

Documento de diseño técnico de la arquitectura de IA



Paradox Studios:

Moltó Ferré, Enrique

Muñoz Periñán, José Luis

Pérez Cristo, Rubén

Rebollo Berná, Antonio

Zamora Pastor, Julio

Máquina de estados

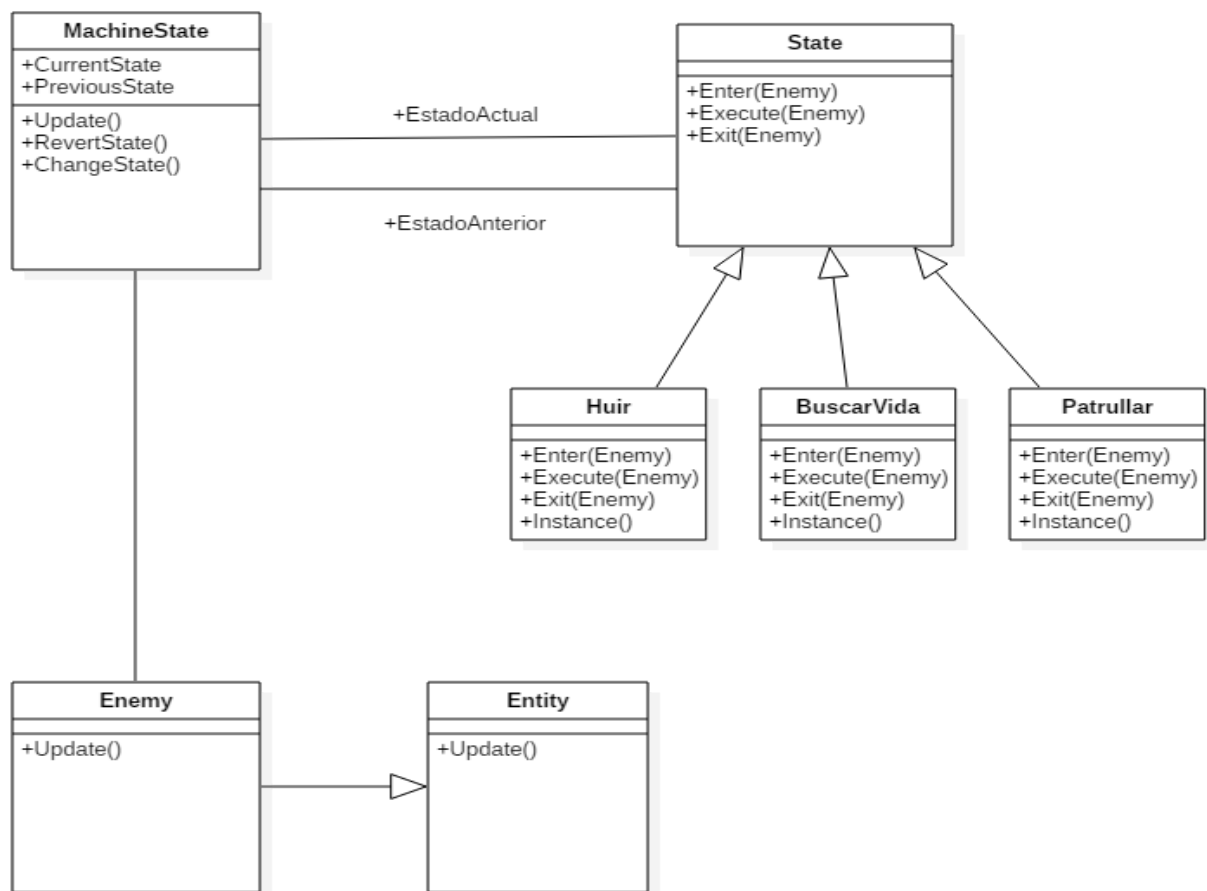
En la IA incorporaremos una máquina de estados de la IA, que la organizaremos de la siguiente forma:

Tendremos una clase *State*, de la que heredarán las clases estado, como *Patrullar*, *Huir* o *BuscarVida*. La clase *State* tendrá los métodos virtuales *Enter()*, *Execute()* y *Exit()*.

Un enemigo tendrá un estado actual *currentState*, de modo que en el *update()* del enemigo se hará *currentState()->Execute()*. Además, siempre que se vaya a cambiar de estado se ejecutará el método *Exit()* del estado del que se sale y el método *Enter()* del nuevo estado, si es necesario.

