Le monde de Reeborg

Reeborg est un environnement de programmation d'un petit robot utilisant un langage très rudimentaire, qui va nous permettre de nous familiariser avec le principe de Python. Pour l'utiliser, il suffit de se rendre sur la page `https://reeborg.ca/index_fr.html`.

Reeborg est un robot évoluant dans un monde en deux dimensions. Il est matérialisé par un robot humanoïde (mais on peut aussi changer son apparence en astromobile). Il se déplace de case en case, selon le quadrillage, en laissant une petite trainée verte. Les traits rouges sont des murs, que Reeborg ne peut traverser. Il peut à loisir prendre des objets présents sur le terrain ou en déposer s'il en a dans son sac.

Pour diriger Reeborg, on écrit un code, en utilisant les commandes connues par Reeborg, ou celles que l'on définit soi-même. Reeborg peut ainsi avancer, tourner sur lui-même, tester son environnement (présence ou absence de murs, d'objets à ses pieds...). Le plus simple est que vous constatiez de vous-même ce que Reeborg peut faire en démarrant les mondes d'introduction.

À faire : Cliquer sur le lien "monde de Reeborg" sur la page d'accueil, puis résoudre ensuite plusieurs mondes (cliquer sur "Description" pour comprendre ce qui est demandé) en ajoutant des commandes dans l'encart "Code Python" (on peut les choisir avec le "Clavier de Reeborg").

Les commandes de Reeborg sont expliquées en anglais dans le lien "Autres options", "Documentation" puis "G. Référence du Monde de Reeborg". Il y a principalement 3 types de commandes pour contrôler Reeborg, résumées ici en français :

- les commandes primitives :
  - `avance()` : avancer d'une case vers l'avant,
  - `tourne_a_gauche()` : tourner d'un quart de tour vers la gauche,
  - `prend()` : ramasser un objet (s'il n'y a pas d'objet par terre, Reeborg s'arrête en hurlant),
  - `depose()` : déposer un objet par terre (si Reeborg a plusieurs types d'objet dans son sac, il s'arrête en hurlant),
  - `construit_un_mur()` : construire un mur entre sa case et celle devant lui (s'il y avait déjà un mur à cet endroit, Reeborg s'arrête en hurlant),
  - `termine()` : s'éteindre,
  - `pense(n)()` : changer le temps entre deux actions (utile pour de longs programmes),
- les tests :
  - `mur_devant()` : vrai s'il y a un mur devant,
  - `mur_a_droite()` : vrai s'il y a un mur à droite,
  - `objet_ici()` : vrai si la case comporte un objet,
  - `transporte()` : vrai si Reeborg transporte au moins un objet,
  - `est_face_au_nord()` : vrai si Reeborg regarde vers le nord,
- les structures :
  - la conditionnelle : `if` <condition> : <actions> qui exécute les actions si la condition est vraie, et rien sinon,
  - la boucle itérative : `repeat` <nombre> : <actions> qui exécute les actions le nombre de fois indiqué,
  - la boucle conditionnelle : `while` <condition> : <actions> qui exécute les actions tant que la condition reste vraie,
  - la définition de fonction : `def` <fonction> : <instructions> qui définit fonction comme un raccourci pour la séquence des instructions.

À penser : Un tutoriel est disponible en cliquant sur le lien "Autres options", "Documentation" puis "A. Tutoriel de base". Vous pouvez vous y référer de temps en temps, et vous pouvez bien sûr m'appeler à n'importe quel moment.