**GUIDE DE SEANCE**

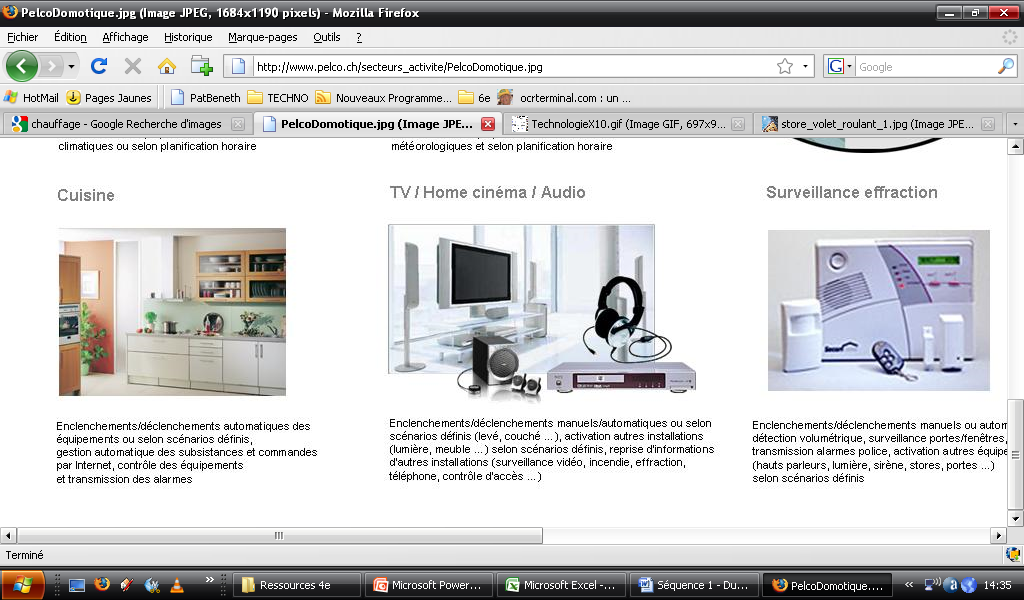
**Niveau : 4e**

**SEQUENCE 3**

*Comment l’objet technique est-il programmé ?*

**4 séances**

Confort et Domotique

***Séance n°1***

*« Si* ***Wall-E©*** *fonctionne de manière autonome et réalise des tâches, c’est parce qu’il a été programmé pour le faire. Qu’est-ce que la programmation ? Comment programmer ? »*

|  |  |  |
| --- | --- | --- |
| **CAPACITES** | **NIVEAUX** | **CONNAISSANCES** |
| Traitement du signal :  algorithme, organigramme, programme. | 1 | Identifier les étapes d’un programme de commande représenté sous forme graphique. |
| Traitement du signal :  algorithme, organigramme, programme. | 2 | Modifier la représentation du programme de commande d’un système pour répondre à un besoin particulier et valider le résultat obtenu. |
| 1. Commande d’un objet technique et logique combinatoire de base : ET, OU, NON. | 2 | Identifier une condition logique de commande. |

|  |  |
| --- | --- |
| **COMPETENCES B2i mises en œuvre** | |
| 1.1 | Je sais m'identifier sur un réseau ou un site et mettre fin à cette identification. |
| 4.5 | Je sais sélectionner des résultats lors d'une recherche (et donner des arguments permettant de justifier mon choix). |

|  |  |
| --- | --- |
| **COMPETENCES SOCLE COMMUN mises en œuvre** | |
| 1 | Maîtrise de la langue française |
| 4 | Maîtrise des techniques usuelles de l'information et de la communication |
| 7 | Autonomie et initiative |

***Problème posé :***

Comment faire pour que le robot aspirateur réalise sa fonction de nettoyage quelque soit l’environnement et la surface à parcourir ?

