

# DOCUMENTACIÓN

Clase: Inteligencia Artificial

Práctica: 1 [Contagio Zombi]

Alumno: Miguel Rodríguez Gallego

## Motivos de mi práctica

Quise tratar de hacer una buena y compleja IA de seguimiento mediante steering behaviours que detectase obstáculos y se adaptase, pero solo logré hacer lo básico para la fecha de entrega, aun así, he dejado en la carpeta de scripts en “\_UnusedAtTheEnd” los scripts que traté de usar para hacer mucho mejor todo, entre los que traté de usar el árbol de decisiones del docente, pero por x problemas lo terminé dejando de lado.

Entonces, al final es solo una práctica en la que sigue directamente cogiendo la posición del target que detecta mediante una esfera con x rango, añadiendo los objetivos que están catalogados por layers y tags, para así poder coger y medir dentro del array de colliders de targets de cada IA el que está más cerca y moverse hacia/en sentido contrario de él.

## Proceso de creación

Comencé por el movimiento de la cámara (extra) que se activa/desactiva al presionar espacio para poder moverte libremente por la escena y probar bien todo sin problema.

Tras esto hice la infección de un ciudadano mediante un raycast desde la cámara cambiando a un prefab.

Luego empecé con la lógica de perseguir, como dije buscando información de cómo hacerlo mediante steering behaviours, principalmente investigando en estas dos fuentes, pero no logré implementarla correctamente en ningún momento porque soy nulo:

<https://www.youtube.com/watch?v=tlfC00BE6z8>

[https://edirlei.com/aulas/game-ai-2020/GAME\\_AI\\_Lecture\\_07\\_Steering\\_Behaviours\\_2020.html](https://edirlei.com/aulas/game-ai-2020/GAME_AI_Lecture_07_Steering_Behaviours_2020.html)

Tras esto, empecé a hacerlo a mi manera, tal y como se me ocurrieron en mi cabeza...

Simplemente decidí crear un collider de una esfera en cada personaje (citizen y zombi) que detectase los colliders de los ciudadanos y de los zombis mediante layers. Estos los registran en un array de colliders, y tomo como target el más cercano para que lo persiga/escape de él la entidad.

## Dificultades y como se resolvieron

Del resultado final, lo que más me costó fue una tontería del acercamiento del zombie al target más cercano, que no lo cogía bien y no se movía bien hacia él, pero por el resto, no fue tampoco realmente complicado. Simplemente no normalizaba la dirección y no le restaba la del target al transform del player correctamente, una tontería de principiante que no me di cuenta vamos.

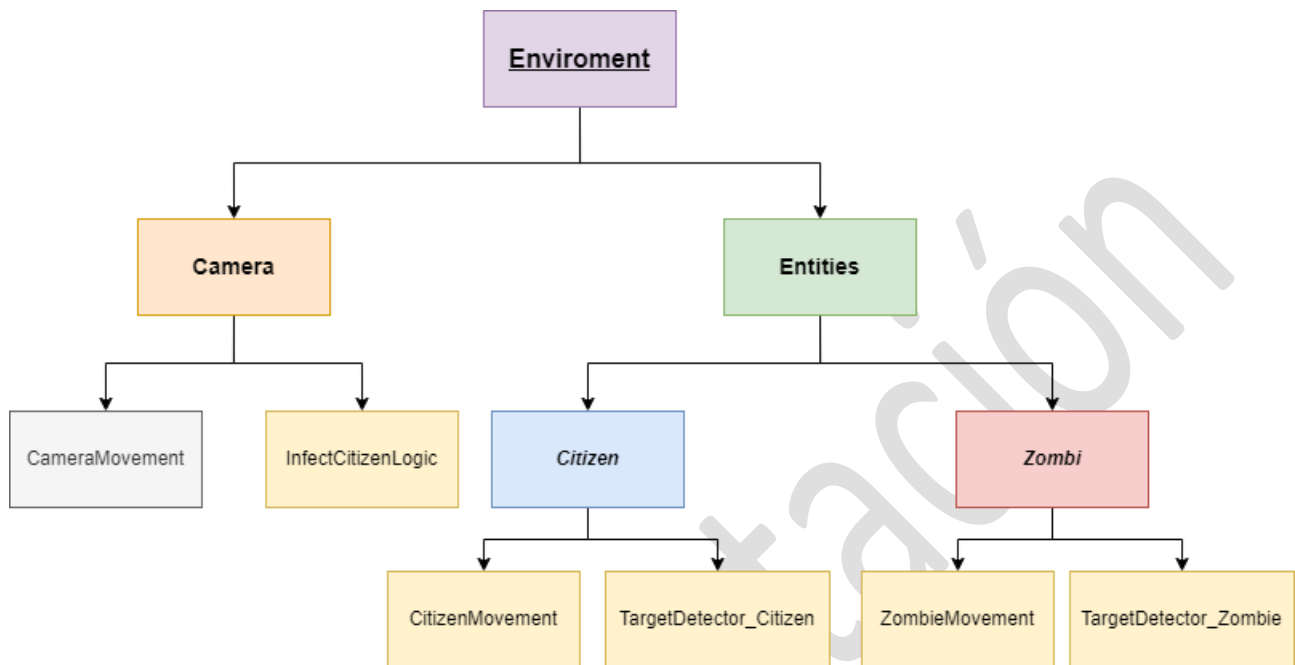
Luego para que escapase el ente sano, era lo mismo, pero al revés...

Del tema de los steering behaviours que no logré implementar correctamente en la práctica en ningún momento como ya dije.

Aparte, me fue un gran problema el hacer que entre los “citizens”, o sea, las entidades sanas, escapasen de la entidad más cercana corriendo del zombi. Debería ser fácil, pero no logré implementarlo porque usé layers para detectar a las entidades sanas, y se detectaba a si mismo, y no logré eliminar o que no se detectase a si mismo, así que lo terminé dejando fuera. Por el resto era fácil, simplemente era crear otro array de colliders pero cogiendo la

layer de las entidades sanas y mandando este target al movimiento, y activando el movimiento de escapar si esta entidad cercana identificada tenía una variable del estilo "isScared == true". Nada loco vamos.

## Diagrama de clases



- **CameraMovement:** Poder mover la cámara cuando presionas espacio y tmb pausarla con la misma tecla para que salga el cursor y puedas interactuar con el ambiente.
- **InfectCitizenLogic:** Infectar a un personaje sano al clickarlo mediante raycast.
- **CitizenMovement:** Movimiento del personaje sano, ya sea con target o en una rutina chill que imita el comportamiento tranquilo.
- **TargetDetector\_Citizen:** Detectar los targets/infectados más cercanos mediante un collider de una esfera. También calcula el target más cercano.
- **CitizenMovement:** Movimiento del Zombie, ya sea con target o en una rutina chill que imita el comportamiento tranquilo.
- **TargetDetector\_Citizen:** Detectar los targets/personajes sanos más cercanos mediante un collider de una esfera. También calcula el target más cercano.

## Autocrítica

El código que he hecho me parece muy malo, y se que se puede refactorizar y dejar mil veces mejor, por lo que no estoy demasiado orgulloso de esa parte. Sin embargo, he logrado hacer lo básico que me propuse y decía el enunciado, así que estoy contento con el resultado.