



Inteligencia Artificial para Videojuegos

Grado en Desarrollo de Videojuegos

Prácticas del curso

Departamento de Ingeniería del Software e Inteligencia Artificial

Facultad de Informática

Universidad Complutense de Madrid



Práctica 3: El fantasma de la ópera

Revisión de documentación, resultados y final: 24 de marzo, 7 y 8 de abril de 2022

Importante: Las revisiones y calificaciones se realizan sobre *el grupo en su conjunto*, en tiempo y forma, asumiendo que se documente y acredite un reparto justo del trabajo. Los profesores deben ser los únicos con acceso a vuestro repositorio IAV22-GXX-P3 dentro de la organización de GitHub IAV22-GXX, siendo *XX* vuestro número de grupo con dos dígitos. En el repositorio debe encontrarse la documentación, el ejecutable y el código fuente del proyecto. Toda la documentación debe estar en el *README.md* de la raíz del repositorio y contener vuestros datos, un resumen del enunciado, la descripción del punto de partida proporcionado, el diseño de la solución (prototipo explicado con diagramas y/o pseudocódigo) y el enlace a un video oculto de YouTube titulado IAV22-GXX-P3 con la grabación personalizada de todas las *pruebas realizadas*, convenientemente rotuladas y comentadas, de menos de 3 minutos de duración. El ejecutable para Windows de 64bits IAV22-GXX-P3.zip debe estar publicado como *lanzamiento* descargable en el repositorio. El código fuente debe incluir en carpetas separadas cualquier recurso o *plugin* de terceros necesario para la correcta compilación del proyecto.

1. Introducción

“El escenario de la famosa novela de Leroux El fantasma de la ópera (1910) tiene su origen en una ópera real de París sobre la que el autor había escuchado rumores desde que finalizó la construcción del edificio. Los detalles sobre el Palacio Garnier de París y los rumores que lo rodean están estrechamente vinculados en el relato de Leroux. El lago subterráneo sobre el que escribe en su novela es exacto al que se encuentra bajo esta casa de la ópera, y que aún se utiliza para enseñar a los bomberos de la ciudad a nadar en la oscuridad. El infame accidente de la lámpara que acontece en la historia también resultó ser cierto. En general, es verdad que los misterios que Leroux narra en su novela acerca del Fantasma siguen siendo misterios. Sin embargo, él defendió aquellos rumores como ciertos, incluso en su lecho de muerte.”

En esta ocasión, la novela gótica más popular del periodista y escritor francés Gastón Leroux (Figura 1) nos sirve como pretexto para abordar el tema de la Decisión en Inteligencia Artificial para Videojuegos. El personaje que da nombre a la novela es, más que un villano, un antihéroe. Erik es un músico deforme que vive escondido en los subterráneos del Palacio Garnier, la casa de la ópera. Suele encerrarse en una sala secreta para componer su gran obra, y sueña con que sea interpretada por la gran diva del canto Christine Daaé, con la que está obsesionado. Para ello, no dudará en secuestrarla y obligarla a memorizar el libreto si hace falta: tiene hasta la celda preparada. Aunque existen dos obstáculos importantes para los planes del monstruo. El primero es el público de la ópera pues, debido a su aspecto, Erik odia ser visto por otras personas y prefiere mantener su autoría en el anonimato. Por eso se desplaza utilizando pasadizos ocultos y pequeñas barcas con las que cruza las zonas inundadas de los sótanos del

edificio, e incluso es capaz de derribar enormes lámparas en el patio de butacas para ahuyentar a los espectadores. El segundo obstáculo en los planes de Erik es el vizconde Raoul de Chagny, que además de ser un joven y atractivo pretendiente de Christine, hará todo lo posible por frustrar los tejemanejes del fantasma.



Figura 1. Una de las cinco acuarelas de André Castaigne que ilustraron la primera edición americana de *El fantasma de la ópera*, en 1911.

