

La mission : algorithmique débranchée

Une séquence du projet *1,2,3... CODEZ !*

Résumé

Dans un premier temps, les élèves sont invités à donner des instructions pour piloter un véhicule à distance. Pour ce faire, ils définissent un langage de programmation et explorent la différence entre une langue naturelle et un langage. Ils découvrent également la notion de bug. Ils sont ensuite confrontés à la nécessité d'encoder un message avec des nombres, puis des informations en binaires. Enfin, ils appliquent leurs découvertes à l'encodage d'un court message textuel en binaire, puis au décodage d'un message en binaire qu'ils reçoivent.



Séance 1 – Comment diriger un véhicule à distance ?

Résumé	Les élèves doivent donner des instructions pour piloter un véhicule à distance. Pour ce faire, ils définissent un langage de programmation et explorent la différence entre une langue naturelle et un langage. Ils découvrent également la notion de bug.
Notions <i>(cf. scénario conceptuel, page 212-213)</i>	<p>« Machines »</p> <ul style="list-style-type: none">• Les machines qui nous entourent ne font qu'exécuter des ordres (instructions). <p>« Langages »</p> <ul style="list-style-type: none">• En informatique, on invente et on utilise des langages.• Pour donner des instructions à une machine, on utilise un langage de programmation, compréhensible à la fois par la machine et par l'être humain.• Un langage de programmation est différent d'une langue naturelle<ul style="list-style-type: none">– Il possède très peu de mots et de règles de grammaire– Il ne laisse place à aucune ambiguïté– Il est compréhensible par certaines machines.• Il existe de nombreux langages de programmation, adaptés à différents usages .• Un bug est une erreur dans un programme.• Un bug minime en apparence peut avoir des conséquences énormes.
Matériel	Pour la classe <ul style="list-style-type: none">• Fiche 28, page 220• Système de vidéoprojection• Une salle informatique, avec une connexion Internet (facultatif)
Lexique	Langage de programmation, instruction, bug
Durée	1 h 30 (éventuellement en 2 x 45 minutes)

Situation déclenchante – présentation du projet

L'enseignant explique à la classe que le projet consiste à simuler une mission d'exploration sur une planète lointaine. Pour le moment, la classe va préparer la mission : réfléchir à la façon dont on va se déplacer, communiquer... Dans un second temps, la classe va « jouer » la mission, à travers un programme de simulation que chacun va pouvoir créer (petit jeu vidéo).

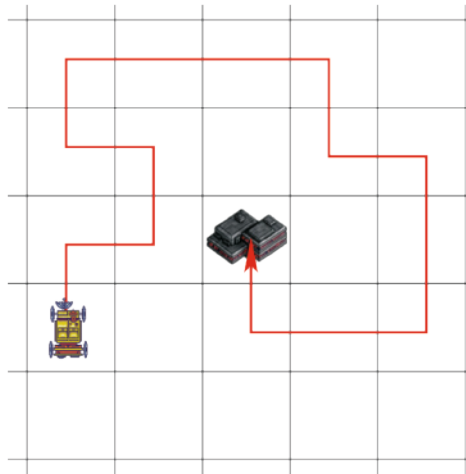
La mission est habitée et, sur la planète, l'équipage dispose déjà d'une base et d'un véhicule terrestre (de type « rover »). L'environnement est hostile, donc lors des sorties d'exploration une personne doit toujours rester à la base par sécurité. Si les personnes qui sont sur le terrain ne sont plus en mesure de piloter le rover (par exemple, si elles ont perdu connaissance), la personne d'astreinte doit pouvoir diriger le rover à distance pour le ramener à la base, sans avoir besoin de parler à l'équipage. Les ordres de déplacement sont donnés au rover sous forme d'ondes, mais il faut inventer un langage pour donner ces ordres.

La question est donc : quel langage utiliser pour piloter un rover à distance ?

