

Robot « idiot » (LEC CM- 6^{ème})

I Généralités :

1) Objectifs pédagogiques :

Cette activité **d'algorithmique débranchée** a été réalisée dans le cadre de la Liaison Ecole Collège (LEC).

Elle est inspirée d'un travail présenté par Marie Duflot sur le site [pixees](https://pixees.fr/dis-maman-ou-papa-cest-quoi-un-algorithme-dans-ce-monde-numerique-%E2%80%A8/) (<https://pixees.fr/dis-maman-ou-papa-cest-quoi-un-algorithme-dans-ce-monde-numerique-%E2%80%A8/>)

Objectifs : faire découvrir la notion d'algorithme et les principes qui y sont associés (bog/debogage ; multiplicité des solutions de programmation ; optimisation ; réutilisation de bouts de programmes....)

2) En pratique :

Idée générale : donner des instructions à un robot pour qu'il sorte d'un labyrinthe matérialisé par des plots.

Conditions matérielles :

- Un espace extérieur à la classe (salle polyvalente, gymnase, préau, cour de récréation...)
- Pour matérialiser un labyrinthe :
 - ❖ Plots plastiques (petites assiettes en carton si on ne travaille pas en extérieur) : 36 plots pour un labyrinthe (grille 5x5),
 - ❖ Un objet pour matérialiser une clé
 - ❖ 8 cordes/bâtons/vêtements pour matérialiser les murs
- Formation de groupes de 4 élèves : 2 élèves de CM et 2 élèves de 6°
- Eventuellement : ardoise, feutre, stylo pour faciliter la recherche écrite des algorithmes

Prévoir 2 labyrinthes pour une classe, soit 4 labyrinthes pour une rencontre CM-6° : cela permet plus de souplesses pour les vérifications *in situ* des algorithmes.

Attention : prévoir 10 minutes pour la matérialisation des labyrinthes (sans les murs ou la clé) et la constitution de groupes.

II Première partie : introduction

1) Conditions :

Conditions : en plénière ; les élèves regroupés par groupe, sont assis devant un labyrinthe.

2) Déroulement :

- Présentation de l'activité (sur un labyrinthe vierge)

Le robot idiot ne comprend que trois consignes (mimé en même temps par le professeur):

- *avancer d'un pas (pour la programmation cette action sera notée A)*
- *tourner d'un quart de tour vers la droite (pour la programmation cette action sera notée D)*
- *tourner d'un quart de tour vers la gauche (pour la programmation cette action sera notée G)*

On peut faire remarquer la nécessité d'un langage commun si on veut pouvoir à la fois être efficace et se comprendre d'un groupe à l'autre.

On montre les mouvements que ne peut pas faire le robot (exemple : un pas vers la droite, ou en diagonale...)

Pourquoi dire que ce robot est « idiot » ? Car il ne réfléchit pas, il ne fait qu'exécuter les ordres qu'on lui donne...

On demande un volontaire élève pour jouer le robot idiot : on interroge différents élèves qui à chaque fois donnent un ordre que le robot exécute. Eventuellement, renouveler l'expérience avec un autre élève volontaire pour faire le robot idiot...

Un élève cobaye teste les consignes de ses camarades (zone vierge sans obstacles).

Distribution de la fiche d'activités.

Question 1 :

Objectif : faire comprendre comment vérifier un programme, c'est-à-dire travailler le « pas à pas ».

Un élève lit la première question de la partie 1.

Le professeur se positionne dans la zone Q pour montrer aux élèves la position de départ du robot. Il laisse un temps de réflexion aux groupes (quelques minutes), puis pour la mise en commun, il va mettre les élèves dans la dynamique de groupe attendue d'eux pour les parties 2, 3 et 4 à savoir : il invite un élève à le rejoindre pour jouer le robot, un autre pour lui dicter les autres et un autre pour contrôler les déplacements du robot.

On en profite pour montrer comment simplifier le code : ainsi AAA peut se noter A3. Cela permet d'aborder de faire un tissage pour la programmation future sous scratch : « avance de ... ».

On peut ainsi aussi compter le nombre d'instructions (voir question2).

Question 2 :

Objectifs :

- faire comprendre qu'il y a plusieurs programmes possibles pour arriver à une même sortie
- notion d'optimisation du programme : avoir le moins d'instructions possibles

Déroulement :

Un élève lit la première question de la partie 2.

Le professeur laisse un temps de réflexion aux groupes (quelques minutes).

Pour la mise en commun :

Il interroge un élève sur le nom de la sortie du robot.

Pour la vérification, un élève vient jouer le robot, un autre lui dicte les autres et un autre contrôle les déplacements du robot.

Question 3 :

Objectifs :

- faire comprendre quel peut être le facteur d'erreur en programmation (notion de bog)
- notion de débogage (travail par essais/erreurs)
- montrer la différence entre déplacement absolu et déplacement relatif

Déroulement :

Un élève lit la première question de la partie 3. Il s'agit de vérifier un programme supposé permettre de sortir par la zone J (ce qui n'est pas le cas).

Le professeur laisse un temps de réflexion aux groupes (quelques minutes).

Pour la mise en commun :

Il interroge un élève sur le nom de la sortie du robot.

Pour la vérification, un élève vient jouer le robot, un autre lui dicte les autres et un autre contrôle les déplacements du robot.

