INTELIGENCIA ARTIFICIAL

Grado en diseño de productos interactivos

juan.raigada@live.u-tad.com

GOAP:
Planificación activa

Planificación de acciones

Hasta ahora hemos visto Como trabajar a nivel de reglas, agentes y sistemas para generar una sensación de oposición al jugador.

Si bien estos sistemas son válidos para juegos suficientemente directos, hay comportamientos más complejos que es difícil generar simplemente con los métodos estudiados hasta ahora.

La mayor dificultad es generar comportamiento a largo y medio plazo que no esté directamente relacionado con la oposición al jugador, sino con **la inmersión del agente en el mundo**.

Planificación de acciones

Una de las aproximaciones más sencillas a estos agentes inmersivos es la creación de pequeños scripts de horarios. Sin embargo esta aproximación se rompe en situaciones suficientemente dinámicas y escala mal según aumenta la complejidad.

Planificación de acciones

Es entonces donde entramos a pensar en la planificación. Es decir, la consecución de tareas a largo y medio plazo que no pueden ser evaluadas directamente.

Se trata de romper acciones y procesos largos en pequeños procesos a corto plazo que puedan ser continuamente evaluados y mantenidos.

También nos sirve para planificar la IA a nivel de sistema o grupo de agentes (las acciones de los grupos de agentes y sistemas suelen estar diseñadas a largo plazo), permitiendo la **emergencia de estrategias complejas**.

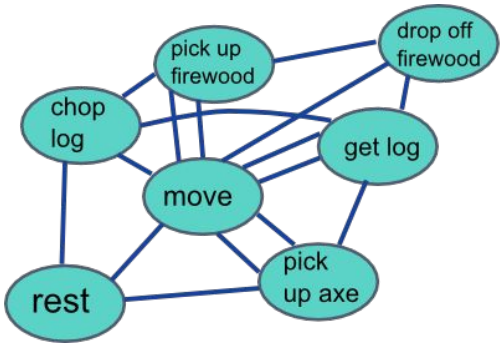
En realidad se trata de generar un sistema de máquina de estados (FSSM) genérico que permita flexibilidad y reuso de tareas.

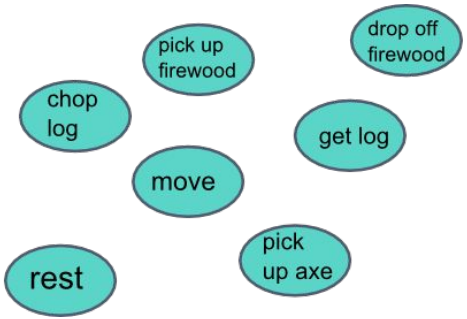
GOAP:
Goal Oriented Action Planning

GOAP

El esquema más común de gestión de planificación es el **Goal Oriented Action Planning**.

La idea a alto nivel es tratar la lógica de un agente como una máquina de estados finita, pero simplificar las transiciones entre estados y hacerlas genéricas.

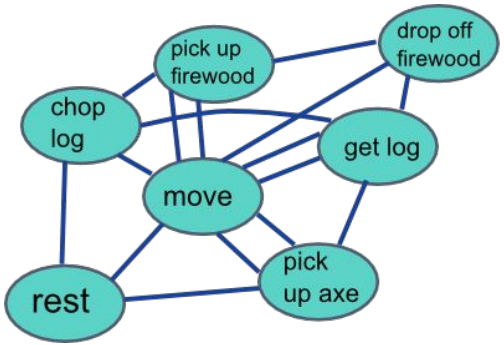


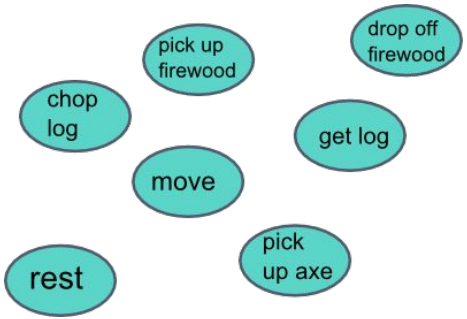


GOAP

El esquema más común de gestión de planificación es el **Goal Oriented Action Planning**.

La idea a alto nivel es tratar la lógica de un agente como una máquina de estados finita, pero simplificar las transiciones entre estados y hacerlas genéricas.





GOAP

Los estados tienen asociadas una serie de **condiciones** y una serie de **efectos**.