L'enseignant affiche ou projette une carte de la zone d'exploration (Fiche 28). Cette zone est quadrillée et un parcours est dessiné de façon à ce que le rover puisse rentrer à la base en évitant les zones dangereuses. On ne peut pas faire de raccourci : il faut absolument suivre tout le parcours dessiné, dans le sens de la flèche.

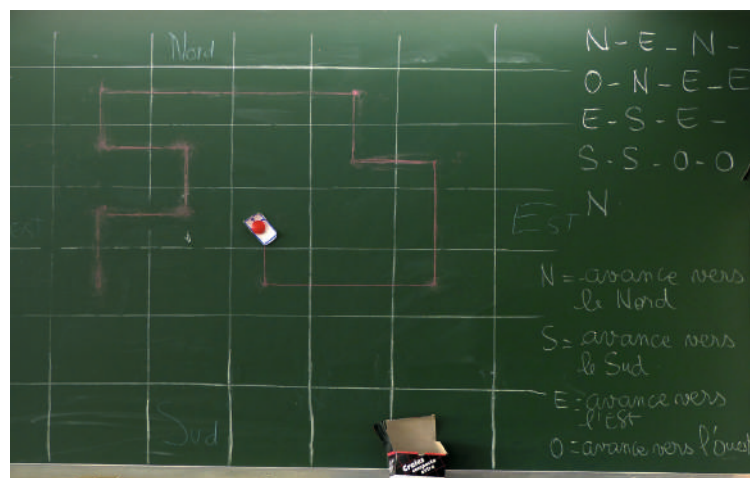
Recherche : définition d'un langage (par binômes)

L'enseignant distribue la Fiche 28 aux élèves, répartis par binômes. Ils doivent définir quels types d'instructions donner au rover pour lui faire suivre le parcours imposé afin de rejoindre la base. Les déplacements se font carreau par carreau, et pas en diagonale.



Mise en commun

Après quelques minutes, la classe met en commun les travaux des différents binômes. On peut, par exemple, dessiner ou projeter le parcours au tableau et, lorsqu'un binôme présente sa solution, vérifier collectivement qu'elle est correcte en l'exécutant au tableau (on prend un objet quelconque pour représenter le rover, cet objet doit suivre scrupuleusement les instructions données).



Classe de CM1 de Carole Vinel (Paris)

On remarque qu'il existe (au moins) deux logiques pour diriger le rover. On peut lui donner des directions « absolues » (va au nord, va à l'ouest...) ou, au contraire, des directions relatives, c'est-à-dire qui dépendent de l'orientation du rover (tourne vers la droite, avance, tourne vers la gauche, recule...). Remarquons qu'il est préférable de découper l'instruction « avance d'une case vers la droite » en 2 instructions bien distinctes : tourne vers la droite (sous-entendu : en restant sur place), puis avance d'une case.

Notes pédagogiques

- La première logique (Nord, Ouest...) est dite « allocentrée » tandis que la seconde logique (droite, gauche...) est dite « autocétrée ».
- Les élèves n'ont pas besoin de connaître ces termes, qui ne seront plus utilisés par la suite.
- Une troisième logique peut (plus rarement) être proposée : il s'agit de donner des coordonnées aux cases (A1, A2, B1...) et, comme dans un jeu de bataille navale, coder les déplacements en donnant le nom de la case de départ et d'arrivée. Exemple, « va de A1 vers A2 ». À noter : le chemin « A1 vers A2 » n'est pas ambigu car les cases sont adjacentes. En revanche, le chemin « A1 vers B7 » est ambigu (et, donc, non satisfaisant) : il y a plusieurs façons d'aller de la case A1 à la case B7. Nous ne détaillons pas cette méthode dans ce qui suit.

Il est probable que les 2 méthodes aient été trouvées par les uns ou les autres. Si cela n'est pas le cas, l'enseignant introduit l'autre méthode lors de cette mise en commun.

