

IA-P3: El Fantasma de la Ópera

Preparación	1
Desarrollo	2
Entorno virtual y Raoul de Chagny	2.a
Lámparas	2.b
Christine Daaé	2.c
Árbol de comportamiento del Fantasma	2.d
Ampliaciones	3
Utilizar técnicas de movimiento	3.a
Crear un escenario con geometría compleja	3.b
Sofisticada gestión sensorial	3.c
Pruebas	4

1. Preparación

Para esta práctica, dado que integrar todas las nuevas carpetas en las carpetas de las prácticas anteriores era una tarea ardua, hemos decidido poner en una carpeta distinta todo lo referente a Unity de esta práctica. Sin embargo, la estructura de todo lo demás sigue siendo la misma, como los apuntes, enunciados y documentaciones, que siguen en el mismo sitio. El repositorio, como es de suponer, vuelve a ser el mismo de GitHub: https://github.com/jeramauni/IA_2020.

2. Desarrollo

a. Entorno virtual y Raoul de Chagny

Todo el escenario ha sido creado **íntegramente** por nosotros. Está compuesto en su mayoría por cubos y cilindros deformados, pero además de esto, contiene modelos creados en **blender**, principalmente hechos para la iluminación de las estancias. Estos modelos son: las *lámparas* del teatro y las *antorchas*. Así mismo, también hemos creado el modelo de un *botón* con el diseño de un semáforo, el cual cuando está en verde significa que se puede tirar la lámpara, y cuando está en rojo significa que la lámpara no se puede tirar, pues ya está caída (aunque esto es visualmente obvio).

Para el diseño de la escena hemos tenido en cuenta cómo sería a grandes rasgos un teatro real, y, dado que nos interesaba que para dotar de *dinamismo* a las interacciones de los personajes, hemos creado **estancias a varios niveles** conectados por escaleras, pasarelas, y rampas. Todo este entorno está recubierto de una malla de navegación, la malla que nos otorga Unity para poder crear automáticamente los algoritmos de búsqueda. Todos los tejemanajes de la malla y demás objetos los iremos explicando a continuación sala por sala:

Patio de Butacas

El **patio de butacas** se compone de dos sectores: el sector *inclinado* y el sector *llano*. El sector inclinado es donde se sitúan todos los espectadores de la ópera, mientras que en el sector llano en principio no hay nada más allá de elementos decorativos, o pequeñas escaleras para subir al escenario.

Los **espectadores** están regidos por dos scripts, de los cuales el principal es *MovEspectadores.cs*. Este script es el que maneja qué deben hacer los espectadores cuando se cae alguna de las lámparas. Y no solo eso, si no que solo cuando es estrictamente necesario, o sea, se haya modificado el estado de alguna lámpara, manda un mensaje de que hagan la acción pertinente.

Por por otro lado están las **lámparas**, de las cuales se hablará en el apartado (b), se sitúan en la parte alta de la zona inclinada.

Escenario

En el **escenario** hay *elementos decorativos*, como las cortinas delanteras y rojas, simplemente decorativas, y las cortinas traseras y negras, que sirven para separar el escenario de las bambalinas. Además de esto, detrás de las cortinas rojas de la izquierda hay una trampilla pequeña representada por un cuadrado plano de color marrón.

También hay que destacar que el *escenario* está **elevado** sobre el patio de butacas 1 m. Esto hace que cualquier agente que quiera pasar de uno a otro solo tenga dos opciones: las *escalerillas* de la derecha o las de la izquierda.

Palco Oeste

Al palco se accede desde el *escenario* por una **escalera compensada**, las cuales a su vez cuando se observan de lejos con las escaleras simétricas del palco este intentan simular unas **escaleras imperiales** de teatro, y se complementan por unas *barandillas decorativas*.

En la parte superior hay un palco circular con unos muros a forma de valla que recubren todo el habitáculo. Además de esto, también encontramos un botón que puede tirar la lámpara, del cual se hablará más adelante, en el apartado (b).

Por último hay otras escaleras, unas **escaleras en L**, que conectan el palco con el sótano oeste, con sus respectivas barandillas.

Palco Este

Simétrico al *palco oeste*.

Bambalinas

Se sitúan detrás de las *cortinas negras*, y de ella sale una rampa en 45 grados hacia el *sótano este* y unas **escaleras simples** que van hacia el *sótano oeste*.

Sótano Oeste

Además de las dos entradas mencionadas, tiene un *NavMeshLink* que hace que los agentes puedan saltar de esta estancia hasta la estancia de la cárcel.

