

Inteligencia Artificial II

Final

Criterio de evaluación

1. Debe cumplir con la consigna implementada 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.

Grupos

- Los grupos pueden ser de **hasta** 3 alumnos.

Consigna

Integre al juego ya desarrollado para el Primer y Segundo Parcial el algoritmo de Goal-Oriented Action Planning (GOAP) visto en clase y agregarle las siguientes características:

- 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 con variables de distintos tipos. 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.

Según convenga, es posible cambiar una variable de tipo string a enum si tiene más sentido para el uso que se le va a dar.

- Precondiciones y efectos **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 “ $m \Rightarrow 2 \cdot (m.mp + m.xp) < fv$ ”
 - 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.**

Se le **tomará mucha importancia al diseño de la inteligencia artificial y su interacción con el jugador**, tanto el level design como game design puede ayudar a llevar esto a cabo, pero se prefiere que se haga hincapié al diseño de la IA antes que a otra cosa. Así mismo, animaciones, modelos y **arte** en general no se tomarán en cuenta para la nota, **solo aquellos elementos que formen parte del feedback** y sean considerados necesarios para que el jugador comprenda lo que la IA está haciendo.

Por ejemplo: si su juego es un juego Stealth, no hay problema si el nivel es un plano, las paredes son cubos y los personajes (enemigo y jugadores) son cápsulas, pero es importante poner al jugador un color distinto a los enemigos (más aún si hay distintos tipos de enemigos, en ese caso un color para cada uno), mostrar un ícono y/o reproducir un sonido cuando los enemigos cambian de estado (de patrulla a alerta porque vieron al jugador).

Entrega

- La carpeta del proyecto (sin Library ni Temp) y el ejecutable deberá ser entregado en un .Rar con el nombre de los integrantes del grupo y la comisión a través de DVPanel. **TODOS los integrantes deben entregar en el DVPanel o de lo contrario el sistema no habilita la subida de notas.**
- Además se ha de entregar un informe escrito sobre el trabajo realizado, con la explicación de los controles y acciones que puede realizar el jugador y cómo funcionan los agentes.
- Se recomienda entregar **por e-mail antes de la fecha** de final para poder darles una devolución temprana y que puedan mejorar su entrega para la entrega final. Corroboen que el e-mail fue enviado con éxito.