***Déroulement de la séance****(environ 1h20)* ***:***

5’

1. Présenter la situation déclenchante :

* Les élèves reformulent la situation pour se l’approprier ;
* Les élèves formalisent la problématique ;

20’

1. Investigation n°1 :

* A partir d’une représentation graphique proposée (parcours simple), du robot en fonctionnement (logiciel **RobotProg**) et avec l’aide du tutoriel, les élèves repèrent les différentes actions effectuées par le robot ;
* Les élèves repèrent également les informations qui permettent l’enchainement des actions ;

|  |  |
| --- | --- |
| Exemple de représentations graphiques | |
| Représentation sous forme d’algorithme | Représentation sous forme de grafcet |

***Ce qui est attendu*** *(pistes)****:***

* Identification des étapes ***(actions associées)*** du programme et la façon dont elles sont traitées ***(événements de mise en service et d’arrêt).***
* Lecture d’un programme simple et de sa représentation graphique: (organigramme ; algorithme ; grafcet) ;
* Reconnaissances des symboles définissant les étapes et actions associées ;
* Identification des ***conditions d’évolutions*** de fonctionnement entre les étapes ;
* …

30’

1. Investigation n°2 :

* Que se passe t-il si on place un obstacle dans l’environnement du robot ? Le programme permet-il de répondre au nouveau besoin ?
* Modifier le programme pour que le robot réagisse en fonction de la rencontre d’un obstacle dans son environnement ;
* Tester le programme modifié ;

***Ce qui est attendu*** *(pistes)****:***

* La modification du programme pour obtenir un parcours sans blocage du robot. Cette modification de programme peut engendrer une modification de structure d’écriture et de représentation du programme : ***structure linéaire ou structure en boucle*** ;

25’

1. Bilan de fin de séance :

* Présentation des solutions proposées par les différents îlots ;
* Mise en évidence qu’il n’y a pas de solution unique à la réalisation d’un programme ;
* Lecture d’une représentation graphique d’un programme avec :
  + *Identification de la structure du programme*
  + *Identification des actions.*
  + *Identification des conditions d’évolutions entre les actions.*
* Mise en évidence des informations qui permettent l’enchainement du programme sont souvent des conditions liées à l’état d’un capteur, une valeur attendue, une position ou tout simplement le résultat vrai ou faux d’une condition ;
* Mise en évidence des conditions logiques dans le programme (OUI, NON, ET, OU)
* Mise en évidence des différentes structures possibles (une seule fois ; répétitive ; en boucle) ;
* ***…***

|  |  |
| --- | --- |
| ***Différentes structures*** | |
| linéaire | En boucle |

|  |  |
| --- | --- |
| ***Conditions d’évolutions*** | |
| Une fonction logique NON | Test à plusieurs fonctions logique |

***Ressources :***

* <http://www.physicsbox.com/Tutoriel%20RobotProg/tutoriel.html>
* Logiciel « Robotprog » : <http://www.physicsbox.com/indexrobotprogfr.html>
* Fichiers de travail pour le logiciel « Robotprog » ;

***Auto-évaluation[[1]](#footnote-2)****(à remplir en fin de séance)****:***

|  |  |  |
| --- | --- | --- |
| ***A la fin de la séance, je suis capable de …***  *(je fais une croix dans la case qui correspond à ce que je me sens capable de faire)* | ***seul*** | ***avec de l’aide*** |
| Identifier les étapes d’un programme de commande représenté sous forme graphique. |  |  |
| Modifier la représentation du programme de commande d’un système pour répondre à un besoin particulier et valider le résultat obtenu. |  |  |
| Identifier une condition logique de commande (ET, OU, NON). |  |  |

