

## - Esquisse Environnement

- ↳ rajouter quelques méthodes
- ↳ ex faire bouger un pt
- ↳ avoir une classe robot, donner un ordre, le déplacer
- ↳ classe simulation, classe robot
- On doit tjrs penser aux tâches par rapport au projet
  - ↳ au lieu de faire des ent raînements abstraits, les lier au projet
- Faire une réunion midr semaine: vendredi / samedi c'est le meilleur.
  - ↳ Bien planifier les réunions dès le début pour prendre le rythme
- Regarder comment mieux découper les tâches en sous-tâches
  - ↳ 20~ tâches / sem
  - ↳ tâches le + indépendantes possibles, tâches unitaires au max
- Trouver une librairie graphique est aussi une tâche, il faut savoir justifier la librairie / env. choisie
- Il faut connaître les besoins / limites avant de chercher une solution

## BRAINSTORMING:

## - Structure

- ↳ • Robot
- ↳ • Environnement

↳ Méthode se déplacer

↳ Un robot muni d'un certain nb de méthode, placer ce robot dans un env, montrer le déplacement avec un print par ex

- Comment modéliser un robot/env? il faut connaître les limites de ce qu'on a / veut caractéristiques:

- capteur distance
- roues
- taille
- servo
- caméra
- gyroscope
- moteur (g/d)
- (→ batterie)
- taille

robot

→ on peut le représenter par une matrice

→ on le représente par un rectangle (ou autre forme)

→ temps

environnement

• rectangle : 4 points

• 2 pts

• largeur + longueur en taille

• robot représenté par ses coordonnées : robot.x robot.y

• représentation continue de l'espace

• fidèle à la réalité (+)

• représentation des obstacles (-)

• 3D (-)

- apprendre à tracer des rayons

- matrice:

- int : robot.x / robot.y
- nb colonnes / nb lignes
- une échelle pour savoir ce que ça représente dans la vraie vie
  - ↳ avantage: + flexible si on veut changer l'échelle
- représentation discrète de l'espace
  - ↳ désavantage: les distances vont être arrondies (-)
  - ↳ mauvaise représentation dynamique
- on peut facilement mettre des obstacles dedans (+)
- plus facile de gérer la 3D (pour suivre la balise avec la caméra) (+)

Possibilité de fusionner les deux représentations (D & C) pour tirer le meilleur des deux (changer les coords du robot int  $\rightarrow$  double)

- robot:

- coordonnées  $x$  et  $y$
- taille / forme
- direction
- $\vec{v} \times dt = \text{distance}$

Ne pas faire de class pour rien. Pas besoin le plus souvent.

Différents types de code (spaguetti, lasagne, ravioli)

(Diapo du cours)

- ↳ Révisions scalaires
- ↳ Infs en +
- ↳ A REGARDER!