

## Trabajo práctico final

**Consigna** Integrar a un juego el algoritmo de Goal Oriented Action Planning (GOAP) visto en clase y agregarle las siguientes características:

- Además de hacer el plan, ejecute las acciones (mueva al agente, destruya objetos, recoja items, etc.). Para la ejecución de las mismas (incluyendo pathfinding), implementar una corrutina o una máquina de estados, lo que resulte más cómodo.
- El agente debe moverse a través de un grafo (o grilla) utilizando A\* funcional lazy (visto en clase) como algoritmo de pathfinding.
- En vez del modelo de mundo basado en un diccionario de booleanos visto en clase, se desea operar sobre un modelo del mundo de juego (clonable, pero minimalista para evitar overhead excesivo de memoria). Este debe tener al menos un estado de cada uno de los siguientes tipos: **float**, **int**, **bool**, **string**.

*Ejemplo:* vida (float), monedas (int), vivo (bool), arma (string = “espada”, “lanza”,...). Evalúe y aplique la estructura de datos que tenga más sentido para su juego.

- **Debe** haber por lo menos **5 acciones suficientemente distintas** en lo que causan sus **efectos**.
- Precondiciones y objetivos **formulados con lambdas**, por ejemplo, para una variable de estado “m” del modelo del mundo de juego,
  - una precondición podría ser “  $m \Rightarrow m.oro > 10$  ” o “  $m \Rightarrow 2*(m.mp+m.xp) < fv$  ”
  - un efecto puede ser “  $m \Rightarrow m.oro += 4$  ” o “  $m \Rightarrow m.hp *= 1.4f$  ”
- Nótese que **todo debe estar implementado de manera funcional y lazy con LINQ y/o generators como se ve en la cursada.**

### Opcional

:

- Se entregará **1 punto extra** por cada uno de los siguientes ítems que sean

*implementados, siempre y cuando su implementación sea correcta y el trabajo ya esté aprobado. (hasta una nota máxima de 10)*

- Aplique “time-slicing” a GOAP y A\*, para ejecutar con una cantidad de tiempo determinada por frame los algoritmos mencionados.
- Aplique el algoritmo de subdivisión espacial (grilla) visto en clase.

## Grupos

- Los grupos pueden ser de **hasta** 2 alumnos.

*Inteligencia artificial II*

*Profesor: Ramiro Spitaleri*

## Entrega y defensa

- Contener un .txt con los datos de los alumnos (nombre y apellido, comisión, turno, etc)

## Criterio de evaluación

1. Debe cumplir con todas las consignas implementadas correctamente (no perfectamente) para poder aprobar.
2. Debe tener cada punto implementado perfectamente para conseguir un 10.
3. Si se copia, le queda un 1.