Nerea Castellanos Rodríguez

Catherine Castrillo González

Sandra Fraile Infante

Stoycho Ivanov Atanasov

Julia Martínez Valera

Gaspar Rodríguez Valero

SKYSCRAPERS   Ver.: 2.0

Sistema de Decision



# Introducción

El sistema de decisiones es aquel que se encarga de gestionar y decidir que realizara en la siguiente actualización del bucle general del juego. Para hacemos uso de dos sistemas de decisión, una máquina de estados y behaviour trees. Nuestra IA dispone de varios estados y dependiendo de uno u otro gestiona un árbol diferente. Así creando un comportamiento complejo en situaciones concretas.

# NPC Agresivos

Son los NPC del mapa cuyo comportamiento hacia el jugador es negativa y buscan la muerte del mismo, los enemigos.

# Behaviour Tree

Los behaviour trees se encargan de gestionar y seleccionar cual es la decisión que tomara la IA y en consecuencia realizar las acciones que dicha decisión acompañan. El sistema anterior comprendía un total de 4 arboles gestionados por 4 estados distintos lo que provocaba una complejidad innecesaria para las acciones que la IA debía tomar. Por ello se ha realizado un único árbol que gestionara las acciones de la IA haciendo el sistema más simple y con la misma cantidad de acciones y decisiones.

Este sistema es susceptible de ser modificado en el futuro por posibles mejoras o cambios en futuras iteraciones. Aquí se puede ver el nuevo árbol y los nuevos nodos que lo conforman.

En este árbol los tipos de nodos han sido modificados a otros muy distintos por ello a continuación vamos a describir los distintos tipos de nodo:

* Nodo Secuencia: Ejecuta de forma ordenada los nodos hijos que les pertenece, pero a la primera respuesta negativa termina su ejecución o en un estado running.
* Nodo Acción (VERDE): Se trata de un nodo final donde se lleva la ejecución de la acción.
* Nodo Condición: Controla si cumple los requisitos para continuar adelante por la rama del árbol.
* Nodo Secuencia Positiva (ROSA): Ejecuta cada uno de sus hijos hasta alcanzar un nodo hijo que devuelva una respuesta positiva o un estado running.
* Nodo Condición Secuencial (AZUL): Es la unión de un nodo de secuencia y uno nodo secuencia, busca que primero se pase la parte de la condición y después ejecutar sus hijos hasta que uno le devuelva un valor negativo o un estado running.
* Nodo Subrutina (AMARILLO): es un nodo acción un poco más complejo pues se trata de una acción más larga que el resto.

# NPC Amigables

En el caso del nuestro proyecto el único NPC amigable se trata del rehén de cada nivel el cual desea ser rescatado. Toda la parte redactada a continuación se encuentra en desarrollo, se tienen notas y planteamientos teóricos, pero aún estamos con la primera iteración.

# Máquina de Estados

Buscamos que el rehén fuera gestionado por una máquina de estados simple ya que la cantidad de acciones o estados del mismo son reducidas. Hemos decido que tendrá tres estados simples:

* **Reposo**: En este estado se encuentra cuando está esperando al NPC a lo largo de la partida o cuando tras ser salvado por el jugador, el mismo entra en combate, el rehén huya y vuelva ese estado hasta que se encuentre de nuevo con el jugador.
* **Huida**: Cuando el rehén se encuentra involucrado en un combate, esta huira hasta encontrar sitio seguro.
* **Seguimiento**: En este estado, el rehén seguirá de forma continua al jugador hasta la posible donde terminará la partida.

# Notas

## Versión 1.0

Primera versión funcional del sistema de decisión, muchas de las decisiones se hallan deshabilitadas a espera de la creación de un Trigger System y un Blackboard dinámico con su correspondiente arquitectura. En esta versión se han comprobado:

* Arboles de decisión probados (Orden correcto y gestión del estado running optimizado)
* Primera versión del nodo mover (nodo critico que gestiona mucha de las decisiones del sistema)
* Comprobada la máquina de estados para la gestión de árboles.
* Árbol estándar testeado en sistema 3D
  + Se cura, alimenta o bebe cuando lo requiere
  + Realiza su rutina cuando debe.

## Versión 2.0

Segunda versión funcional del sistema de decisión, las decisiones respecto al sistema de combate se hallan sin implementar. Se ha implementado los distintos blackboard y el trigger system. Se ha eliminado la parte de los estados y los arboles han sido reducidos a un único árbol. En esta versión se ha comprobado:

* Arbol de decisión comprobado, tanto orden de ejecuccion, como el estado running y el reset del mismo.
* Segunda versión del nodo mover (gestión de múltiples posiciones a modo de pila) y cálculos simplificados.

# Bocetos y Anotaciones



