Pràctica d’Agents

Assignatura: Intel·ligència Artificial (Grup 1)

Professor: Dr. Ramon Mas

Integrants:

Alejandro Rafael Bordón Duarte

Joan Sansó Pericás

Joan Vilella Candia

Contenido

[1. Comportament general del nostre agent 3](#_Toc86933150)

[2. Particularitats de comportament 3](#_Toc86933151)

[2.1. Moviment 3](#_Toc86933152)

[2.2. Anar a per recursos nostres 3](#_Toc86933153)

[2.3. Disparar recursos enemics 4](#_Toc86933154)

[2.4. Disparar enemics 4](#_Toc86933155)

[2.5. Gestionar dispar enemic 4](#_Toc86933156)

# Comportament general del nostre agent

La principal estratègia que sigui el nostre agent és la recol·lecció dels propis recursos i la denegació d'aquests als enemics. Hem considerat que un comportament més enfocat a la recol·lecció de recursos és la més aconsellable, ja que en disparar amb precisió a enemics és una acció força complicada i amb molt de risc associat. Abans de comentar els aspectes del comportament, comentarem l'elecció d'atributs inicials.

////

En primer lloc, comentarem els aspectes generals de com va el nostre agent. La nostra màxima prioritat és que el Bestio mai estigui en col·lisió, o almenys el menor temps possible. És per això que està situada al primer condicional a l'hora de caminar. En cas de col·lisió generem una sèrie de moviments aleatoris per sortir com més aviat millor d'aquesta situació. Tot això ja que considerem la col·lisió com l'un dels estats més crítics (sinó el que més), ja que quedem a la mercè dels nostres rivals.

Després el nostre “Bicho” camina de manera diferent atenent la distància que està de les parets. Si la distància és prou llunyana, podríem dir que fa camí d'una manera més preventiva sense fer canvis bruscs. Però en canvi quan hi ha una paret busca fa canvis més bruscos per esquivar-la com més aviat millor.

Finalment, el nostre agent fa girs de reconeixement cada cert temps per intentar evitar les situacions on no ve cap mena de recurs durant un temps considerable.

La manera com mirem els recursos del nostre entorn està completament relacionada amb la nostra estratègia, de fet, és la funció que més influeix de cara al nostre plantejament de la partida.

La nostra prioritat és sempre agafar els nostres recursos, a més per a ahorrar més energia triarem sempre per ordre de distància (té més prioritat el més proper). Cal destacar que no hem considerat els escuts tan importants com els recursos. Ja que, degut a la durada de la partida, amb una quantitat relativament baixa, considerem que són suficients.

És per això que moltes vegades es podrà observar el nostre agent ignorar escuts que estén més a prop nostre per anar a buscar recursos més llunyans. Ara bé, si tenim pocs escuts (en el nostre cas menys que 5) la prioritat serà la mateixa que amb els recursos (la més propera fam). També cal destacar que, encara que tinguem més de 5 escuts, si al nostre camp de visió no hi ha cap recurs amb més prioritat, anirem a buscar-lo.

# Particularitats de comportament

## Moviment

El mètode camina s'encarrega del desplaçament del nostre agent a través de l'entorn:

En primer lloc, comprovem si estem en col·lisió, en cas positiu incrementem un comptador i cridem al mètode *gestionaColisio()* on s'encarrega de gestionar les accions que realitza l'agent mentre l’agent està en col·lisió, primer verifiquem el comptador de col·lisió, és a dir, si el nostre agent dur aproximadament dos segons col·lisionant activem l'hiperespai, sinó intentem evitar la col·lisió, realitzant un gir i un moviment aleatori.

