

Projet C : Robot

Le sujet

Le projet consiste à déplacer à l'aide de *commandes* un *robot* dans un *carré* et lui faire *dessiner* son chemin dans cet environnement.

Ce robot n'est pas sans rappeler la célèbre et ancestrale *tortue Logo* dont vous avez peut-être déjà entendu parler. Regardez la page [fr.wikipedia.org/wiki/Logo_\(langage\)](http://fr.wikipedia.org/wiki/Logo_(langage)) pour plus d'information.

Chaque coté du carré est égal à $\text{MAXCOTE } u$ (unité). Au départ, le robot est placée au *centre* du carré. Les coordonnées du centre sont $(0, 0)$.

L'interface graphique représentera ce carré par une zone de dessin de la bibliothèque *libsx* et créé par la fonction `MakeDrawArea` (voir la documentation de *libsx*). Notez que *libsx* définit une *Drawing Area* comme une zone rectangulaire de hauteur et de largeur exprimées en *pixels*, et dont le point de coordonnées $(0, 0)$ est en haut à gauche. Il vous faudra donc convertir les coordonnées du robot exprimées en unités u en coordonnées *libsx* exprimées en *pixels*.

On définit un petit langage de commandes qui permet de faire évoluer le robot dans le carré. Les commandes sont les suivantes :

- **AVANCE** n : avance de n unités u ;
- **RECULE** n : recule de n unités u ;
- **GAUCHE** n : tourne sur la gauche de n degrés ;
- **DROITE** n : tourne sur la droite de n degrés ;
- **LEVE_CRAYON** : lève le crayon ;
- **BAISSE_CRAYON** : baisse le crayon ;
- **COULEUR** c : met la couleur courante c ;
- **NETTOIE** : efface l'écran et revient à la position $(0, 0)$;
- **ALLERA** $x y$: place le robot en coordonnées $u (x, y)$;
- **CACHE** : cache le robot ;
- **MONTRE** : montre le robot ;

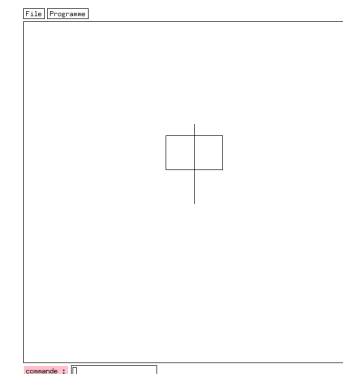
Votre programme devra permettre l'exécution de commandes écrites dans le langage défini précédemment. Le robot est muni d'un crayon. Chaque déplacement, si le crayon est baissé, provoque le tracé de ce déplacement dans le carré. Notez qu'un déplacement en dehors du carré est considéré comme une erreur et le déplacement ne pourra se faire.

Par exemple, la suite de commandes :

```
NETTOIE
LEVE_CRAYON
ALLERA -50 100
BAISSE_CRAYON
AVANCE 100
DROITE 90
AVANCE 60
DROITE 90
AVANCE 100
DROITE 90
AVANCE 60
LEVE_CRAYON
```

ALLERA 0 120
BAISSE_CRAYON
RECULE 140
CACHE

permettra faire le dessin suivant dans le carré :



L'utilisateur aura la possibilité de fournir les commandes à exécuter par le robot :

1. soit de façon interactive par l'intermédiaire de l'interface graphique ;
2. soit lues à partir d'un fichier. *Toutes* les commandes contenues dans le fichier seront alors exécutées.

Les limites du projet ne sont pas définies de façon précise afin de laisser libre cours à votre imagination quant aux possibilités fonctionnelles du robot. En particulier vous pouvez enrichir le langage de commandes, mais ce dernier doit obligatoirement contenir les commandes données plus haut.

1 Travail à réaliser

Vous devez programmer le *robot* avec toutes les fonctionnalités données ci-dessus. Votre application devra être nécessairement écrite en C et munie d'une interface graphique **obligatoirement** programmée avec la bibliothèque *libsx*.

Pensez à bien structurer votre application et à la répartir dans plusieurs fichiers qui représentent les grandes fonctionnalités de votre application. Il en sera tenu bien évidemment tenu compte dans la notation.

D'autre part, chaque fonction doit avoir un commentaire de description, et être correctement indentée. Évitez les fonctions trop longues. Utilisez des noms de fonction/type/variable/constante explicites et significatifs, et respectez l'emploi à bon escient des minuscules et majuscules.

Ce projet doit montrer votre autonomie et votre compréhension tant dans la conception du programme que dans sa réalisation. Si les codes de plusieurs projets se trouvent être identiques, ou être copiés depuis le web, tous les projets concernés seront immédiatement sanctionnés par un zéro.

2 Remise du projet

Votre projet est à faire en binôme. Il est à rendre au plus tard :

Le 18 juin 2017, minuit

sous forme d'une archive `robot-n1-n2.tar.gz` avec `n1` et `n2` sont les deux derniers chiffres de numéro d'étudiant, et que vous enverrez à vos enseignants aux adresses suivantes : `vg@unice.fr`, `jyliu@xidian.edu.cn` et `dmei2006@xidian.edu.cn`.

2.1 Cette archive devra contenir :

- les fichiers sources (`.c` et `.h`) correctement documentés (chaque fonction doit avoir un commentaire, les invariants de boucle doivent être marqués), indenté, et codé (les noms de variables explicites, éviter les trop longues fonctions) ;
- le fichier `Makefile`
- un fichier `Documentation` au format *pdf* et décrivant le fonctionnement général du programme, les algorithmes, ainsi que les choix de programmation ;
- la compilation avec les options `-Wall -pedantic` ne doit pas donner de *warning*.

Bon travail et bon courage