El prototipo que vamos a desarrollar se centra en la toma de decisiones del fantasma, que será el agente inteligente con el comportamiento más complejo de todos. Su objetivo es secuestrar a la cantante, llevarla a la celda secreta que tiene preparada para ella, encerrarla allí, y poder así seguir trabajando confiado en su gloriosa (y a la vez interminable) obra maestra. La diva, por su parte, trabaja sobre el escenario aunque en el descanso entre las escenas, se retira a las bambalinas. El fantasma la aterroriza y es incapaz de ofrecer resistencia alguna si la captura. Su amigo el vizconde, por el contrario, la ayuda a volver a las tablas. Precisamente el avatar que controla el jugador es el vizconde, capaz de moverse por todas partes y poner remedio a todos los males que haya podido causar el fantasma (además de ponerle trabas a este). El vizconde recoloca lámparas caídas (viles atentados que en ocasiones realiza el fantasma para expulsar al público de la ópera), rescata a la cantante de su celda, e incluso puede ensañarse a veces con la guarida del fantasma, golpeando los muebles (¡y el piano!) que este guarda en la sala de música, haciendo enfurecer terriblemente a esta malvada figura. Si, como jugadores, no intervenimos frente a las tropelías del fantasma, este no tardará en secuestrar a nuestra ‘prima donna’, tras ahuyentar al público, y seguir impunemente con su febril actividad artística.

Este prototipo servirá para probar la técnica de toma de decisiones tanto de las máquinas de estado como de los árboles de comportamiento, las dos más populares de la actualidad. Además se complementará con la búsqueda de caminos mediante mallas de navegación, los comportamientos de dirección y hasta la gestión sensorial, aunque esta vez aprovechando en todo lo posible las herramientas que ya trae integradas Unity.

2. Planteamiento del proyecto

Desarrolla un prototipo de IA para Videojuegos, dentro de un entorno virtual que represente la ópera de París (véase Figura 2), con un agente inteligente (el fantasma) que decide, se mueve y actúa según lo que encuentra en sus diferentes estancias, otros agentes más simples como la cantante y el público, y un avatar, el vizconde -némesis del fantasma-, controlado por el jugador.

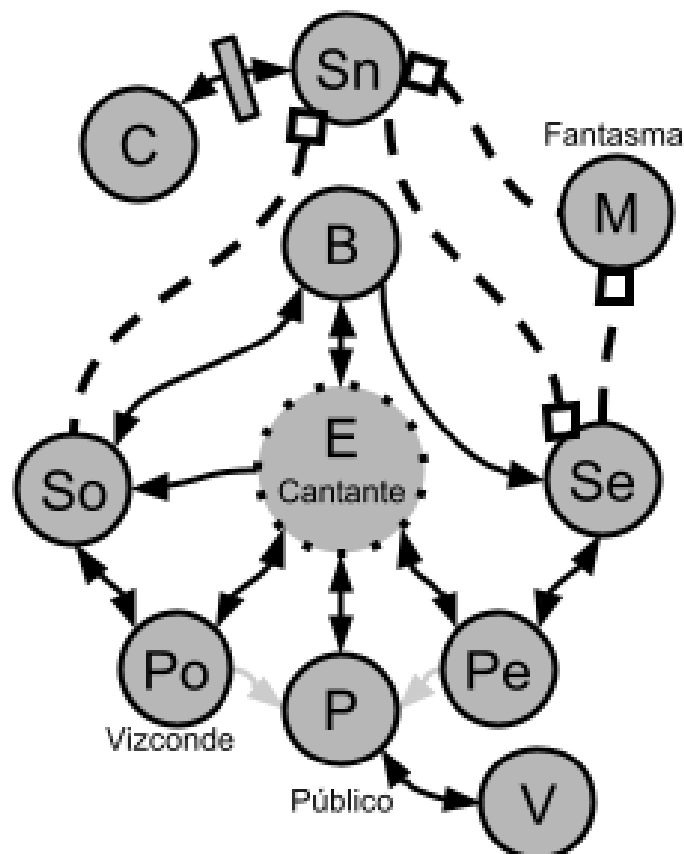


Figura 2. Esquema con la topología de las distintas estancias de la ópera. Los nodos del grafo representan estancias identificadas por una abreviatura, las aristas dirigidas representan visibilidad entre ellas, e incluso navegabilidad si las aristas son negras.

