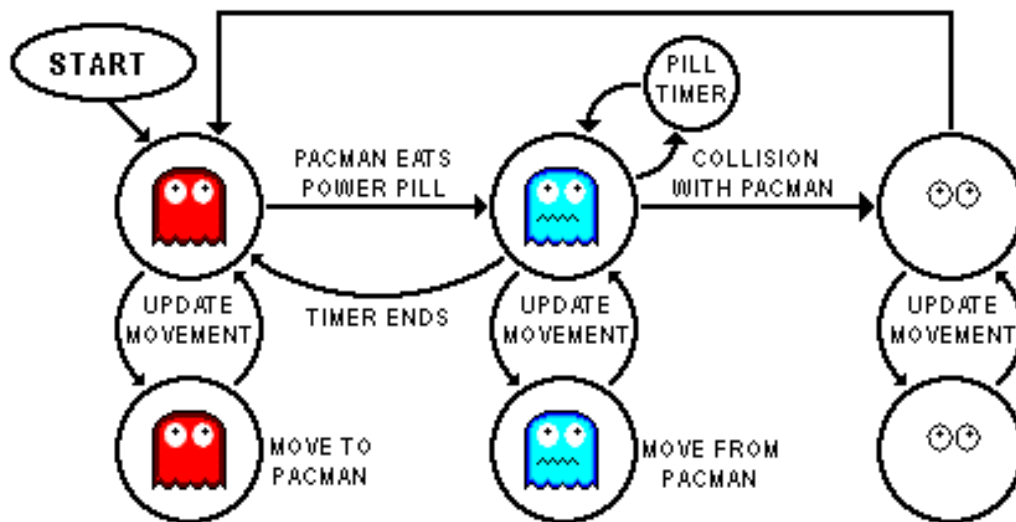


- Máquinas de estados finitos (continuación)
 - Ejemplos: Un esquema muy simple





- Máquinas de estados finitos (continuación)
 - Ej: Fantasma PacMan





- Máquinas de estados finitos (continuación)
 - No todos los estados pueden alcanzarse desde el estado **actual**, como en otros mecanismos de decisión (ej.: *árboles de decisión*): sólo los alcanzables, una vez se cumple su precondition.
 - Su rendimiento es de orden lineal, $O(m)$, siendo m el número de transiciones por estado.
 - En general, los juegos que utilizan máquinas de estados finitos, presentan un código menos claro que los que usan otras técnicas para la toma de decisión.



Índice:

1. Introducción a la IA para Videojuegos

2. Modelos de IA en Videojuegos

2.1 Juegos con adversario

2.2 Búsqueda de caminos

2.3 Mecanismos de toma de decisión

2.3.1 Máquinas de estados finitos

2.3.2 Árboles de decisión

2.4 Árboles de comportamiento

3. Bibliografía y Lecturas Complementarias



2.3.2 Árboles de decisión

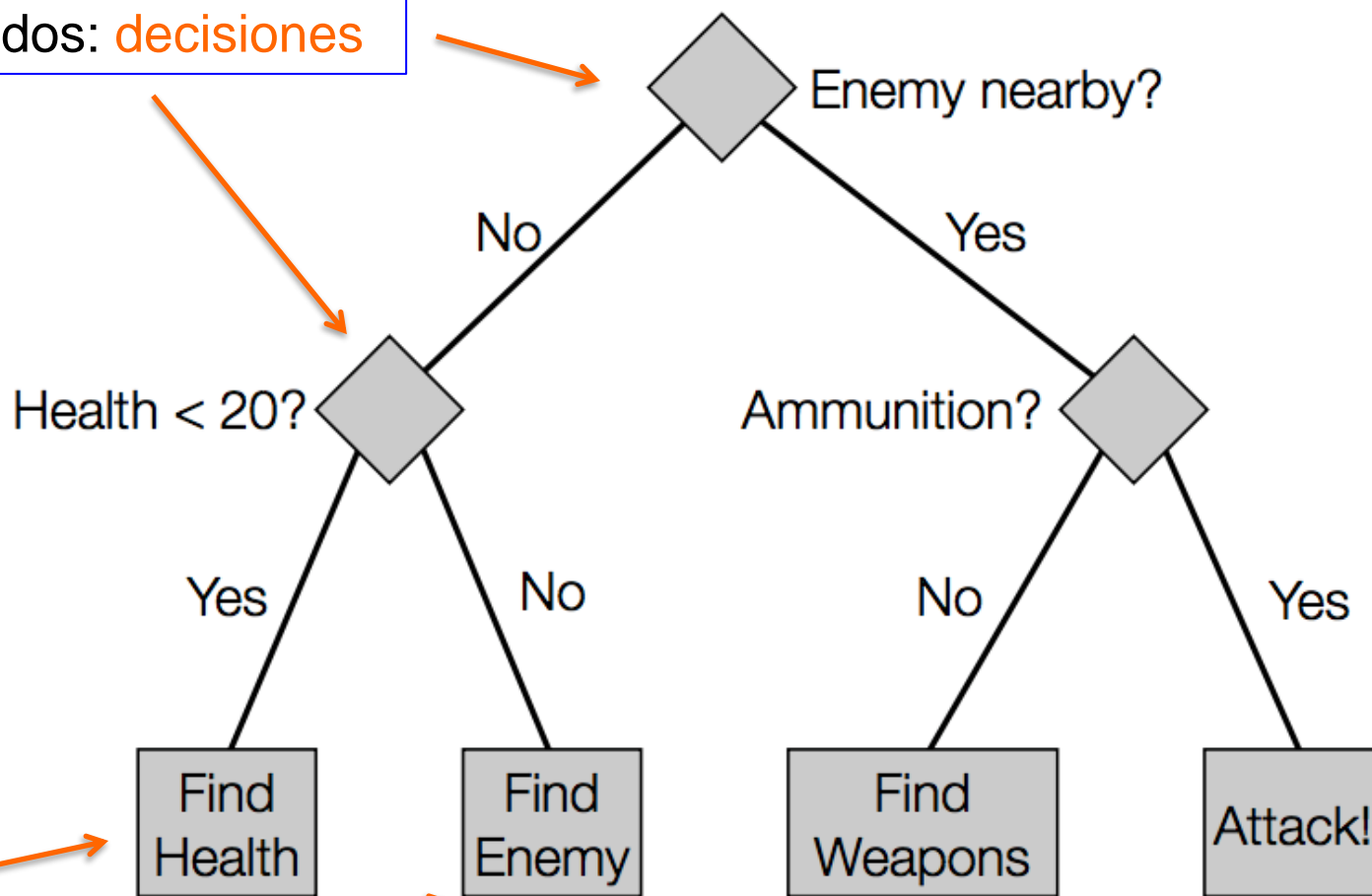
- **Mecanismo de decisión** que consiste en una estructura de conocimiento que permite **asociar acciones** con **selectores basados en conocimiento interno y externo**.
- Es un mecanismo **sencillo de implementar**, **rápido**, y **fácil de entender**, utilizado en los videojuegos tanto **para tomar decisiones**, como para el **control de animaciones**.





- Árboles de decisión (continuación)

Nodos: **decisiones**



Hojas: **Acciones**

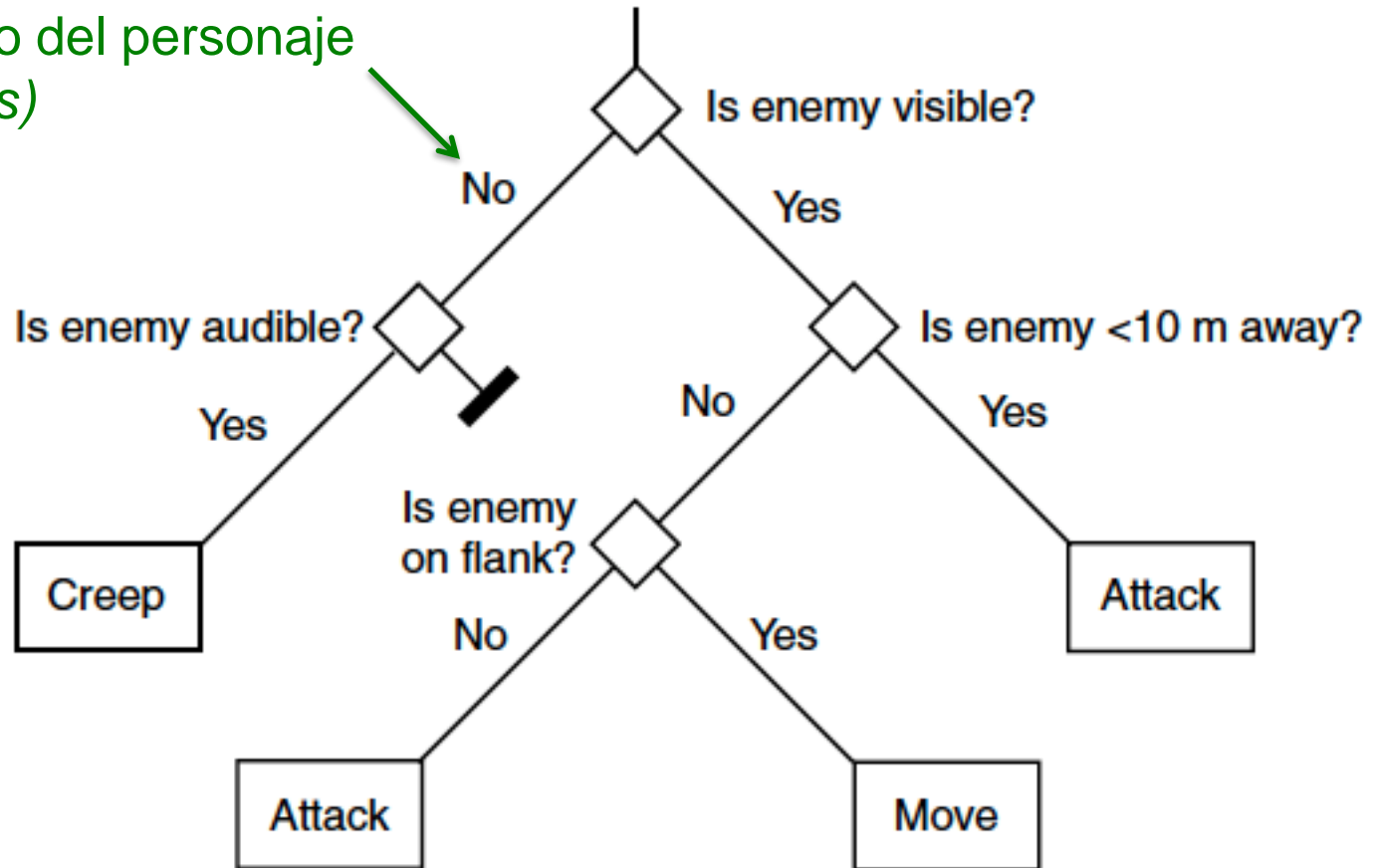


- Árboles de decisión (continuación)
 - Cada elección se basa en el conocimiento del personaje.
 - Cuando se alcanza una decisión, su acción asociada es ejecutada de modo inmediato.
 - La mayoría de los árboles de decisión se utilizan para tomar decisiones muy simples, entre sólo un par de opciones:
árboles de decisión binarios
 - La misma acción puede utilizarse en distintas decisiones, es decir, se puede alcanzar la misma acción desde diferentes condiciones previas.
 - No necesariamente sólo pueden ser respuestas booleanas



