

## Compte rendu du TME9 :

Faire attention aux noms des méthodes (`get_distance_parcourue`) sinon `update` si elle met juste à jour.

Problème avec radians par seconde et degré par seconde, il faut le régler et se mettre d'accord sur une même valeur.

Si cela prend plus pour la boucle `run` (les 10 millisecondes), le temps écoulé ne va pas être bon ! Et c'est mauvais, il faudrait passer dans `update` le temps réel. Pourquoi on passe un `dt` théorique ? On devrait mettre `dt = get.time()` par exemple, mais enlever.

Le temps du robot est utilisé par `distance parcourue` qui est utile par le proxy et pas la simulation : problème on a un `update` pour faire la mise à jour alors que c'est le proxy qui l'utilise et cela va faire un décalage et on ne veut pas ça, on doit juste avoir `last update = distance parcourue` à sa fin avant de faire le `return`.

Il y a un truc calculé par robot et l'autre par proxy, on doit choisir et le mettre dans un, les deux fonctions doivent être appelées en même temps qui doit se faire via un `update` par le proxy qui l'appelle et cela n'a rien à voir avec la simulation. Actuellement, on devrait tout mettre dans le robot, `distance_parcourue` devrait être une partie du proxy et pas robot. On enlève tous les `dt` qui n'ont pas leur place ailleurs...

Dans la boucle, on peut dire si on a été trop rapide, on peut attendre un peu, de façon à ce que ça n'influence pas le reste.

`Run` fait `update` IA som et gui, si on veut désactiver l'un on va pas pouvoir, donc c'est un problème on doit. Si notre gui est lent, on va l'attendre, et c'est pas bien. L'angle augmente de 0.1 à chaque fois et cela peut être lié à cause de ça.

On doit avoir un thread pour l'ia et la simulation et gui.

Problème de module et d'organisation. Tout est dans un module et ça ne sert à rien alors, on `robot_virtuelle` et `robot_reel` alors qu'on devrait l'appeler `proxy_reel` et `proxy_virtuel` on va devoir les laisser les noms de base.

On a un `update` qui met à jour le temps, or il devrait aussi calculer la distance parcourue, on risque de l'appeler plusieurs fois pour rien, dans contrôleur on a un `get vitesse` et `distance parcourue`, c'est chiant de se souvenir de l'appeler à chaque fois pour mettre à jour cela devrait être au proxy.

Proxy doit appeler les ordres, à chaque ordre on appelle un proxy qui met à jour et `time.time` ainsi cela ne sera pas grave de l'appeler plusieurs fois d'affilée, dans tourner on met aussi un `update`...

Stratégie angle ligne 56, le vrai robot attend une vitesse angulaire on n'a pas à donner autre chose.

On pourrait mettre un `start` dans les stratégies pour l'initialiser, donc ça serait bien de mettre un `start` si on a le temps.

Dans le main, on a un problème car quand on veut changer on va devoir la changer or on veut pas que l'utilisateur y touche, le `run` a juste besoin de prendre un proxy ! C'est le core qui devra créer, le main est trop chargé on crée tous.

Problème car on a un `dt` dans le robot réel donc corriger aussi les attentes se passent uniquement dans le main.

**Bilan :** On est dans le bon sens, des petits problèmes à corriger rapidement.