***Séances n°2 et 3***

*« Notre aspirateur-robot doit à présent être programmé pour être capable d’effectuer sa tâche. »*

|  |  |  |
| --- | --- | --- |
| **CAPACITES** | **NIVEAUX** | **CONNAISSANCES** |
| Traitement du signal :  algorithme, organigramme, programme. | 1 | Identifier les étapes d’un programme de commande représenté sous forme graphique. |
| Traitement du signal :  algorithme, organigramme, programme. | 2 | Modifier la représentation du programme de commande d’un système pour répondre à un besoin particulier et valider le résultat obtenu. |
| 1. Commande d’un objet technique et logique combinatoire de base : ET, OU, NON. | 2 | Identifier une condition logique de commande. |

|  |  |
| --- | --- |
| **COMPETENCES B2i mises en œuvre** | |
| 1.1 | Je sais m'identifier sur un réseau ou un site et mettre fin à cette identification. |
| 4.5 | Je sais sélectionner des résultats lors d'une recherche (et donner des arguments permettant de justifier mon choix). |

|  |  |
| --- | --- |
| **COMPETENCES SOCLE COMMUN mises en œuvre** | |
| 1 | Maîtrise de la langue française |
| 4 | Maîtrise des techniques usuelles de l'information et de la communication |
| 7 | Autonomie et initiative |

***Problème posé :***

Comment faire pour que le robot aspirateur réalise sa fonction de nettoyage quelque soit l’environnement et la surface à parcourir ? *(Il s’agit de réaliser quelques séquences de programmation répondant au besoin futur de fonctionnement.)*

***Déroulement des séances****(environ 2h40)* ***:***

5’

1. Présenter la situation :
   * Les élèves reformulent la situation pour se l’approprier ;
   * Les élèves formalisent la problématique ;

2h

1. Investigation:

* Etudier les programmes donnés en exemple ;
* Ecrire différents programmes d’essais à l’aide de blocs fonctions[[2]](#footnote-3) :
  + - *Faire déplacer le robot en même temps qu’il aspire*
    - *Déplacer le robot jusqu’à la rencontre d’un objet (capteur tactile)*
    - *Déplacer le robot et l’arrêter avant l’objet (capteur ultrasons)*
    - *Déplacer le robot et l’arrêter avant qu’il ne tombe (escalier, fin de surface)*
    - *Faire arrêter et reculer le robot devant un objet*
    - *Orienter le robot dans une direction libre d’objet à moins d’une distance définie.*
* Raccorder les composants capteurs et moteurs à l’unité centrale ;
* Transférer les programmes dans l’unité centrale et tester;

|  |
| --- |
| Voici des programmes *à structure linéaire où les actions et informations sont traitées l’une à la suite de l’autre*. La description graphique de ce genre de programme devient vite volumineuse. |
| Exemple de programme: ***Déplacement du robot en marche avant*** |
|  |
| Exemple de programme: ***Tourner à droite, arrêt, attente, tourner à gauche*** |
|  |

|  |
| --- |
| Les programmes avec ***boucle*** permettent de réaliser une ou ***des actions permanentes tant qu’une condition n’est pas acquise*** (information de capteur, résultat de comptage, information de temps).  La boucle offre l’avantage de limiter les suites répétitives d’actions. |
| Exemple de programme en boucle: ***Le robot tourne en permanence*** |
|  |
| Exemple de programme avec boucle: ***Le robot tourne par angle défini jusqu’à détection d’une distance > à une valeur*** |
|  |
| Exemple de programme avec boucle: ***Le robot avance jusqu’à une distance définie puis il recule durant un temps défini*** |
|  |

|  |
| --- |
| Dans la structure d’un programme il est parfois nécessaire de créer ***une divergence ou commutation*** selon le résultat d’un test, de façon à effectuer une ou plusieurs actions différentes en fonction de ce résultat.  ***Il s’agit de structure à divergence en OU.*** |
| Exemple de programme avec divergence ou séquence commutation NXT |
|  |

|  |
| --- |
| Cette structure de programme permet d’envisager que le robot fasse ***plusieurs actions en même temps*** ou simultanément. Il s’agit d’une ***structure à divergence en ET***. |
| Exemple de programme avec séquences simultanées |
|  |

***Ce qui est attendu :***

* + Lire des programmes simples et comprendre leur fonctionnement à leur lecture ;
  + Différencier une structure linéaire, d’une structure divergente qu’elle soit en ***ET (actions simultanées)*** ou ***en OU (actions différentes selon la condition) ;***
* Ecrire différents programmes d’essais à l’aide de blocs fonctions en utilisant la ou les structures adaptées :