En el entorno virtual encontramos las siguientes estancias, describiendo también los elementos más relevantes que podemos encontrar en ellas, y su comportamiento:

1. **Patio de butacas (P).** Es la estancia inicial del *público*, dividido en dos bloques (este y oeste). Cada bloque permanecerá en su sitio salvo que caiga la lámpara gigante del techo correspondiente (hay *lámpara este* y *oeste*), en cuyo caso ese lado del patio de butacas se oscurecerá y los espectadores huirán despavoridos al *vestíbulo*. No regresarán hasta que no se vuelva a colocar su lámpara. Esta estancia está conectada con el *escenario* (visibilidad y navegabilidad, como indica la arista negra), con el *vestíbulo*, y es visible desde los *palcos este* y *oeste*, aunque el público no puede alcanzar a ver bien si hay alguien en los palcos, ni aunque estén las dos lámparas encendidas.
2. **Vestíbulo (V).** Es la zona más externa de la ópera, donde van los bloques de público cuando se asustan. Simplemente conecta con el patio de butacas.
3. **Escenario (E).** Es la estancia inicial de la cantante, donde se dedica a su oficio, aunque lo intercala (cada pocos segundos) con un descanso que realiza tras las *bambalinas*, una estancia contigua. Además también conecta con el *patio de butacas* y los *palcos*, y es posible dejarse caer por una trampilla al *sótano oeste*, aunque no sea posible regresar. El fantasma no pisará ninguna estancia como esta mientras haya *público* mirando. Eso sí, tanto en esta estancia como en otras puede “capturar” (coger al hombro) a la cantante, incluso compartiendo estancia con el vizconde y llevársela a donde quiera, soltándola por voluntad propia o porque se sienta intimidado por el “choque” con nuestro héroe. Si la cantante acaba en una estancia que no esté conectada con el escenario, se sentirá confusa y navegará entre estancias aleatoriamente, dejándose “rescatar” por el vizconde en caso de que lo vea cerca, con la esperanza de que la lleve hasta una estancia que conozca, para poder retomar así su ritmo normal de trabajo.
4. **Bambalinas (B).** Estancia donde suele descansar la *cantante* y que conecta con el *escenario*, el *sótano oeste* y que permite deslizarse por una rampa algo oculta al *sótano este*, sin posibilidad de regresar.
5. **Palco oeste (Po).** Estancia inicial del *vizconde*, personaje que controla el jugador y que gusta disfrutar desde aquí de la función. El palco tiene una palanca que se puede usar para dejar caer la lámpara oeste del patio de butacas. Conecta con el *escenario*, con el *sótano oeste* y permite ver el *patio de butacas*, aunque debido a la altura no existe visibilidad en el sentido opuesto. El vizconde puede moverse con libertad, como el fantasma, también sobre las barcas cercanas. Puede usar palancas y chocar contra el fantasma, intimidándolo y haciendo que retroceda durante unos pocos segundos (y suelte a la cantante si la llevaba). Puede golpear muebles, como los de la sala de música, haciendo un ruido tremendo que se escuchará en toda la zona subterránea. Puede interactuar con una lámpara caída, para arreglarla automáticamente (colocándola en su sitio), y también con la *cantante*, cogiéndola en brazos automáticamente y llevándola consigo, o dejándola en el suelo a voluntad (interactuando sin que haya una palanca delante ni otra cosa así).
6. **Palco este (Pe).** Estancia similar al *palco oeste*, con una palanca que se puede usar para dejar caer la lámpara este del patio de butacas. Conecta con el *escenario*, con el *sótano este* y permite ver el *patio de butacas*, aunque sin visibilidad en el otro sentido.
7. **Sótano oeste (So).** Estancia que conecta con el *palco oeste*, con las *bambalinas* y con el *sótano norte*, aunque para recorrer esta conexión hace falta subirse a una barca. Sólo una persona (tal vez con otra al hombro o en brazos) puede montarse sobre la barca a la vez y sólo si esta se encuentra atracada en la orilla de esa estancia. Por defecto, la barca que se necesita aquí comienza atracada en la otra orilla, en la del sótano norte, y aunque en todas las orillas siempre hay una palanca que permite acercarla, el proceso de “recuperar” la barca es algo lento (mayor coste). Se puede llegar a esta estancia desde el escenario, pero no al revés.