Toda la lógica se realizará en la clase *MachineState*, el enemigo tendrá un objeto *MachineState* que ejecutará los diferentes estados y los cambios entre los mismos.

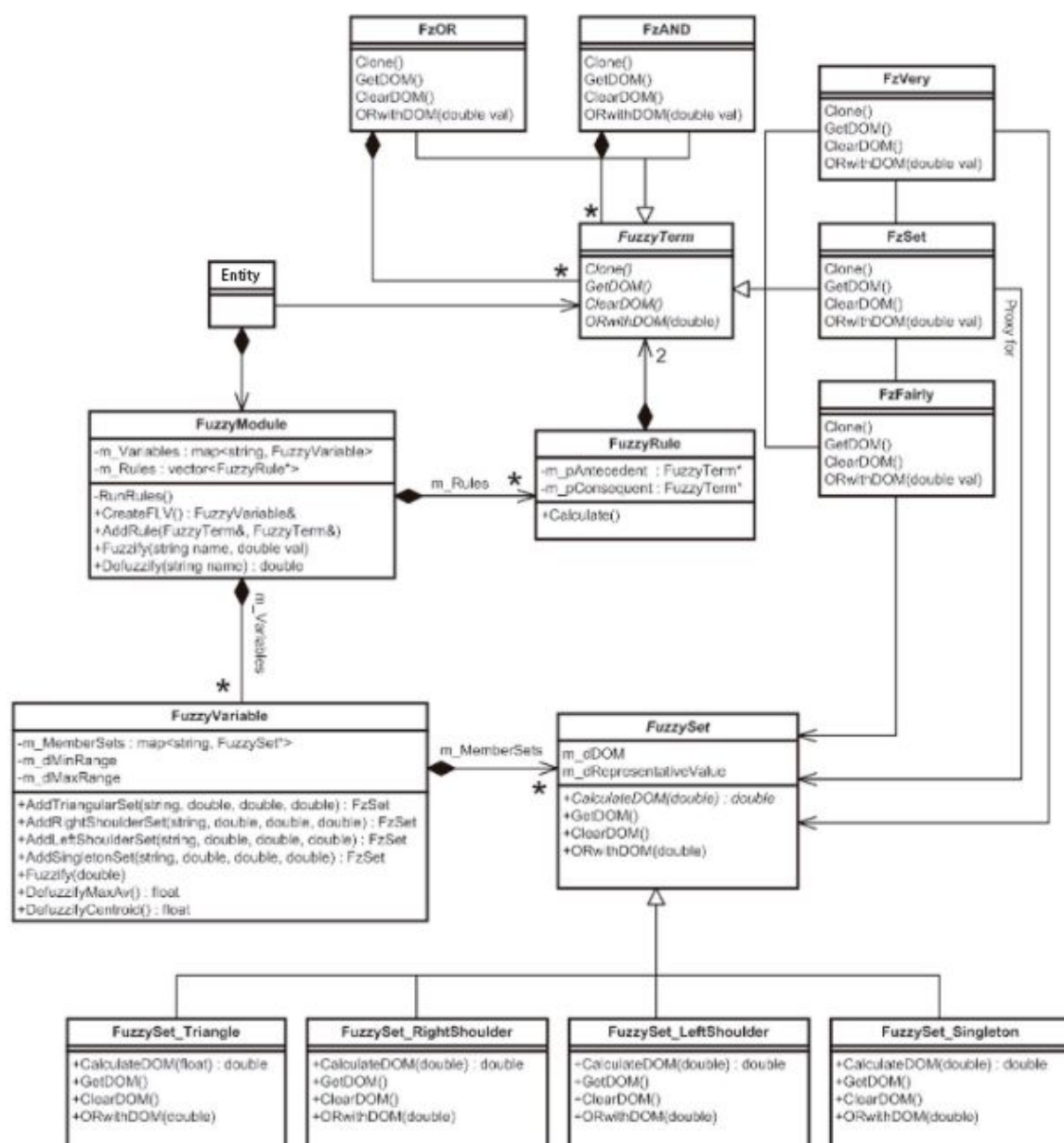
El diagrama de clases sería el siguiente:



En cada *Update()* de *Enemy* se llamará al *Update()* de su objeto *MachineState*, y cuando se quiera cambiar de estado se llamará al método *changeState()* pasándole el nuevo estado usando singleton.