Questionnements :

Est-ce que le robot sort par la sortie J ?

Pourquoi ça ne marche pas ? Erreurs du robot ? (non car un élève a vérifié les mouvements !)

Erreurs de programmation ? Si oui, quelle(s) instruction(s) modifier ?

L'erreur de programmation pointe ici un problème que risque de rencontrer certains groupes : le problème d'orientation absolue ou relative du robot. Le robot étant en position de descente, pour le faire aller vers la droite du labyrinthe, il faut lui dire de tourner à gauche !

Un élève propose la modification, et un nouveau test est fait pour entériner la réponse.

III Parties en autonomies :

1) Conditions :

Pour les parties 2, 3 et 4 les conditions sont les suivantes :

- Par groupe
- Durant la phase de réflexion les élèves regroupés par groupe, sont assis devant un labyrinthe.
- Pour vérifier leur programme, ils se lèvent et utilisent le labyrinthe.
- Le labyrinthe a été modifié : des « murs » ont été matérialisés

2) Partie 2 : sortir du labyrinthe

Objectif : proposer une suite d'instructions pour permettre au robot de sortir à la sortie prévue, et ceci en optimisant le nombre d'instructions

Déroulement :

Les élèves choisissent le niveau de difficultés auquel ils veulent travailler.

Différenciation proposée :

Niveau découverte : pas de problème d'orientation absolue ou relative car le robot est dans le sens de marche. Nombre maximum d'instructions : 5 (non communiqué aux élèves a priori)

Niveau standard : la gestion d'orientation absolue ou relative n'arrive que sur la fin du programme.

Nombre maximum d'instructions : 7 (non communiqué aux élèves a priori)

Niveau expert : la gestion d'orientation absolue ou relative est centrale car le robot n'est pas dans le sens de marche. Nombre maximum d'instructions : 9 (non communiqué aux élèves a priori)

Les élèves utilisent leur ardoise, ou la feuille distribuée pour rédiger, par groupe, la liste des instructions la plus courte possible (ils indiquent le nombre d'instructions utilisées).

Lorsqu'ils sont tous d'accord, ils vont vérifier sur le labyrinthe si leur programmation est correcte :

- ~ Un élève joue le robot
- ~ Un lui dicte les instructions, sous le contrôle d'un autre
- ~ Le dernier vérifie si les mouvements du robot sont conformes aux ordres donnés

Si la programmation est incorrecte, ils font le débogage de leur programme en restant sur le labyrinthe si nécessaire

Si la programmation est correcte, il appelle le professeur pour vérification finale (instructions et optimisation)

3) Partie 3 : collecter et sortir du labyrinthe

Le labyrinthe reste identique mais une clé a été déposée dans une des cases...

Objectif : Savoir réutiliser des bouts de programme existant afin de le réinvestir dans une nouvelle programmation.

Les élèves vont devoir proposer une suite d'instructions pour permettre au robot de récupérer la clé puis de sortir à la sortie prévue, et ceci en optimisant le nombre d'instructions

Déroulement : Le déroulement est identique à celui de la partie 2.

Une nouvelle instruction est rajoutée : « ramasser » ; elle est codée R. Le robot ne ramasse que ce qui est à ses pieds !

L'idée est que les élèves réinvestissent le programme fait en partie 2 pour rajouter le détour puis le ramassage de la clé : ils travaillent donc en priorité dans le même niveau de difficulté choisi en partie 2.

4) Partie 4 : lancer un défi aux autres !

Détermination de la mission. Les élèves choisissent :

- ~ le point de départ
- ~ le point de sortie
- ~ l'ajout d'objets à récupérer (en plus de la clé)

Les élèves doivent proposer une programmation la plus efficace possible, la tester sur le labyrinthe. Ils mettent ensuite au défi un autre groupe de leur choix : ils doivent trouver une programmation au moins aussi efficace que la leur ! Le groupe qui rédige une liste d'instructions la plus courte possible gagne !

5) Autres formes de différenciations envisageables

- Modification du labyrinthe : condamner certaines cases ; permettre au groupe de rajouter des murs : attention, il faut alors vérifier que le trajet existe bel et bien !
- Demander de récupérer les objets dans un certain ordre (exemple : d'abord la clé, puis le livre, puis l'ardoise...)
- Imposer que le robot ne peut porter qu'un objet à la fois (on pourra alors rajouter la commande « poser »)
- Chercher à récupérer tous les carrés verts
- Utiliser aussi les carrés violets : le robot ne doit pas s'arrêter sur une case contenant un carré violet...
- Imposer un nombre maximal d'instructions (il faut alors vérifier que c'est réalisable !)

IV Retours sur l'expérience menée :

Un grand intérêt de la part de tous les élèves CM comme 6°.

La différenciation envisagée leur a plu : certains ont testé plusieurs niveaux pour chaque partie (d'abord « standard » puis « expert »).

Ils se sont laissés prendre au jeu et ont beaucoup aimé lancer des défis aux autres groupes.

La durée nécessaire est de plus d'une heure (1h30) car le temps d'installation est nécessaire : si possible, installer les labyrinthes à l'avance (3 complets, et un vierge pour la démonstration de la partie1), afin d'optimiser cette durée.

En prolongement, une séance de programmation avec scratch serait très intéressante (voir documents fournis) : les élèves pourraient alors finaliser leur programmation en passant d'un langage à l'autre.