Las condiciones son necesarias para poder entrar en un estado, mientras que los efectos cambian el **mundo**, y por tanto las condiciones de entrada en otros estados.

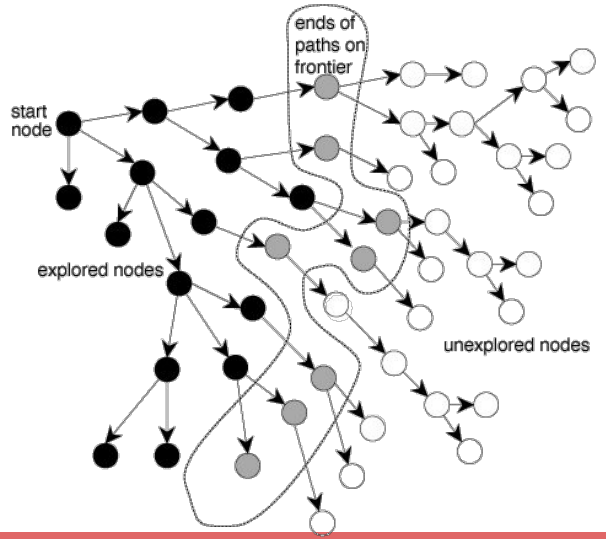
Asimismo, cada estado puede tener asociado un coste (que puede ser abstraído como coste de tiempo o de energía, pero depende de la metáfora del juego).

GOAP

Los agentes (o grupos de agentes) deciden una **META** siguiendo estrategias tradicionales (evaluación de necesidades).

Cada cierto tiempo, esa meta se evalúa, tratando de encontrar un camino en el grafo de estados de tal modo que se termine de cumplir la condición.

Esta búsqueda se puede hacer siguiendo diferentes heurísticas (amplitud, profundidad, A*).



GOAP

Este sistema de planificación y búsqueda requiere de una estructura de código relativamente compleja (es un sistema de resolución de problemas genérico), pero existen multitud de librerías abiertas para su implementación (y versiones de pago con editores gráficos).

https://github.com/luxkun/ReGoap

Existen también complicaciones en la implementación de **descriptores de estado de mundo** suficientemente genéricos y abstractos para poder ser empleados por este algoritmo, así como la **modificación de estos estados por agentes ajenos** al agente ejecutando la búsqueda.

El problema es, pues, de alta complejidad inicial, tanto de diseño como de implementación.

GOAP

Las ventajas del modelo son:

- Reuso: los estados son reusables entre diferentes agentes, pues son genéricos. Si se necesita ser más específico, cada agente puede tener una distinta implementación del estado (por ejemplo, moverse a diferentes velocidades) o evaluación de este dependiendo de las precondiciones, y también se pueden usar estas para prohibir estados a agentes.
- Adaptabilidad: El comportamiento de los agentes responde a los cambios en el mundo.
- Mantenibilidad: Es fácil añadir complejidad a los agentes, añadiendo nuevos estados y sensores.
- Claridad hacia el jugador: usando metas muy diferentes para distintos agentes es sencillo crear comportamientos que el jugador perciba como coherentes con los agentes representados por la metáfora.

GOAP

Las limitaciones del modelo son:

- La evaluación de metas. El cambio de estado del juego puede necesitar de un cambio de metas en un momento determinado, sin embargo, el cambio con demasiada frecuencia puede llevar a resultados inconsistentes.
- La reevaluación de caminos. Si un camino es imposible, el agente debería deber poder cambiar el camino en cuanto se diese cuenta de ello. Sin embargo, el cambio demasiado frecuente tampoco es bueno.
- El rendimiento si las metas son a muy largo plazo puede ser un problema en determinadas arquitecturas.

Una solución parcial a lo anterior es la **división jerárquica de metas en sub-metas de diferentes dominios** y la creación de un **modelo de conocimiento del agente** que limite sus sensores (y por tanto su percepción del cambio del mundo).

También es útil introducir estados que pueden **interrumpir** la consecución de la meta, o tener varias metas simultáneas y modificar dinámicamente su prioridad.

GOAP:
FEAR

FEAR

El videojuego F.E.A.R. fue uno de los primeros en usar GOAP como base de su IA.