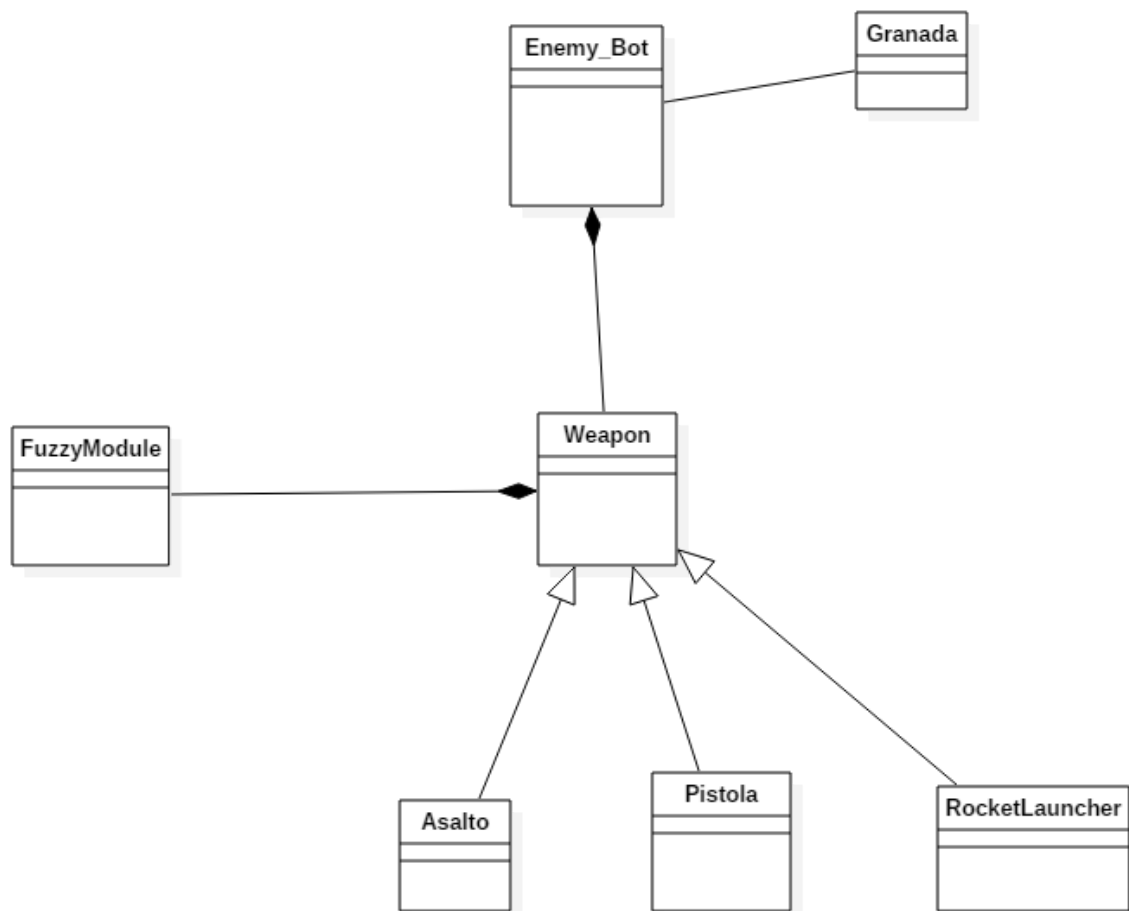
Módulo de Lógica Difusa

Utilizaremos lógica difusa diseñando algunas FLV's y sus reglas para determinar el comportamiento de la IA. El diagrama de clases será el siguiente.



Cada entidad del juego que desee utilizar un módulo de lógica difusa podrá utilizar **composición** y utilizar la interfaz del módulo para crear sus FLV's y reglas.

De esta forma el diagrama de clases de la IA quedaría de la siguiente forma si añadimos al enemigo bot una serie de clases de armas, cada una de las cuales tendría un objeto de tipo FuzzyModule:



De esta forma se utilizará Lógica Difusa para establecer que arma debe tener el bot enemigo.

