



Controlador de inteligencia artificial multicapa

Controlador de inteligencia artificial multicapa

03 Introducción

Breve resumen de la intención y motivación del proyecto y planteamiento del problema a resolver.

04 Controlador

Exposición del modelo de datos y del funcionamiento del controlador propuesto.

06 Resultados

Análisis y discusión de los resultados obtenidos con la aplicación del controlador descrito.

INTRODUCCIÓN

En videojuegos, los personajes no controlados por el jugador suelen poseer una inteligencia artificial compleja formada por varios sistemas receptores y varios sistemas de toma de decisiones. Esto solo es posible si se organiza bien el sistema multicapa con unas comunicaciones rápidas que agilicen los cálculos y reutilicen información ya calculada.

En un videojuego en el que el jugador lleva un estilo de juego frenético matando enemigos se espera que los enemigos también actúen con la misma rapidez, sacrificando en ocasiones premeditación. Del mismo modo, en juegos tácticos se espera de la IA que funcione con la misma madurez de análisis y toma de decisiones que aquella que se espera de un jugador humano.

Para conjugar estas características se necesitan diversas capas que trabajen a diferente velocidad en el ciclo de juego y que busquen objetivos distintos. Así se puede abarcar el máximo nivel de manejo de información sin tener que sacrificar demasiado rendimiento.

En este documento se expone la creación de un controlador de Inteligencia Artificial aplicado a un videojuego de tipo shooter. Usando algoritmos de búsqueda y heurísticas en las tomas de decisiones, este controlador permite el funcionamiento de los enemigos de un escenario que combinan la búsqueda de coberturas y armas con identificar la posición del jugador para poder dispararle.

Título

Controlador de inteligencia artificial multicapa

Autor

Jesús Fermín Villar Ramírez

Asignatura

Ingeniería del conocimiento: IA

Dicente

Luis Peña Sánchez

Grado

Diseño y Desarrollo de Videojuegos, ESNE

Fecha

Mayo 2019

Fotografía de portada

<https://www.pexels.com/photo/brown-egg-shell-on-white-surface-929774/>

En Madrid, a mayo de 2019

Jesús Fermín Villar Ramírez

Diseño y Desarrollo de Videojuegos, ESNE

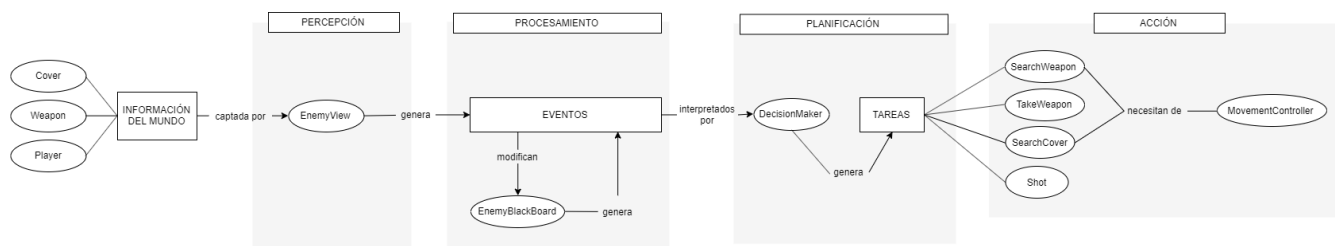


Figura 1. Esquema del controlador

El controlador propuesto consta de diferentes capas. Una primera capa que recibe la información del mundo, o capa de percepción; una capa de procesamiento donde se analiza esta información percibida; una capa de planificación donde se toman decisiones en base a la configuración del mundo percibida y procesada; y una última capa de acción que ejecuta las decisiones tomadas.

Percepción

La capa de percepción es la manera que tiene el agente de recibir la información del mundo. El funcionamiento de esta capa se basa en una percepción visual mediante el casteo de rayos en un campo visual (figura 2). Este comportamiento está controlado por la clase EnemyView. Estos rayos, al colisionar con elementos del escenario (coberturas, armas, el jugador...) lanzan eventos para analizar dicha información.

Procesamiento

Los eventos lanzados por el sistema perceptor modifican distintos parámetros recogidos en el EnemyBlackBoard. Esta clase es un contenedor de la información actualizada que posee el enemigo de sí mismo y del entorno en todo momento. Por ejemplo, en

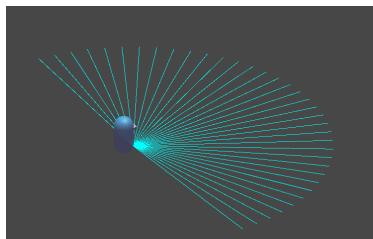


Figura 2. Esquema de percepción

el caso de que el enemigo perciba una nueva cobertura, esta será añadida a la lista de sus coberturas conocidas; si ve al jugador modificará su estado de percepción del jugador... Estas modificaciones en el EnemyBlackBoard también generan, a su vez, eventos que devuelven el control al DecisionMaker para poder planificar la/ las acción/ acciones que realizará el enemigo.

