

Trabajo práctico final

Consigna: Integrar a un juego-simulación 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 como algoritmo de pathfinding.
- Se debe operar sobre un modelo de mundo que debe tener al menos una variable 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 => m.oro>10**” o “**m => 2*(m.mp+m.xp) < fv**”
 - un efecto puede ser “**m => m.oro += 4**” o “**m => 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**.

Grupos

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

Entrega y defensa

- Se ha de presentar un **Informe escrito sobre el trabajo realizado**.
- Contener un .rar con el proyecto de Unity

- Contener un .txt con los datos de los alumnos (nombre y apellido, comisión, turno, etc)
- Contener una copia de las consignas (este archivo)
- Defensa oral, opcionalmente grupal, en la fecha de entrega.

Criterio de evaluación

1. Debe cumplir con todas las consignas implementadas correctamente (no perfectamente) para poder aprobar.