

ROBOCODE

Version 1 de Chartiedor

par Dorian, Thierry et Charly

Stratégie recherchée pour la version 1 de notre robot

- Récupérer nos coordonnées X, Y au départ d'un round pour se sortir tout de suite de zone jugée éventuellement dangereuse : éviter de rester dans la zone centrale au milieu du champ de tir, éviter d'être coller au mur où l'on a moins de liberté de mouvements.
- Pour bien se défendre, être le plus souvent possible en déplacement
- Pour attaquer efficacement, ne pas gaspiller trop d'énergie en ratant les tirs, on évite de tirer de trop loin et l'on préfère se rapprocher du robot scanné
- Dès que l'on est placé à bonne distance, on oriente le canon vers le robot adverse et on tire tout en se déplaçant de manière à l'encercler.
- On inverse le sens de déplacement, lorsque notre robot est bloqué par un mur ou un autre robot, c'est à dire lorsque notre vitesse est égale à 0.
- Lorsque l'on percute un robot, on oriente le scan vers lui et on le prend comme cible.

Principes de bases utilisés (expliqués précisément dans le rapport)

- Alignement du radar vers le robot adverse de manière à ne pas le perdre de vue.

```
double angleAlignementRadar =  
getHeadingRadians() + e.getBearingRadians() -  
getRadarHeadingRadians();  
setTurnRadarRightRadians (Utils.normalRelativeAngle  
(angleAlignementRadar));
```

- Se rapprocher jusqu'à 100 pixels du robot scanné s'il est à plus de 200 pixels

```
if (e.getDistance() > 200) ; setAhead(e.getDistance() - 100);
```

- Orienter le canon en direction du robot adverse

```
double angleAlignementCanon =  
normalRelativeAngleDegrees(e.getBearing() + (getHeading() -  
getRadarHeading()));  
setTurnGunRight(angleAlignementCanon);
```

- Orienter le châssis vers la cible adverse

```
setTurnRight(e.getBearing()) ;
```

- Encercler le robot adverse

```
setTurnRight(e.getBearing() + 90);  
setAhead(1000 * inverserDirection);
```

- Bloqué, inverser le sens de déplacement

```
if (getVelocity() == 0)  
inverserDirection *= -1;
```

- correction de l'angle de visée en tir rapproché en fonction de la direction et de la vitesse du robot adverse

```
double angleAbsoluAdverse = getHeadingRadians() +  
e.getBearingRadians();  
setTurnGunRightRadians(Utils.normalRelativeAngle(angleAbsoluA  
dverse - getGunHeadingRadians() + (e.getVelocity() *  
Math.sin(e.getHeadingRadians() -angleAbsoluAdverse) / 11)));
```

Analyse des statistiques des combats

Les tableaux statistiques des résultats de cette première série de « battles » avec nos premières versions de robots montrent que notre char robotisé Chartiedor1 récoltent les meilleurs résultats sur les champs de batailles les plus petits (environ < à 1000 sur 1000) y compris avec d'autres robots proposés par l'application. Par contre dès que le champ de bataille s'agrandit, notre char est trop statique et la portée du radar insuffisante. Il peut être longtemps sans rencontrer d'adversaire et cela le pénalise énormément. Lorsqu'il est le meilleur, son score se répartit de façon équilibré entre les différentes catégories de gains (points survie, points obus, points collisions...)

Vérification de notre code

Contrairement à ce que nous voulions mettre en place pour cette version, et en s'apercevant que notre char tirait au départ d'un scan un obus n'importe où, nous avons corrigé notre oubli en rajoutant les méthodes qui désolidarisent le radar du canon et le canon du châssis.

Analyse des robots adverses

	omega	kfc	groupe1
généralités	AdvancedRobot il tire beaucoup et se protège avec les murs.	Robot se déplace sans tenir compte des autres, tourne assez souvent son canon.	AdvancedRobot se déplace de manière saccadée en tournant beaucoup son canon. Ne cherche pas à se rapprocher des autres
Déplacement initial	Se place en contact avec le mur, et tourne autour du champ de bataille vers la droite	Avance s'il n'y a pas d'événements, et tourne son canon sur 360° dès un événement. Ou avance, tourne à gauche, tourne son canon sur 360°, recule et tourne encore son canon sur 360°	Tourne à droite d'un peu plus de 45° et tourne son canon sur 360° puis se déplace...
Sur robot scanné	Orienté son canon sur le robot scanné et tire plus ou moins fort	S'il est près tire une fois à 3, s'il est moyennement distant, tire 3 fois à 2 s'il est loin tire à 1	Tire de plus en plus fort s'il a beaucoup d'énergie
Touché par un obus	À partir d'un certain nombre de touches, essaie d'éviter en faisant un zig-zag en continuant d'avancer	Recule et tourne à gauche	
Touche un mur	Tourne 90° à droite	Tourne pour éviter le mur	Tourne à droite moins de 90 degrés et avance
Touche un robot	Orienté son canon sur le robot et tire plein feu	Tire plein feu un certain nombre de fois	

Conclusion : évolution de notre stratégie

Pour l'instant, notre robot est bien placé face à ses adversaires directs des autres groupes et face aux robots de bases. Mais il nous semble nécessaire :

- d'optimiser sa réaction face à un robot longeant les murs
- de lui imposer de changer de sens juste avant de percuter un mur
- de le faire tirer d'un peu plus loin en essayant d'optimiser l'angle de tir
- de le faire avancer vers le robot scanné sans utiliser la trace directe
- de le faire avancer sur les grands champs de bataille pour en rencontrer d'autres