INTEL·LIGÈNCIA ARTIFICIAL

PRÀCTICA 1

30 DE SETEMBRE DE 2023

PAU SEGURA BAÑOS
DAVID VILAJOSANA GARRIGA

Introducció:

Com a introducció al primer treball de l'assignatura intel·ligència artificial volem dir que tenim moltes ganes d'aprendre sobre aquest tema que ens sembla tan interessant. En aquesta primera pràctica haurem d'aplicar un algorisme A* el qual hem vist a la classe de teoria i ens ha semblat molt interessant la forma que té de calcular els costos mitjançant l'heurística. Nosaltres en aquesta pràctica no l'haurem de programar, però, així i tot, intentarem entendre com funciona.

Algorisme A*:

Com ja ens van explicar a la classe de teoria, sabem que l'algorisme A* és un algorisme de cerca. L'objectiu d'aquest algorisme és trobar el camí òptim a un estat final, en aquest cas, el *checkmate.*
L'algorisme compta amb una llista anomenada *frontera* i un diccionari *dictPath.* A la llista frontera s'emmagatzemaran en forma de tupla, cada un dels possibles moviments de les peces a partir d'un estat, juntament amb la seva heurística, és a dir la seva distància al node final i a *dictPath* emmagatzemem una tupla que conté informació sobre el pare del node i el cost del pare
L'algorisme comença amb el node inicial amb el qual s'inicialitza el taulell. Aquest s'afegeix a la frontera i a *dictPath.* Com és el primer node, no té pare.
Entrem en la iteració principal. Aquest algorisme s'executarà mentre hi hagi contingut a la llista *frontera.* En cada iteració, obtenim l'element amb valor mínim d'heurística de la *frontera,* què és la clau de l'algorisme A*, l'afegim a una llista d'estats visitats i l'eliminem de la frontera. Simulem el moviment de les peces per reproduir l'estat i actualitzar el taulell i mirem si és *checkMate*, ja que és una condició de parada de l'algorisme. En cas de ser escac i mat, hem d'imprimir el camí amb els moviments realitzats per arribar a aquesta situació.
Mitjançant un bucle *for* iterem sobre els possibles estats a partir de l'estat actual i si aquest no ha estat visitat o si sí que ha estat visitat, però el cost fins a aconseguir l'estat és menor que l'anterior guardat l'afegim a la frontera i actualitzem el diccionari *dictPath*.

A la pràctica també se'ns demana que si canviem la posició del rei a la [7,7] com afecta a la resolució de l'algorisme i si hem hagut de canviar alguna part del codi perquè ara funcionés correctament. En el nostre cas ha funcionat correctament un cop canviada la posició i no hem hagut de canviar res del codi.

També volem mencionar que al principi ens va semblar curiosa la forma que té d'avançar, ja que primer va el rei en diagonal a l'esquerra i després en diagonal cap a la dreta, quan podria fer el camí recte. Hem arribat a la conclusió que tots els camins tenen el mateix cost i aquest és el primer que troba perquè a l'hora de buscar estats veïns comença pels situats a l'esquerra.

Conclusió:

Gràcies a aquesta pràctica hem pogut veure l'ús de l'algorisme A* en aplicacions pràctiques. Ens ha semblat molt útil, ja que creiem que començar per algorismes així ens pot ajudar molt a entendre el món de la intel·ligència artificial, un món pel qual tenim molt entusiasme.

Volem destacar les dues parts que ens hem encallat més realitzant la pràctica:

- La primera ha sigut la d'entendre com funcionava el diccionari dictPath que emmagatzema l'estat pare i el cost fins a arribar a aquest, ja que sense aquest ens estava sent impossible poder veure quin era el camí més eficient perquè no podíem comparar el cost en dos estats iguals, però que hi havíem arribat de diferent forma.
- La segona, en canvi, és sobre la forma d'emmagatzemar els estats, ja que teníem un problema perquè a vegades es guardava primer una figura i a vegades primer una altra. Ho vam solucionar ficant un sorted() a cada un dels estats per així tenir-los sempre igual ordenats.