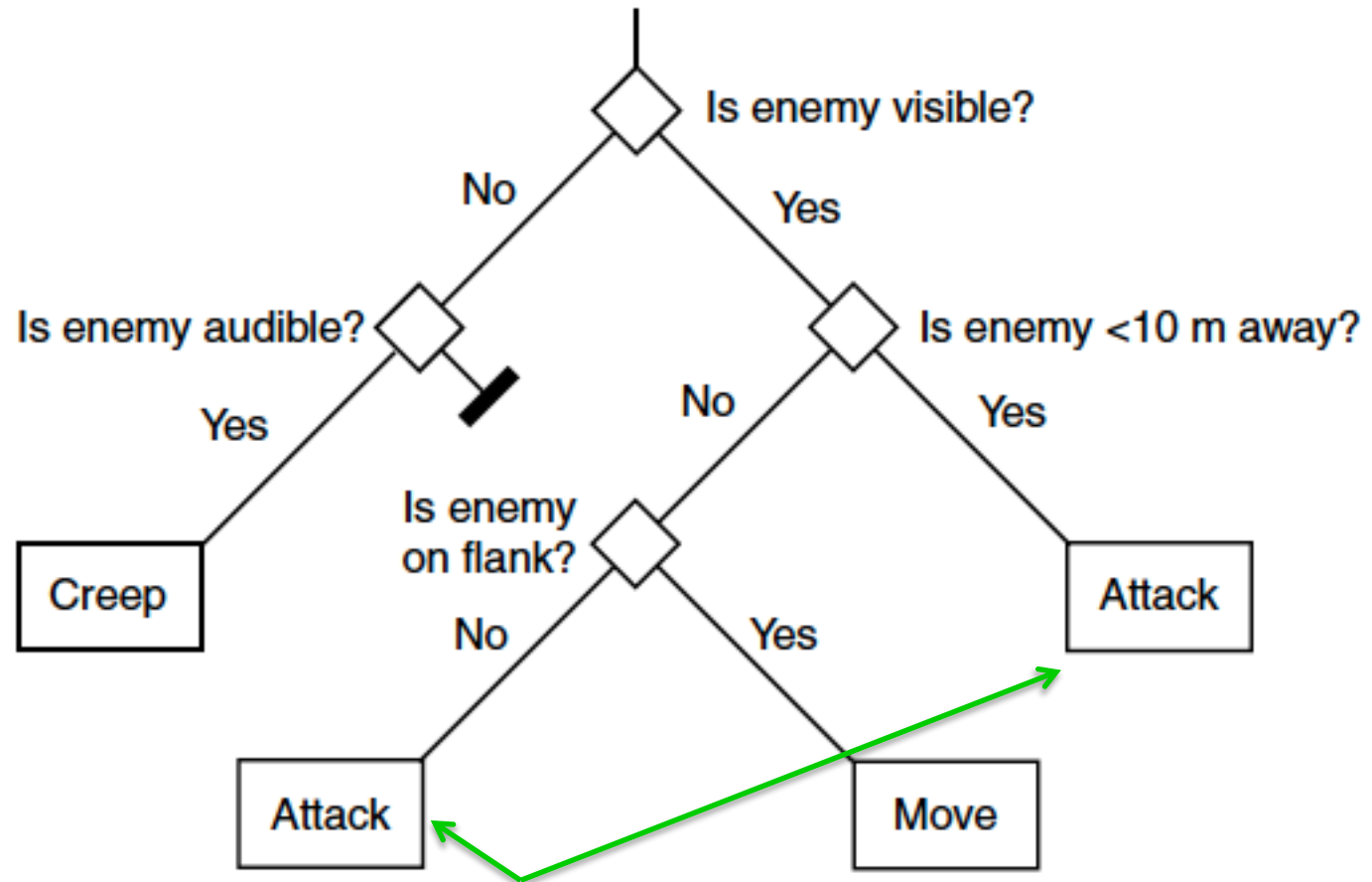
- Árboles de decisión (continuación)

Conocimiento del personaje
(en las ramas)





- Árboles de decisión (continuación)

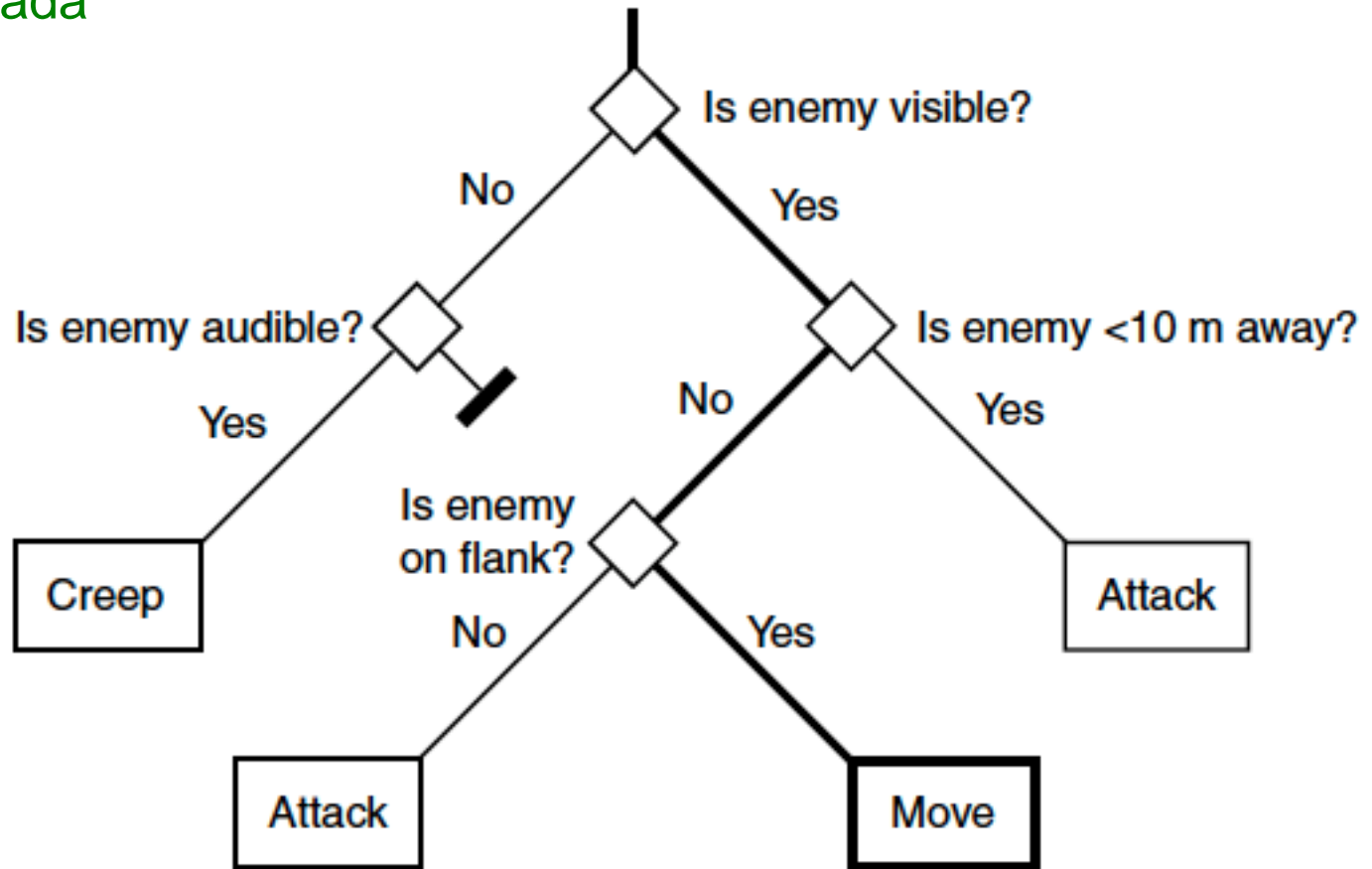


Misma acción en distintas *hojas*



- Árboles de decisión (continuación)

Decisión tomada





- Árboles de decisión (continuación)
 - Los árboles de decisión **pueden ser ampliados** fácilmente para contemplar nuevas situaciones: pueden construirse sencillos inicialmente, y completarlos posteriormente
 - El rendimiento es $O(\log_2 n)$ con el número de nodos si consideramos que cada decisión necesita el mismo tiempo, y que el árbol está idealmente balanceado (mismo número de hojas en cada rama).
 - Si no está balanceado, el rendimiento es $O(n)$
 - Es preferible tener árboles balanceados para que no sean más lentas unas decisiones que otras.