L'enseignant fait remarquer que les instructions sont exprimées dans un langage particulier, avec un vocabulaire très restreint, et non ambigu : chaque instruction est parfaitement explicite et ne peut pas donner lieu à plusieurs interprétations. Il s'agit d'un langage de « programmation ».

en haut - à droite - en haut - à gauche - en haut
à droite - en bas - à droite - en bas - à gauche -

↑ → ↗ ← ↖ → ↘ ↙ ↕ ↗ ↘ ↙ ↖ ↗ ↘ ↙ ↖ ↗ ↘ ↙ ↖

① = avancer
② = tourner à droite
③ = tourner à gauche
④ = reculer

tout droit, droite, tout droit, gauche, tout droit, droite, droite,
droite, bas, droite, bas, bas, gauche, gauche, tout droit,
TD, D, TD, G, TD, D, D, D, B, B, B, G, G, TD,
1PTD, 1PD, 1PTD, 1PG, 1PTD, 3PD, 1PB, 1PD, 2PB,
2PG, 4PTD.

Avancer d'un carreau.
Tourner à droite et avancer d'un carreau.
Tourner à gauche et avancer d'un carreau.
Tourner à gauche et avancer d'un carreau.
Tourner à droite et avancer d'un carreau.
Tourner à droite et avancer de trois carreaux.
Tourner à droite et avancer d'un carreau.
Tourner à gauche et avancer d'un carreau.
Tourner à droite et avancer de deux carreaux.
Tourner à droite et avancer de deux carreaux.
Tourner à droite et avancer d'un carreau.

Quelques propositions d'élèves (en haut à gauche, langage allocentré sans gestion de la longueur du déplacement; en haut à droite, langage allocentré mais les légendes sont autocétrées; en bas à gauche : évolution de la proposition d'un élève pendant la phase de recherche; en bas à droite, langage autocétré entièrement rédigé. Classe de CM2 d'Anne-Marie Lebrun (Bourg-la-Reine)

Le langage choisi peut encore être simplifié. Par exemple, il est inutile de dire « Va vers l'est » quand on peut simplement dire « Est » ou « Va à droite » quand on peut simplement dire « Droite » (si on a bien défini au préalable ce que l'on entend par « Droite », par exemple, « va d'une case vers la droite » et non pas « pivote sur toi-même d'un quart de tour vers la droite »).

Collectivement, la classe explicite le vocabulaire de 2 langages, par exemple :

Langage allocentré (ou « absolu »)	Langage autocentré (ou « relatif »)
<ul style="list-style-type: none"> • Nord (signifie « avance d'une case vers le nord ») • Sud • Est • Ouest 	<ul style="list-style-type: none"> • Avancer (signifie « avance d'une case droit devant toi ») • Droite (signifie « pivote sur place d'un quart de tour vers la droite ») • Gauche

On remarque que le langage allocentré nécessite 4 mots de vocabulaire tandis que le langage autocentré n'a besoin que de 3 mots. Certains élèves pourront proposer l'instruction « Recule », mais on peut remarquer que le rover se retrouve dans la même case s'il recule d'une case ou s'il fait « Droite, Droite, Avance ». Dans ce dernier cas, il a changé d'orientation. Si on souhaite qu'il reprenne son orientation initiale, il faut écrire « Droite, Droite, Avancer, Droite, Droite ».

On remarque également qu'il est possible de réduire encore le lexique de ce langage autocentré. « Gauche » par exemple peut se dire « Droite, Droite, Droite ». Ainsi, 2 mots peuvent suffire. Pour plus de clarté, on peut décider de garder 3 ou 4 mots, selon ce qui sera décidé par les élèves.

L'enseignant fait remarquer que la grammaire est également très simple. Il n'y a pas de genre, de nombre, de mode, de temps... La seule règle présente ici est celle de la séquence : quand 2 instructions se suivent, par exemple « Droite Avance », cela signifie qu'elles doivent être exécutées l'une après l'autre. Pour plus de clarté lors de l'écriture et la lecture, on peut décider (ou non !) d'introduire une règle supplémentaire, comme séparer les instructions par des virgules.