*- Faire déplacer le robot en même temps qu’il aspire ;*

*- Déplacer le robot jusqu’à la rencontre d’un objet (capteur tactile) ;*

*- Déplacer le robot et l’arrêter avant l’objet (capteur ultrasons) ;*

*- Déplacer le robot et l’arrêter avant qu’il ne tombe (escalier, fin de surface) capteur de luminosité ;*

*- Faire arrêter et reculer le robot devant un objet ;*

*- Orienter le robot dans une direction libre d’objet à moins d’une distance définie ;*

*- …*

* Utilisation rationnelle du logiciel de programmation (cohérence avec le besoin) ;
* Transmission des programmes dans l’unité centrale et tests;

15’

1. Bilan de fin de séance :

* Présentation du travail réalisé et des solutions proposées par les différents îlots ;
* Mise en évidence de la complexité de la programmation de l’aspirateur-robot ;
* Possibilité de réaliser des sous-programmes ;

***Ressources :***

* Logiciel « Mindstorms NXT » ;
* Fichiers de travail pour le logiciel « Mindstorms NXT » ;
* Manuel d’utilisation des LEGO Mindstorms NXT (PDF ou papier) ;

***Auto-évaluation1****(à remplir en fin de séance)****:***

|  |  |  |
| --- | --- | --- |
| ***A la fin de la séance, je suis capable de …***  *(je fais une croix dans la case qui correspond à ce que je me sens capable de faire)* | ***seul*** | ***avec de l’aide*** |
| Identifier les étapes d’un programme de commande représenté sous forme graphique. |  |  |
| Modifier la représentation du programme de commande d’un système pour répondre à un besoin particulier et valider le résultat obtenu. |  |  |
| Identifier une condition logique de commande (ET, OU, NON). |  |  |

***Séance n° 4***

*« Notre aspirateur robot peut se déplacer et aspirer en même temps. Il faut maintenant qu’il effectue sa tâche dans une pièce »*

|  |  |  |
| --- | --- | --- |
| **CONNAISSANCES** | **NIVEAUX** | **CAPACITES** |
| Traitement du signal :  algorithme, organigramme, programme. | 1 | Identifier les étapes d’un programme de commande représenté sous forme graphique. |
| Traitement du signal :  algorithme, organigramme, programme. | 2 | Modifier la représentation du programme de commande d’un système pour répondre à un besoin particulier et valider le résultat obtenu. |
| 1. Commande d’un objet technique et logique combinatoire de base : ET, OU, NON. | 2 | Identifier une condition logique de commande. |
| Processus de réalisation (fabrication, assemblage, **configuration**) d’un objet technique. | 3 | Réaliser tout ou partie du prototype ou de la maquette d’un objet technique. |
| Contraintes liées aux procédés et  modes de fabrication :  - formes possibles ;  - précision accessible.  Contraintes liées aux procédés de contrôle et de validation. | 2 | Préparer un protocole de test et/ou de contrôle en fonction des moyens disponibles. |

|  |  |
| --- | --- |
| **COMPETENCES B2i mises en œuvre** | |
| 1.1 | Je sais m'identifier sur un réseau ou un site et mettre fin à cette identification. |
| 4.5 | Je sais sélectionner des résultats lors d'une recherche (et donner des arguments permettant de justifier mon choix). |

|  |  |
| --- | --- |
| **COMPETENCES SOCLE COMMUN mises en œuvre** | |
| 1 | Maîtrise de la langue française |
| 4 | Maîtrise des techniques usuelles de l'information et de la communication |
| 7 | Autonomie et initiative |

***Problème posé :***

Notre robot peut-il parcourir seul et sans blocage l’ensemble de la pièce ? Comment remédier aux problèmes rencontrés ?

