Compte rendu du TME4 :

Dans la classe simulation on gère toute l'interface graphique et il ne **faut surtout pas.**

Code : Dubitatif : on mélange l'aspect simulation et projet on fait un retour en arrière, on doit séparer la visualisation et la simulation elle gère beaucoup trop. Tout ce qui est graphique doit être ailleurs. Le module, on doit le renommer on doit l'oublier et le renommer comme (simulation, Toolbox). Les noms des modules ne sont pas clairs, on doit les renommer.

Dans Environnement on a la méthode generer_obstacle() qui est trop lourd on doit découper les tests en une partie génère et une autre qui vérifie si l'emplacement est bel et bien libre.

Les variables x et y ne servent à rien utiliser plutôt un fstring()

On doit absolument séparer la visualisions et la simulation. La visualisation est différente de la réalité, elle ne doit pas pas à gérer, nous ne sommes pas sûr que l'affichage reflète la réalité. Tourner : ordre alors que sedeplacer() non.

Gestion affichage/ et l'affichage soit dans un thread séparer c'est un OBSERTVATEUR.

Simulation : mise à jour / donner des ordres des robots. On peut faire une fausse clase IA qui prend les ordres qu'elle va donner au robot.

Revoir la fonction run pour qu'elle marche en dehors de l'utilisation de Tkinder.

Tests unitaires : teste TOUT même les constructeurs par exemple quand une fonction n'a rien on peut testez l'initialisation.

Avoir une simulation du robot qui ressemble à la vraie en donnant uniquement la vitesse des deux roues. Et séparer bien la visualisation de la simulation.

Se renseigner sur l'api du robot

Les ordres on les donne dans une classe qui fera office d'IA

Github : Message de commits sont clairs.
Tout le monde touche au code de tout le monde donc ne pas revenir dessus. On a trop de tâtonnement on doit évitez.