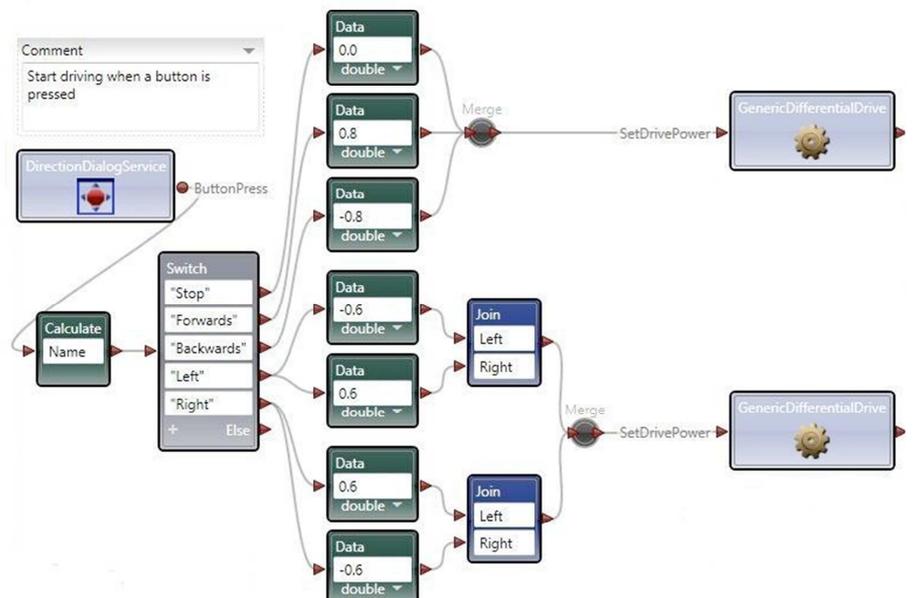
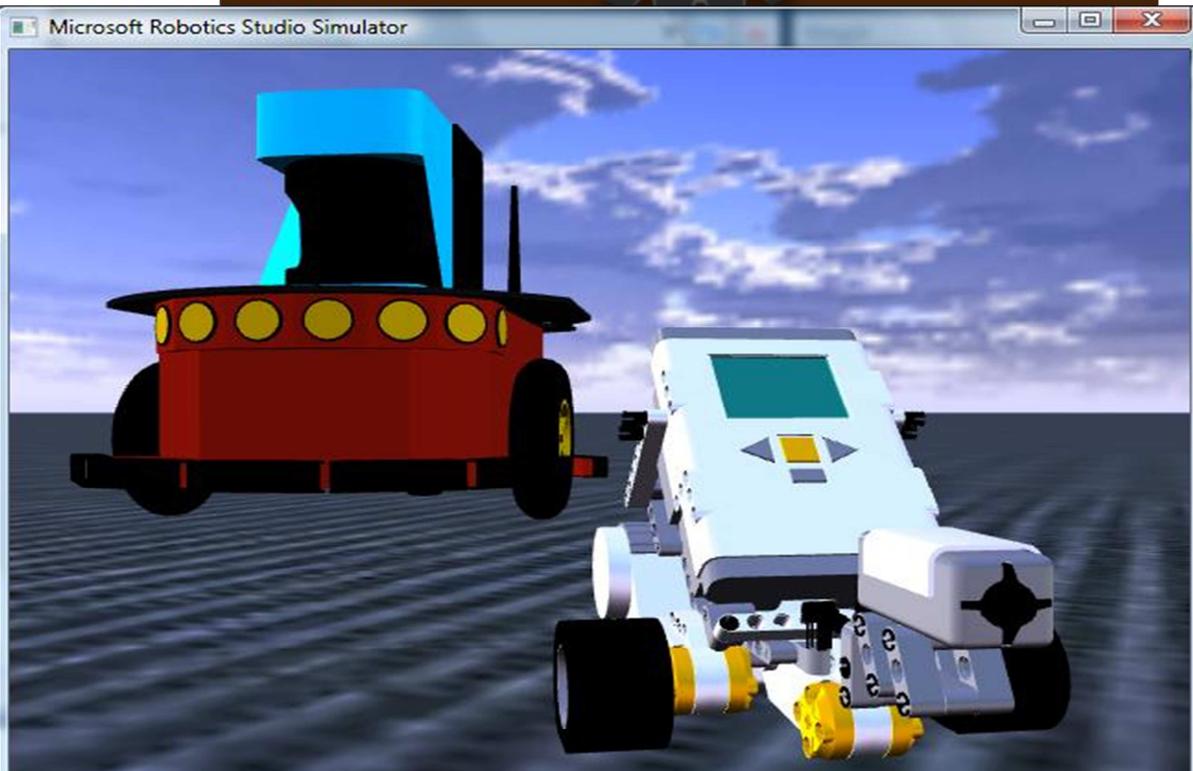


Microsoft® ROBOTICS STUDIO

Microsoft
Visual Programming Language

© 2010, Microsoft Corporation. All rights reserved.



Sommaire

Exemples de programmes en VPL

1 - Prise en main des outils logiciels	3
2 - Premier programme	6
3 - Pilotage manuel d'un Robot Lego réel à partir du PC	9
4 - Utilisation des services Lego (éviterement d'obstacles)	10
Annexe : Première utilisation d'un robot Lego NXT	18
Webographie	19

1 - Prise en main des outils logiciels

Source

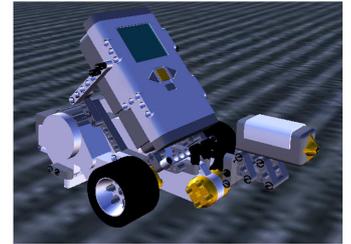
D'après un exemple du site « Génération Robot ».

Cible

Robot Lego Nxt Tribot déplacé dans Visual Simulation Environnement.