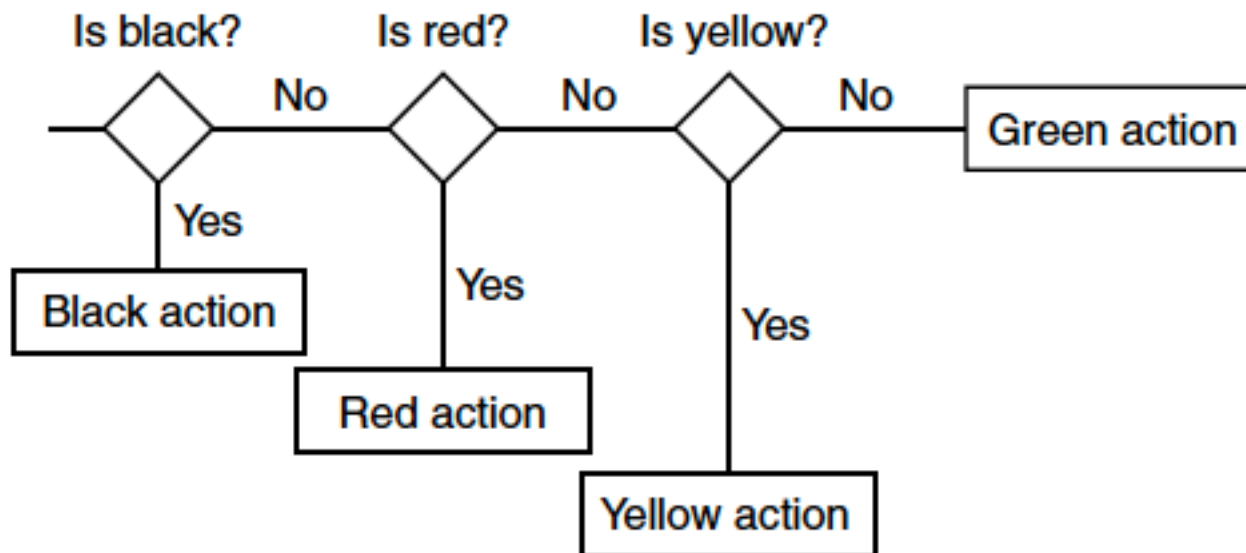
- Árboles de decisión (continuación)
 - Si tenemos evaluaciones que son muy costosas de realizar, entonces debemos **sacrificar** tener un árbol balanceado y directamente recurrir implementar mecanismos **que prioricen evaluar lo que es menos costoso**. Ejemplo:
 - Si una de las evaluaciones de nodos intermedios es la distancia de todos los enemigos, entonces quizás deben evaluarse antes las demás posibilidades que tener que decidir en base a esto, que nos llevará más esfuerzo.





- Árboles de decisión (continuación)

Otra forma de representar
un árbol de decisión binario

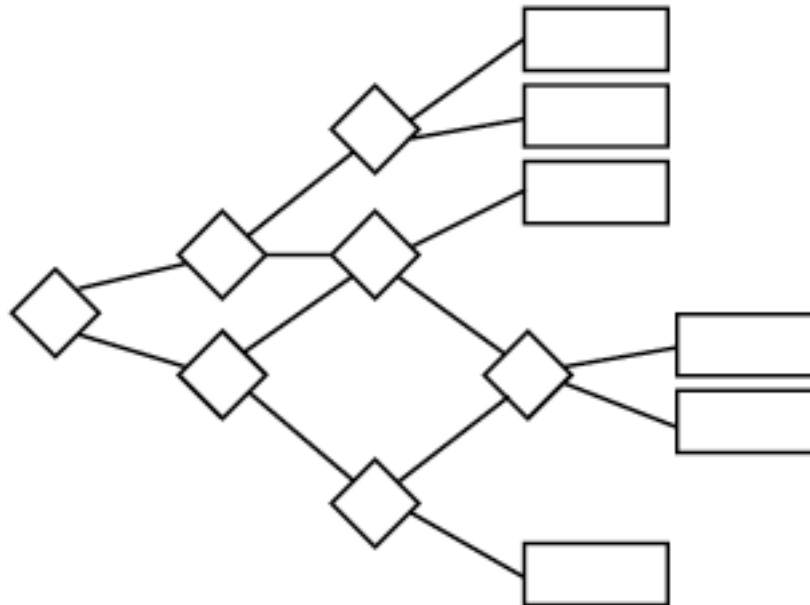




- Árboles de decisión (continuación)

Se pueden crear también este tipo de estructuras u otras:

Cuidado con los ciclos



Los ciclos podrían surgir con determinados enlaces hacia atrás...

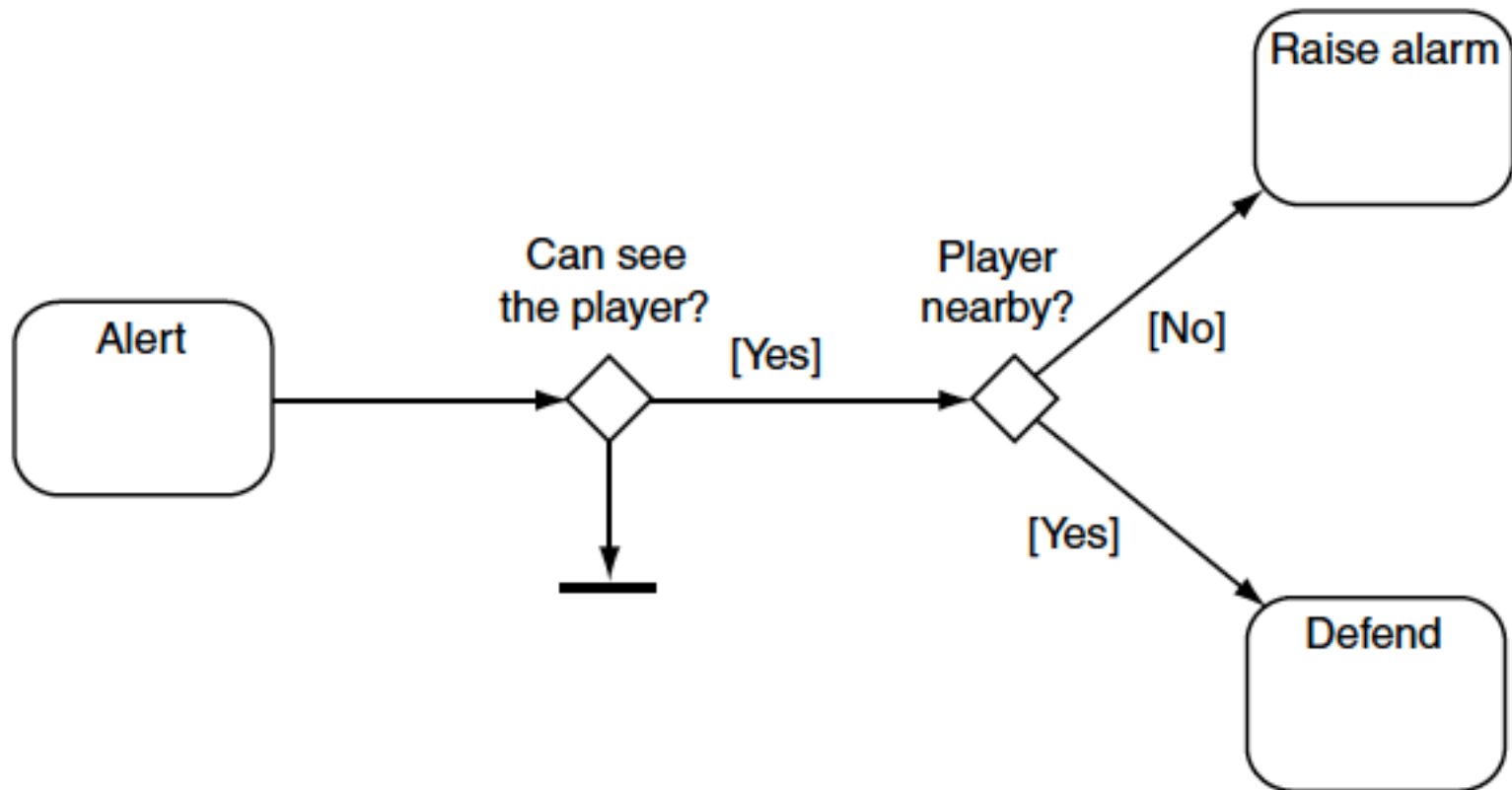


- Árboles de decisión (continuación)
 - Para poder conseguir que la actuación del personaje sea **menos predecible**, se pueden implementar nodos en los que exista **aleatoriedad**, o **una distribución de probabilidad de no tomar la mejor decisión**.
 - Este mecanismo puede ser regulable
 - **Riesgo**: Puede producir **actitudes titubeantes** en el personaje si no se controla adecuadamente:
 - Cuando en cada iteración, la misma evaluación tiene diferente resultado y el personaje actúa haciendo una cosa y la contraria.



- Árboles de decisión (continuación)

Se pueden **combinar** árboles de decisión con máquinas de estados finitos: Autómatas con árboles de decisión entre los estados





- Árboles de decisión (continuación)
 - Ejercicio Propuesto. Redacte ahora su comparación entre ambos mecanismos: *máquinas de estados finitos y árboles de decisión*, en el contexto en el que nos encontramos.





- Árboles de decisión (continuación)
 - ¿Cómo utilizar Máquinas de EF en Unity 3D?
 - Implementándolos por nosotros mismos.
 - Adquiriendo el *plug-in* (asset) en la Asset Store

Requires Unity 4.5.0 or higher.
Finite State Machine

This is a straight forward and robust state based controller developed to provide programmers with a light weight option for implementing a finite state machine in their applications. It is easy to use and does not require the GUI builder if you do not want to use it.

The state controller allows you to build state machine controllers for a wide range of applications.

