

<b>ISN</b>	<b>Langages et programmation</b>  <b>TP 1 Initiation à la programmation avec RURPLE</b>
------------	---

Après avoir lancé le logiciel **Rurple**, allez dans le menu « *Scène/Nouveau labyrinthe...* » afin de créer une scène carrée de 4 carreaux de côté. (Redimensionnez si nécessaire les différentes zones d'affichage avec la souris).

Allez ensuite dans le menu « *Scène/Nombre de billes...* » et placez quatre billes dans la poche du robot.

Le robot **Rurple** obéit aux ordres élémentaires suivants :

- « avance » (touche A) : le robot avance d'une case devant lui ;
- « gauche » (touche G) : le robot tourne sur lui-même de 90° vers la gauche ;
- « dépose » (touche D) : le robot dépose une bille sur la case où il se trouve ;
- « prends » (touche P) : le robot prends une bille sur la case où il se trouve.

1. Manipulez le robot et faites lui déposer ses quatre billes aux quatre coins de la scène.

À tout moment vous pouvez réinitialiser la scène en cliquant sur l'icône  ou en appuyant sur la séquence de touches CTRL-R). Ceci a pour effet de remettre les billes à leur place et de replacer le robot sur sa case initiale. Faites le maintenant !

2. Plutôt que d'utiliser les touches vous pouvez écrire dans la fenêtre de gauche un texte qui dicte au robot les ordres qu'ils doit exécuter. Ce texte est appelé un **programme**.

(a) Entrez le texte ci-contre dans la fenêtre de gauche. N'oubliez pas les parenthèses.

(b) Appuyez ensuite sur la touche F5 ou cliquez sur l'icône  pour exécuter le programme.

Chaque ligne est une instruction. Le robot exécute ces instructions séquentiellement, c'est-à-dire les unes à la suite des autres jusqu'à la fin du texte.

(c) Réinitialisez la scène, puis remplacez le texte par un programme qui dirige le robot de façon à ce qu'il dépose ses quatre billes aux quatre coins de la scène.

```
depose ()
avance ()
avance ()
depose ()
gauche ()
avance ()
avance ()
```

3. Pour vous souvenir plus tard de ce que fait exactement ce programme il est très commode de pouvoir ajouter à votre texte quelques lignes explicatives.

(a) Ajoutez au début de votre programme les cinq lignes ci-dessous :

```
""""Scène : Salle carrée 4x4. Le robot a quatre billes en poche.
      Il est en bas à gauche et regarde vers l'Est.
Action : Le robot dépose ses quatre billes aux quatre coins de la
salle.
""""
```

Veuillez à bien encadrer le texte par trois guillemets double ("). Cette partie de texte apparaît alors en rouge foncé. La présence de ces guillemets permet de signaler au robot qu'il ne s'agit pas d'instructions qu'il doit analyser et exécuter mais de *commentaires* qu'il doit ignorer.

(b) Pour retrouver facilement vos programmes il est important de choisir des noms qui soient pertinents et de les ranger avec logique dans des dossiers.  
Créez un dossier « *purple* » sur votre ordinateur ainsi qu'un sous-dossier « *premiers-pas* » et enregistrez votre programme dans ce sous-dossier sous le nom « *quatre-billes-1.py* » (allez dans « *Programme/Enregistrer sous...* »).

Le robot RURPLE dispose de capteurs dont les valeurs sont Vraies ou Fausses ( en anglais **True et False** ). Il existe des procédures `mur_devant()`, `bille_au_sol()`, `depose()`, `prends()`, ... Le robot perçoit son environnement et sait à tout moment s'il y a un mur devant lui, s'il y a une bille au sol, etc.

(

4. Dans le programme ci-contre nous rencontrons pour la première fois le mot « **while** ». C'est un mot du langage que nous rencontrerons souvent désormais. Si je vous dis que ce mot se traduit par « tant que » en français, peut-être devinez vous l'effet que produit l'exécution de ce programme ?

```
while not bille_au_sol():
    avance()
while bille_au_sol():
    prends()
    avance()
while bille_en_poche():
    depose()
    avance()
```

- a) Créez un couloir identique à celui-ci :



- b) Ecrivez ce programme dans l'éditeur. N'oubliez pas les « : » à la fin des lignes qui débutent par des « **while** » et pensez bien à indenter la ligne qui suit.
- c) Exécutez alors le programme et essayez de comprendre chacune des parties.
- d) Revenez à la situation :  
-salle carrée 4x 4 et 4 billes en poche.
- e) Reprenez le programme « *quatre-billes-1.py* » et modifiez-le pour écrire une version plus courte.

Vous appellerez « *quatre-billes-2.py* » cette version du programme qui utilise le mot « **while** ».

**Attention :** Cette nouvelle syntaxe doit vous permettre de créer un programme qui fonctionne correctement pour *n'importe quelle scène rectangulaire*, c'est-à-dire de dimensions quelconques. Lorsque votre programme est au point, réécrivez les commentaires en tête du programme et enregistrez-le sous le nom « *quatre-billes-3.py* ».