Description

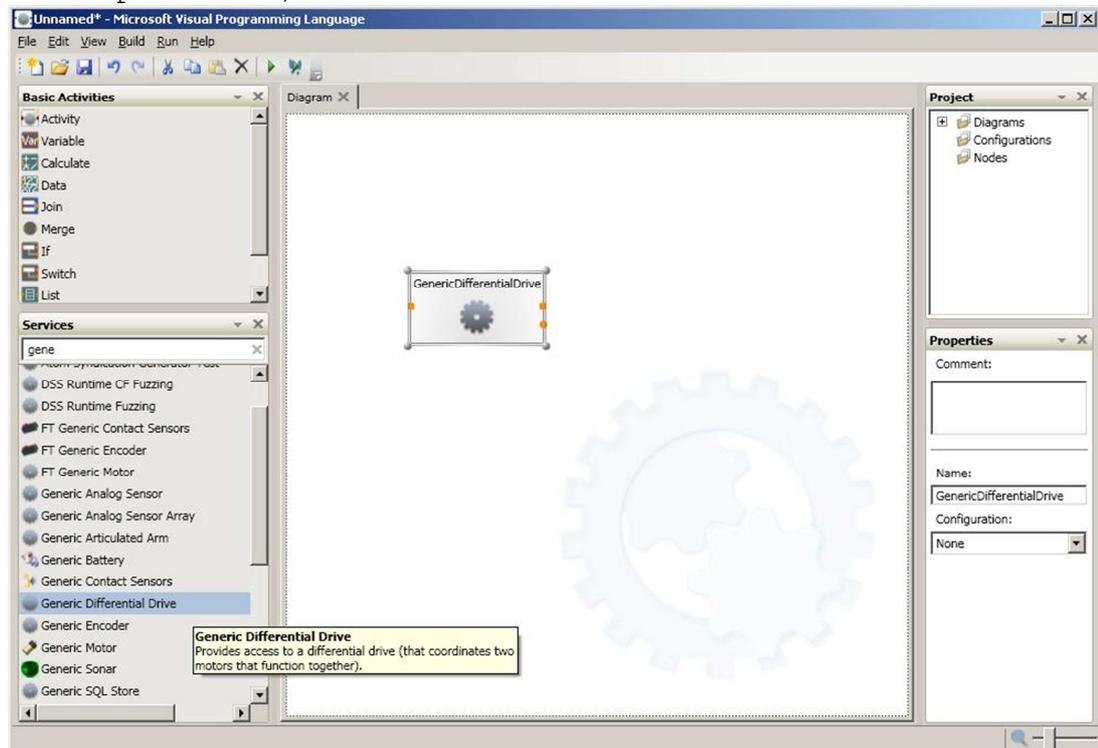
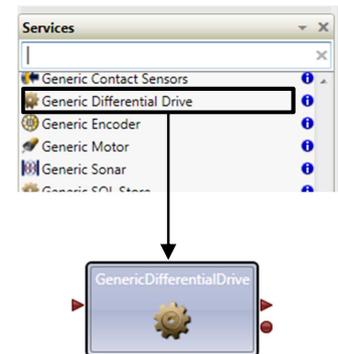
Déplacement d'un robot Lego NXT Tribot dans l'environnement de simulation Visual Simulation Environment. Le code est généré par Visual Programming Language.



Création du diagramme VPL

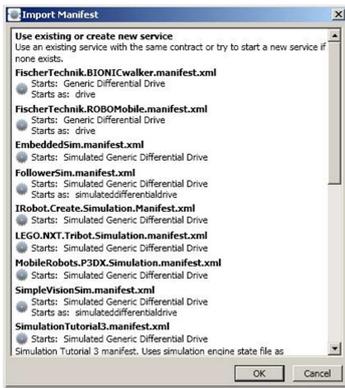
- Mise en place du service « Generic Differential Drive »**
 Démarrez l'outil Visual Programming Language depuis le menu démarrer de votre PC. Dans l'onglet service à gauche, sélectionnez le service *Generic Differential Drive* et glissez le sur le diagramme de travail (il s'agit de la partie centrale de l'écran VPL).

Le service *Generic Differential Drive* permet de piloter un robot ayant deux roues et se dirigeant ainsi à l'aide d'une conduite différentielle (c'est-à-dire qu'au lieu d'orienter les roues à l'aide d'un volant comme sur une voiture, l'orientation du robot est réalisée en faisant varier les vitesses et le sens de rotation sur chaque des deux roues indépendamment).



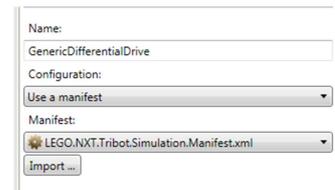
Lorsque la boîte correspondant au *Generic Differential Drive* est sélectionnée, dans la colonne des propriétés, à droite, il existe une liste déroulante nommée Configuration.

Sélectionnez « Use a manifest ».



Ceci indique que pour ce service *Generic Differential Drive*, dont l'objectif est de piloter un robot ayant deux moteurs et deux roues (un sur chaque roue), nous allons utiliser un fichier XML de paramétrage (un *manifest*) existant.

Cliquez sur le bouton *Import* qui est apparu. La fenêtre suivante est lancée : Cette fenêtre liste tous les *manifests* trouvés sur votre PC. Il s'agit de fichier XML que vous avez réalisé ou bien installé avec MSRDS. Le *manifest* indique la liste des services qui devront être démarrés.



Choisissez `LEGO.NXT.Tribot.Simulation.Manifest.xml`. Ce manifest indique à MSRDS que le robot lancé sera le tribot de Lego et que celui-ci sera lancé dans l'environnement de simulation. Notez que si nous avons choisi `LEGO.NXT.Tribot.xml`, cela aurait indiqué que le robot est un robot réel. Comme vous le voyez, passer de l'environnement de simulation à l'environnement réel est aussi simple que cela, il suffit de changer de manifest ! (Celui-ci doit être correctement paramétré !) (Voir le paragraphe 3)

- **Mise en place du service « Simple Dashboard »**
 Dans la liste des services, dans le menu de gauche, trouvez le service *Simple Dashboard* et glissez-le sur le diagramme.



Ce service affiche une fenêtre sur votre PC qui vous permet de piloter le robot depuis cette interface (que ce soit le robot dans l'environnement virtuel ou bien un robot réel).

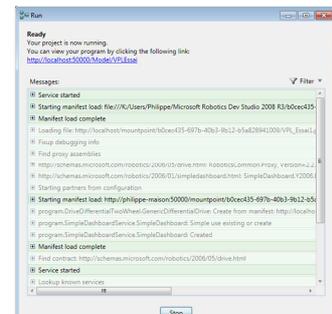
La page « Diagram » doit ressembler à ceci :



Génération du projet

Sauvegardez votre travail et cliquez ensuite sur le bouton *Start* en forme de flèche verte.

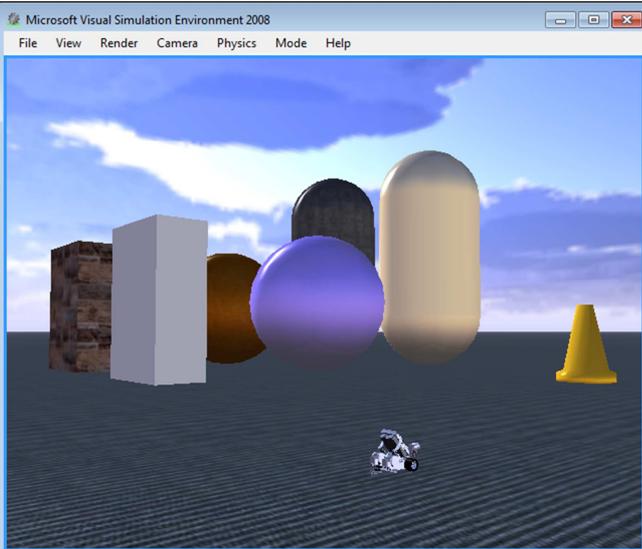
Cela lance tout d'abord la fenêtre RUN qui vous montre les étapes techniques de votre programme.



