TME SOLO - 1h30

Dans ce TME SOLO, vous pouvez (et devez) réutiliser tout ce que vous avez fait durant le projet. Lorsque l'on parle de centimètre dans l'énoncé, il s'agit d'une unité de mesure constante quelconque dans votre arène simulée.

Avant de commencer le TME SOLO, créez une nouvelle branche dans votre dépôt `github` qui porte votre nom. Vous travaillerez uniquement dans cette branche. A la fin du TME, vous enverrez un email à `[nicolas.baskiotis|olivier.sigaud]@sorbonne-universite.fr` qui contiendra l'adresse du `github`, le nom de la branche et la liste des questions traitées **en indiquant obligatoirement les questions auxquelles vous avez répondu et quels fichiers ont été modifiés pour ces questions**.

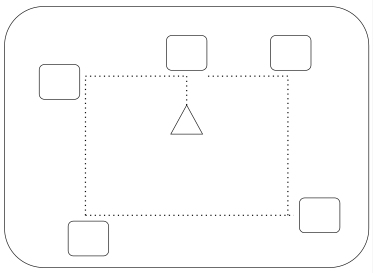
En dernier recours, si vous n'arrivez pas à sauver votre travail dans la branche, vous enverrez par email votre code.

Dans un fichier `tmesolo.py` à la racine de votre dépôt, vous regrouperez les fonctions `qX.Y()` qui permettent d'exécuter le code correspondant aux questions `X.Y` ainsi que les affichages graphiques si vous en avez.

Les parties sont indépendantes, vous pouvez les traiter dans l'ordre que vous voulez. Si votre plateforme ne dispose pas de certaines fonctionnalités, vous pouvez les supposer implémentées.

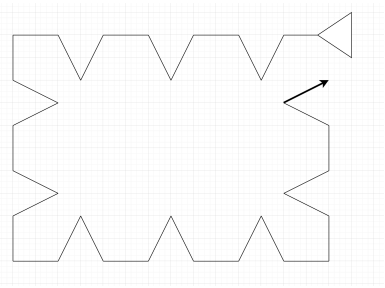
Exercice 1 – Manipulations simples

Q 1.1 Fabriquez une simulation avec 5 obstacles (de forme quelconque, carré, rectangle ou rond) placés comme sur le diagramme sur la droite et un robot au milieu de la simulation.



Q 1.2 Changez la couleur des obstacles en orange (par exemple `#FFA500` ou $(255, 165, 0)$ en RGB).

Q 1.3 On veut équiper le robot d'un crayon qui peut dessiner sa trace quand il est abaissé (et ne dessine rien quand il est levé). Ajoutez une commande `dessine(Bool: b)` au robot et modifiez le nécessaire dans votre affichage de telle façon qu'à partir du moment où `dessine(True)` est appelée, la trace de votre robot est affichée dans la simulation. Un appel à `dessine(False)` permet de désactiver la trace dans l'affichage graphique.



Q 1.4 Implémentez une stratégie qui va tout droit jusqu'à rencontrer un obstacle, tourne à gauche de 90 et continue d'aller tout droit jusqu'à rencontrer un nouvel obstacle et répète ces actions 5 fois. Testez sur la simulation de la question précédente (ne perdez pas de temps si la trajectoire n'est pas conforme à ce qui est indiqué sur le diagramme).

Q 1.5 Implémentez une stratégie (la plus courte possible en termes de code) qui permet d'effectuer approximativement la trajectoire du diagramme sur la droite.

Exercice 2 – Modification de l'environnement

On considère une simulation sans obstacle. On considère un nouvel objet, un ballon, et un nouvel obstacle particulier rectangulaire - un but. Le robot peut cogner le ballon et l'objectif est de l'envoyer dans le but. La modélisation physique du ballon est très simpliste : lorsque le robot le cogne à un temps t_0 alors que le robot a un vecteur vitesse $v^R_{t_0}$, la vitesse du ballon devient $v^B_{t_0+\delta t} = 2 * v^R_0$ (sa vitesse juste après l'impulsion). Si le ballon avait déjà une vitesse $v^B_{t_0}$, alors les deux vecteurs s'additionnent $v^B_{t_0+\delta t} = 2 * v^R_0 + v^B_{t_0}$. Sans interaction, le ballon subit des frottements et la norme de

la vitesse est modifiée selon l'équation suivante : $\|v_{t+\delta t}\| = max(0, \|v_t\| - \delta t(0.01\|v_t\|^2 - 0.06\|v_t\|))$. La direction ne change pas. Si le ballon atteint le but, alors sa vitesse devient nulle.

Q 2.1 Codez un objet ballon et sa dynamique. Pour simplifier, vous pouvez représenter le ballon par un carré. Testez la dynamique avec une vitesse initiale donnée.

Q 2.2 Codez l'interaction entre le ballon et le robot : si le robot touche le ballon, alors la vitesse est transférée selon l'équation de l'énoncé.

Q 2.3 Codez une stratégie qui permet au robot d'aller cogner le ballon. On supposera que le robot connait à tout instant la position du ballon et son vecteur vitesse.

Q 2.4 (non obligatoire) Si le ballon se cogne à un obstacle ou à un mur, alors la norme de la vitesse n'est pas changée, mais sa direction est réfléchie par rapport au mur. Dans le cas horizontal (resp. vertical), l'opération est simple : la composante en ordonnée du vecteur vitesse est inversée (resp. en abscisse). De manière générique, le nouveau vecteur vitesse est : $v - 2(v.n).n$ (produit scalaire), avec n le vecteur normal au mur/obstacle.

Q 2.5 (question ouverte mais importante) Donnez une stratégie pour que le robot marque un but. On supposera que le robot connait les coordonnées du but. Vous expliquerez l'intuition de votre stratégie en commentaire ou dans l'email.