

PRÁCTICA 02: AGENTE PI

Alumnos: Pablo Padial Iniesta y David Ferreras Díaz

1. Descripción detallada del entorno usando REAS

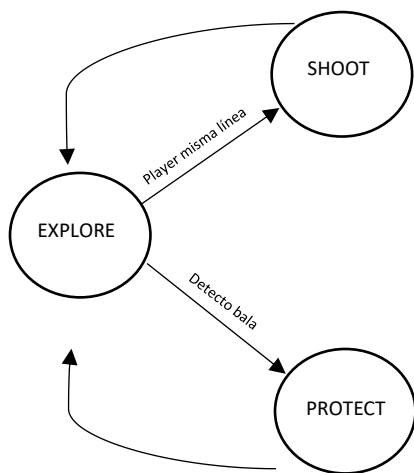
- **Rendimiento:** El agente siempre cumple con su misión principal, destruir el Command Center. Tiende a hacer pocos movimientos innecesarios debido a como está desarrollado el algoritmo. Es capaz de matar al jugador con rapidez si está cerca o lo percibe además de ser incapaz de que el jugador lo pueda alejar del Command Center atrayéndolo por el mapa.
- **Entorno:** El entorno está compuesto por distintos tipos de elementos:
 - **Player** (tanque verde): Es el tanque que controla el jugador y puede moverse libremente, su misión es destruir el resto de tanques antes de que destruyan la Command Center o a él. Tiene 1 vida.
 - **Nuestro agente** (tanque rojo): Sobre el que hemos desarrollado toda la práctica, su misión es ir a destruir la Command Center si el player se interpone en su camino atacarle. Se mueve con “inteligencia”. Tiene 2 vidas.
 - **Tanque random** (tanque azul): Es el otro tanque enemigo y en este caso únicamente tiene un movimiento aleatorio y dispara todo el rato. Tiene 2 vidas.
 - **Command Center:** La misión del player es protegerla del ataque de los tanques, siempre está en la misma posición y no se puede mover.
 - **Balas:** Todos los tanques pueden disparar una bala, mientras esta bala siga existiendo no puede disparar otra. Con la bala puede quitar una vida al tanque que le impacte. Puede destruir los bloques destructibles.
 - **Bloques destructibles:** Se encuentran por el tablero impidiendo el avance de los tanques. Con un disparo se destruyen.
 - **Bloques indestructibles:** Estos bloques a diferencia de los anteriores no se pueden destruir de ninguna manera.
- **Acciones:** El agente modifica el entorno con cada acción ya sea únicamente su movimiento, disparando balas a los bloques para destruirlos y modificar el entorno o disparar al player para tratar de destruirlo.
- **Sensores:** El agente percibe el entorno a través del vector `perception[]` que contiene toda la información de los objetos que le rodean, es decir, es sabe dónde está la Command Center en todo momento aunque no sabe cómo es el mapa.

2. Qué tipo de agente es:

Se trata de un **Agente reactivo basado en modelo** ya que toma decisiones en cada ciclo (Update) en función de las percepciones actuales como la posición, presencia de obstáculos, enemigos, etc. Está implementado mediante una máquina de estados donde cada uno de sus estados refleja el conocimiento parcial de la situación. La meta del agente es siempre la misma, ganar yendo a la Comand Center o si tiene posibilidad, destruyendo al Player.

3. Diseño preliminar del agente:

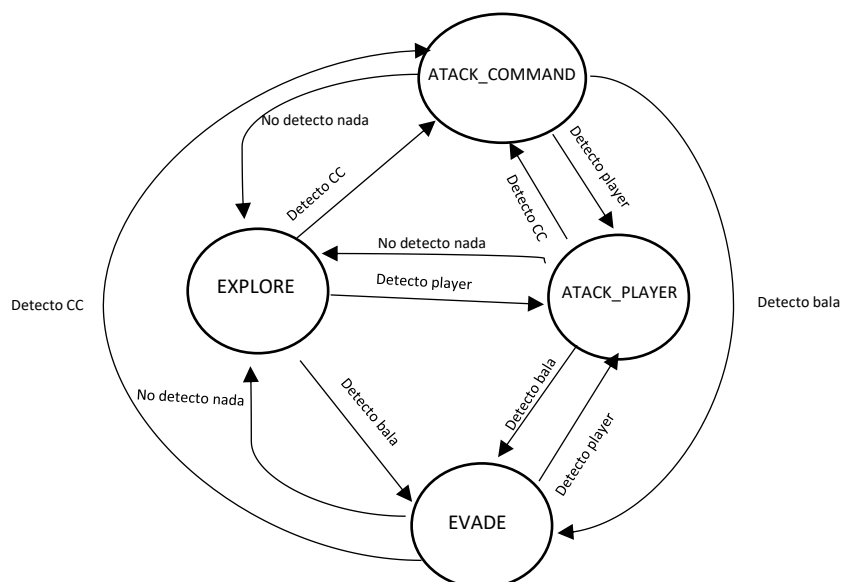
Inicialmente teníamos pensado un esquema más simple, pero a medida que avanzábamos en el desarrollo de la practica nos dábamos cuenta que teníamos que contemplar más puntos.



- EXPLORE: Busca la forma más rápida de llegar a la Comand Center.
- SHOOT: Si el player se encuentra en la misma X o Y entonces el agente gira hacia él y dispara
- PROTECT: Si el agente detecta una bala en su misma X o Y entonces esquiva o dispara una bala para destruirla.

4. Diseño final del agente:

Finalmente, vimos conveniente hacer algún cambio en el desarrollo de la máquina de estados y añadimos uno extra.



- EXPLORE: Iniciamos siempre en el estado explore que buscará la ruta más rápida hasta la Command Center (CC) usando los algoritmos de A* y Manhattan. Si detecta al player en la percepción pasara a estado ATTACK_PLAYER, si detecta la CC pasará a ATTACK_COMMAND y si detecta una bala pasara al estado EVADE.
- EVADE: Se encargará de esquivar y disparar cuando en la percepción detecta una bala. Si tras esto no detecta nada pasara al estado EXPLORE de nuevo y continuará la ruta hacia la CC, si detecta al player pasara al estado ATTACK_PLAYER y si detecta la CC al ATTACK_COMMAND.
- ATTACK_COMMAND: Se encargará de disparar cuando detecte la CC destruyendo si es necesario los bloques que haya. Si detecta una bala pasará al estado EVADE, si detecta al player pasa a ATTACK_PLAYER y si en algún momento deja de detectar la CC vuelve al estado EXPLORE.
- ATTACK_PLAYER: En este estado se encargará de atacar al player cuando la percepción lo detecte. Además, si detecta una bala pasará al estado EVADE para tratar de evitarla, si detecta la CC irá al estado ATTACK_COMMAND para tratar de destruirla y si deja de detectar algún objeto (player, CC o bala) volverá al estado EXPLORE.