Planificación

En función de la información recogida en el BlackBoard, la clase DecisionMaker planificará la acción siguiente. Por ejemplo, si el enemigo ve al jugador pero este [el enemigo] no está armado, buscará un arma para poder dispararle después; mientras que si está armado buscará una cobertura si ve que el jugador le está mirando para poder dispararle desde ahí (o directamente le dispara si se da cuenta que el jugador todavía no se ha percatado de su presencia). Esta planificación se realiza en función de tareas. El enemigo tiene a su disposición diferentes tareas: buscar una cobertura, buscar un arma, tomar un arma o disparar.

Acción

Una vez tomada una decisión, el control pasa a la respectiva tarea. Algunas de estas tareas son más complejas que otras. Por ejemplo la tarea SearchCover o la tarea SearchWeapon implementan un algoritmo de búsqueda A* antes de ejecutar el movimiento. Este algoritmo de búsqueda se utiliza para calcular qué arma o cobertura es la más adecuada. Se tienen en cuenta diversos parámetros a

la hora de calcular la heurística, como la distancia a la cobertura o al arma, el daño y rango en el caso del arma, o el grado de ocupación y el estado de visibilidad por parte del jugador en caso de las coberturas.

Una vez elegida la cobertura o el arma, el MovementController se encarga de realizar la acción locomotriz y, una vez en el destino, genera los eventos necesarios para devolver el control al DecisionMaker.

