
Intel·ligència Artificial - Pràctica 2

Tutor de la pràctica : Llorenç Burgas

Marc Sànchez Pifarré, GEINF (UDG-EPS)

2019-04-03

Introducció

En aquest informe destacaré els mètodes més rellevants així com els patrons i el plantejament per a la solució del problema.

Funcionament global

Petita adaptació.

Referent a l'estat del codi inicial s'hi ha fet una petita adaptació. S'ha generat una herència per part de tots els elements que intervenen al tauler d'una classe Position.

Aquesta herència s'ha realitzat per aconseguir una generització a l'hora de fer el tractament de les distàncies. Tots els elements comparteixen els mètodes de càlcul euclidià.

Operacions dels Robots ()

Cada un dels elements mòbils, Robot, pot moure's en 8 sentits. NO, N, NE, E, SE, S, SO. Existeix una sola operació moure que en funció d'un destí es desplaça cap aquest destí 1 casella tant en horitzontal, com en diagonal, com en vertical. **Es preveuen les colisions** dels robots desviant la trajectòria dels mateixos. Definim crash com la casella on hi van a parar 2 drons de cop.

```
1  /**
2   * En funció d'una serie de posicions conflictives intentem desviar el
3   * dron.
4   */
5   public void setNextPosition(List<Position> crashes){
6       SortedSet<Voyage> desviaments = new TreeSet<Voyage>();
7       desviaments.add(new Voyage(new myRobot(from.x, from.y+1, this.from.
8           escena), target));
9       desviaments.add(new Voyage(new myRobot(from.x, from.y-1, this.from.
10          escena), target));
11      desviaments.add(new Voyage(new myRobot(from.x-1, from.y, this.from.
12          escena), target));
13      desviaments.add(new Voyage(new myRobot(from.x+1, from.y, this.from.
14          escena), target));
15      desviaments.add(new Voyage(new myRobot(from.x+1, from.y+1, this.
16          from.escena), target));
17      desviaments.add(new Voyage(new myRobot(from.x+1, from.y-1, this.
18          from.escena), target));
```

```
13     desviaments.add(new Voyage(new myRobot(from.x-1, from.y+1, this.  
14         from.escena), target));  
15     desviaments.add(new Voyage(new myRobot(from.x-1, from.y-1, this.  
16         from.escena), target));  
17     Voyage desv = desviaments.first();  
18     while(!desviaments.isEmpty() && crashes.contains(desv.from)){  
19         desviaments.remove(desv);  
20         desv = desviaments.first();  
21     }  
22     nextX = desv.from.x;  
23     nextY = desv.from.y;  
24 }
```

El que es fa és generar tots els possibles moviments del dron i seleccionar-ne un que no entri en conflicte amb cap crash.

Viatge

Definim viatge que la trajectòria que realitza un Robot des del punt en el que està fins al seu target. És una classe que conté 4 camps, el robot que es mou, l'objectiu a assolir, la següent posició x i la següent posició y.

Amb aquesta classe es domina el que seria el tractament dels possibles destins als que pot arribar un Robot. A l'hora d'ordenar els viatges s'utilitza una llista i el mètode Collections.sort() que crida al mètode compareTo del Voyage. Dins del compareTo s'implementa el càlcul de punts que estipula lo bo que és aquest viatge.

Càlcul de la puntuació.

Tant per el mètode cooperatiu com el competitiu s'estipula la mateixa puntuació per definir l'objectiu a apagar. Per realitzar el càlcul es mira tots els focs i s'ordenen per una puntuació.

```
1  
2 @Override  
3 public int compareTo(Voyage o) {  
4     return (calcul(this)).compareTo(calcul(o));  
5 }  
6  
7 /**  
8  * Es separa en un mètode per poder-lo cridar des d'altres punts.  
9  */
```

```
10 private Double calcul(Voyage o){
11     Double dist = o.target.getEuclidian(o.from);
12     Double arbres = from.escena.nombreArbresVoltant(o.target.x, o.
        target.y);
13     if(arbres == 0){
14         dist += 13;
15     }
16     else if(arbres > 0 && arbres < 3){
17         dist += 5;
18     }
19     else if(arbres >= 3 && arbres < 6){
20         dist += 2;
21     }
22     // Prioritzo per sobre de tot els que tenen més de 5 arbres
23     return dist;
24 }
```

La distància euclidiana del viatge es desa a la variable dist mentre que la variable arbres conté el nombre d'arbres que poden ser cremats per el foc target. El càlcul de la puntuació està pensat de la següent manera :

Seguin les següents premises.

- El Robot realitza el càlcul a cada pas que dóna i avança de 1 en 1 per tant a mesura que vagi avançant el càlcul serà més ajustat.
- Si la distància és molt llarga en relació a la puntuació assolida per el nombre d'arbres ens interessa que primi la distància que el nombre d'arbres.

Al final la suma d'un valor extret del nombre d'arbres i la distància euclidiana serà el que permetrà al Robot decidir el seu destí. Inicialment S'utilitzava la série de fibonacci, número per número, és a dir, dist + fibonacci(arbres). La precisió aportava desviació i s'ha acabat ajustant a utilitzar la proporció de fibonacci però saltant-se els parells i l'1.

- 0 arbres = fibonacci(7) = 13
- 1 o 2 arbres = fibonacci(5) = 5
- 3, 4 o 5 arbres = fibonacci(3) = 2
- 6 o més arbres = fibonacci(0) = 0

Llavors a mesura que es vagi acostant al foc anirà descartant els focs que no son perillosos i primarà anar a apagar els focs que estan amb més risc de propagació.

S'utilitza la série de fibonacci per que el nombre més alt de la série també serà el que tindrà la major distància amb el seu antecessor, també la vaig utilitzar a la pràctica del quarto.

Funcionament Cooperatiu

Per establir el protocol de comunicació el que s'ha fet ha sigut dotar els drons d'una "bústia de viatges". Cada dron implementa les interfícies Subject i Observer que li permeten enviar i rebre els viatges que han programat els drons que té al voltant. Per fer-ho a cada pas que han de donar es subscriuen els drons i es dessubscriuen al final del torn.

La idea és subscriu, notifico, dessubscriu.

Cada cop que un Robot decideix viatjar cap a un target ho comunica als demés. Per tant quan un dron comença a calcular ja sap on aniran els altres drons que ja han pres una decisió, d'aquesta manera pot decidir si anar cap al mateix lloc on van o no i realitzar el càlcul també entrant els destins i els moviments dels altres a l'equació (crashes).

Funcionament competitiu

Es deshabilita el pas de missatges. S'utilitza la mateixa estratègia per definir els destins sense tenir en compte la decisió dels altres ja que l'objectiu dels drons ha de ser apagar tots els focs intentant apagar més que els demés però que es prioritzin les propagacions és l'objectiu principal i ha d'estar juntament amb el càlcul de les distàncies.

Guinyo a en Jordi Regincós.

S'ha definit un patró strategy per realitzar el canvi de comportament entre cooperatiu i competitiu, s'ha afegit un botó a sota de tot de la botonera que permet en temps d'execució canviar el comportament. Per defecte sempre comencen en cooperatiu, un click canvia a !cooperatiu és a dir a competitiu i si es torna a clicar canvien a cooperatiu, i així successivament.