Sótano Este

<p>Simétrico al sótano oeste, pero además tiene un <i>NavMeshLink</i> extra, un camino va al <i>sótano norte</i>, y el otro va a la <i>sala de música</i>.</p>
<p>Cárcel</p> <p>La cárcel tienen una celda, y además unas escaleras de piedra que suben a un segundo nivel en el que está el sótano norte. La celda está compuesta por varios barrotes, los cuales <i>solo son atravesables</i> los del lado sur.</p>
<p>Sala Norte</p> <p>Esta sala es simplemente un <i>pasaje elevado</i>, que conecta con la <i>cárcel</i> y el <i>sótano este</i>.</p>
<p>Sala de Música</p> <p>La sala de música tiene un piano, el cual cuando el jugador interactúa con él, lo destroza, haciendo que el Fantasma quiera dejar inmediatamente cualquier cosa que esté haciendo.</p>

Los *NavMeshLink* no eran parte de los recursos del proyecto, así que fué necesario incluir una nueva librería para que pudiera haber conexiones entre zonas.

Hay que aclarar cómo funcionan los *NavMeshLink* de los ríos. En primer lugar son **unidireccionales**, por lo que solo pueden ir de arriba a abajo o viceversa. En segundo lugar, en cada río hay una barca situada inicialmente en un lado, acorde con la dirección del *link*, y cuando un agente llega al salto, y si este salto se puede realizar en esa dirección, navegará por el *link*. Mientras navega, la barca **se hace hijo** de este agente para que la posición de la barca sea la misma que la del agente. Cuando el agente llega al otro lado, un *collider* hace que la barca se suelte del padre e **invierta** el *link*, de forma que esté listo para un nuevo trayecto en la dirección inversa.

Dicho esto, ahora aclararemos cómo es el comportamiento del jugador. El **jugador** se controla por cursores y navega por la *NavMesh*, de tal forma que puede considerarse un *NavMeshAgent*.

b. Lámparas

Las **lámparas** las hemos modelado por blender, y que tienen dos luces integradas, la *luz direccional*, que apunta hacia el suelo, y la *spotlight*, que da una luz ambiental más completa. Como es de suponer, hemos decidido quitar la luz del “sol” de unity, para así dar una sensación mayor de interior.

Cuando las lámparas se caen, desactivamos la *spotlight* para que parezca que el teatro se ha quedado en penumbra, pero dejamos la *luz direccional*, que cuando está caída apunta al escenario, **pues la lámpara ha volcado**.

El comportamiento de la lámpara se rige por el script *Lamp.cs*, la cual se puede **reparar**, o **sabotear**. Para sabotearla, hay una **palanca**. Cuando la palanca se activa, visualmente y mediante el script *Lever.cs*, la luz que tiene se pone de color rojo en señal de

que la lámpara está caída, y no volverá a ponerse en verde hasta que el jugador repare la lámpara.

La forma de *sabotear la lámpara* es que tanto el Fantasma como Raoul vayan al lugar donde hay situada alguna de las palancas y al entrar en el collider:

1. Si eres *Raoul*: le des a la 'F' para interactuar con la palanca
2. Si llega el *Fantasma*: interactúa con ella directamente.

La forma de reparar la lámpara es única, y solo puede suceder cuando Raoul entra en el collider de la lámpara

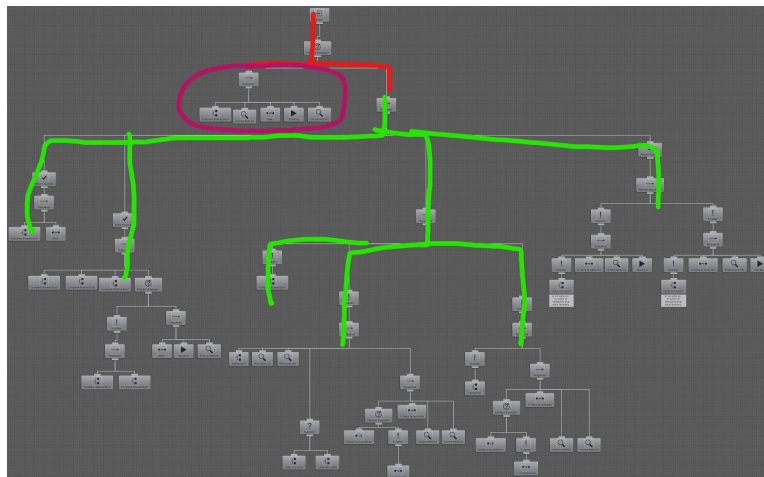
c. Christine Daaé

Christine tiene un árbol de comportamiento que trata todas sus acciones, es capaz de salir al escenario a cantar y si por algún casual tiene una crisis se dirige detrás del escenario a refugiarse en las bambalinas para llorar, esperando que el vizconde venga a consolarla y poder cantar de nuevo. Si por algún casual el fantasma consigue atraparla, dejará de cantar de inmediato ya que es amordazada por éste.