Enfin, la classe remarque que ces langages ne permettent pas de faire autre chose qu'un déplacement sur un quadrillage (on ne peut pas afficher du texte ou faire des calculs) : les langages de programmation sont spécialisés. L'enseignant peut faire remarquer qu'il existe d'autres langages similaires (peu de « mots », peu de règles de grammaires, peu ou pas d'ambiguïté...) : la notation musicale par exemple.

Sensibilité aux erreurs

L'enseignant demande aux élèves ce qui se passe si on introduit une erreur dans un programme (par exemple, si on oublie une instruction). On peut prendre un exemple concret, en se basant sur le parcours initial du rover (Fiche 28). Que se passe-t-il si l'on saute une instruction ? On s'aperçoit que quel que soit le langage utilisé, on rate l'objectif. On remarquera qu'une erreur dans un langage autocentré peut conduire plus loin de l'objectif qu'une erreur dans un langage allocentré. Cependant, dans les deux cas, il s'agit d'un bug et on notera deux choses. Premièrement, l'objectif n'est pas atteint, donc c'est un échec aussi important dans un cas que dans l'autre. Deuxièmement, si le terrain présente des obstacles (crevasses...), alors on ne veut pas se tromper... même pas un tout petit peu. Les deux bugs sont donc aussi problématiques l'un que l'autre !

La classe discute des différentes origines possibles d'un bug. Il peut s'agir d'une erreur dans l'algorithme (la méthode), ou d'une erreur dans le programme (l'expression de l'algorithme dans le langage choisi : une erreur de syntaxe, par exemple), ou encore d'une erreur matérielle (liée, par exemple, à une panne d'un élément de la machine ou à une erreur dans la transmission des instructions, comme ici).

Note pédagogique

Nous avons choisi d'utiliser l'orthographe anglaise *bug* (qui signifie « insecte » en anglais) plutôt que l'orthographe francisée « bogue » de façon à respecter l'origine historique de ce mot. C'est en effet Grace Hopper qui a popularisé l'expression « bug », inventée par Thomas Edison, lorsqu'elle a découvert l'origine d'une panne dans son ordinateur : un insecte grillé au cœur des circuits. Voir l'éclairage scientifique sur Grace Hopper page 14.

Conclusion et traces écrites

La classe synthétise collectivement ce qui a été appris au cours de cette séance :

- En informatique, on invente et on utilise des langages.
- Pour donner des instructions à une machine, on utilise un langage de programmation, compréhensible à la fois par la machine et par l'être humain.
- Un langage de programmation est différent d'une langue naturelle :
 - Il possède très peu de mots et de règles de grammaire.
 - Il ne laisse place à aucune ambiguïté.
- Il existe de nombreux langages de programmation, adaptés à différents usages.
- Un bug est une erreur dans un programme.
- Un bug minime en apparence peut avoir des conséquences énormes.

Les élèves notent ces conclusions dans leur cahier de sciences. L'enseignant, quant à lui, prépare une affiche intitulée : « Qu'est-ce que l'informatique ? » Cette affiche sera remplie au fur et à mesure du projet et permettra de dresser un panorama général de cette science (notions de langage, d'algorithme, de programme, de machine, d'information...). Il commence par recopier ce que la classe a appris sur la notion de langage au cours de cette séance.

Exercice en ligne

Si l'école dispose d'une salle informatique, on peut prolonger cette séance par une courte activité branchée permettant de retravailler la notion de bug. L'activité (disponible sur le site Web du projet, voir page 349) peut aussi être menée collectivement au vidéoprojecteur ou sur TNI, quelques jours après cette séance, de façon à en renforcer les acquis.

Prolongement débranché (cycle 4)

Cette séance peut être prolongée par un exercice de traduction d'un langage vers un autre langage, à faire en binôme :

- Traduire la séquence d'instructions « Nord, Ouest, Nord, Est, Est, Est, Sud, Est, Sud, Ouest, Ouest, Nord, Nord » (langage allocentré) en une expression autocentrée. Vérifier sur une feuille quadrillée que les 2 expressions conduisent bien au même résultat.

