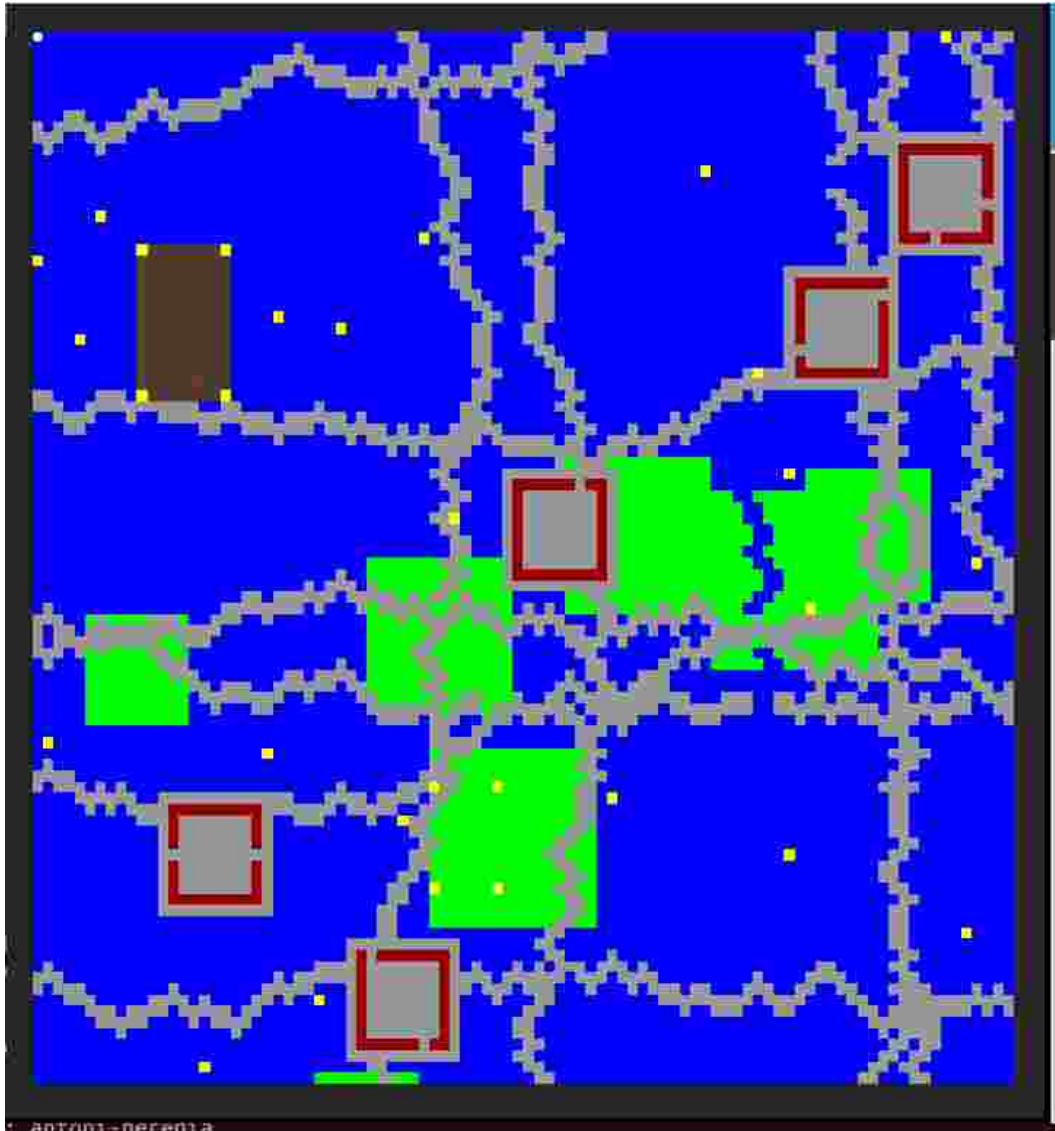


# Inteligencia Artificial

## Practica 2



Los mundos de Belkan

## Nivel 1

En el nivel 1 implemento un algoritmo de búsqueda en anchura. El algoritmo de anchura busca todas las posibilidades que se puede dar. Siempre nos dará en el nivel 1 el mejor camino.

Básicamente el algoritmo mira una a una si la casilla que esta es el objetivo y la marca como visitada. Para cada casilla guarda la lista de movimientos que tiene que hacer en una pila. Al final, cuando se encuentra con el objetivo en plan guarda la lista de movimientos que tiene al frente de la cola. En caso de no encontrar ningún plan, se quedaría parado el agente.

La mayoría de errores que me ha dado en este nivel se debían a que me pasaba de los bordes del mapa en la búsqueda y por lo tanto me salía de la matriz.

## Nivel 2

En el nivel 2 lo que realiza el agente, es que si tiene el plan preparado pero a la hora de realizar una acción de moverse hacia delante tiene un aldeano delante, lo detecta y vuelve a buscar un plan. El plan esta vez a la hora de buscar el camino, tiene en cuenta que delante hay un aldeano, para ello, lo primero que hago es marcar como visitada la casilla de enfrente del agente. En caso de que el aldeano, tape el único camino que haya para llegar al punto, el agente se queda parado esperando que se aparte de en medio y en ese momento sigue el plan.

Aquí tuve mas problemas, ya que a la hora de re calcular el plan cuando tenia un aldeano delante, no me di cuenta de que no lo tenia previsto en el pathfind y por tanto me seguía dando el mismo plan, pero al estar el aldeano no se podía mover.

## Nivel 3

Para el nivel 3, tengo una busqueda al azar del punto PK. Una vez que encuentra el punto PK genera un plan, para ello permite calcular el plan atraves de las ?. Cuando en un plan generado con las interrogaciones, tiene en delante algo por donde no se puede pasar, regenera el plan. Asi poco a poco va llegando al objetivo.

Para rellenar el mapaResultado tengo una especie de algoritmo que se basa en dos for, uno va hasta 4, que es la lejanía con la que ve el bot y dentro de ese for hay otro que va desde - el indice del for padre hasta + el indice del for padre. Para recorrer así cada una de las columnas de la visión del agente. El mapa resultado se va rellenando según la orientación que tenga el bot.

Los mayores problemas de este nivel es que no he sido capaz de generar un algoritmo eficiente para que encuentre el punto PK. Entonces puede no encontrarlo.