8. **Sótano este (Se).** Estancia que conecta con el *palco este*, y tanto con el *sótano norte* como con la *sala de música* donde compone su obra el *fantasma*, aunque para recorrer estas dos últimas conexiones hacen falta barcas. Por defecto, la barca que lleva al *sótano norte* sí está en esta orilla, pero la que lleva a la *sala de música* está en la orilla contraria. Aunque se puede llegar a esta estancia desde las bambalinas, por una trampilla, desde aquí no se conecta con las bambalinas.
9. **Celda (C).** Estancia donde el *fantasma* deja a la *cantante* para completar su secuestro con éxito, usando una palanca que activa unas rejas que la impiden salir (y que por supuesto el vizconde podrá desactivar). Conecta con el *sótano norte*.
10. **Sótano norte (Sn).** Estancia que conecta con la *celda*, además de con la *sala de música*, el *sótano este* y el *sótano oeste* a través de sus correspondientes tres barcas.
11. **Sala de música (M).** Estancia inicial del *fantasma*, donde le gusta pasar tiempo componiendo su ópera. Conecta mediante una barca con el *sótano este*, y con otra con el *sótano norte*. El fantasma tiene el objetivo principal de secuestrar a la *cantante*, para lo que intentará buscarla en las *bambalinas*, en el *escenario* o si no logra dar con ella, explorando las demás estancias meticulosamente por si estuviera “perdida” por allí. No puede acceder al escenario si hay público mirando, de modo que, como objetivo secundario, necesita tirar las dos lámparas del techo para vaciar del todo el patio de butacas. Sea como sea, una vez atrapada la *cantante*, la llevará consigo hasta la *celda*, intentando usar siempre el camino con menor coste (recordando la última posición de las barcas y del vizconde que conoce, y eligiendo la ruta con menor coste, la que tenga más barcas a su favor y que evite al héroe de esta historia). Cuando llega hasta la *celda* la soltará allí, activará las rejas e irá hasta la *sala de música*, permaneciendo allí indefinidamente. Lo único que desconcentra al fantasma cuando está componiendo es escuchar a su musa cantar de nuevo en el escenario, reavivando sus deseos de secuestrarla y encerrarla otra vez en su celda. Por otro lado, si el fantasma llega a percibir el ruido de los golpes del vizconde a su piano, abandonará lo que esté haciendo (soltando a la cantante) y correrá enfurecido hasta allí para dedicar unos segundos a arreglar semejante estropicio.

La **documentación** del prototipo deberá ser la adecuada (datos correctos, resumen completo del enunciado, descripción del punto de partida con suficiente detalle, investigación de posibles referentes, diseño de la solución mediante diagramas y/o pseudocódigo...) [2,5 ptos.].

El **ejecutable** será práctico y fácil de usar, comprendiendo -con la calidad mínima esperable en el prototipo de un videojuego- las siguientes funcionalidades:

- A. Mostrar el entorno virtual (la casa de la ópera), con un esquema de división de *mall*a de *navegación* proporcionado por Unity, donde se ubiquen todos los elementos descritos anteriormente. El vizconde será controlado libremente por el jugador mediante los *cursores* y una única tecla de acción para interactuar con otros elementos. Aunque haya cámaras que sigan a cada uno de los personajes, conviene que haya una adicional que nos dé la vista general del entorno [0,5 ptos.].
- B. Hacer que parte del público huya tras la caída de una lámpara, y regrese en cuanto esté arreglada. Será una navegación y un movimiento trivial, sin apenas decisión [0,5 ptos.].
- C. Representar a la cantante como un agente inteligente basado en una *máquina de estados* que pasa del escenario a las bambalinas cuando toca, que puede ser “llevada” por los otros dos personajes hasta otra estancia, que navega algo desorientada cuando está en

las estancias subterráneas, y que se deja llevar por el vizconde, con la esperanza de reencontrar el escenario y continuar su rutina allí. Tiene navegación, movimiento y percepción sencillos, y decisión mediante máquina de estados [1 ptos.].

- D. Desarrollar el *árbol de comportamiento* completo del fantasma, para que busque a la cantante, tire las lámparas, la capture, la lleve a la celda, active las rejas, etc. [1 pto.]
- E. Usar un sistema de gestión sensorial para que el fantasma reaccione *realmente* a lo que ve (en la propia estancia o estancias vecinas visibles) y lo que oye (el canto de su musa y el ruido de la sala de música), sin tener que recurrir a información privilegiada (únicamente recordando lo que ha visto anteriormente) [1 pto.].

