

Projet PodRacer

Le but de ce TD est de programmer quelques algorithmes de pilotage automatique de podracers dans une course virtuelle façon SW (<https://www.youtube.com/watch?v=Dqus0aXiAVE>).

Vous trouverez dans le répertoire commun une archive compressée « PodRace.zip » regroupant tous les fichiers nécessaires au fonctionnement du jeu.

L'idée de ce TD est empruntée au jeu *Coders strike back* du site internet (de jeu :-) CodinGame : voir <https://www.codingame.com/multiplayer/bot-programming/coders-strike-back>.

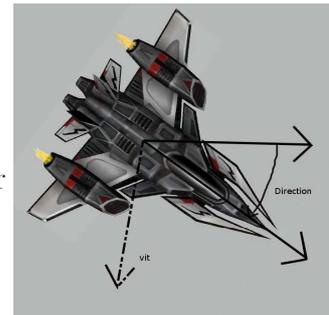
I. Énoncé

1. Copier l'archive « PodRace.zip » dans votre répertoire réseau et la décompresser.

Tous vos algorithmes devront être des fichiers indépendants avec les caractéristiques suivantes :

- le fichier se trouve dans le sous-dossier `Algos` du répertoire PodRace ;
- le fichier est un fichier Python disposant d'une fonction `algo(pos, vit, direction, adv, vit_adv, porte, portes, debut)` où (unité de mesure des distances : pixel)

`pos` est un tableau `numpy` représentant votre position (x, y)
`vit` est un tableau `numpy` représentant votre vitesse (v_x, v_y)
`direction` est un flottant représentant l'angle entre le vecteur \vec{v} et votre propre direction (voir figure ci-contre)



`adv` est un tableau `numpy` représentant la position de l'adversaire (A_x, A_y) dans le cas d'une partie à deux joueurs. Si la partie ne compte qu'un seul joueur, cet argument vaut `None`.

`vit_adv` est un tableau `numpy` représentant la vitesse de l'adversaire (w_x, w_y) dans le cas d'une partie à deux joueurs. Pour une partie à un seul joueur, cet argument vaut `None`.

`porte` est un tableau `numpy` représentant la position de la porte (p_x, p_y)

`portes` est une liste donnant les positions de toutes les portes dans l'ordre où elles doivent être passées

`debut` est un paramètre booléen qui vaut `True` si c'est le premier tour de la partie, `False` sinon. Il peut être utile pour initialiser des variables d'état en début de partie.

- votre fonction doit renvoyer :
 - 0 pour accélérer au prochain tour
 - 1 pour tourner à gauche au prochain tour
 - 2 pour tourner à droite au prochain tour
 - 3 pour utiliser un *boost* au prochain tour sachant que deux boosts seulement sont disponibles au cours de la partie.

Pour vous aider à commencer, une ébauche d'algorithme est déjà présente dans le dossier PodRace : il s'agit du fichier `naif.py`.

2. Copier le fichier `naif.py` dans le dossier `Algos` puis

compléter les emplacements marqués par `???` de sorte à ce que cet algorithme :

- calcule l'angle α entre le vecteur \vec{v} et le vecteur joignant votre Pod à la prochaine porte à passer ;
- calcule la différence `ecart` entre votre direction et l'angle α

- si la valeur absolue de cette différence est faible (par exemple inférieure à $\frac{\pi}{9}$), accélère au prochain tour ;
 - sinon tourne dans le bon sens pour que *ecart* se rapproche de 0 au prochain tour.
3. Programmer dans un fichier `OK.py` un algorithme basé sur `naif.py` mais où les coordonnées de la porte sont modifiées à l'aide de votre vitesse pour prendre ***la valeur relative qu'elles devraient avoir au prochain tour.***
 4. Programmer dans un fichier `suiveur.py` un algorithme appliquant l'algorithme précédent aux coordonnées de l'adversaire à la place des coordonnées de la porte. C'est aussi l'occasion de tenter d'utiliser vos boosts (au bon moment!).
 5. Commencer à programmer votre propre algorithme!

Quelques précisions utiles :

- toutes les mesures de distance sont effectuées dans le jeu en pixels ;
- la direction du robot est un multiple de $\frac{\pi}{10}$ (autrement dit, le robot tourne par tranches de $\pi/10$ rad) ;
- le diamètre d'une porte est environ de 80 pixels. Si le bord de votre robot passe à moins de 40 pixels du centre de la prochaine porte, il valide cette porte et se dirige alors vers la suivante ;
- chaque tour de jeu correspond à 1/20ème de seconde de temps de course ;
- le terrain de jeu mesure 1600x900 (largeur par hauteur), le coin inférieur gauche ayant pour coordonnées (0; 0), le coin supérieur droit ayant pour coordonnées (1599; 899). Les robots peuvent sortir de l'aire de jeu sans pénalité mais ne seront alors plus visibles sur les images et la vidéo de course ;
- pour tester votre robot, démarrer le fichier `PodRace.py`, choisir (au besoin) de réinitialiser joueurs et circuit, prendre l'un des circuits (aléatoire, rectangulaire ou triangulaire) et demander une course solo. La vidéo de la course sera enregistrée dans le dossier `VideoCourse` si `moviepy` est bien installé, sinon, on peut voir les images de la course dans le dossier `ImagesCourse`.