Training Videos

<https://www.youtube.com/watch?v=J5V-LtONJ54>

<https://www.youtube.com/watch?v=Zhx-rH7-54A>

<https://www.youtube.com/watch?v=iXtCDYmXSDY>

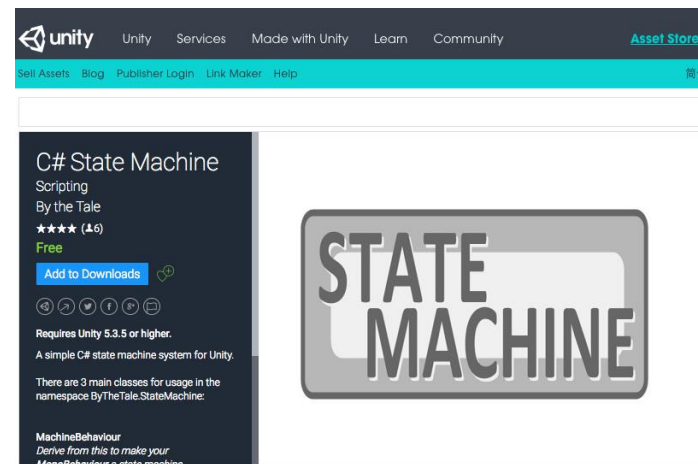
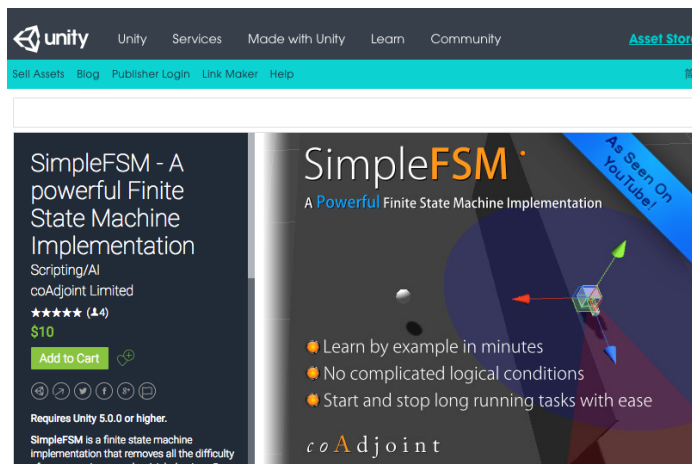
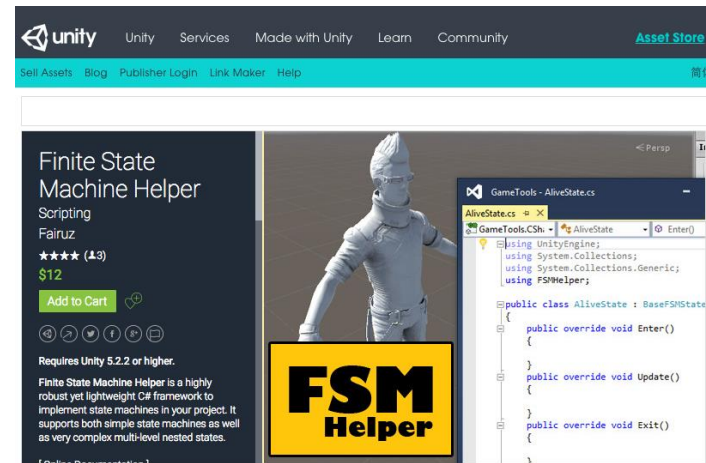
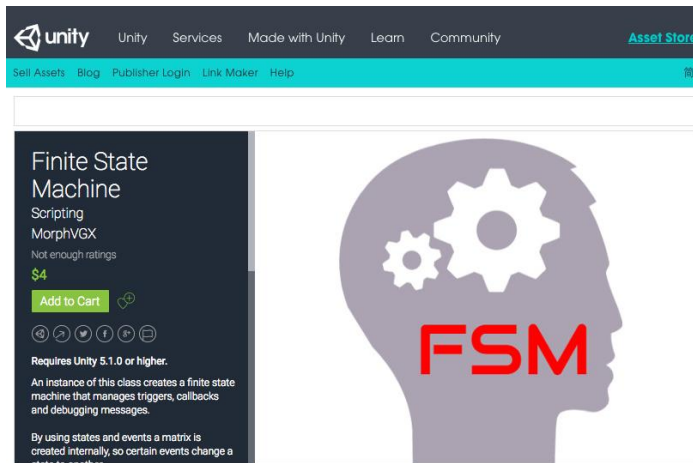
Please note that this is not a visual scripting tool designed to allow non-programmers to graphically build applications. It is an asset that allows you to quickly and easily build the underlying application structure. As such it is not recommended for people unfamiliar with programming in Unity.

The controllers contain callbacks for each of the states that fire when the state is entered by the controller. ...

The screenshot shows the Unity Asset Store interface. At the top, there's a navigation bar with links like 'unity', 'Unity', 'Services', 'Made with Unity', 'Learn', 'Community', and 'Asset Store'. Below this is a search bar. The main content area displays the 'Finite State Machine' asset by Nerdicus Rex LLC. The asset is priced at \$10 and has an 'Add to Cart' button. The description states: 'This is a straight forward and robust state based controller developed to provide programmers with a light weight option for implementing a finite state machine in their applications. It is easy to use and does not require the GUI builder if you do not want to use it.' To the right of the text is a diagram of a finite state machine with three states represented by rounded rectangles. Arrows indicate transitions between the states: a horizontal arrow from left to right, a vertical arrow from top to bottom, and a curved arrow from right to top.



- Árboles de decisión (continuación)
 - ¿Cómo utilizar Máquinas de EF en Unity 3D? – Diversos assets





- Árboles de decisión (continuación)
 - ¿Cómo utilizar **Árboles de Decisión** en Unity 3D?
 - Implementándolos por nosotros mismos.
 - Adquiriendo el *plug-in (asset)* en la Asset Store

Requires Unity 5.4.1 or higher.

Decision Tree Toolkit provides you with a pre-built kit that you can use to build decision trees for your NPC and gameplay AI and is optimized for decision making speed.

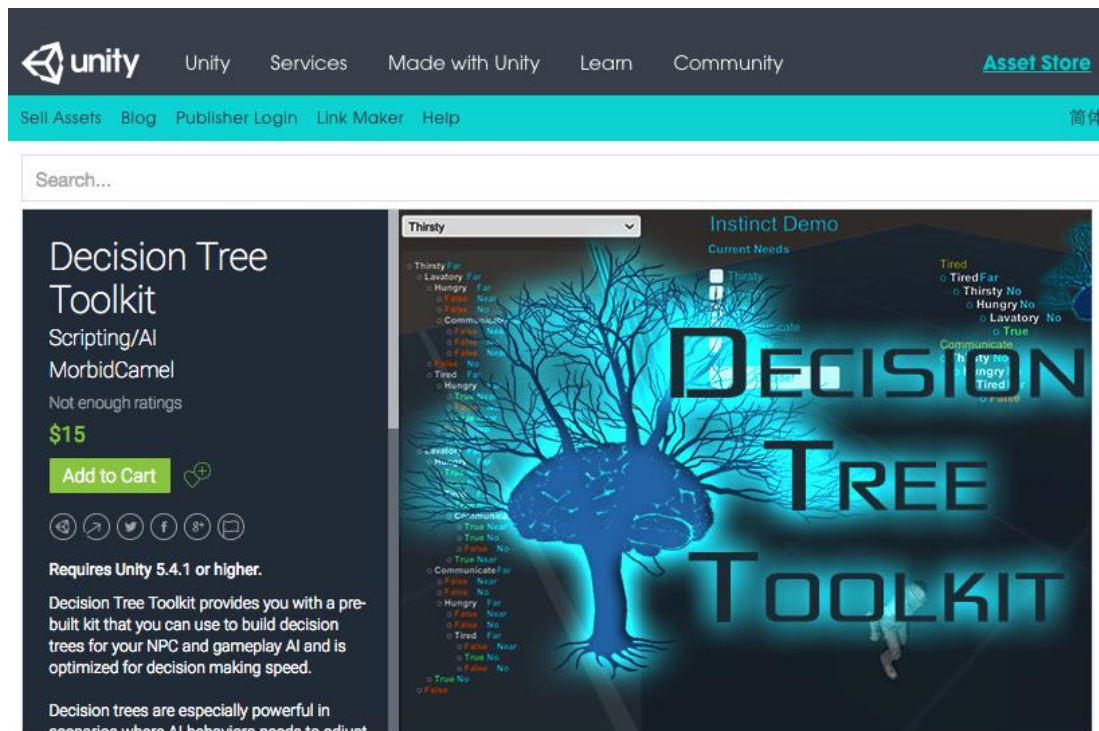
Decision trees are especially powerful in scenarios where AI behaviors needs to adjust dynamically during gameplay and a must have for RPG and strategy games that will keep the player guessing. Decision trees are also very useful in FPS bot learning to dynamically learn a player's habits and therefore make it even more challenging for a player.

For decision tree learning an implementation of the ID3 (Iterative Dichotomiser 3) algorithm is used to generate a decision tree from a dataset.

The dataset is in the form of a SQLite database that you ship with your game and the toolkit provides a pre-built database that can be customized for your needs. You can also update the dataset dynamically and rebuild the decision trees to have dynamic intelligence in your game or to implement machine learning based on the players actions.

Decision trees are fast because they only have to scan the dataset once and then derive a tree from it.

The toolkit provides a way to build multiple decision trees from a single dataset using configurable outcomes and makes it easy to nest decision behaviors for more complex learning solutions.





