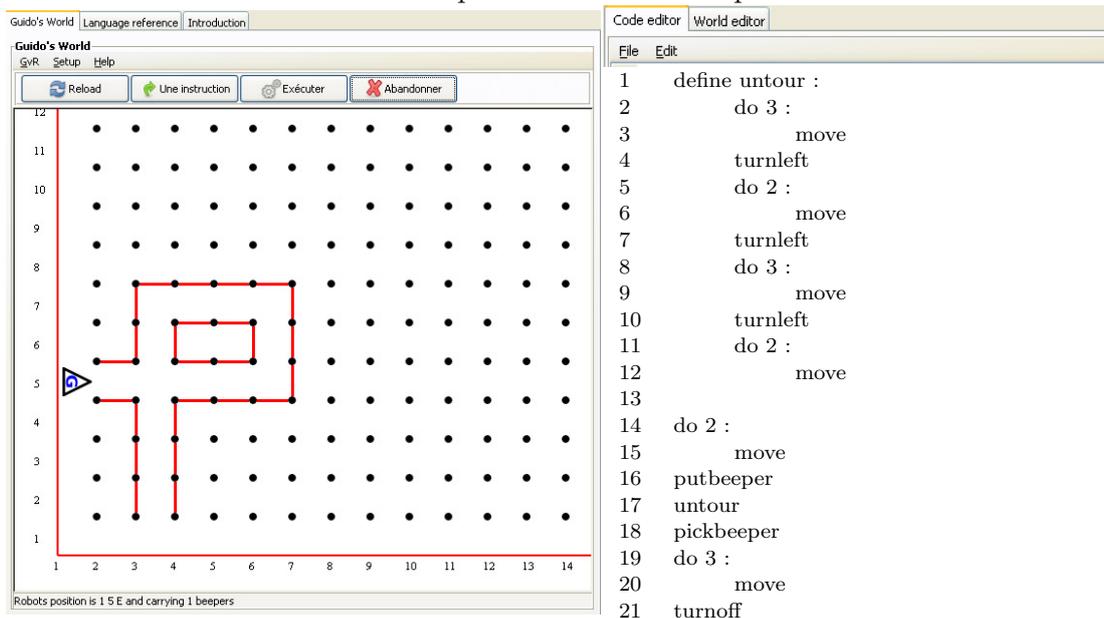


1 Présentation de GvR

Guido van Robot 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 le télécharger, vous pouvez copier sur clef le dossier Guido ou bien le télécharger : gvr.sourceforge.net (Puis Download sur la colonne de gauche, Download GvRng, GvR GTK, 4.4, GvRng-4.4-win32-standalone.zip)

Guido est un robot évoluant dans un monde en deux dimensions, constitué de rues et d'avenues. Il est matérialisé par une triangle dont l'orientation indique dans quelle direction il regarde. Il se déplace de case en case, les coins des cases étant matérialisés par les points noirs. Les traits rouges sont des murs, que Guido ne peut traverser. La barre de statut indique sa position, sa direction et le nombre de sonnettes qu'il transporte. Il peut à loisir prendre des sonnettes présentes sur le terrain ou en déposer s'il en a dans son sac. Voici un exemple de situation de départ :



```

1  define untour :
2      do 3 :
3          move
4          turnleft
5          do 2 :
6              move
7          turnleft
8          do 3 :
9              move
10             turnleft
11             do 2 :
12                 move
13
14 do 2 :
15     move
16 putbeeper
17 untour
18 pickbeeper
19 do 3 :
20     move
21     turnoff
  
```

Pour diriger Guido, on écrit un code, en utilisant les commandes connues par Guido, ou celles que l'on définit soi-même. Guido peut ainsi avancer, tourner sur lui-même, tester son environnement (présence ou absence de murs, de sonnettes...). On peut lire ci-dessus un exemple de code. Le plus simple est que vous constatiez de vous-même ce que ce programme fait effectuer au robot.

A faire : Copier le fichier "Guido.zip" depuis le réseau sur la session, à un endroit quelconque. Extraire les fichiers, puis exécuter "gvrng.exe", et aller dans l'onglet "Guido's World" dans la fenêtre de gauche.

Ensuite, dans la fenêtre de droite, cliquer sur l'onglet "World editor", puis sur "File" et "Ouvrir". Aller dans le dossier où Guido a été extrait, puis sélectionner (sans l'ouvrir) le dossier "Algorithmique". Rajouter ce dossier dans les Raccourcis, en cliquant sur "Ajouter" car on chargera tous les fichiers à partir de ce dossier. Une fois cela fait, ouvrir le dossier "Algorithmique" et ouvrir "exo0.wld". Enfin, cliquer sur l'onglet "Code editor" et ouvrir le fichier "exo0.gvr" au même endroit.