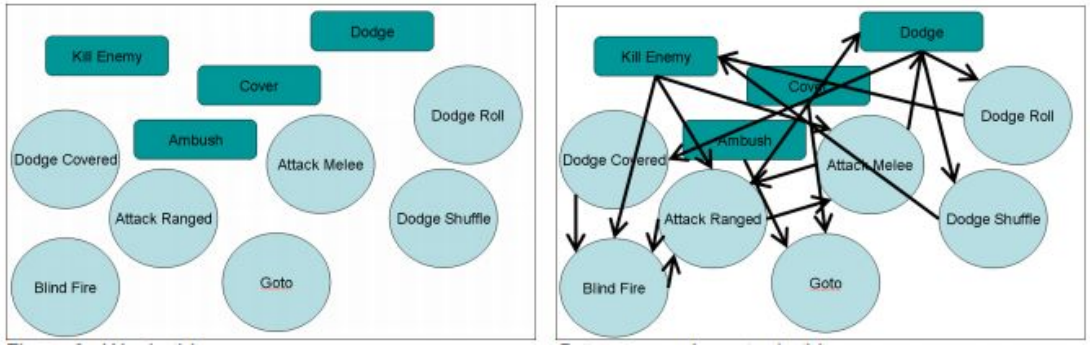
FEAR

Esta implementación dotaba a cada agente de un set de metas y un set de acciones.

Las interacciones entre el set de metas (objetivo a largo plazo) y el set de acciones (qué puede hacer cada agente para satisfacer o no las metas) generaba el comportamiento.

FEAR

El sistema no permite simultaneidad de metas. Si bien las metas están diseñadas por capas, sólo hay una meta activa en cada momento.



FEAR

Las diferencias respecto a un resolutor de problemas genérico son:

- Coste por acción: las acciones tienen un coste y son recorridas por el algoritmo A*.
- El estado del mundo está configurado por arrays fijos. Esto quiere decir que los agentes tienen que usar un sistema por encima de las reglas para priorizar a qué prestan atención. Sin embargo, el proceso de búsqueda es mucho más rápido.
- Condiciones procedurales: Hay condiciones que tienen su propio sensor adjudicado (por ejemplo, puedo escapar?). Este sensor solo se utiliza cuando se necesita emplear la acción para cumplir una meta.
- Efectos procedurales: los efectos crean cambios en el mundo que luego son detectados por los sensores.

FEAR

Ventajas del modelo:

- El comportamiento de los agentes respecto al core gameplay (combate táctico) es emergente y adaptable.

Limitaciones del modelo:

- La planificación se usa para cada agente particular, pero las escuadras usan un sistema de gestión por slots con acciones sencillas.
- La comunicación entre agentes es inexistente (es puramente cosmética).

EJEMPLOS DE SISTEMAS - MARK OF THE NINJA

EJEMPLOS DE SISTEMAS - MARK OF THE NINJA

Mark of the Ninja es un juego de stealth donde una de las mecánicas core es la manipulación del comportamiento de los enemigos.

-La implementación usada es sencilla.

-Los enemigos tienen dos tipos de sensores: auditivos y visuales.

-En el mundo suceden eventos interesantes para los agentes.

-Cuando un agente percibe un evento interesante a través de sus sensores, procede a investigarlo.

EJEMPLOS DE SISTEMAS - MARK OF THE NINJA

-Los intereses están priorizados:

A igual prioridad,
el más nuevo.

```
INTEREST_PRIORITY_LOWEST = 0
INTEREST_PRIORITY_BROKEN = 1
INTEREST_PRIORITY_MISSING = 2
INTEREST_PRIORITY_SUSPECT = 4
INTEREST_PRIORITY_SMOKE = 4
INTEREST_PRIORITY_CORPSE = 4
INTEREST_PRIORITY_NOISE_QUIET = 4
INTEREST_PRIORITY_NOISE_LOUD = 4
INTEREST_PRIORITY_BOX = 5
INTEREST_PRIORITY_SPIKEMINE = 5
INTEREST_PRIORITY_DISTRACTIONFLARE = 10
INTEREST_PRIORITY_TERROR = 20
```

Cada interés provoca distintos comportamientos en el enemigo.

-Cada enemigo solo tiene un interés activo, y no puede ser interesado dos veces por la misma fuente de interés.

-Gestión de grupo: una misma fuente solo afecta a un agente de un grupo de agentes cercanos, mientras que los demás visten su comportamiento para indicar dinámicas de grupo. El interés asigna papeles a los enemigos.