Índice:

1. Introducción a la IA para Videojuegos

2. Modelos de IA en Videojuegos

2.1 Juegos con adversario

2.2 Búsqueda de caminos

2.3 Mecanismos de toma de decisión

2.3.1 Máquinas de estados finitos

2.3.2 Árboles de decisión

2.4 Árboles de comportamiento

3. Bibliografía y Lecturas Complementarias



2.4 Árboles de comportamiento (*behaviour trees*)

- Técnica de moda para **modelar IA** en videojuegos
- Se usan para **organizar el comportamiento** de agentes, y facilitar así el trabajo organizativo (en videojuegos pero también en robots).
- Populares por su uso en:

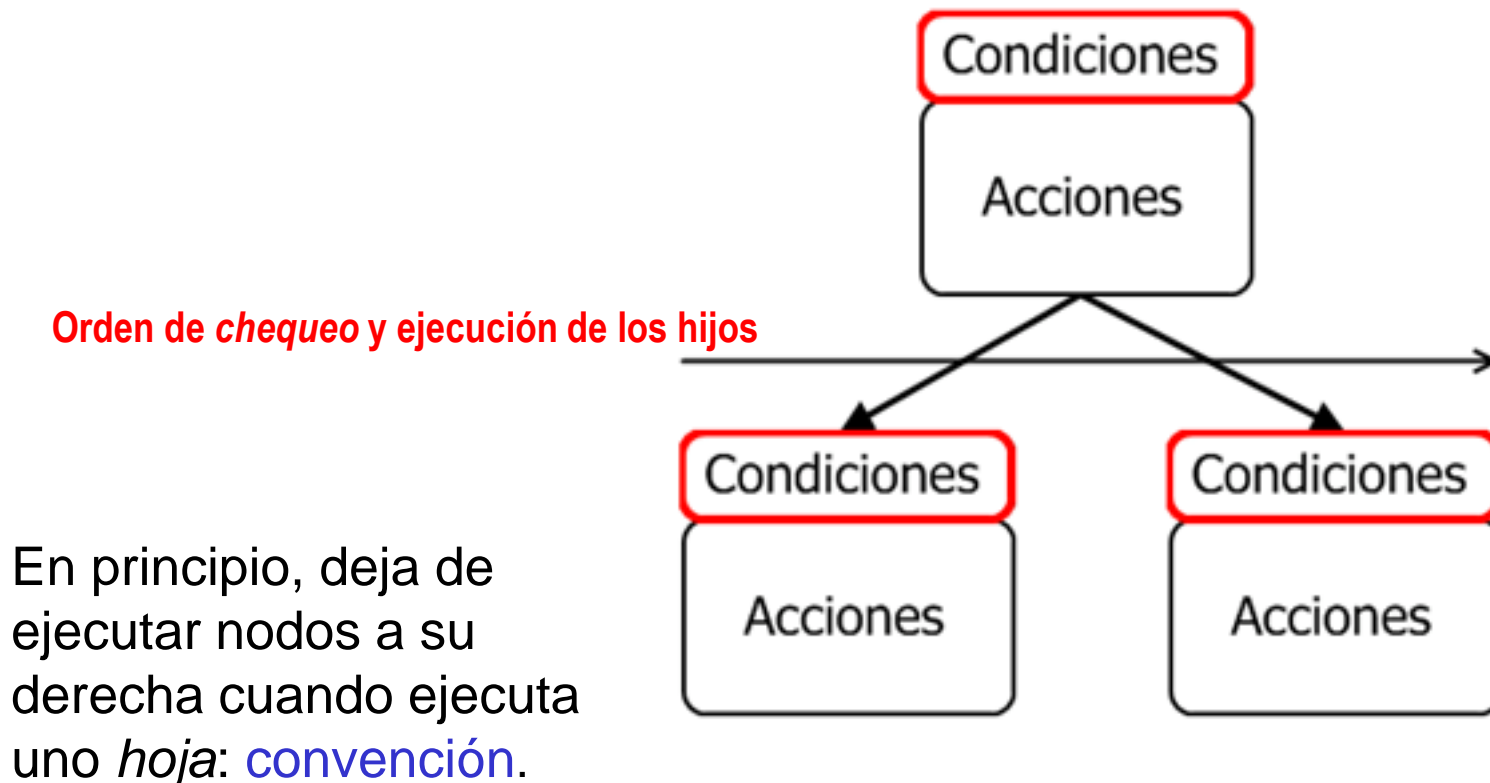




- Árboles de comportamiento (continuación)
 - En concreto,
 - Permite hacer **más sencilla la inclusión de nuevos comportamientos** que añadan complejidad a las respuestas de nuestros agentes inteligentes.
 - **Simplifican el trabajo de creación** de los comportamientos
 - Permiten **prototipar** y **probar** para **optimizar** el comportamiento
 - El **nodo**:
 - Representa cada **comportamiento** con sus **acciones** y **pre-condiciones**.
 - Los **arcos**:
 - Marcan la jerarquía de comportamientos y su evolución



- Árboles de comportamiento (continuación)
 - Los nodos tienen sólo un padre, pero pueden tener varios hijos





- Árboles de comportamiento (continuación)

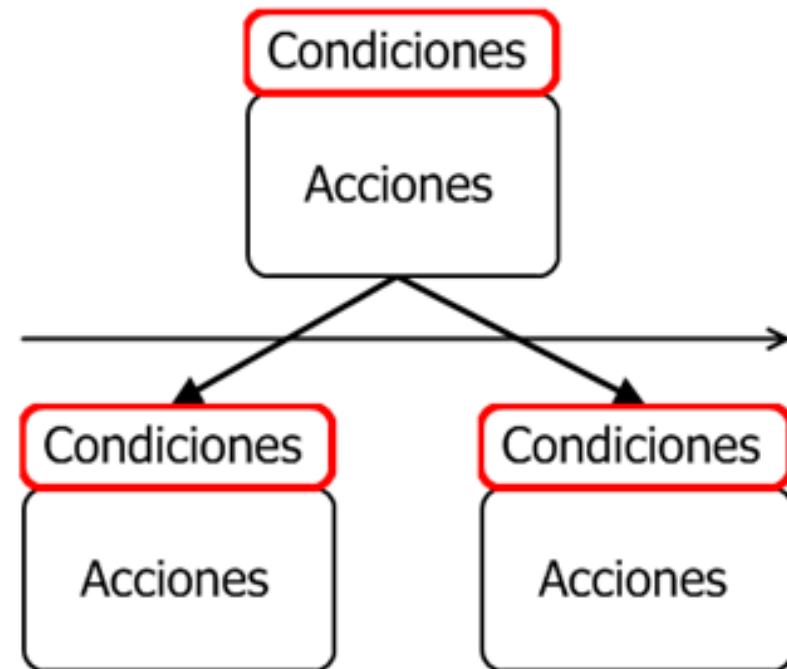
- Condiciones:

- Si Veo_Enemigo==1

- Acciones:

- Alertar
 - Esconderse

- ☐ Las condiciones pueden ser complejas, combinando varias...
- ☐ Internamente, las acciones se detallan o concretan en otro lado





- Árboles de comportamiento (continuación)
 - De modo general:
 - Se pasa de un nodo a otro nodo descendiente si se cumplen sus precondiciones **completamente**.
 - El procesamiento se detiene cuando se alcanza un nodo hoja y se realizan sus acciones.
 - Es posible, aunque menos frecuente, tener nodos ejecutándose **en paralelo**.



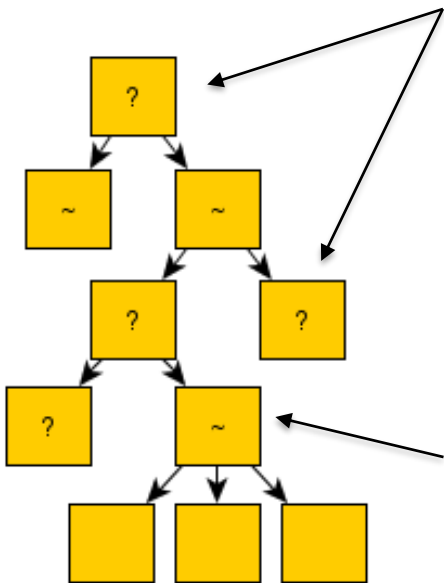
- Árboles de comportamiento (continuación)

- De modo particular:

- Se usan dos tipos de nodos de control:

- **Selectores**: Seleccionan qué nodo hijo ejecuta en función de sus condiciones: a cada nodo hijo (*subtareas*) se le cede el control, y éste devuelve éxito o fallo en función de si las ha completado o no (cumple o no cumple sus precondiciones). Decide con ello, ceder el control al siguiente hijo o no.

- **Secuenciadores**: Simplemente están para secuenciar el orden de ejecución de los nodos hijos: son los que ceden el control sistemáticamente de izquierda a derecha.

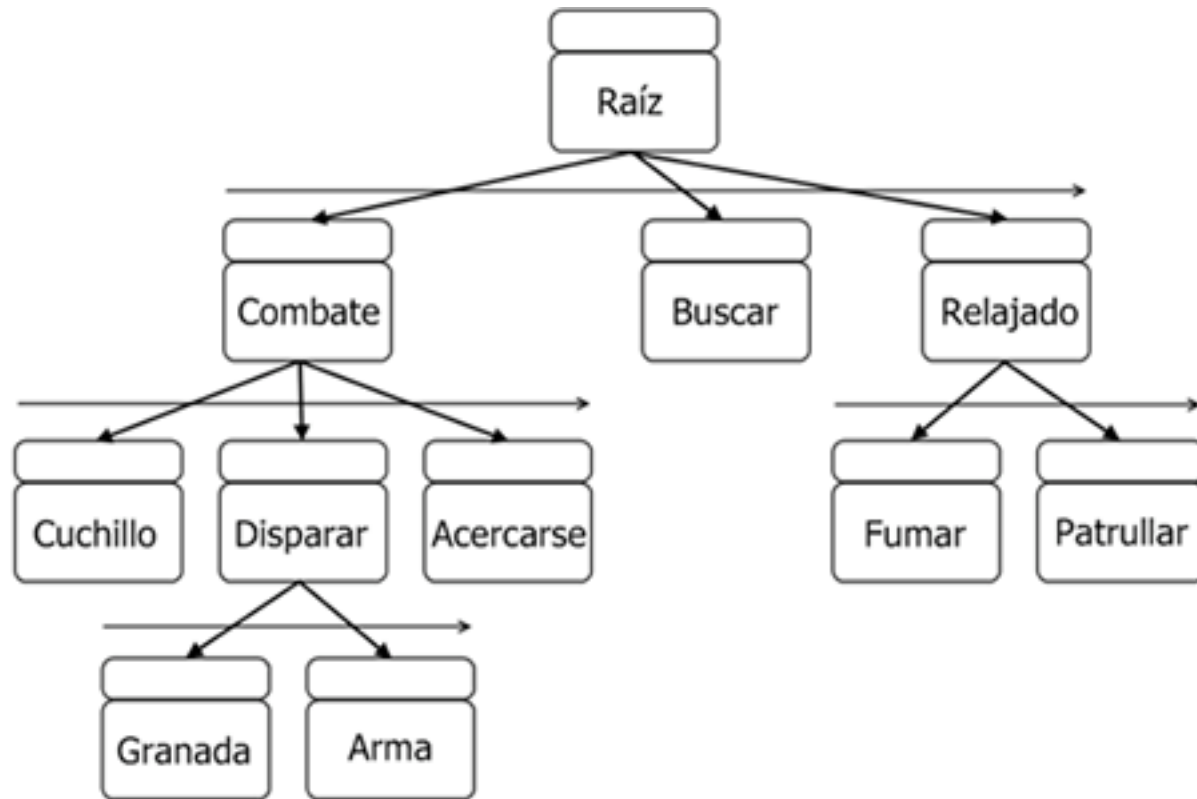




- Árboles de comportamiento (continuación)
 - Obsérvese que:
 - Los comportamientos así representados no son conductas generales, sino que se trata más bien de unidades finalistas (tareas o grupos de tareas), es decir, se codifica con ellos actuaciones concretas y específicas que cubren el instante inmediato y actual del juego (Ej: un giro de cabeza del personaje...).
 - Por tanto, los nodos hoja son los que suelen incluir fundamentalmente las actuaciones, aunque los nodos intermedios pueden tener también alguna actuación, común a los comportamientos codificados en sus nodos hijos.