Les 12 premières lignes ne sont pas exécutées, elles constituent une définition de fonction permettant de découper la tâche à effectuer en sous-tâches plus simples à appréhender. La première commande exécutée par le robot est une boucle do 2, qui lui demande de répéter deux fois les instructions suivant la ligne do qui sont indentées. Ensuite, il dépose sa sonnette (putbeeper), exécute l'ensemble des instructions définissant la fonction untour, ramasse sa sonnette (pickbeeper), avance encore de trois cases, et s'éteint.

Pour bien comprendre ce qui se passe, le plus simple est de modifier une ligne ou deux, pour voir le nouveau résultat. Et s'il se passe quelque chose qui n'est pas prévu, tenter de comprendre en suivant pas-à-pas le déroulement des instructions !

2 Les commandes de GvR

Elles sont expliquées en anglais dans l'onglet "Language reference" du logiciel. En voici un résumé en français :

Les commandes permettant de contrôler Guido sont de 3 types :

- les commandes primitives :
 - ◇ move : avancer d'une case vers l'avant,
 - ◇ turnleft : tourner de 90° vers la gauche,
 - ◇ pickbeeper : ramasser une sonnette,
 - ◇ putbeeper : déposer une sonnette,
 - ◇ turnoff : s'éteindre ;
- les tests :
 - ◇ front_is_clear : vrai si il n'y a pas de mur devant,
 - ◇ front_is_blocked : vrai si il y a un mur devant,
 - ◇ left_is_clear : vrai si il n'y a pas de mur à gauche,
 - ◇ left_is_blocked : vrai si il y a un mur à gauche,
 - ◇ right_is_clear : vrai si il n'y a pas de mur à droite,
 - ◇ right_is_blocked : vrai si il y a un mur à droite,
 - ◇ next_to_a_beeper : vrai si la case comporte une sonnette,
 - ◇ not_next_to_a_beeper : vrai si la case ne comporte pas de sonnette,
 - ◇ any_beeper_in_beeper_bag : vrai si Guido transporte au moins une sonnette,
 - ◇ no_beeper_in_beeper_bag : vrai si Guido ne transporte aucune sonnette,
 - ◇ facing_north : vrai si Guido regarde vers le nord,
 - ◇ not_facing_north : vrai si Guido ne regarde pas vers le nord,
 - ◇ facing_south : vrai si Guido regarde vers le sud,
 - ◇ not_facing_south : vrai si Guido ne regarde pas vers le sud,
 - ◇ facing_east : vrai si Guido regarde vers l'est,
 - ◇ not_facing_east : vrai si Guido ne regarde pas vers l'est,
 - ◇ facing_west : vrai si Guido regarde vers l'ouest,
 - ◇ not_facing_west : vrai si Guido ne regarde pas vers l'ouest ;
- les structures :
 - ◇ la conditionnelle : if <condition> : <actions> qui exécute les actions si la condition est vraie, et rien sinon,
 - ◇ la boucle itérative : do <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 : define <fonction> : <instructions> qui définit fonction comme un raccourci pour la séquence des instructions.

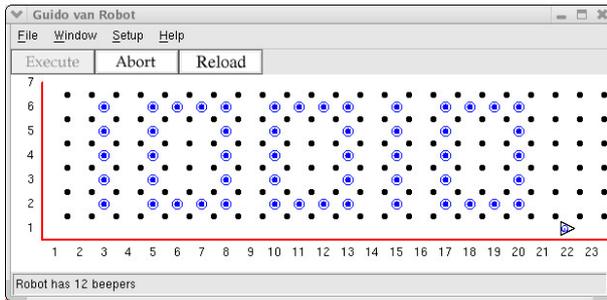
A penser : Un tutoriel est disponible en cliquant sur "Help" puis GvR Lessons, en anglais. Vous pouvez vous y référer de temps en temps, et vous pouvez bien sûr m'appeler à n'importe quel moment.

3 Exercices niveau 1

Exercice n°1 : Guido est majeur !

Guido vient d'atteindre ses 18 ans, et aimerait le montrer au reste du monde. Pour ce faire, il aimerait écrire son âge dans les ruelles du monde dans lequel il évolue. Puisque Guido ne comprend que les nombres binaires, il écrira ainsi 10010.