La cantante es a su vez un *NavMeshAgent* que se desplaza por el teatro gracias a la malla de navegación incorporada en Unity. Esto le permite acceder a todos los comportamientos implementados para los agentes.

No contentos con esto, hemos incorporado un archivo libre de *copyright* para la voz de *Christine*, para que los personajes puedan oírla.

d. Árbol de comportamiento del Fantasma

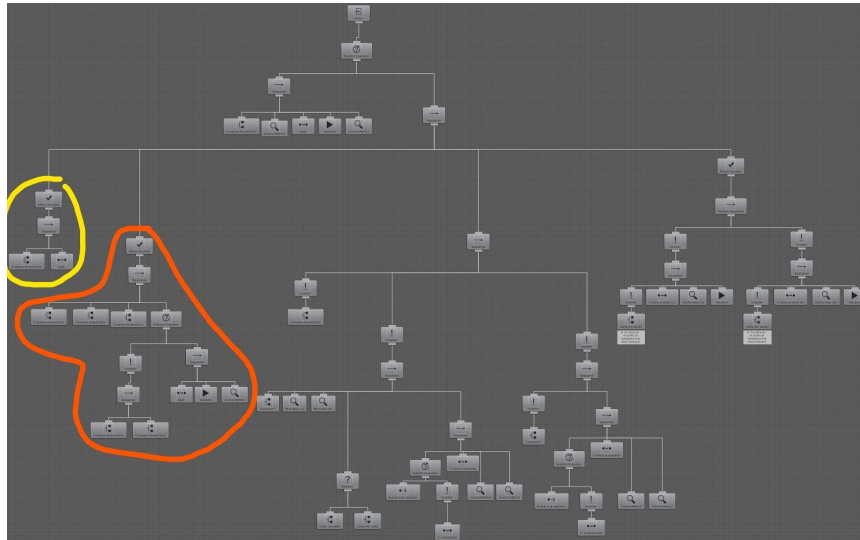


La estructura del árbol del fantasma de la ópera sigue un patrón fijo. En primer lugar, las ramas hijas de la raíz u otras ramas se dividen en dos partes:

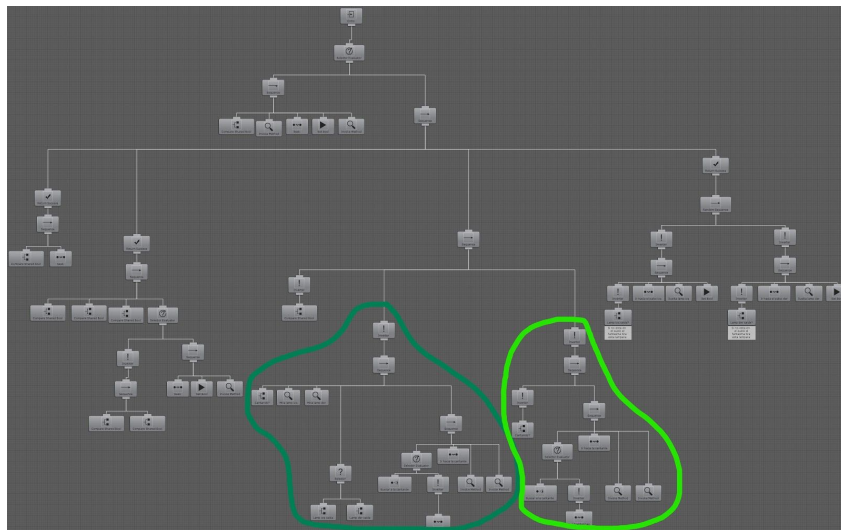
- Izquierda: condiciones a cumplirse para la ejecución de la derecha.
- Derecha: Conjunto de acciones a realizar y punto de nacimiento de una nueva rama.

Teniendo en cuenta lo mencionado previamente damos paso a la explicación de las partes de nuestro árbol de comportamiento.