Deux fenêtres apparaissent :

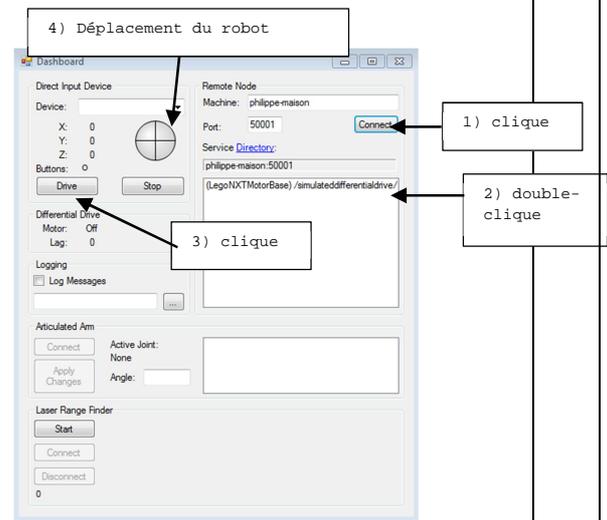
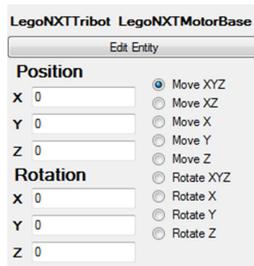
Microsoft Visual Simulation Environment
(puisque spécifié dans le manifest)

Le **dashboard** (le tableau de bord). Afin de piloter le robot Tribot à l'aide du Dashboard, il faut indiquer au Dashboard où trouver le robot. Pour cela, dans le champ Machine, saisissez l'adresse IP locale de votre machine (127.0.0.1) puis cliquez le bouton *Connect*. Vous devez voir le Dashboard tel que présenté dans la figure ci-dessous :



↓↑ : zoom

Si le robot se retourne (en cas de choc avec un obstacle), sélectionnez
Mode → Edit
puis LegoNXTMotorBase



Dans la zone de texte au milieu à droite, le DashBoard a découvert le Tribot. Double-cliquez dessus, puis cliquez sur le bouton Drive qui se trouve à gauche. Vous y voilà, à l'aide de la boule de direction en haut à gauche, pilotez le Tribot dans l'environnement de simulation. Constatez le respect des règles physique dans l'environnement de simulation en fonçant sur le plot par exemple.

2 - Premier programme

Source

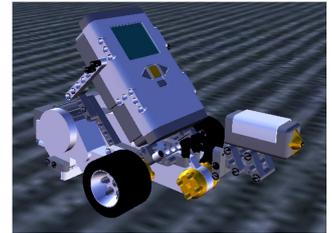
D'après un exemple du site « Génération Robot ».

Cible

Robot Lego Nxt Tribot déplacé dans Visual Simulation Environnement.

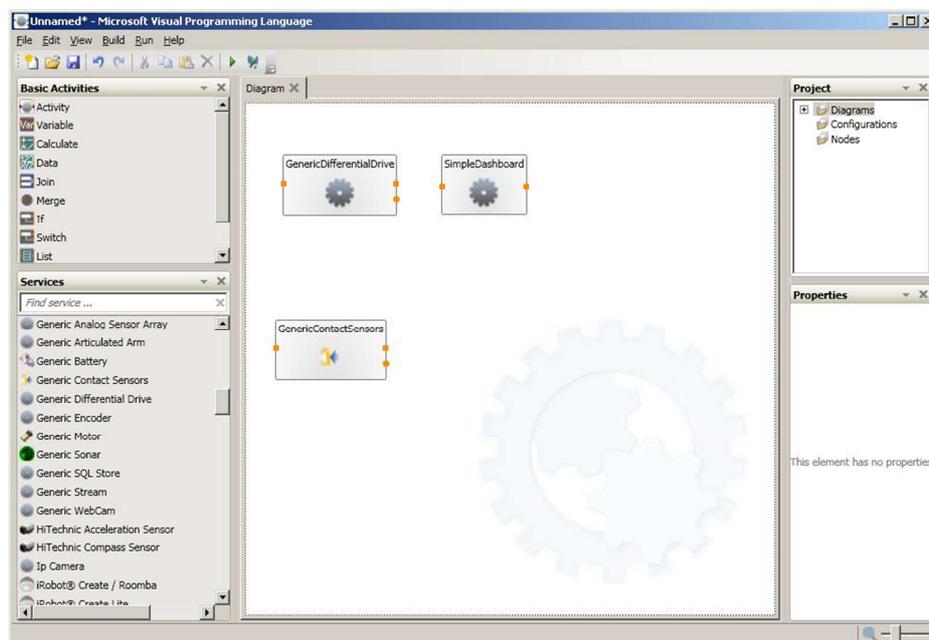
Description

Affichage d'une fenêtre Windows lorsque le robot Lego Nxt Tribot touche un obstacle.

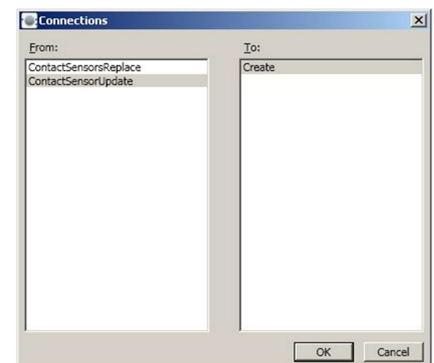


Création du programme de pilotage de robot

- Glissez-déposez les services **Generic Differential Drive**, **Simple Dashboard** et **Generic Contact Sensor** sur le diagramme principal comme illustré ci-dessous.



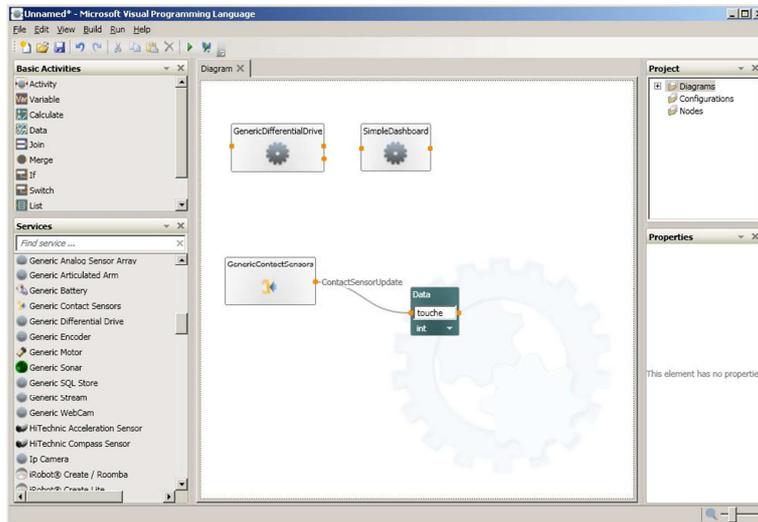
- Faites glisser ensuite une activité **Data** depuis la zone 1 vers le diagramme. Reliez ensuite la sortie **Notification** du service **Generic Contact Sensors** à l'entrée de l'activité **Data**. Pour cela, placez votre curseur sur le bouton orange en forme de cercle qui se trouve en bas à droite du **Generic Contact Sensors** et tout en maintenant la souris enfoncée, rejoignez la case orange se trouvant à gauche de l'activité **Data**. Cela fait apparaître une boîte **Connections**. Cette boîte vous permet d'indiquer quelles sont les **paramètres de la liaison** entre les deux cases que vous reliez sur le diagramme.



Dans notre cas, cliquez dans la zone de gauche sur **ContactSensorUpdate** et dans la zone de droite sur **Create**.