***Déroulement de la séance****(environ 1h20)* ***:***

5’

1. Présenter la situation :
   * Les élèves reformulent la situation pour se l’approprier ;
   * Les élèves formalisent la problématique ;

55’

1. Investigation:

* A l’aide du prototype de robot LEGO mis à disposition (programme complet en mémoire), élaborer un protocole de test de fonctionnement ;
* Réaliser les tests prévus ;
* Repérer les problèmes ;
* Modifier la solution pour résoudre les problèmes rencontrés ;

***Ce qui est attendu :***

- Protocoles de test cohérents *(fiches de test, parcours de test, …)* ;

- Utilisation rationnelle du matériel et du logiciel ;

- Identification des problèmes ;

- Propositions de solutions aux problèmes rencontrés *(repositionnement des capteurs, modification du*

*programme, forme du prototype, …)* ;

20’

1. Bilan de fin de séance :

* Présentation du travail réalisé et des solutions proposées par les différents îlots ;
* Enumération des différents problèmes rencontrés ;
* Pour un même problème, on remarque que les solutions peuvent être multiples *(Exemple : en cas de chute dans un escalier : modification du programme ou modification de la forme du prototype, …)* ;
* Pour plusieurs problèmes, la solution peut être unique *(Exemple : en cas de collision avec un obstacle et d’une mauvaise mesure de la distance : modification de la position du capteur à ultrasons)* ;

***Ressources :***

* Logiciel « Mindstorms NXT » ;
* Fichiers de travail pour le logiciel « Mindstorms NXT » ;
* Manuel d’utilisation des LEGO Mindstorms NXT (PDF ou papier) ;

***Auto-évaluation1****(à remplir en fin de séance)****:***

|  |  |  |
| --- | --- | --- |
| ***A la fin de la séance, je suis capable de …***  *(je fais une croix dans la case qui correspond à ce que je me sens capable de faire)* | ***seul*** | ***avec de l’aide*** |
| Identifier les étapes d’un programme de commande représenté sous forme graphique. |  |  |
| Modifier la représentation du programme de commande d’un système pour répondre à un besoin particulier et valider le résultat obtenu. |  |  |
| Identifier une condition logique de commande (ET, OU, NON). |  |  |
| Modifier une partie du prototype d’un objet technique pour répondre à un besoin. |  |  |
| Préparer un protocole de test et/ou de contrôle en fonction des moyens disponibles. |  |  |

***Eléments de synthèse active*** *(Pistes)* ***:***

Quelque soit le type de programmation d’un objet technique, on y trouve des actions et des conditions.

Les conditions doivent être validées pour passer à l’action suivante.

Les conditions sont souvent liées à l’état d’un capteur qui va renvoyer une valeur (obstacle détecté, son perçu, …).

La structure d’un programme peut être linéaire (en ligne) ou former des boucles, des répétitions.

Une boucle permet de faire réaliser une action tant qu’une condition est (ou n’est pas validée). Exemple : tant que le robot ne détecte pas d’obstacle, il avance d’une case ;

Dans une structure, on rencontre des **conditions logiques** :

* Lorsqu’il y a un choix, une question, cette condition logique est un **« ou »** (c’est l’une **ou** l’autre des solutions). Exemple : y a-t-il un mur en face ? La réponse sera Oui **ou** Non.. Ca ne peut pas être Oui et Non;
* Lorsque 2 conditions doivent être validées en même temps pour passer à l’action suivante, c’est un **« et »** (c’est l’une **et** l’autre). Exemple : s’il n’y a pas de mur en face **et** qu’il n’y a pas d’escalier, alors le robot continue d’avancer. Si l’une des 2 conditions n’est pas vérifiée, c’est que le robot risque d’entrer en collision avec un mur ou qu’il risque de tomber dans les escaliers;

Chaque fonction peut être programmée séparément. On appelle cela un sous-programme. En assemblant les sous-programmes, on peut obtenir le programme complet.

On peut représenter le fonctionnement à l’aide d’une description graphique simple :

* + Type algorithme
  + Type organigramme
  + Grafcet
  + Littérale (texte)

1. Le document d’auto-évaluation est distribué en début de séquence. [↑](#footnote-ref-2)
2. Les différentes fonctions peuvent être réparties entre les îlots pour la programmation et les tests. [↑](#footnote-ref-3)