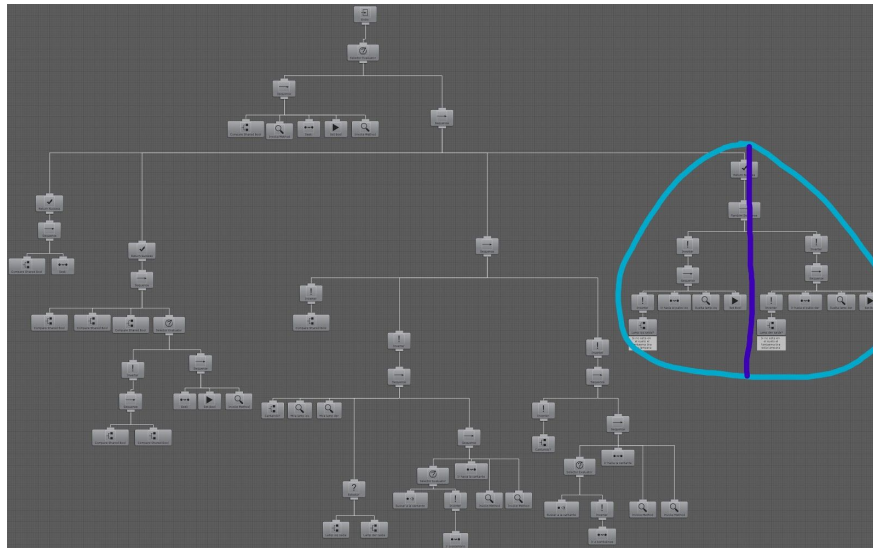
Como podemos apreciar en la imagen superior el tronco del árbol (**rojo**) se divide en la *condición maestra* (**Magenta**) y la *rama principal* (**Verde**). La condición maestra es capaz de parar toda la ejecución de la *rama principal* (**Verde**) y comprueba en este caso que el vizconde no haya destrozado la sala de música.



Las dos ramas menores izquierdas de la *rama principal* son: La que se encarga de llevar al fantasma hasta la sala de música una vez haya encarcelado a su víctima la cantante (**Amarilla**). y la que se encarga de llevar a la cantante a la cárcel una vez ha sido capturada (**Naranja**).



La rama mayor de todo el árbol tras la *rama principal* es la rama de búsqueda y captura dividida a su vez en dos ramas menores: La rama de búsqueda y captura en escenario (**Verde oscuro**) y la rama de búsqueda y captura en bambalinas (**Verde claro**). Ambas se encargan de, si las condiciones son las adecuadas; ir, buscar y capturar a la cantante.



La subrama más a la derecha del árbol (**Azul**) es la encargada de tirar las lámparas para dotar al fantasma de acceso al escenario. Esta rama se divide en dos, cada una de las partes encargada de su lámpara correspondiente.

3. Ampliaciones

a. Utilizar técnicas de movimiento

El movimiento de los personajes está controlado por el **NavMesh** de Unity, y como el escenario está compuesto de diversos obstáculos y caminos de un solo sentido como es el caso de rampas o trampillas, los personajes deciden la ruta más óptima para viajar hasta el punto donde desean llegar, en el caso del fantasma por ejemplo es consciente de que en los ríos solo puedes pasar a través de ellos si hay una barca varada en la orilla en la que se encuentra en caso contrario deberá buscar un camino mejor.

b. Crear un escenario con geometría compleja

El escenario se compone de múltiples formas combinadas para poder crear un escenario realista y con un gameplay más puro, metiendo diferentes tipos de escaleras, formas de salas, diferentes alturas y múltiples detalles como la iluminación, las barandillas y antorchas situadas por el mapa.

Tanto la lámpara, las antorchas y los botones están hechos en blender, por lo que ha sido necesario el uso de recursos externos para la realización de la práctica.

c. Sofisticada gestión sensorial

En cuanto a la gestión sensorial, el fantasma tiene *vista* y *oído*, es capaz de escuchar cuando la cantante llega al escenario a **recitar su melodía** (incorporada mediante su archivo .ogg), y en ese momento entrar en acción para poder raptarla de manera efectiva.

En cuanto a la vista, el fantasma es **capaz de ver** a la cantante, de esta forma puede trazar las rutas más óptimas para distraer y asustar a los espectadores, y poder hacerse con su víctima de la forma más eficaz posible.

4. Pruebas

Durante el desarrollo del juego hemos ido debugueando mediante puntos de interrupción en *Visual Studio*, *Debug.Log()* en Unity y observación de los árboles de *Behavior Designer*. Es destacable mencionar que hemos creado **una cámara para cada jugador**, y así saber qué hace cada uno en cada momento, y así se pueda hacer más fácil la corrección y visualización de la práctica.

Cuando se ejecuta el juego, observamos cómo el fantasma procede a ir por la derecha o por la izquierda a romper alguna lámpara. Si la cantante va a bambalinas, entonces el fantasma termina de hacer su acción para ir a por ella. Cuando la coge el Fantasma, es necesario que Raoul le haga un placaje o que rompa la sala de música. Es **muy importante** que **no** rompa la sala de música cuando lo Erik y Christine están cruzando con una barca, pues esto hará que Erik suelte en el río a Christine y se quede quieta ahogándose.