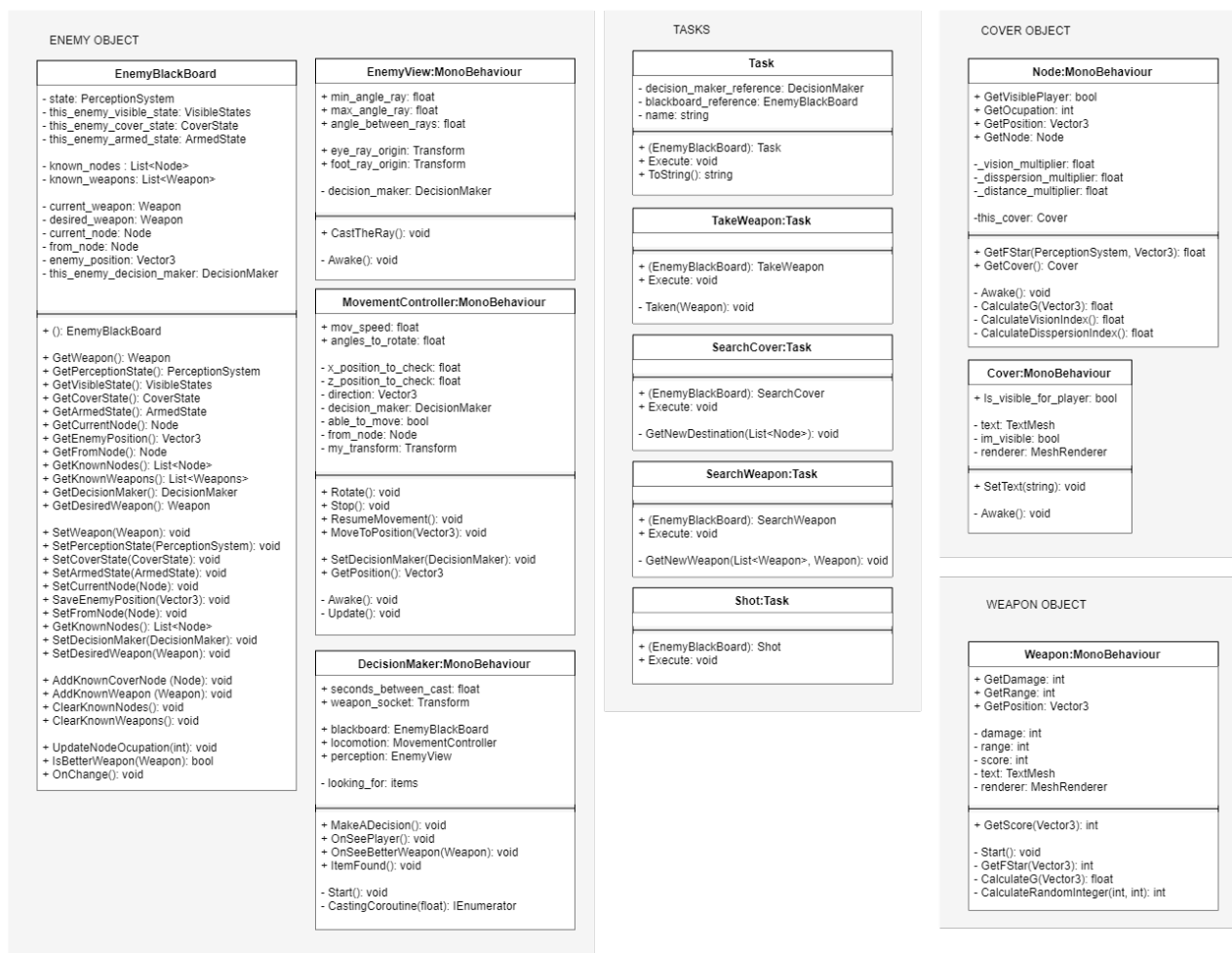


Figura 3. Diagrama de las clases principales del proyecto

Representación de la información

Para facilitar el análisis se ha generado una capa de depuración en la cual se indica qué tarea tiene asignada en cada momento el enemigo así como se coloca, sobre las coberturas correspondientes, el cálculo heurístico de cada uno de ellas (solo de las expandidas por el enemigo) y de las armas vistas por el enemigo.

El panel con fondo negro va mostrando el avance en el mapa, dejando tras de sí un rastro histórico con las posiciones del enemigo. Esta información es útil para conocer los cambios de ruta y ver a qué se deben,

En el panel con fondo gris podemos observar en la franja superior la tarea actual del enemigo. Las coberturas de color verde hacen referencia a las coberturas no visibles por el jugador, mientras que las rosas hacen referencia a aquellas sí visibles. A su lado, en negro, se escribe su valor heurístico en el momento que se calcula. El resto de puntos son las armas, con diferentes colores (blancas, grises,

azules y moradas). Estas también muestran a su lado, en azul, el valor heurístico calculado.

Búsqueda del arma

La primera tarea que se le presenta al enemigo (figura 4) es encontrar un arma ya que está desarmado. El enemigo ha visto al jugador pero ve que le da la espalda por lo que no prioriza esconderse. En su lugar busca el arma con mejor cálculo heurístico. Esta es el arma con valor 18. Esta decisión está justificada por los pesos de cada variable en la heurística. Cambiando dichos valores se ve un considerable cambio en la toma de decisiones en cuanto a elección de armas. Actualmente se priorizan las estadísticas del arma antes que la distancia a la misma.

Un arma mejor

Al vislumbrar un nuevo arma mejor (figura 5), el enemigo cambia su ruta para dirigirse hacia ella. Esto es debido al evento lanzado por el sistema de percepción que avisa al controlador de que la información del mundo percibida por él ha cambiado.

Coberturas

Una vez armado (figura 6) el enemigo va patrullando hasta que encuentra una cobertura desde la cual puede disparar al jugador sin estar al descubierto.

Cambiar el arma

Los enemigos incluso responden al evento de cambiar de arma cuando ya tienen un arma. En la figura (figura 7) se puede ver como el enemigo amarillo al principio llevaba un arma azul consigo hasta que la cambió por una morada.

Más de un enemigo

Cuando metemos varios enemigos en la escena comprobamos que es necesario algún tipo de comunicación entre los enemigos para evitar que den varias vueltas por los mismos sitios que han sido recientemente visitados por otro enemigo. En el escenario actual del controlador (figura 8) se puede observar cómo al no existir esa comunicación los enemigos dan muchas vueltas sobre los mismos lugares.