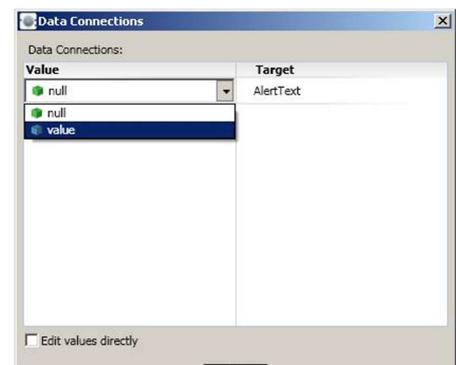
Vous indiquez par ces choix qu'un événement sera déclenché lorsque l'état du capteur de contact change (choix de gauche) et qu'à chaque fois, cela va créer une valeur (choix de droite).

Dans la boîte *Data*, remplacez le 0 par le mot *Choc*. Dans la liste déroulante, changez le type de la valeur utilisé : choisissez « string » au lieu de « int ». Vous indiquez donc que la valeur qui est créée à chaque fois que le capteur de contact est activé est une valeur de type chaîne de caractère et qu'elle prend la valeur du mot ou de la phrase que vous avez choisie.



- Glissez-déposez depuis la liste des services, un service qui se nomme *SimpleDialog*. Ce service affiche des fenêtres de messages simples. Comme précédemment, reliez la sortie de la boîte *Data* à l'entrée de la boîte *SimpleDialog*.

Dans la boîte *Connections* qui apparaît, choisissez *DataValue* à gauche et *AlertDialog* à droite. Cela indique que la valeur de la boîte *Data* sera passée à une boîte de dialogue, c'est-à-dire une fenêtre simple qui n'affiche qu'un message. Lorsque vous cliquez sur OK, une seconde fenêtre *Connections* apparaît, il s'agit d'une fenêtre *Data Connections* vous demandant de préciser exactement quelle valeur va être passée la boîte de dialogue. Dans la liste déroulante de gauche choisissez « value » au lieu de null.



Vous avez terminé la partie programmation. Sauvegardez votre programme. Vous constatez donc qu'il y a trois flux en parallèle dans ce programme. Trois flux vont donc fonctionner en parallèle, le flux qui gère la conduite du robot, le flux qui gère l'interface de pilotage et enfin le flux qui gère le capteur de contact. Ce petit exemple illustre combien il est aisé de réaliser des programmes multitâche à l'aide de cet environnement.

Choix du robot piloté par le programme

Avant de tester, nous devons préciser quel robot va être utilisé. En effet, Visual Programming Language ne fait pas d'hypothèse sur le robot qui va être utilisé. Mieux, le programme que vous réalisez est compatible avec tout type de robot pour peu qu'il existe des services permettant de le manipuler. Préciser le robot revient à dire quelle est le constructeur du robot en question et quelle est sa configuration (combien et quels capteurs, où sont placés ces capteurs, le nombre de roues...). Toutes ces informations sont stockées dans un fichier manifest. C'est un fichier XML que vous pouvez vous-même créer. Pour notre exemple, nous allons utiliser un manifeste existant et fourni avec Microsoft Robotics Studio.

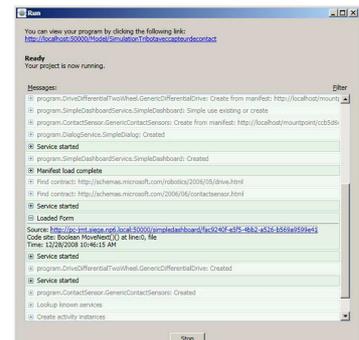
- Cliquez sur la boîte *Generic Differential Drive*, puis dans la zone Propriétés (zone 5), choisissez la valeur « Use a manifest » pour la liste déroulante « Configuration ». Cliquez sur le bouton « Import » et dans la liste qui apparaît, choisissez **LEGO.NXT.Tribot.simulation.manifest.xml**. Ce manifest représente le robot Tribot basé sur Lego Mindstorms NXT, dans l'environnement de simulation de Microsoft Robotics Studio. Cliquez sur OK.

Faire la même manipulation en sélectionnant la boîte *GenericContactSensors* mais cette fois, au lieu d'importer le manifest en cliquant sur le bouton import, sélectionnez le manifest précédent dans la seconde liste déroulante.

Sauvegardez votre programme. Vous êtes à présent prêt à le tester. Pour ce faire, appuyez sur la touche F5.

Utilisation du programme

Déplacez le robot et constatez qu'une fenêtre s'ouvre lorsque son capteur de choc touche un obstacle.



3 - Pilotage manuel d'un Robot Lego réel à partir du PC

Cible

Robot Lego Nxt.

Description

Déplacement du Robot Lego NXT avec le Dashboard.



- Reprenez la démarche du paragraphe 1 pour obtenir le diagramme ci-dessous.



- Sélectionnez le manifest ci-contre pour le service GenericDifferentialDrive.

Name: GenericDifferentialDrive

Configuration: Use a manifest

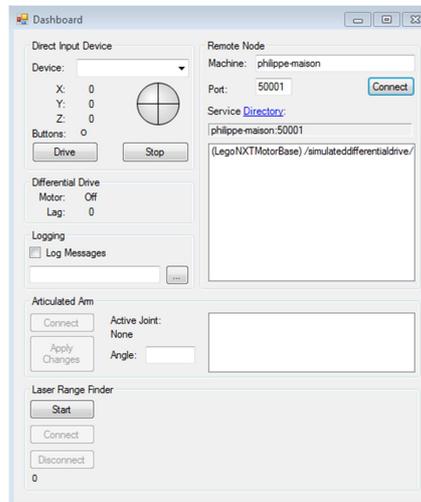
Manifest: LegoNXTDrivev2 in LEGO.NXT.TriBot.Manifest.xml

Import ...

Par défaut, le manifest est configuré comme ci-dessous :

```
<?xml version="1.0" encoding="UTF-8"?>
- <DriveState xmlns="http://schemas.microsoft.com/rob
  <DistanceBetweenWheels>0</DistanceBetweenWheels>
  - <LeftWheel>
    <MotorPort>MotorB</MotorPort>
    <ReversePolarity>>false</ReversePolarity>
    <WheelDiameter>0.056</WheelDiameter>
  </LeftWheel>
  - <RightWheel>
    <MotorPort>MotorC</MotorPort>
    <ReversePolarity>>false</ReversePolarity>
    <WheelDiameter>0.056</WheelDiameter>
  </RightWheel>
  <PollingFrequencyMs>0</PollingFrequencyMs>
</DriveState>
```

La connectique de la brique est à adapter en conséquence ou bien il faut changer les paramètres du manifest.



- Déplacez le robot avec le Dashboard

4 - Utilisation des services Lego

Source

Tutorial MSRDS (Ultrasonic Sensor)

Cible

Robot Lego Nxt réel

Description

Déplacement autonome du robot Lego Nxt en évitant les obstacles.



Les exemples précédents font appel à des services génériques (non dédiés à une cible particulière). Pour la commande d'une brique Lego NXT, il est plus simple de faire appel aux services Lego contenu dans VPL.