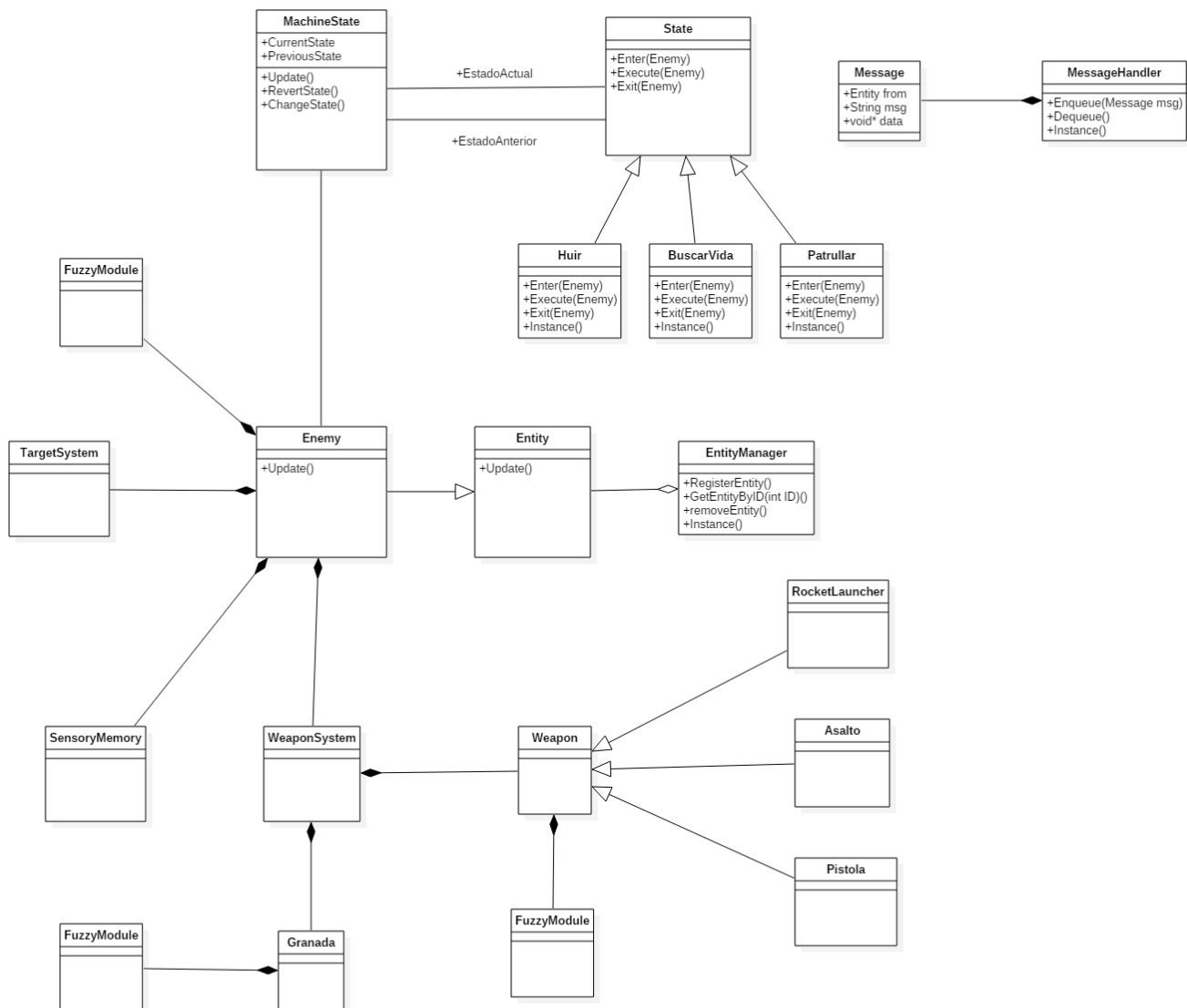
Pathfinding

Cada bot enemigo tendrá que tener acceso a un sistema de pathfinding que le permita navegar por el mapa.

Percepción

Los bots enemigos deben ser capaces de percibir su entorno. Por ejemplo si un objetivo entra en el campo visual del enemigo este debe ser capaz de percibirlo y almacenar este encuentro para su futuro tratamiento. Los bots enemigos también serán capaces de escuchar los disparos o los pasos de los objetivos. y obtener su posición.

Pero es importante que el bot conserve la memoria de todos estos datos que va percibiendo de su entorno, para ello se elaborará un sistema de memoria:



El bot enemigo tendrá un sistema de memoria que irá almacenando cuando ha visto a un enemigo, o cuando escuchó un sonido.

Además el bot enemigo dispondrá de un sistema de selección de objetivo. Este sistema se irá actualizando y hará uso del sistema de memoria para saber si ha visto o ha escuchado

algún objetivo. Si por ejemplo ha escuchado a más de un objetivo seleccionará al más cercano. También indicará si el objetivo está a tiro y no está detrás de ningún objeto del mapa.

Por último, habrá un sistema de manejo de armas que se encargará de controlar el arma activa, así como de controlar la precisión de disparo del bot y del tiempo de reacción.