El *banco de pruebas* realizadas y documentadas en la grabación, cubrirá todas las funcionalidades del prototipo [1 pto.], demostrando sus posibilidades con aquellas *métricas* que permitan valorar la eficiencia de la implementación: en este caso, cantidad de estados en la máquina de estados o tareas en el árbol de comportamiento, tareas ejecutadas con éxito y tareas ejecutadas con fracaso, tiempo tardado en completar con éxito un secuestro [1 pto.].

El **código fuente** y las buenas prácticas de desarrollo software (*commits* frecuentes en la rama principal, organización de recursos en el proyecto y de objetos en la escena, escritura sistemática de los comentarios, etc.) se tendrán en cuenta en la revisión final [1,5 ptos.].

3. Restricciones y consejos

A la hora de desarrollar este proyecto es obligatorio:

- Utilizar únicamente las herramientas de Unity y los *plugins* de terceros *Bolt* y *Behavior Designer* acordados con el profesor, sin reutilizar código ajeno a estos.
- Documentar claramente los algoritmos, heurísticas o cualquier “truco” utilizado.
- Diseñar y programar de la manera más limpia y elegante posible, separando la parte visual e interactiva del juego, del modelo y las técnicas de IA implementadas.
- Evitar, en la medida de lo posible, el uso de recursos audiovisuales pesados o ajenos.

Para realizar las pruebas y facilitar las revisiones de los profesores, intentando aprovechar el esfuerzo de desarrollo, conviene crear una interfaz gráfica cómoda para mostrar distintos escenarios de ejemplo, instrucciones de uso, etc. Por ejemplo, habilitando una “consola de trucos” (o teclas rápidas) que permitan ver los datos de los desarrolladores, reiniciar la ejecución, cambiar la cámara, establecer situaciones específicas para hacer pruebas, hacer invencible al avatar, etc. El manejo debe ser ágil e intuitivo para poder realizar rápidamente todas las pruebas con aquellas variaciones que puedan resultar interesantes.

4. Referencias y ampliaciones

Como base para vuestra investigación, además de la bibliografía de la asignatura, puedes aprovechar las siguientes referencias. En ningún caso debes replicar código que encuentres por ahí; asegúrate de entenderlo y verifica que funciona *exactamente* como se pide que lo haga.

- Opsive, Behavior Designer
<https://opsive.com/assets/behavior-designer/>
- Unity, Bolt Visual Scripting
<https://docs.unity3d.com/bolt/1.4/manual/index.html>

- Unity, Navegación y Búsqueda de caminos
<https://docs.unity3d.com/es/2019.3/Manual/Navigation.html>
- Unity 2018 Artificial Intelligence Cookbook, Second Edition (Repositorio)
<https://github.com/PacktPublishing/Unity-2018-Artificial-Intelligence-Cookbook-Second-Edition>
- Unity Artificial Intelligence Programming, Fourth Edition (Repositorio)
<https://github.com/PacktPublishing/Unity-Artificial-Intelligence-Programming-Fourth-Edition>

Para ir más lejos en tu aprendizaje y alcanzar la máxima calificación, realiza estas ampliaciones:

- Crea un escenario con geometría compleja, con portales que unen distintas zonas y saltos insertados en la malla de navegación.
- Crea un escenario con mecanismos más complejos, como botones que abren y cierran pasadizos, puertas giratorias, ascensores o rampas controlados por temporizadores, etc.
- Mejora el razonamiento del fantasma sobre el estado y la posición de los distintos elementos (las barcas, la cantante, el vizconde, etc.), de modo que pueda tomar decisiones más inteligentes, considerando los efectos causados por otros personajes.
- Mejora la gestión sensorial, de manera que el fantasma vea y oiga a los personajes no necesariamente compartiendo estancia, razonando también con esas percepciones, y mostrando realimentación visual para que quede claro lo que conoce el fantasma.
- Plantea incluir varios fantasmas y varias cantantes, funcionando de manera simultánea.
- Plantea tener un cargador de niveles (construidos modularmente) desde un fichero.