Ils évitent en particulier d'écrire directement dans les fichiers manifest !

- **Mise en place du service « Lego NXT Brick (v2) »**

Les programmes utilisant les services Lego doivent contenir la brique Lego.

Placez le service « Lego NXT Brick (v2) sur la feuille « Diagram »

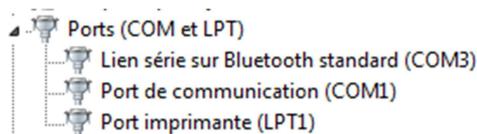


Paramétrage de la brique Lego

Ce service supporte uniquement la connexion bluetooth.

Celle-ci doit être configurée conformément au paramétrage de la liaison série associée à la clé.

Exemple



Sélectionnez la brique puis « **Set initial configuration** » dans **Properties**

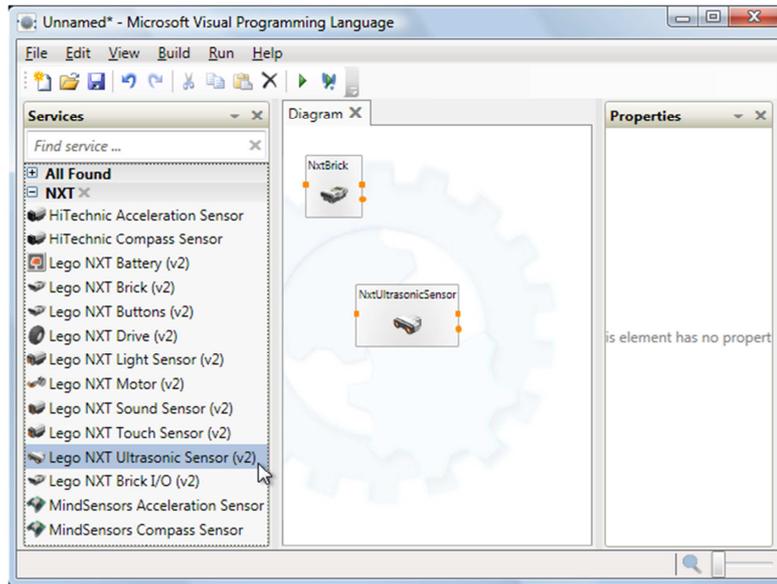
Configuration

Exemple :

[-] Configuration	
SerialPort	3
BaudRate	9600
ConnectionType	Bluetooth
ShowInBrowser	<input checked="" type="checkbox"/>

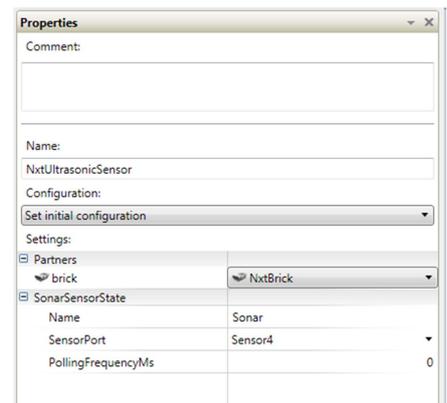
- Mise en place du service « Lego NXT Ultrasonic sensor (v2) »

Placez un capteur à ultrasons sur la feuille « Diagram »



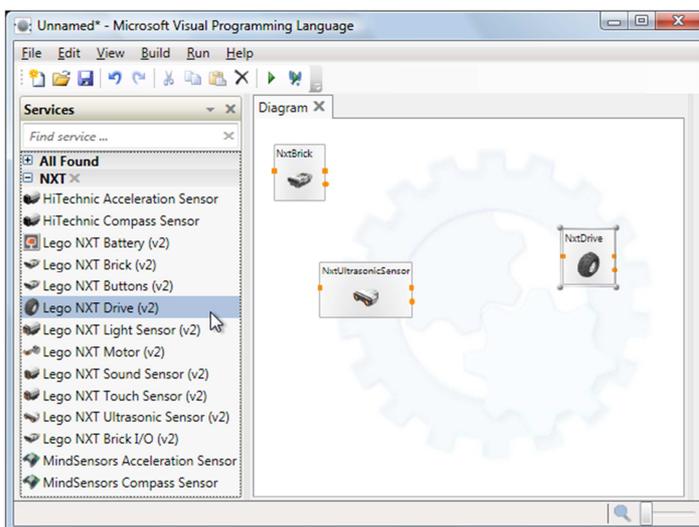
Configurez le capteur à ultrasons

Exemple : Capteur positionné sur l'entrée 4 de la brique NXT.



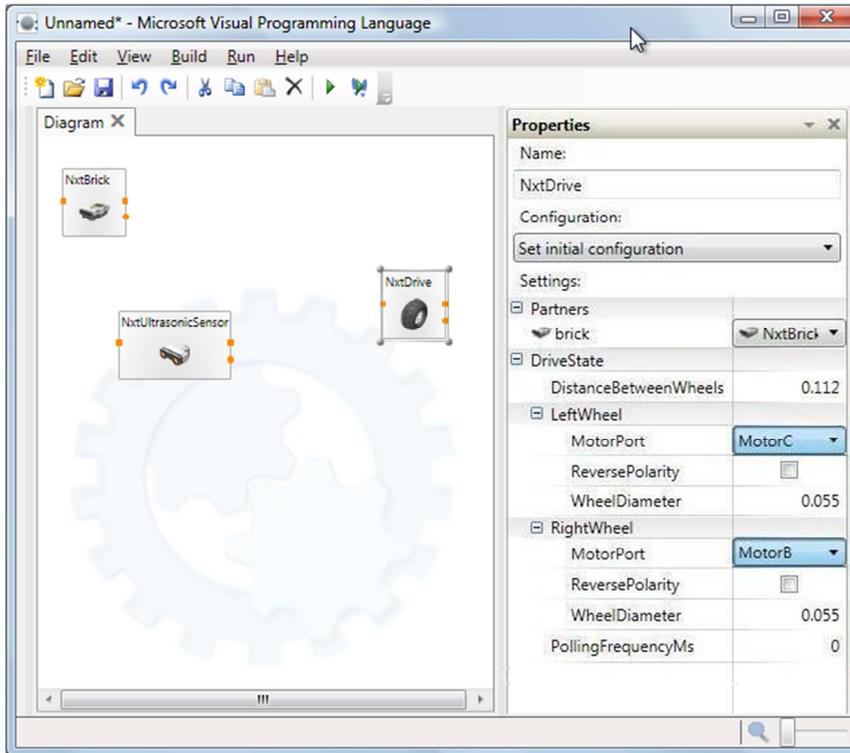
- Mise en place du service « Lego NXT Drive (v2) »

Ajoutez la commande des moteurs en mode différentiel (Lego Nxt Drive v2)



Paramétrez le service (Lego Nxt Drive v2) en spécifiant :

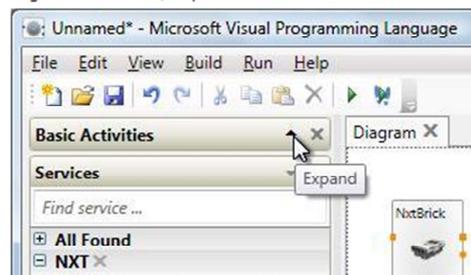
- La distance entre les roues (en m)
- Le diamètre des roues (en m)
- Les ports associés aux roues
- PollingFrequencyMs = 0 par défaut



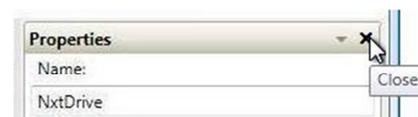
- Programmation de la logique d'évitement d'obstacles

Cette partie est largement détaillée dans l'aide en ligne de MSRDS. Un copier-coller du Tutorial : Ultrasonic Explorer MSDRS est donné ci-dessous.

