

Inteligencia Artificial para Videojuegos

Práctica 5: aliados y enemigos de combate



Grupo 07

Alejandro Ortega Álvarez

Borja Cano Álvarez

Problema a resolver

- La propuesta es rellenar en Unity a partir de un prototipo dado de un shooter en tercera persona ciertas características para darle más jugabilidad al simulador.



- Para ello se piden los siguientes requisitos:
 1. Observar y comprender los árboles de comportamiento de los soldados
 2. Crear varias entidades nuevas de fantasmas, entidades que vuelan por el escenario con un comportamiento de patrullas aleatorias por el mapa, sin perseguir al jugador.
 3. Los fantasmas no mueren, pero se teletransportan a una posición aleatoria del mapa cuando se les dispara.
 4. Los fantasmas hacen daño cuando atraviesan a los soldados.
 5. Los soldados enemigos tienen como prioridad disparar a los fantasmas antes que disparar al jugador.

El juego

La base de juego inicial dispone de:

- Un soldado en 3ª persona con capacidad de desplazarse, correr, saltar, agacharse y disparar, con detección de colisiones para chocarse con las paredes y estar en contacto con el suelo.
- Un mapa cerrado de varios niveles de altura.
- Enemigos y torretas que dañan al soldado y se pueden eliminar.
- Objetos de vida y munición que se pueden recoger.
- Inventario de armamento.



Partes añadidas

1- Añadir Fantasmas

El prefab de los fantasmas lo hemos tenido que pasar del proyecto de IAVP6-BD-TPC. El comportamiento de los fantasmas venía también dado (flock), hemos modificado las variables de dicho comportamiento para que se adapten al entorno:

- Las velocidades normal y angular nos pareció que estaban correctas.
- La distancia que decide si un elemento es vecino o no en principio no influía al comportamiento pero decidimos aumentarla un poco.
- La distancia al objetivo (Look Ahead Distance) sí que fue importante aumentarla, ya que con la que venía los fantasmas trazaban trayectorias demasiado cortas y apenas se movían del sitio. Tuvimos que triplicarla para que realizaran patrullas más reales.
- También disminuimos el radio de obstáculo, porque a veces no cabían por sitios estrechos como escaleras o pasarelas y se quedaban parados.

2- Hacer que los fantasmas hieran al personaje

Simplemente creamos un nuevo script añadido al personaje para que cada vez que entrara en contacto con algún fantasma, llamara a la función `<CharacterHealth>().Damage()` y sufriera daño.

El daño es continuo, cuanto mayor sea el tiempo que está en contacto mayor es el daño.

Para la colisión, proporcionamos un Capsule Collider a cada fantasma adaptada más o menos a su forma y en forma de trigger, para que atravesara sin problema los obstáculos. De esta forma la comprobación del daño en el código se hace con la función `OnTriggerEnter()`.

3- Teletransporte de los fantasmas cuando son disparados

Para ello modificamos el script de `ShootableWeapon`, ya que vimos que era más lógico y eficiente que el fantasma desapareciera al recibir una bala de cualquier arma, no sólo de la del personaje. Éste script es el que se encarga de lanzar el Raycast de cualquier arma para comprobar si ha habido algún obstáculo con el que haya chocado y a qué distancia. En el método de comprobación del Hit del Raycast, es donde le preguntamos si el objeto con el que ha chocado pertenece al layer Ghosts (creado por nosotros, en el layer número 21) y, en caso afirmativo, genera una posición aleatoria dentro del mapa para que el fantasma con el que haya chocado se teletransporte.