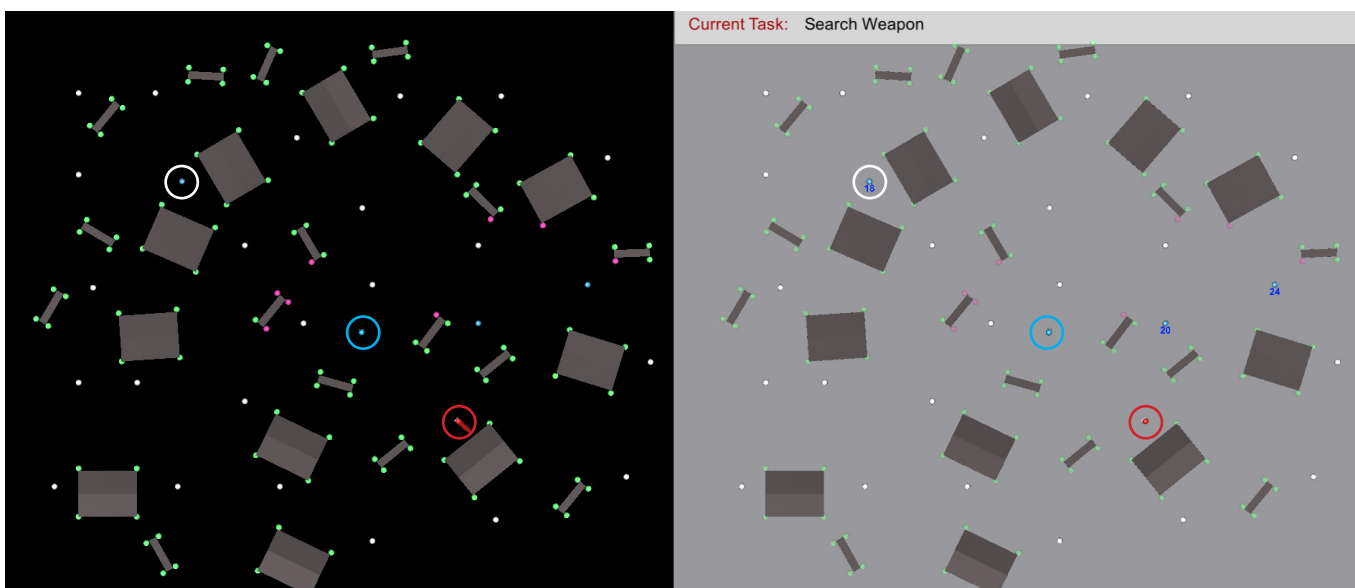


Figura 4. Representación de la búsqueda de arma. Rodeado en rojo está la posición del enemigo, en azul la posición del jugador y en blanco su objetivo.

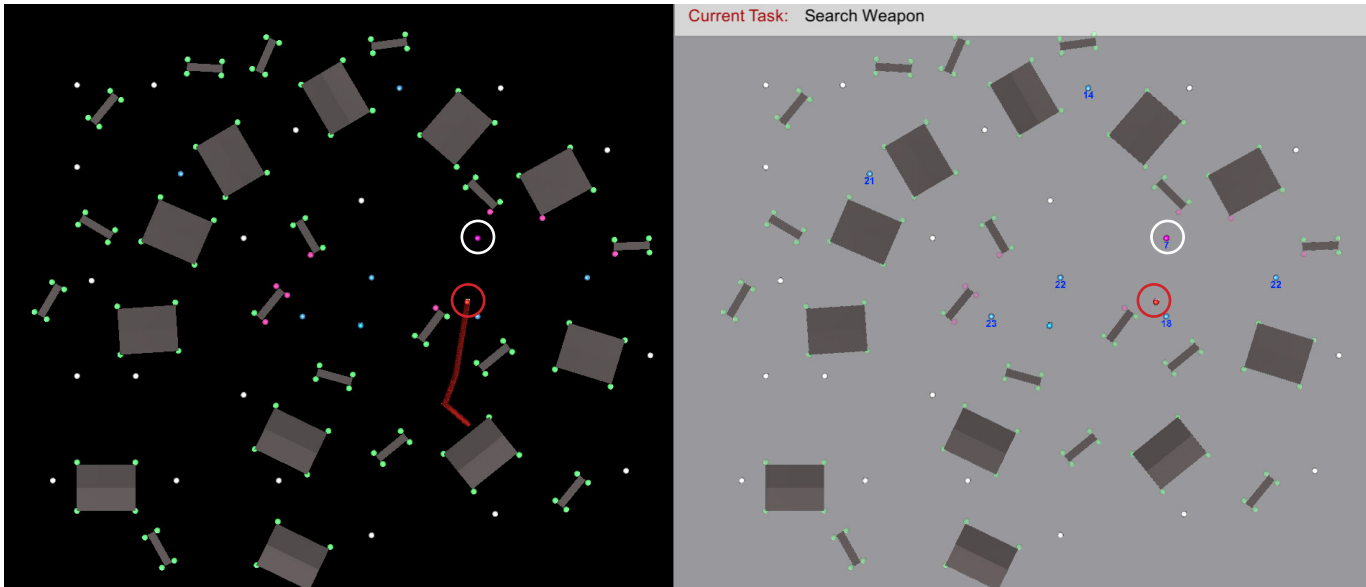


Figura 5. Representación de la búsqueda de arma. Rodeado en rojo está la posición del enemigo y en blanco su objetivo.