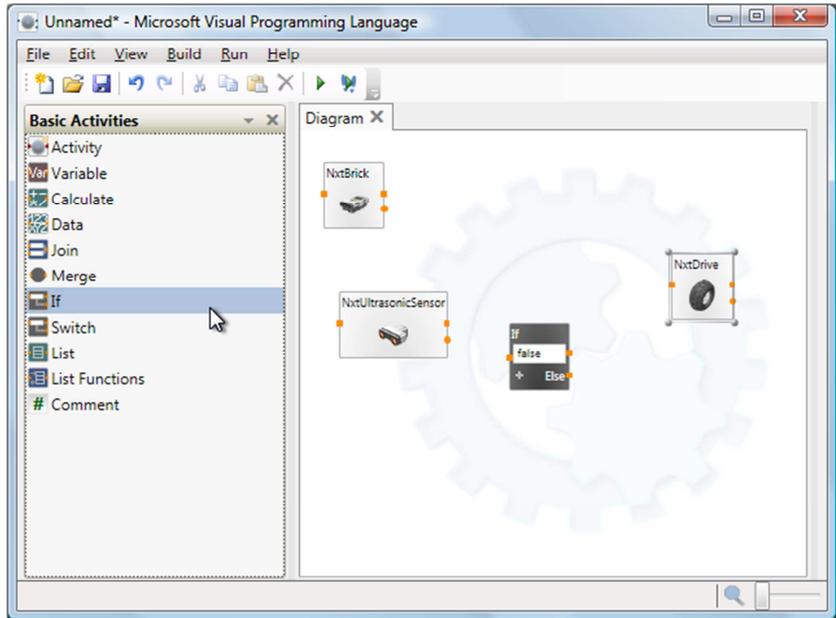
Adding logic to the diagram is straight forward but you may need to make more room for the diagram. To do this, expand the **Basic Activities** window:



You can also close the **Properties** window:

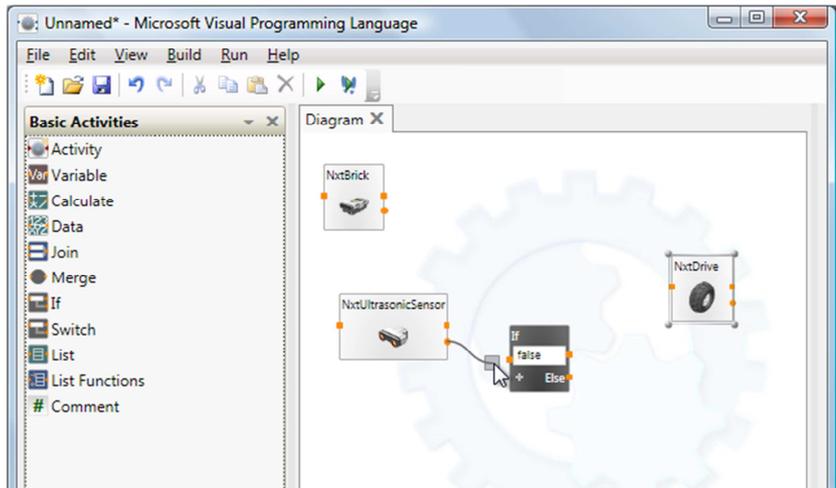


From the **Basic Activities** window, drag an **If**, block on to your diagram.

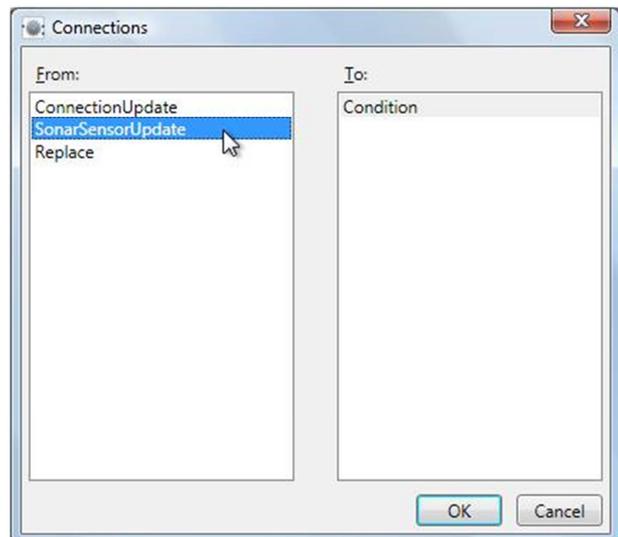


Next you will capture notifications from the **NxtUltrasonicSensor** every time a Distance reading is received. You will use this value to determine how to drive the motors.

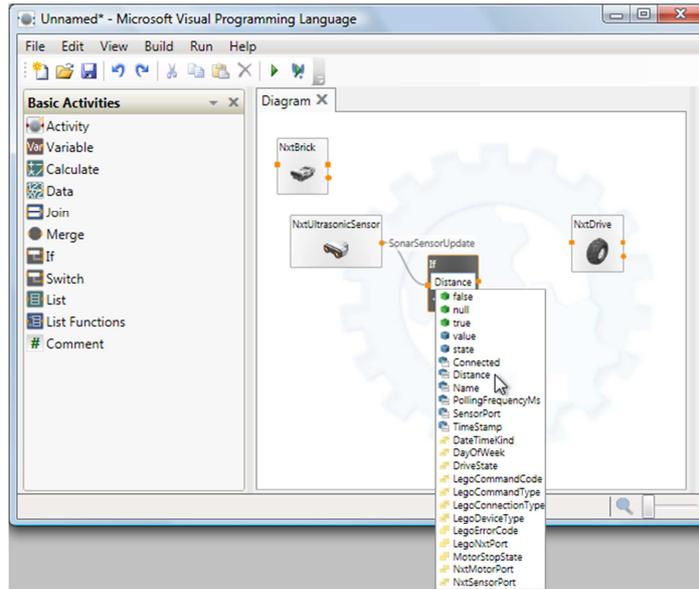
Select the **Round Notification** pin on the **NxtUltrasonicSensor** and drag it on to the **If**, block. This indicates that you will capture notifications from the Touch Sensor and evaluate them.