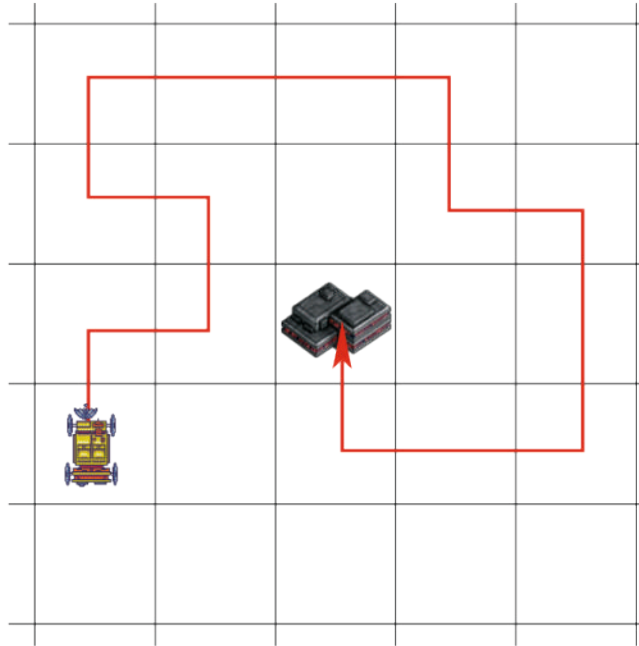
Traduire (pour un rover orienté initialement vers le nord) « Droite, Avancer, Avancer, Gauche, Avancer, Gauche, Avancer, Avancer, Avancer, Avancer, Droite, Avancer » en une expression allocentrée et, de même, vérifier sur une feuille quadrillée le résultat.

Si ce prolongement a été traité, ajouter dans la trace écrite la notion suivante :

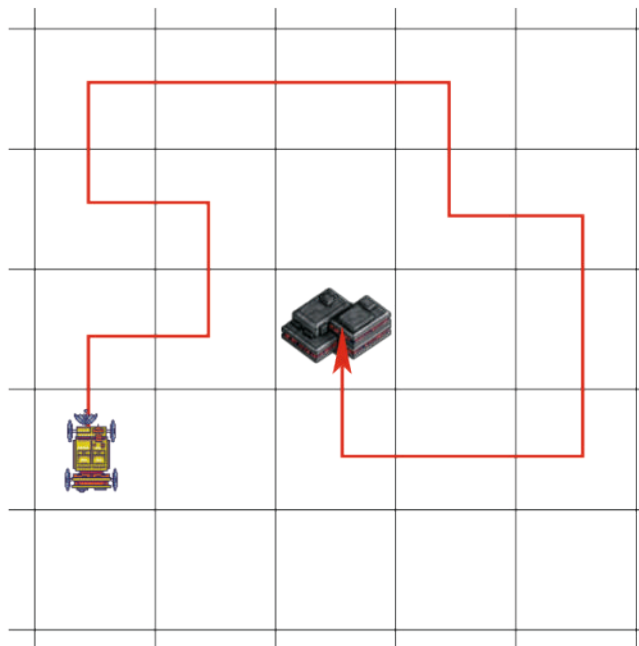
- On peut traduire le même ensemble d'instructions d'un langage vers un autre.

FICHE 28 Piloter un véhicule à distance

Consigne : Écris une suite d'instructions permettant au véhicule de suivre case à case le parcours rouge jusqu'à la base.



Consigne : Écris une suite d'instructions permettant au véhicule de suivre case à case le parcours rouge jusqu'à la base.





Séance 2 – Comment encoder un message avec des nombres ?