1. Ouvrir le monde "exo1.wld"
2. Ecrire la fonction dessiner_un qui écrit un "1" avec des sonnettes (une ligne de 5 sonnettes)
3. Ecrire la fonction dessiner_zero qui écrit un "0" avec des sonnettes (un rectangle de 5 sonnettes de haut et 4 sonnettes de large)
4. Utiliser ces deux fonctions pour que Guido écrive 18 en binaire. Le monde final ressemblera à :

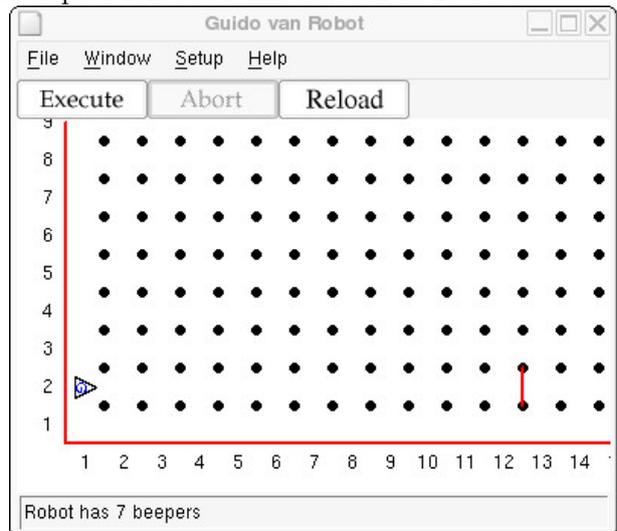
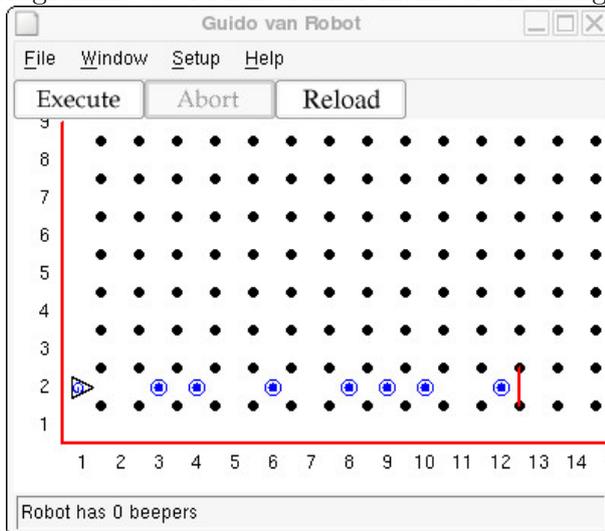


Exercice n°2 : Prise de décision

Guido doit remonter la 2e Rue, du début jusqu'à la 12e Avenue. Sur son passage, il y aura parfois des sonnettes, parfois pas. Le but est de lui demander de ramasser toutes les sonnettes sur son passage. Attention ! Si vous lui demandez de ramasser une sonnette alors qu'il n'y en a pas sous lui, il s'éteindra.

1. Ouvrir le monde "exo2.wld"
2. Faire ramasser à Guido toutes les sonnettes sur son passage, et revenir tout au début de la 2e Rue, orienté vers l'est.

Guido sait qu'il y a 12 cases sur lesquelles il doit éventuellement ramasser les sonnettes, mais il ne faut pas "lui dire" sur lesquelles il y a des sonnettes, et sur lesquelles il n'y en a pas. Ainsi, votre programme devra fonctionner même si on change la disposition des sonnettes sur la 2e Rue.



Exercice n°3 : Prise de décision bouclée

Le programme que l'on vient d'écrire à l'exercice 2 a sûrement un défaut à corriger : si vous avez fait faire à Guido 12 fois la même opération, en recopiant 12 fois la même suite d'instructions, alors modifiez le programme à l'aide d'une boucle do.

4 Exercices niveau 2

Exercice n°4 : Guido à Roland-Garros

1. Guido veut travailler à Roland-Garros comme ramasseur de balles. Le responsable lui fait passer un test : partant de la situation de la figure 1 (monde "exo4.1.wld"), Guido doit ramasser toutes les balles (les sonnettes) disséminées sur le terrain et les remettre dans la corbeille dans le coin inférieur gauche, pour arriver à la situation de la figure 2.

Écrivez un code permettant à Guido de réaliser cette opération. Il devra comporter une boucle do répétant 6 fois le ramassage d'une rangée de balles, cette opération étant elle-même constituée d'une boucle do. Vous pouvez enregistrer votre code dans "exo4.1.gvr".