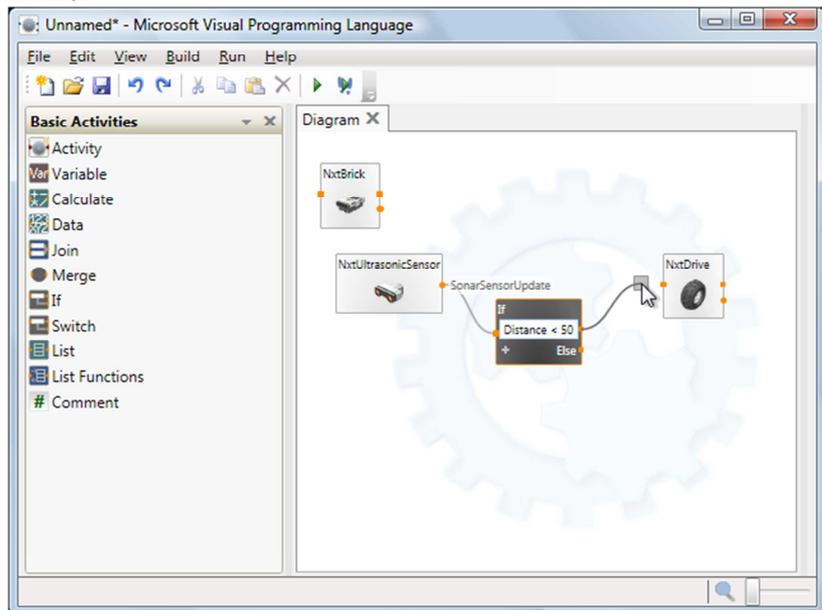
This will open the Connections dialog. Select **From the SonarSensorUpdate** notification and click **OK**.



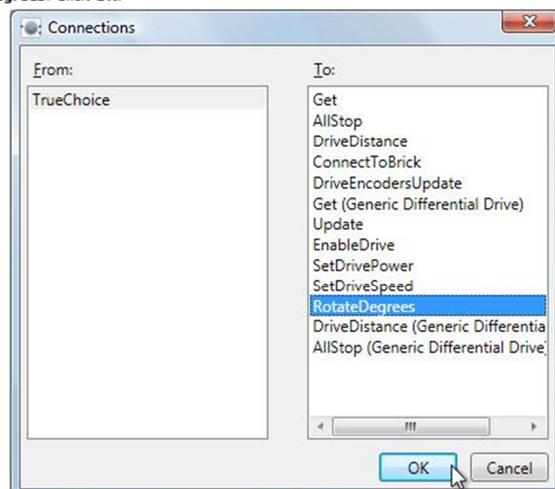
In the **If** expression box, type a space. Notice that a drop-down dialog opens, indicating the values which are available from the sensor. Select or type, **Distance < 50**. This means, when the distance is less than 50 cm.



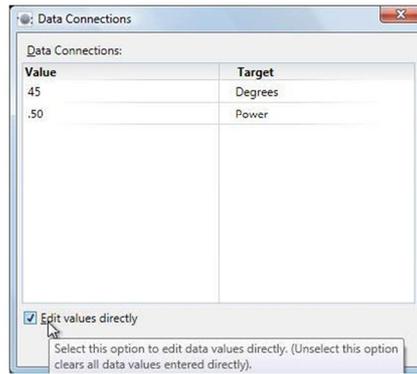
Drag from the pin to the right of the expression over to the **NxtDrive** activity.



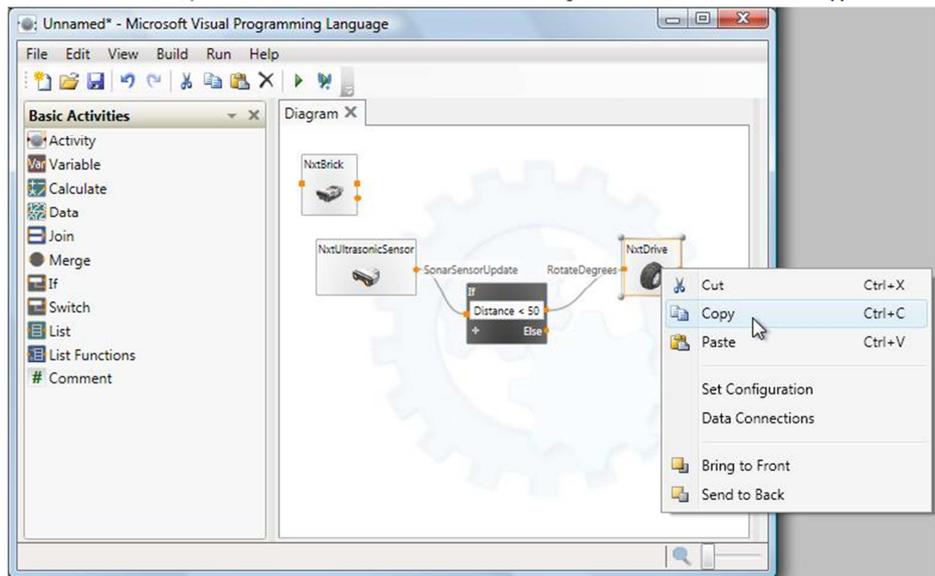
This will open the **Connections** dialog. Select **From TrueChoice** and **To the RotateDegrees**. Click **OK**.



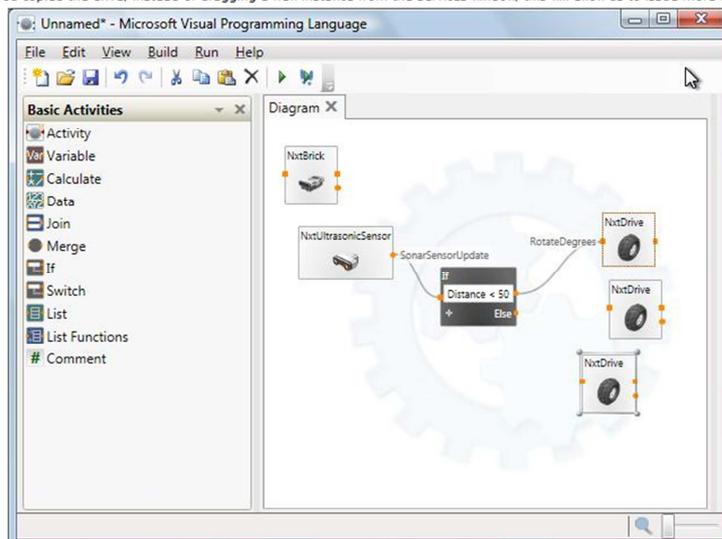
This will open the Data Connections for Rotate Degrees. Select, [x] Edit values directly, then type in, 45 Degrees and .50 Power. Click **OK**. Assuming no further drive commands are issued, this will turn the Tribot 45 degrees to the left, or counter-clockwise.



