

Trabajo Práctico N° 2: Pacman

Enunciado

Se deberá implementar el modelo de un juego tipo pacman que utilice los laberintos diseñados ya almacenados en la plataforma de TPs anteriores.

Funcionalidad a Implementar

Se pide modelar e implementar los siguientes conceptos:

FANTASMA:

El fantasma es un personaje del juego y está caracterizado por poder estar:

- Muerto
- Como fantasma cazador, con distintos niveles de agresividad
- Como fantasma Presa

El fantasma se puede mover dentro de los caminos del laberinto, puede ser comido, puede incrementar su ira. Por naturaleza al comienzo es cazador y puede ser convertido en presa en cualquier momento. Siendo cazador, cada X1, X2 y X3 intervalos incrementa su ira, luego de X3 permanece en ese nivel de ira o agresividad.

Si un fantasma esta muerto no puede incrementar su ira. No puede ser comido ni tampoco puede ser convertido en presa. Al morirse debe reiniciar el nivel de agresividad y en un tiempo X4 debe volver a “revivir” como cazador para comenzar el ciclo.

Si un fantasma es cazador no puede ser comido, puede incrementar su nivel de agresividad (hasta un máximo) y puede ser convertido en presa.

Si un fantasma es presa puede ser comido (y debe estar muerto luego de ser comido), no puede incrementar su nivel de agresividad y no puede ser convertido a presa. Al cabo de un tiempo x (ver aclaraciones) debe volver a un naturaleza cazador.

Los fantasmas en estado cazador pueden desplazarse al menos en alguno de los siguientes 4 criterios de inteligencia:

- Zonzo: Se mueve aleatoriamente (* ver nota posterior) , solamente se lanza a comer al pac-man si este se encuentra a N1 (cuatro por ej casilleros seguidos) lugares de distancia, si este se escapa a mas de N1 lugares continúa moviéndose aleatoriamente.

- **Perezoso:** Se mueve aleatoriamente, solamente se lanza a comer al pac-man si este se encuentra a N2 (ocho por ej) lugares de distancia.
- **Buscador:** Se mueve aleatoriamente, si el pac-man se encuentra a N3 lugares de distancia lo persigue. Si el pac-man se escapa a más de N3 (diez por ej) lugares de distancia el fantasma buscador va hacia la última posición donde lo vio (no se mueve de manera aleatoria) y si en el camino lo ve, se dirige hacia esa posición.
- **Buscador temperamental:** Es igual al buscador anterior, pero si el fantasma incrementa su nivel de ira incrementa en X la visión (inicialmente tiene una visión de N4 lugares(12). Si se reinicia el nivel de ira, entonces también se reinicia la visión).

(*) El movimiento aleatorio mencionado anteriormente tiene que cumplir con los siguientes requerimientos:

- Una vez que entra a un canal del laberinto, sigue hasta llegar a un cruce o bifurcación, recién ahí vuelve a decidir hacia donde ir. Y no es opción regresar por donde vino, salvo en el momento de cambio de presa a cazador o viceversa, solo en este momento el fantasma podría invertir su dirección sin estar en una bifurcación.
- A la llegada de un cruce si no sabe a donde ir dado su visión, entonces define aleatoriamente que camino tomar.
- Siempre se mueven, es decir nunca están quietos.

El fantasma en estado presa debe alejarse del pac-man, es decir que se debe moverse según se describió pero alejándose en los casos que antes se acercaba al pac-man.

Un fantasma presa tarda X tiempo en volver a ser cazador. El X depende de la ira de los mismos. Cada nivel de ira tiene un X distinto.

PACMAN:

Puede moverse libremente por los caminos del laberinto. Tiene la misma velocidad del fantasma inicialmente.

Puede comer fantasmas mientras que los mismos sean presas.

Cuando el pac-man come un fantasma cazador (o es cazado por un fantasma) muere.

Objetivos

- Implementar una versión inicial de la herramienta que cumpla los requerimientos.
- Aplicar las técnicas vistas en la teoría y en la práctica.
- Ganar conocimiento sobre las herramientas vistas en la materia.

Herramientas a utilizar

- Javascript
- Node.js
- Docker
- Git

Restricciones

- Se trabajará sobre el repositorio git de **GitLab**, el cual deberán crear e invitar al usuario **tecnicas7510** como miembro **Maintainer**
- Trabajo realizado en grupos de 3 personas.

Criterios de Corrección

- Cumplimiento de las restricciones
- Diseño del modelo
- Diseño del código
- Documentación entregada (informe, diagramas y tests)
- Correctitud
- Completitud
- Buen uso de las herramientas indicadas
- Uso de buenas prácticas en general

Entrega

- Para el Hito **Entrega TP, via GIT**: Crear un tag "entrega-2" (Campo "Tag Version" en GitLab, comando ``git tag entrega-2 ; git push origin entrega-2``) sobre el commit a entregar
- La entrega deberá incluir un conjunto de test unitarios y de aceptación que validen escenarios válidos e inválidos.

Calendario

FECHA	HITO
Jueves 9/5	Presentación del TP
Jueves 16/5, Jueves 23/5	Revisión estado del TP en clase
Jueves 30/5	Entrega TP, via GIT
Jueves 13/6	Devolución y comentarios de las entregas