2. Ça y est, Guido est engagé! Lors de son premier travail, il doit ramasser les balles laissées sur le terrain après l'entraînement d'un champion. Mais il se rend compte que la situation (figure 3, monde "exo4.2.wld") est un peu différente de celle de son test d'embauche : il y a parfois plusieurs balles au même endroit, et d'autres endroits ne comportent pas de balle.

Modifiez le code de votre premier programme de manière à ce que Guido puisse accomplir cette nouvelle tâche. Vous pouvez enregistrer votre code dans "exo4.2.gvr".

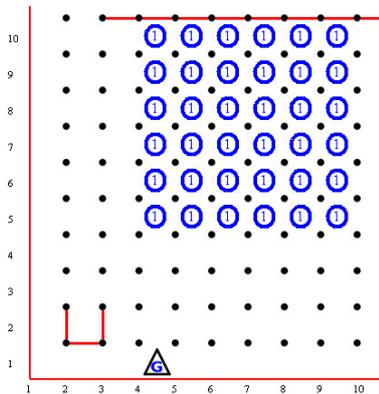


Figure 1

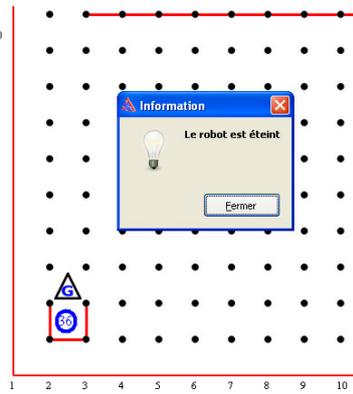


Figure 2

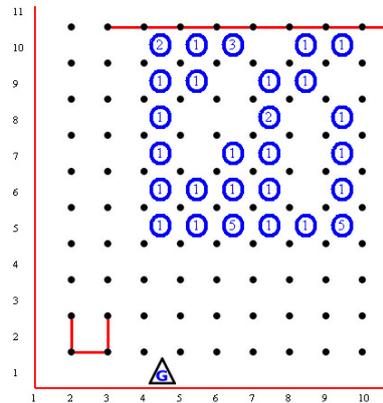


Figure 3

Exercice n°5 : Guido fait la course

1. Guido a décidé de faire du sport : il veut se mettre à la course de haies (figure 4, monde "exo5.1.wld"). Une course se passe de la manière suivante : Guido part du coin inférieur gauche, et doit courir vers la droite en contournant les haies représentées par des murs, puis s'arrêter à la 17^e intersection (figure 5).

- (a) Écrivez une fonction saute_haie : partant de la gauche de la haie, regardant vers l'est, Guido doit se retrouver derrière la haie, regardant encore vers l'est.
- (b) Utilisez cette fonction pour écrire un programme réalisant l'intégralité de la course.

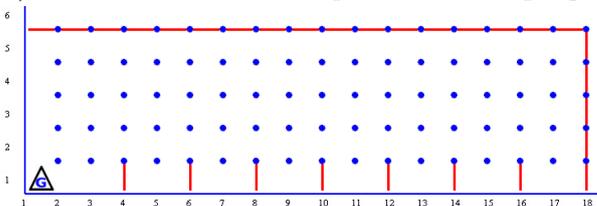


Figure 4

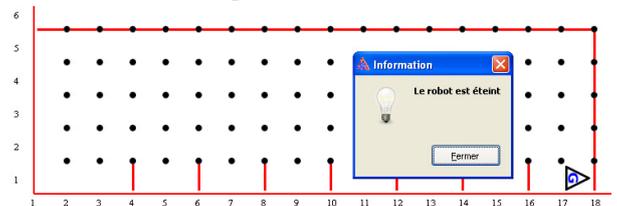


Figure 5

2. Guido trouve la course de haies trop facile, il veut se mettre à la course de murs : les obstacles sont disposés aux mêmes endroits sur la piste, mais ils sont de hauteur variable (figure 6, monde "exo5.2.wld"). Modifiez votre code de manière à permettre à Guido d'exceller dans ce nouvel exercice.

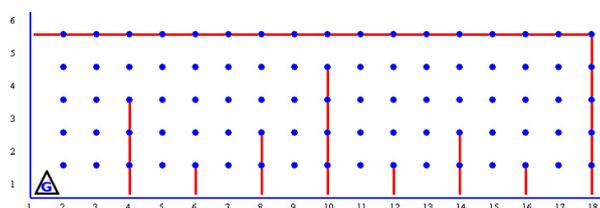


Figure 6