4- Hacer que los fantasmas hieran a los demás soldados

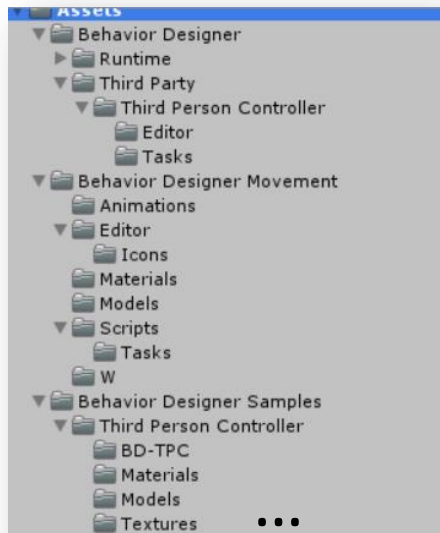
Simplemente hay que añadirle el componente de `GhostCollision` (del punto 3 para herir al agente) a los demás guardias del mapa. Lo malo de este punto es que no hay forma directa de comprobarlo ya que los fantasmas no siguen a los soldados para chocarse con ellos ni al revés, por lo que es difícil que ocurra la situación para verlo. Pero puesto que con el personaje funciona, debería funcionar también.

5- Salir del .exe con el ESC

Que no estaba puesto.

Problemas

-Principal problema el no saber dónde estaban las cosas. Los elementos del prototipo que se nos proporcionaba estaban almacenados en demasiadas carpetas (algunas con nombres muy parecidos o idénticos) y era complicado buscar la ruta si no era fuera de Unity. Aun habiendo encontrado el archivo que buscábamos, era muy difícil detectar el elemento que queríamos tocar, por lo que ha habido apartados que directamente no hemos conseguido lograr por no saber dónde se modificaba.



???

-Demasiado tiempo empleado para 4 modificaciones que tiene la práctica, casi todo el tiempo empleado en tratar de comprender cómo funcionaba y buscar los elementos que necesitábamos. Y sin el tutorial que nos mandaste hubiéramos tenido que gastar el doble de tiempo más.

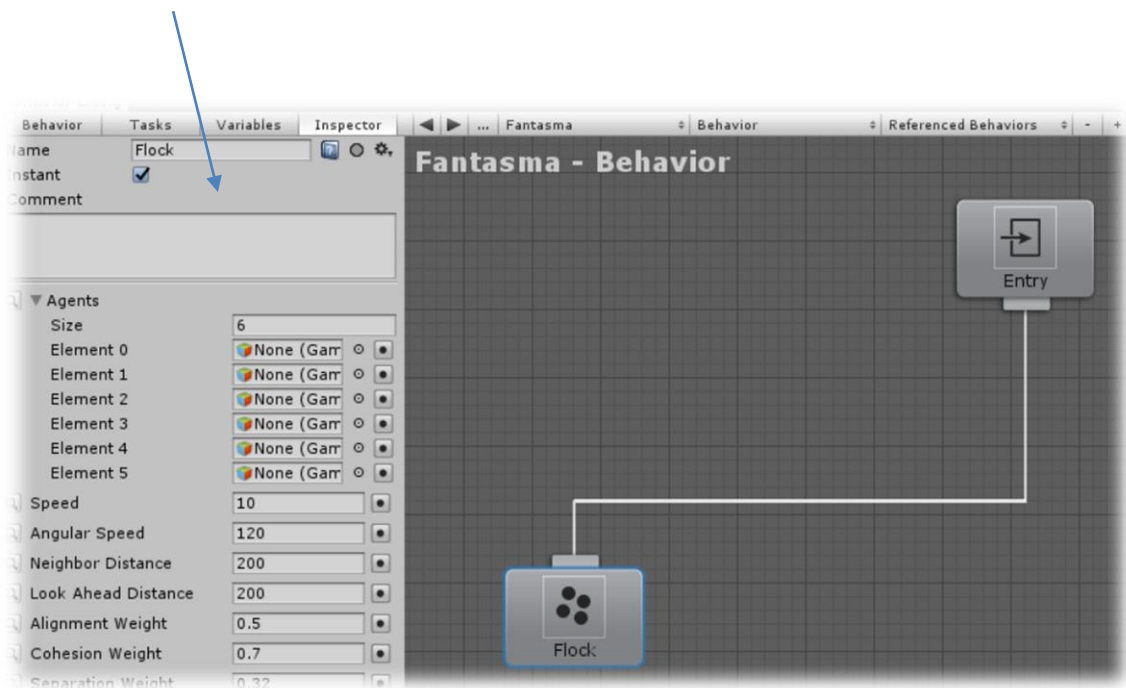
-En el caso de los fantasmas, se podía pasar el prefab de un proyecto a otro pero no con las partes necesarias (sólo nos aparecía el cuerpo y tuvimos que meter a mano los brazos, la cabeza y las texturas).