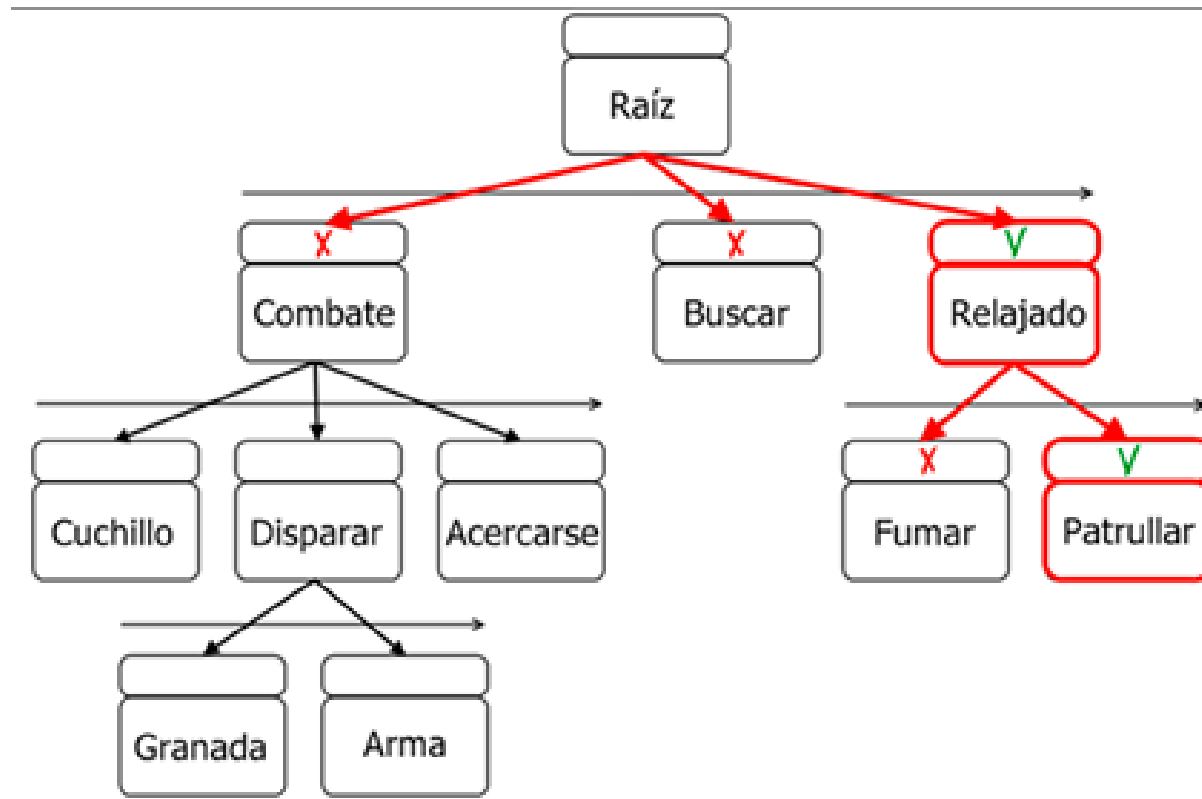


- Árboles de comportamiento (continuación)



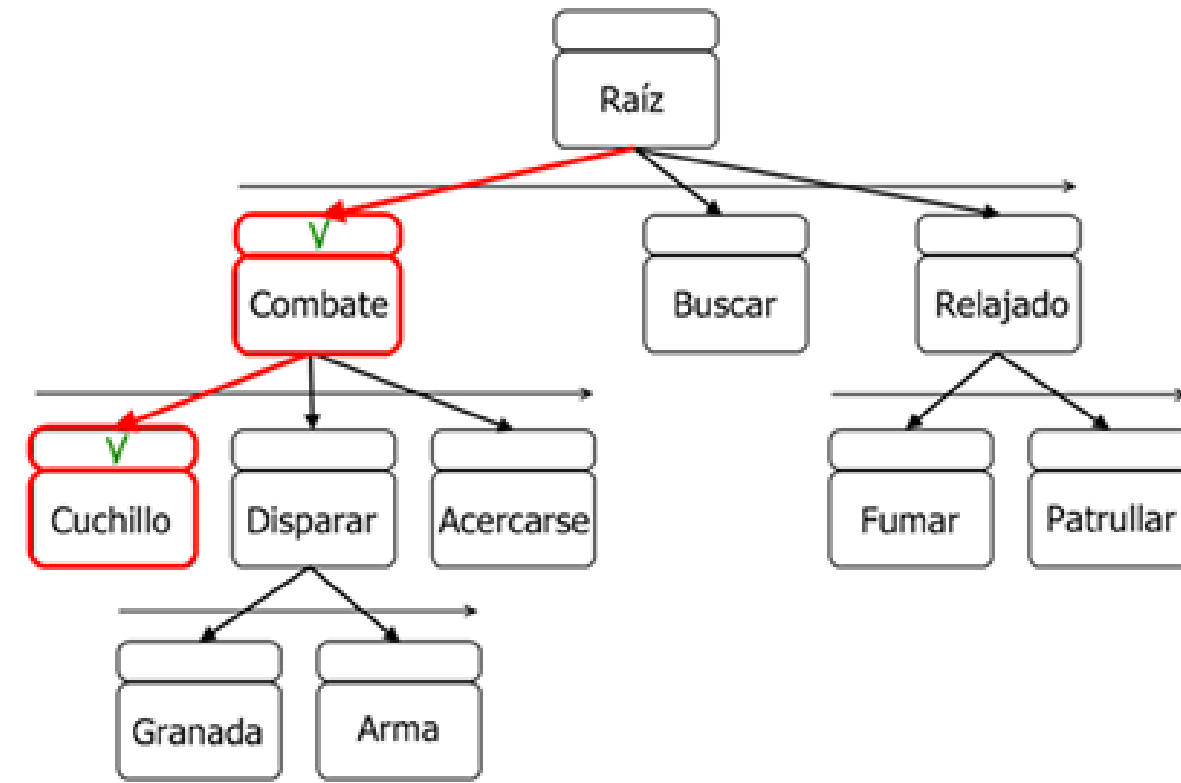


- Árboles de comportamiento (continuación)





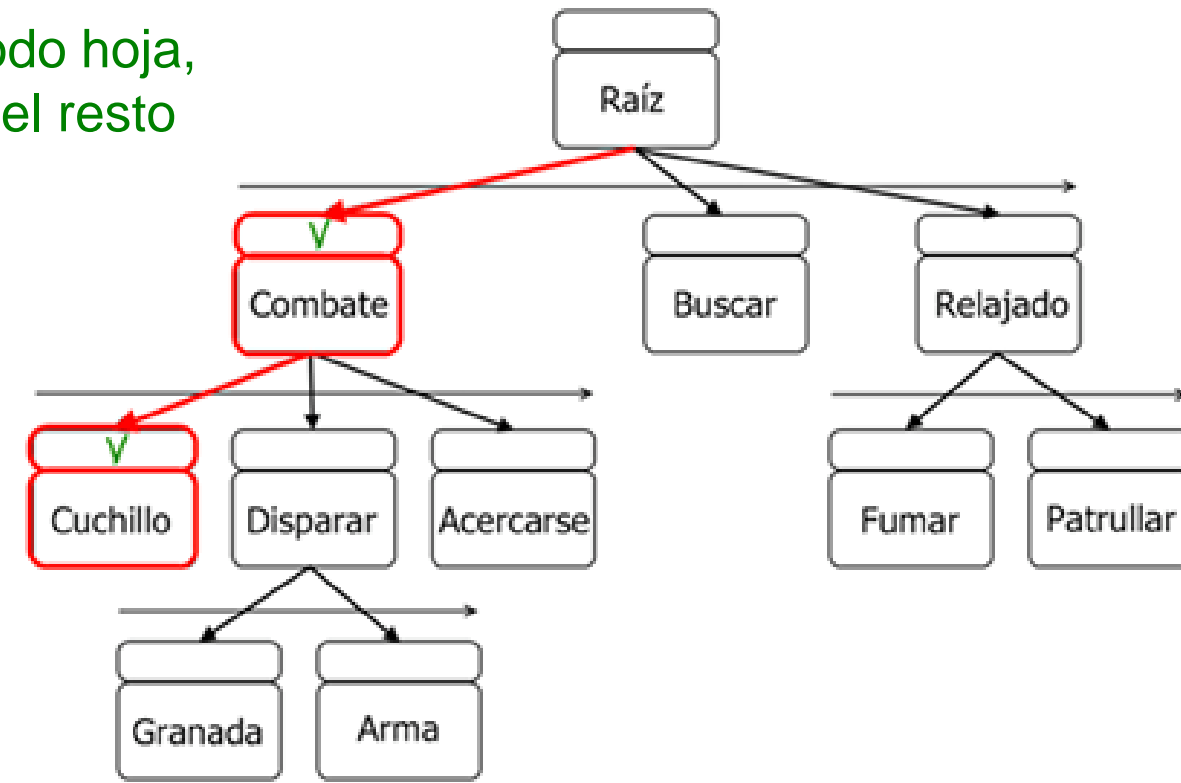
- Árboles de comportamiento (continuación)