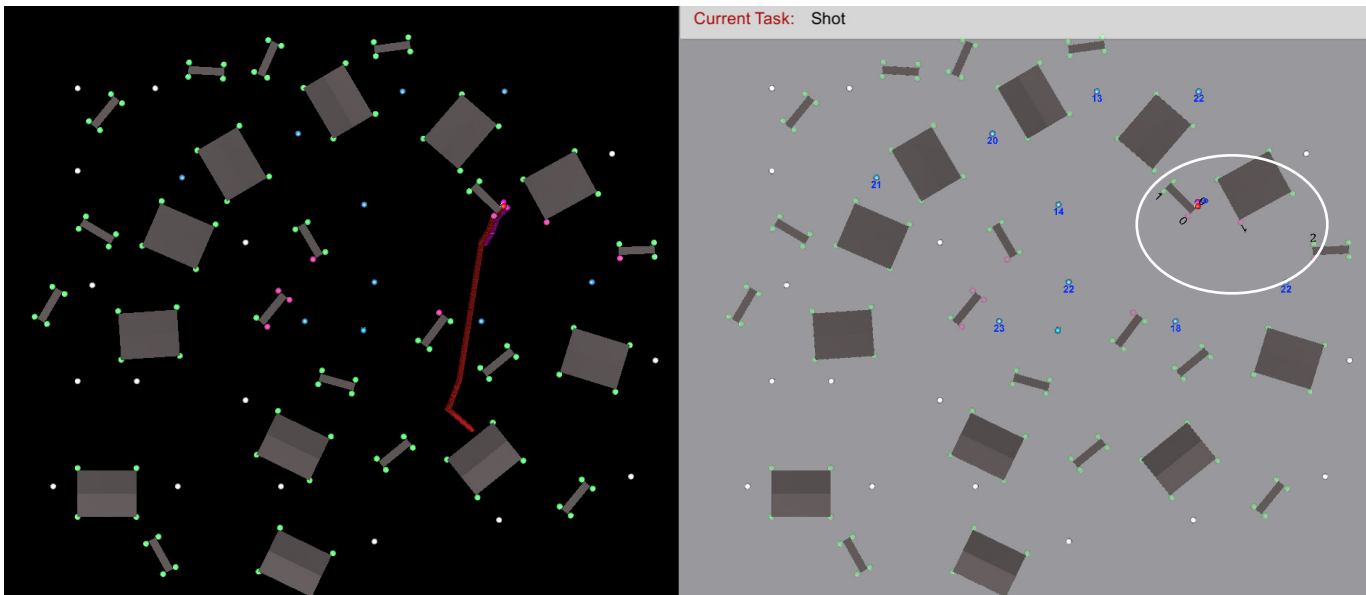


Figura 6. Buscando coberturas y disparo. Rodeado en blanco se ven los valores heurísticos de las coberturas.

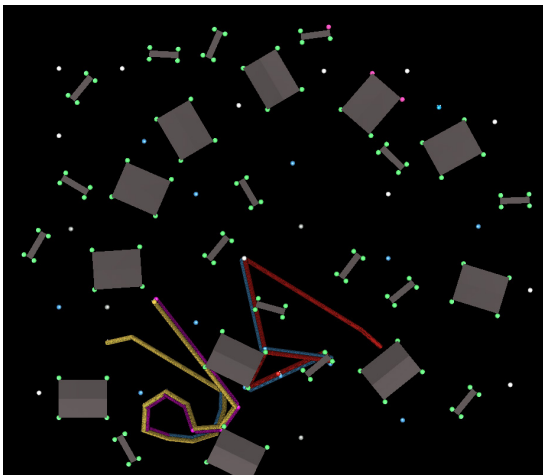


Figura 7. Cambio de arma

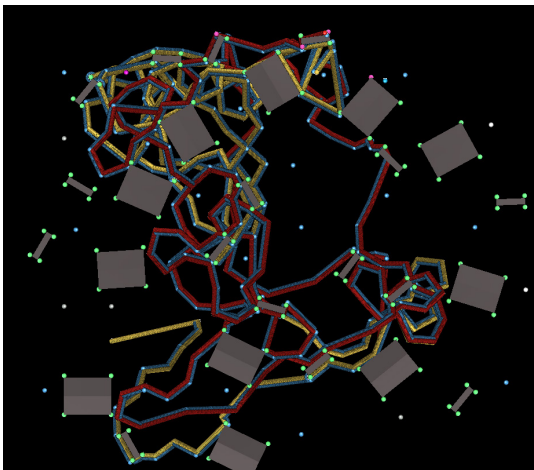


Figura 8. Rutas