-Hicimos perfectamente el pequeño detalle de que “flotaran” los fantasmas, con el seno en el script Flock para mover la variable que posicionaba la distancia con el suelo, pero al pasarlo al otro proyecto se perdió y no conseguimos volver a encontrar la variable.

-Dificultad a la hora de tocar los Scripts que ya venían, por ser demasiado extensos y no conseguir encontrar nada, al final la solución para las implementaciones era crear otro script auxiliar que accediera al que correspondiera.

-Nos pasaban cosas extrañas con el comportamiento de los fantasmas (completamente el mismo que en el otro lado); cuando conseguíamos que funcionara correctamente, nos poníamos a tocar otra cosa y, sin haber tocado el comportamiento, dejaban de funcionar, y hacía cosas raras como que los fantasmas se activasen al pasar un soldado enemigo cerca. Además, en la ventana del comportamiento, cada vez que reiniciamos Unity, se pierden las referencias a los fantasmas y hay que volverlas a poner cada vez que arrancamos la práctica, cosa que nos ha fastidiado bastante porque también se pierde al hacer el ejecutable y hace que no funcione la práctica.



-Solucionamos esto al final en el método OnStart() del Script del comportamiento, buscando todos los objetos con el tag Ghost y metiéndolos en el array Agents.

-En el resultado final, habiendo tocado las distancias del objetivo, con los vecinos y demás hemos conseguido controlar el comportamiento salvo porque a veces se quedan parados unos breves segundos, no nos gusta pero creemos que es porque trata de buscar una ruta imposible y hasta que no le llegue otra no continúa, y eso no sabemos cómo arreglarlo ya que es de la propia implementación del algoritmo.

-El único apartado que nos quedó por hacer fue el de que los soldados disparasen a los fantasmas. Le proporcionamos un nuevo comportamiento a su árbol, más prioritario que el de disparar al jugador que fuera exactamente igual pero disparando a los fantasmas y sin tener que ser dañado para activarse. Pero nos daba referencia nula al objeto pese a haberlo

cambiado y tuvimos que quitarlo para que no diera errores. Además, fue un lío ya que tratamos de crearlo a partir de una copia del comportamiento que ya tenía de disparar al soldado, pero al modificarlo se modificaban los 2 comportamientos y no había forma de crearlo desde 0 ya que había que añadir muchas cosas, al final después de unas cuantas pruebas tuvimos que dejarlo como estaba.

-También problemas con la carga de Unity. Al principio no tanto, pero después de varios cambios al darle al play a la simulación se quedaba pillado un rato bastante largo debido al peso del proyecto, en una ocasión tuvimos que forzar el cerrado de la aplicación y es bastante fastidio si no nos damos cuenta de guardar los cambios.

Resultados

-En el resultado final tenemos el juego base con el añadido de los fantasmas casi a la perfección. Se mueven con el comportamiento indicado, si atraviesan a los soldados les causa daño (a los soldados enemigos también, les hemos añadido el mismo componente) y si se les dispara a los fantasmas se teletransportan a un punto aleatorio del mapa.

-Por lo que tenemos una simulación jugable bastante más complicada que lo que era antes (que por cierto, era ya difícil). El jugador tiene que tratar de eliminar a todos los soldados enemigos sin que le eliminen y además, ir esquivando los fantasmas para que no choquen con él y le vayan quitando más vida (Hay 6 fantasmas en total por el mapa).

Referencias y apoyos del desarrollo de la práctica:

- *Apuntes del Campus Virtual y enlaces de los prototipos*

Repositorio de Github: <https://github.com/alex97ortega/IA.git>