En cas que no es troben en col·lisió, resetegem el comptador a cero i en funció de la distància que estiguem a les parets el nostre agent començarà a girar abans o després, per exemple si els nostres visor veu una paret a una distància propera (35 px), aleshores cridem al mètode *giraAProp()* que s'encarrega de girar uns graus aleatoris entre [10º - 40º], però si el visor central veu una paret a una distancia menor a (70 px) aleshores fem un gir aleatori dins el rang, sinó vol dir que te una paret lluny i aleshores fem un gir de 10º i això ho fem en funció a la distància que es troben els visors de l'esquerra o dreta, ja que si el visor de l'esquerra és més gran a la dreta girem cap a l'esquerra en cas contrari cap a la dreta. Si l’agent veu una paret lluny (85 px) i amb el visor central veu una paret a una distancia inferior (90 px), aleshores cridem al mètode *giraLluny()* que s'encarrega de girar en funció a la distància que es troben els visors de l'esquerra o dreta, es a dir, si el visor de l'esquerra és més gran a la dreta girem a l'esquerra en cas contrari girem a la dreta, i aquest gir ho fem a una velocitat angular que hem definit en el principi de la partida. Si passa un temps en què l'agent no veu cap recurs llavors se crida al mètode *girRecon()* que s'encarrega de fer un gir de 120º a l'esquerra o dreta per tornar a avaluar l'entorn i sinó verifica cap d’aquestes condicions el nostre avança.

Per últim, en cada iteració gestionem les accions que realitza l’agent sobre un agent enemic, ja que se crida al mètode *gestionarEnemic()*

## Anar a per recursos nostres

Per poder anar a per els nostres recursos, primer hem de avaluar que podem cercar algun recurs i que el nostre agent no ha vist cap enemic, si compleix aquesta condició es crida el mètode *mirarRecurs()* que s'encarrega de cercar entre tots els objectes del entorn els nostres recursos de manera prioritària, si hem trobat un recurs nostre el mirem i anem a per ell. Si veiem algun escut i en tenim menys de cinc o directament no veiem cap recurs a prop cerquem els escuts i anem a per ells.

Si no veiem res durant un temps anem decrementant el comptador de gira per així en el mètode *camina()* s’executi el mètode *girRecon()* per fer un gir aleatori i tornar a avaluar l'entorn.

## Disparar recursos enemics

Un de les estratègia que vàrem decidir fer es també anar a per els recursos enemics i dispara-les, de manera que a part de avaluar si disparar a un enemic o no, també avaluem si podem disparar al seu recurs, per tant, per això si veiem un recurs enemic se crida al mètode *evaluarDisparMenjar()* que s’encarrega en primer lloc detectar si hem llançat una bala i si hem vist algun recurs si es aquest cas recorrem tots el objectes fins trobar el recurs enemic que el nostre agent ha vist i per això se ha de complir la condició de que el objecte no es un agent, no es un recurs nostre, es troba a algun dels sectors on el nostre agent el veu i per últim es troba a una distancia recomanable per fer el llançament, una vegada se hagi complit sa condició retornem el objecte en concret i el disparem.

## Disparar enemics

El primer objectiu del nostre agent es avaluar si veu un enemic relativament a prop i disparar-lo, però cada vegada que disparar hem decidit esperar dos segons per tornar a disparar al enemic, ja que si esta molt a prop a un enemic llança molt de bales i a lo millor el enemic te activat el escut, per tant estirem desaprofitant les bales per això hem decidit posar un temps entre llançament de bales a un enemic. Per poder determinar si el nostre agent a vist a un enemic, es crida al mètode *evaluarDisparEnemic()* que s’encarrega de detectar si hem llançat una bala i si hem vist a algun enemic o algun recurs, si es aquest cas recorrem tots el objectes fins trobar al objecte enemic i per això se ha de complir la condició de que el objecte es un agent i es troba a algun dels sectors on el nostre agent el veu i la distancia del enemic es inferior o igual a (120 px) i a més a més la distancia del objecte es menor a la distancia màxima per disparar, per tant si això es compleix retornem el objecte enemic i el disparem.

## Gestionar dispar enemic

En el método de camina, sempre avaluem si un enemic ha disparat, per això se crida al método *gestionarEnemic()* que s'encarrega de detectar si un dels nostres sectors observa un llançament enemic i si no tenim l'escut activat un cop s'hagi complert aquesta condició verifiquem si el llançament se troba a prop nostre (40 px) si és així activem l'escut. Despres, detectem si l'enemic ens ha impacatat un llançament i noltros no l’hem vist, per tant si hen a donat i no tenim activat el escut, l’activam i tornam setear el nombre d’impactes rebuts.