- Árboles de comportamiento (continuación)

Al llegar a un nodo hoja,
ya no evaluaría el resto





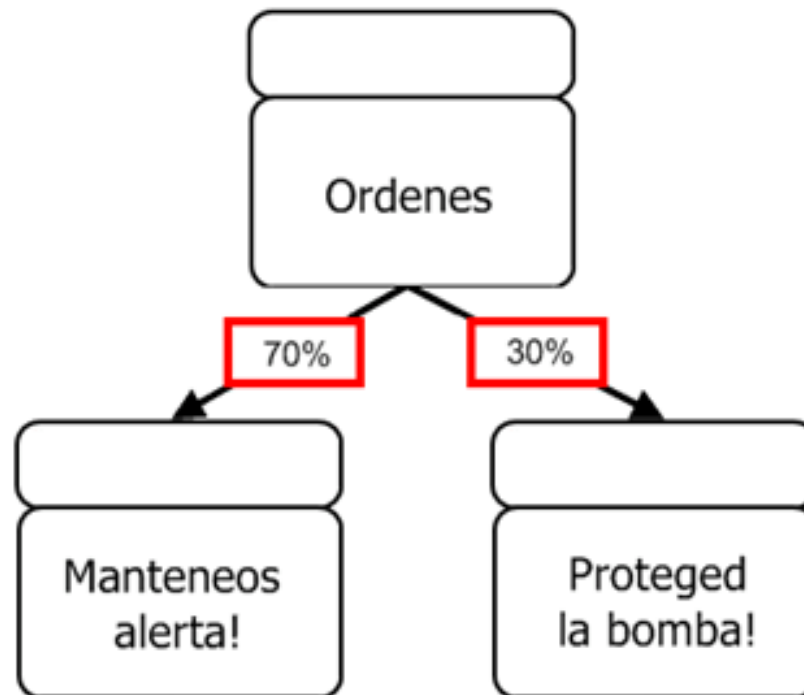
- Árboles de comportamiento (continuación)
 - Las empresas del sector, justifican su uso basado en razonamientos tales como “*si usamos gráficos más realistas, debemos usar una IA más realista*”.
 - Dada su sencillez de uso, pone al alcance de los diseñadores una herramienta para el diseño de los comportamientos, y lo retiran de quienes lo tenían habitualmente: los programadores.
 - Los árboles de comportamiento se muestran para algunos equipos como la “alternativa” a los autómatas de estados.



- Árboles de comportamiento (continuación)
 - El árbol de comportamiento es, en cierto modo, un mecanismo que permite implícitamente a su vez, implementar **toma de decisiones**.
 - La potencia de esta técnica se basa en que combina diferentes elementos: permite implementar agentes inteligentes, gestionar el estado de los mismos (encapsulado), “el conocimiento” del agente... y luego también la capacidad de gestión de cómo debe comportarse (usan así lógica de forma transparente).
 - Los árboles de comportamiento se pueden jerarquizar, y son típicos también modelos híbridos que los utilizan.

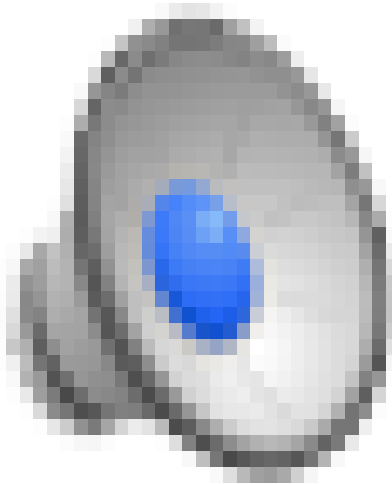


- Árboles de comportamiento (continuación)
 - A veces, para aportar más realismo, no se evalúa de izquierda a derecha sino que se hace aleatoriamente, e incluso con pesos diferentes, etc., para que no parezca sistemático.





- Árboles de comportamiento (continuación)
 - Unreal Engine





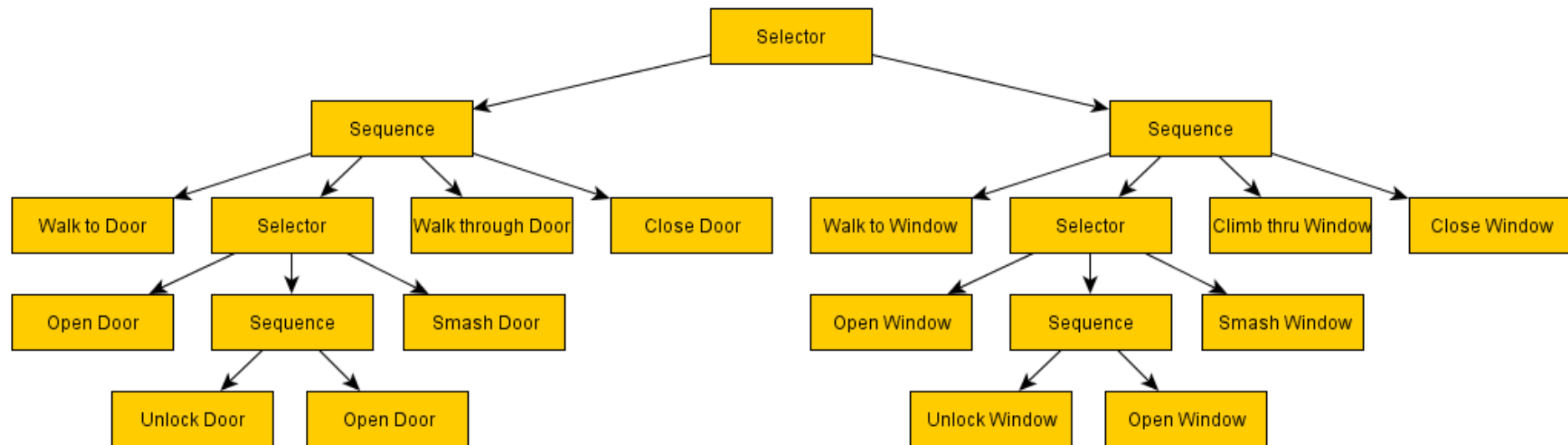
- Árboles de comportamiento (continuación)
 - Tácticas de Grupo:
 - Uso combinado y coordinado de varios agentes
 - Se crean **nodos nuevos específicos** para ello, con unas condiciones determinadas para marcar el agente como disponible para esa táctica conjunta
 - Este mecanismo **actúa como un sistema de señalización** para poder hacer acciones condicionadas y relacionadas de unos y otros agentes.



- Árboles de comportamiento (continuación)
 - Concluyendo:
 - Su potencia está basada en permitir crear comportamientos con tareas muy complejas a base de tareas sencillas.
 - Las implementaciones de las tareas elementales, están fuera del propio árbol, en otro lugar.
 - Tienen similitudes con las máquinas de estados finitos jerárquicas, salvo porque manejan tareas como elementos básicos en lugar de estados.
 - En definitiva, permiten aglutinar y manejar de forma sencilla, la función de otros recursos: arquitecturas de subsunción, descomposición secuencial de comportamientos, árboles de comportamiento, etc.



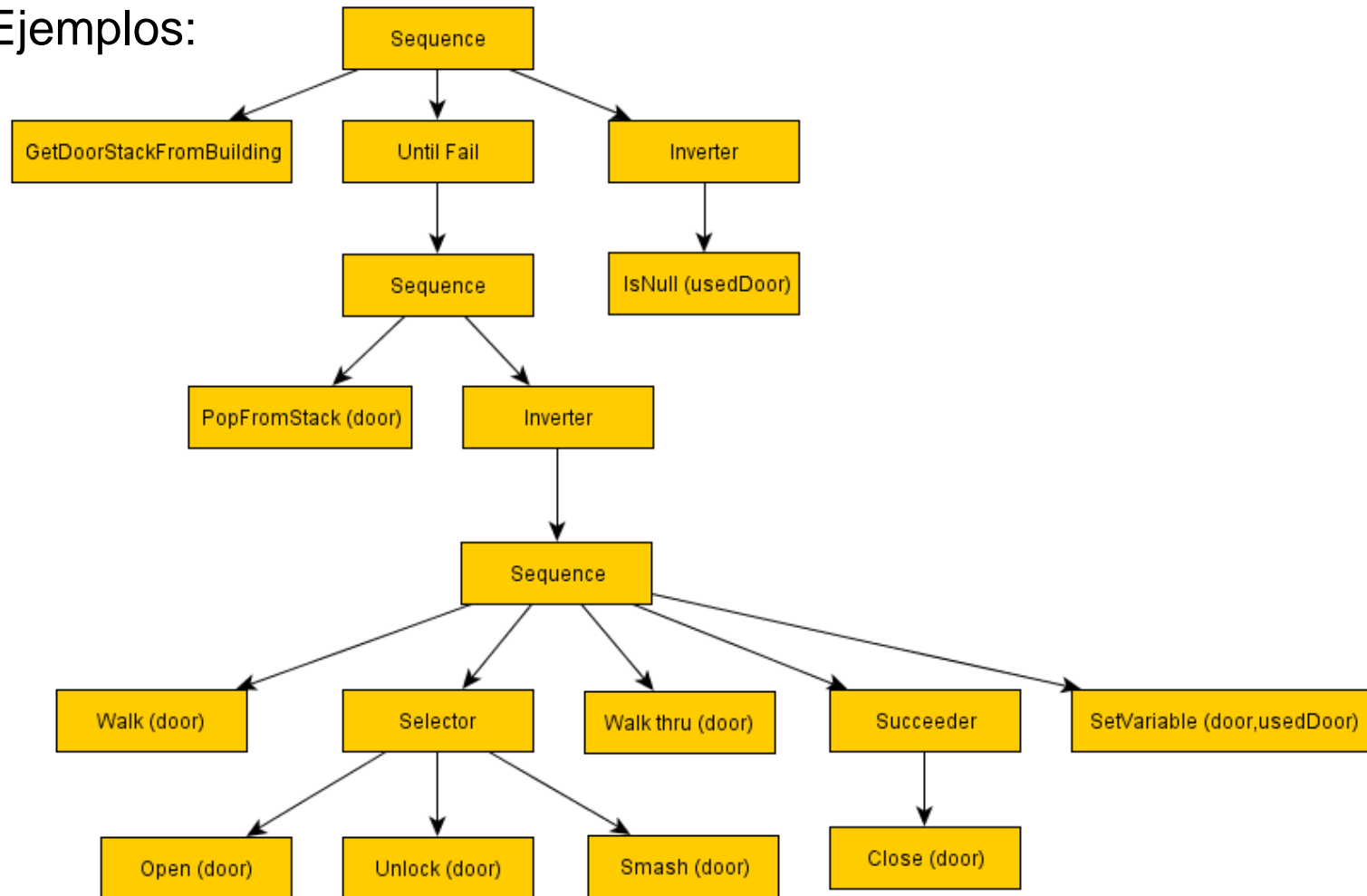
- Árboles de comportamiento (continuación)
 - Ejemplos:





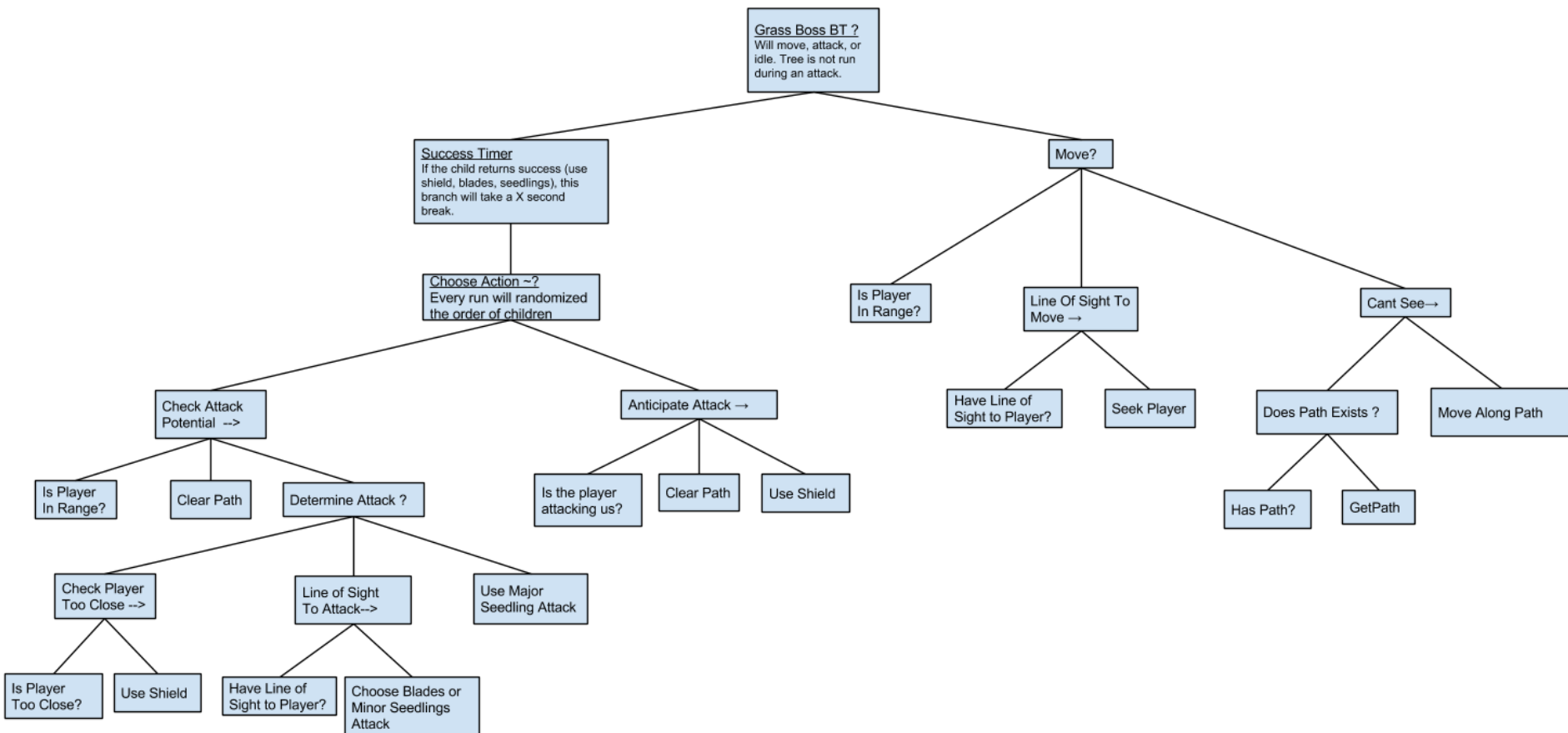
- Árboles de comportamiento (continuación)

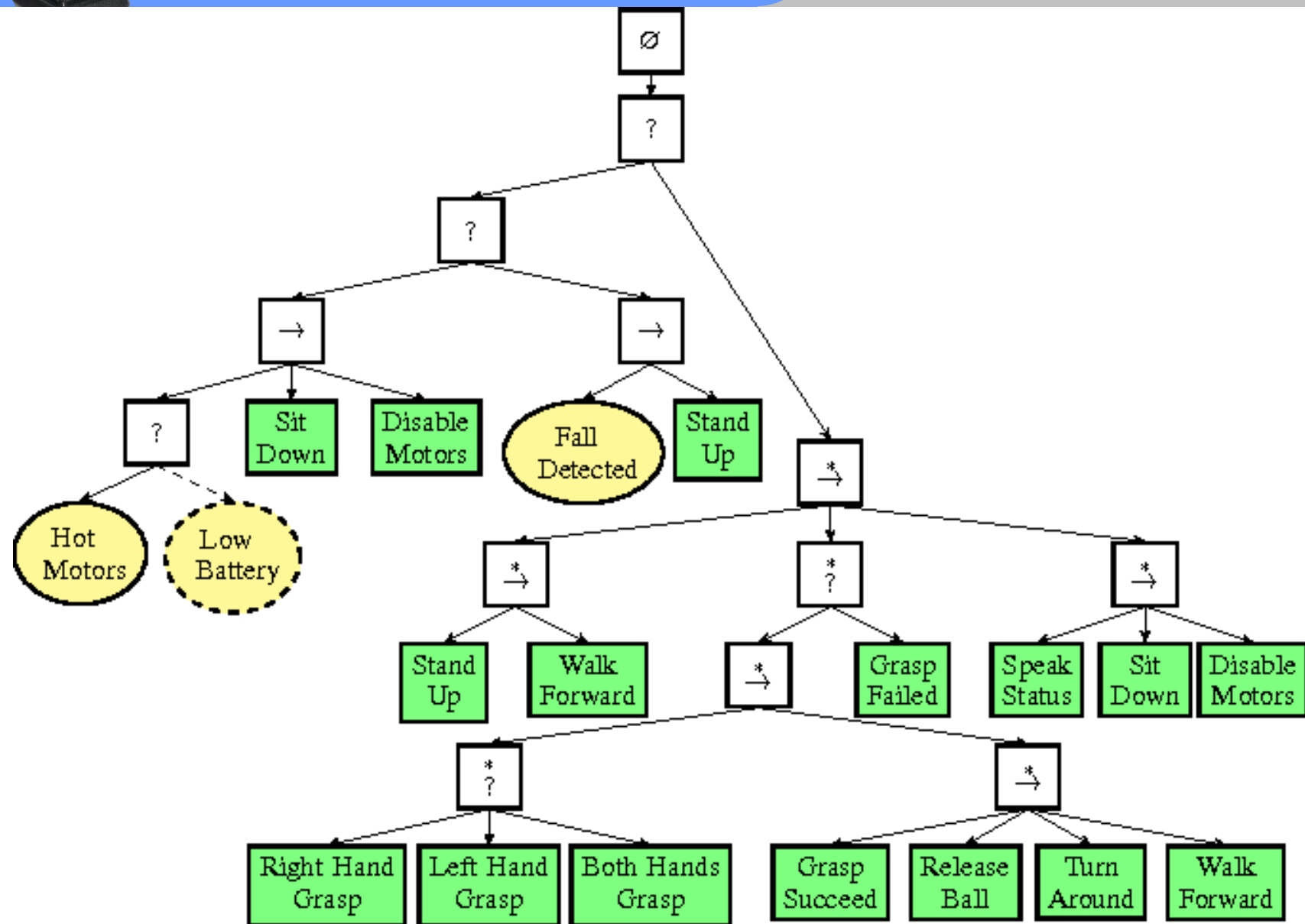
– Ejemplos:





- Árboles de comportamiento (continuación)
 - Ejemplos:







- Ejercicio:
 - Establezca las similitudes y diferencias entre las máquinas de estados finitos y los árboles de comportamiento





4. Bibliografía y Lecturas Complementarias

- AI for Games, Ian Millington, Morgan Kaufman
- *Videocharla sobre Árboles de Comportamientos*. Ricardo Pillosu:
<http://www.youtube.com/watch?v=S3fbHReq48M>
- AI for Game Developers, David M. Bourg, O'Reilly 2004
- AI Techniques for Game Programming, Mat Buckland and André LaMothe (CEO Xtreme Games LLC), Premier Press
- Artificial and Computations Intelligence in Games, S.M. Lucas, M.Mateas, M. Preuss, P. Spronck and J. Togelius
- IA en Videojuegos, José Carlos Cortizo Perez,
(<http://es.slideshare.net/jccortizo/ia-videojuegosd>)



GAME OVER