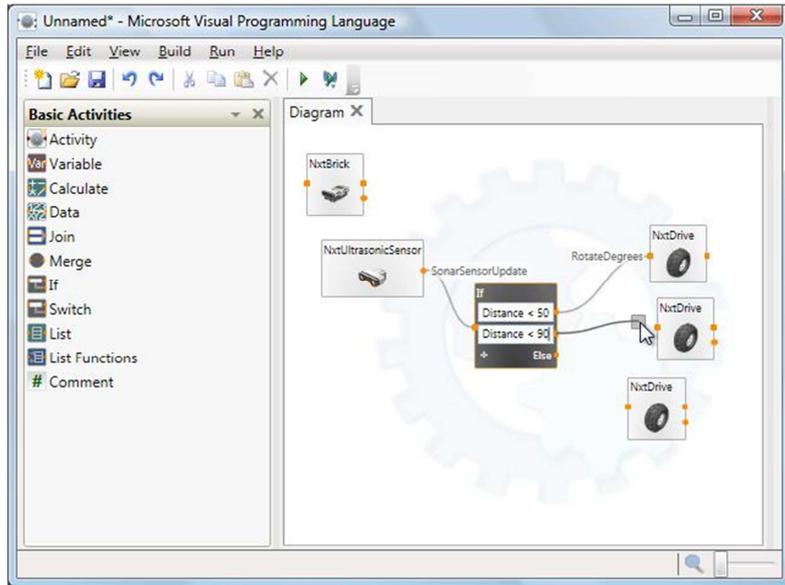
The obstacle avoidance is not sufficient to exploring the environment. To do this, you want to use the same **NxtDrive** two more times. Right-click on the **NxtDrive** and select **Copy**.



Paste two times, use **Ctrl-V** or **Edit/Paste** from the menu. Since you copied the drive, instead of dragging a new instance from the services window, this will allow us to issue more commands to the same drive.



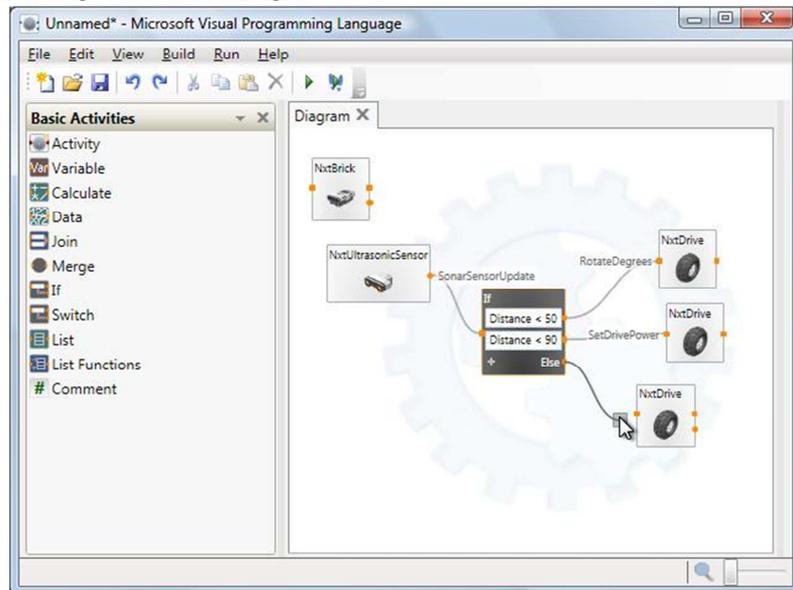
Drag a connection from the second condition, Distance < 90, to the second **NxtDrive**.



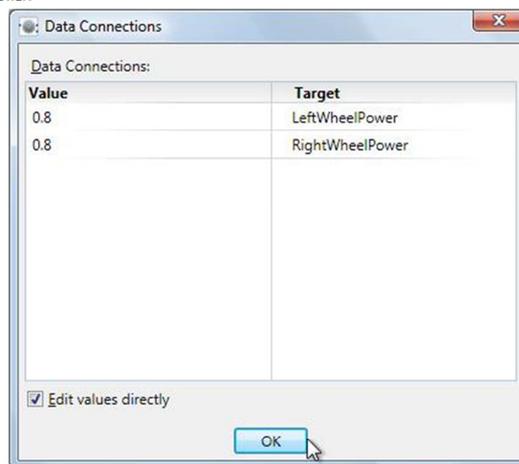
This time, select **SetDrivePower** and enter, 0.3, for **LeftWheelPower** and 0.6, for **RightWheelPower**. When an obstacle is between 50 and 90 cm, this will cause the Tribot to drive in an arc to the left.

Value	Target
0.3	LeftWheelPower
0.6	RightWheelPower

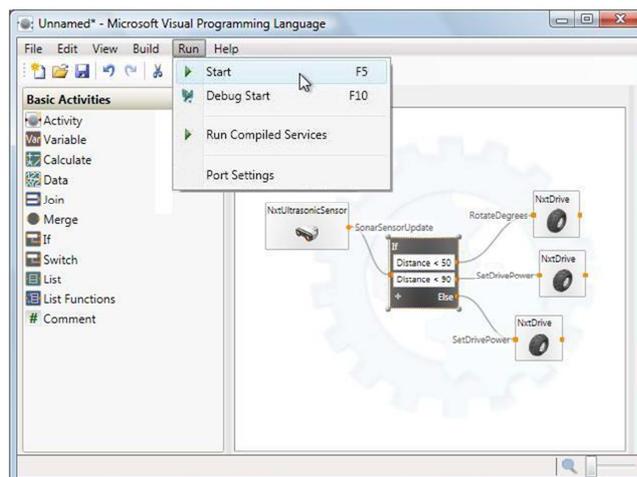
Finally connect the, **Else**, pin to the third **NxtDrive**. In the **Connections** dialog, select, **SetDrivePower**, again.



When no objects are less than 90 cm from the Tribot, blast straight ahead at 80% power.



Now you are ready to run. Make sure your Tribot is turned on, then set it on the floor and select **Run / Start** from the VPL menu. If all goes well your Tribot will run around and every time it gets near an obstacle, it will turn and go in a different direction.



La documentation en ligne de VPL propose des exemples et des tutoriaux sur le langage VPL et le Lego NXT.

Annexe : Première utilisation d'un robot Lego NXT

1) Mettre à jour son software si nécessaire (V1.29 au 06/11) à partir du logiciel « Lego NXT Mindstorms ».

→ Outils

→ Mettre à jour le microprogramme NXT

→ Mise à jour en ligne

→ Vérifier

Sur le site, sélectionner produit, puis mindstorms.

Sur le site anglais sélectionner support puis files et firmware.

2) Configurer la liaison Bluetooth entre le PC et le NXT

2.1) Installer une clé Bluetooth si nécessaire

2.2) Etablir la connexion (voir le chapitre introduction à Lego Mindstorms NXT dans l'aide en ligne de MSRDS (Microsoft Robotics Developer Studio -> Supported Robots)

Webographie

Présentation de MSRDS sur Wikipédia(06/11)

http://fr.wikipedia.org/wiki/Microsoft_Robotics_Developer_Studio

Installation des outils logiciels (06/11)

<http://msdn.microsoft.com/fr-fr/express/aa718373>

<http://msdn.microsoft.com/fr-fr/robotics/aa731520%28en-us%29.aspx>

Un environnement virtuel soigné et un outil de démonstration dans le domaine du virtuel et du réel de la société SympliSim (NXT MSRDS-R3)

<http://www.simplysim.net/index.php?p=gallery&id=11>

Le site de MSRDS

<http://www.microsoft.com/robotics/#About>

Revendeur Lego Mindstorms NXT

<http://www.generationrobots.com/index.cfm>

pour aller plus loin...

Article : Commande Lego en C# ou VPL

<http://channel9.msdn.com/coding4fun/articles/Microsoft-Robotics-Studio-and-Lego-Mindstorms-NXT>