Résumé	Les élèves doivent encoder un message textuel en utilisant seulement des nombres. Pour cela, ils font des propositions puis conviennent d'un tableau de correspondance entre caractères textuels et nombres, commun à toute la classe. Ils utilisent ce tableau pour encoder un message qu'ils doivent envoyer, puis pour décoder un message qu'ils ont reçu.
Notions (cf. scénario conceptuel, page 212-213)	« Information » <ul style="list-style-type: none"> • Un caractère peut être représenté par un nombre. • Un texte, qui est une succession de caractères, peut être représenté par une succession de nombres. • L'encodage du texte est le remplacement de ses caractères par les nombres qui leur correspondent. Le décodage est l'opération inverse.
Matériel	Pour la classe <ul style="list-style-type: none"> • Fiche 29, page 226
Lexique	Table de correspondance, encodage, décodage
Durée	1 h 15

Notes scientifiques préalable sur le vocabulaire

- Dans le langage courant, les termes « coder », « chiffrer », « crypter » sont souvent confondus et employés à mauvais escient. Le terme « coder », par exemple, est parfois utilisé pour signifier « programmer » (on lit aussi « écrire du code »), parfois pour « représenter de l'information » (par exemple, le codage binaire), ou encore pour « altérer un message pour le rendre incompréhensible » (en référence à la notion de « code » secret...).
- Dans tout ce qui suit, nous tâcherons d'employer les mots dans leur signification scientifique, c'est-à-dire :
 - « Coder » signifie représenter de l'information. Au cours de cette séance, les élèves vont représenter un message textuel à l'aide de nombres. Lorsqu'ils transformeront les lettres en nombres, on parlera d'« encodage » et, lorsqu'ils effectueront la transformation inverse, de « décodage ». C'est l'objet de cette séance, puis des deux suivantes sur le binaire.
 - « Chiffrer » un message consiste à le déformer pour le rendre incompréhensible à toute personne non destinataire du message (qui ne possède pas la clef pour le déchiffrer). C'est l'objet de la Séance 4 de la Séquence 3, page 322.
 - Les termes « crypter »/« décrypter » (que nous éviterons ici) sont en général employés comme synonymes de « chiffrer »/« déchiffrer ». La « cryptanalyse » consiste à casser le chiffrement (on devine quelle est la clef alors qu'on n'est pas censé la connaître).

Situation déclenchante

L'enseignant explique à la classe que l'équipe du poste de contrôle communique avec le rover et avec l'équipe de terrain en utilisant des instruments électroniques. Ces instruments permettent d'envoyer et de recevoir seulement des nombres, pas de lettres. Tout message textuel à transmettre doit donc être converti en une succession de nombres avant l'envoi (c'est l'opération d'encodage du message),

puis reconverti en une succession de caractères textuels à l'arrivée (c'est l'opération de décodage du message). Il ne peut pas y avoir d'espace ou de virgule entre les nombres : les nombres arrivent tous « collés » à la suite les uns des autres.

La question est donc la suivante : comment encoder des messages textuels sous forme de nombres, puis les décoder ?

Recherche : stratégie d'encodage et de décodage (par binômes)

L'enseignant annonce que l'équipe de terrain souhaite prolonger sa sortie au-delà de l'horaire initialement prévu. Mais le vent semble se lever. Il faut envoyer au poste de contrôle un message qui commence comme ceci :

ENVOYEZ BULLETIN METEO.

Par binômes, les élèves recherchent une stratégie d'encodage de ce texte en nombres. Il ne s'agit pas pour le moment d'effectuer l'encodage proprement dit, mais seulement d'en proposer les modalités.



Classe de CM2 de Christelle Crusberg (Champigny-sur-Marne)

Lorsqu'un binôme annonce avoir trouvé une stratégie d'encodage, l'enseignant donne en secret à chaque élève un petit papier sur lequel est écrit un court message textuel contenant au moins un espace ou un point : par exemple « PLUS VITE. » ou « TRÈS BIEN. » ou « TOUT VA BIEN. ». Il demande aux élèves d'encoder le message à destination de leur camarade, qui doit ensuite le décoder. L'enseignant rappelle que les nombres du message encodé devront être tous collés les uns à la suite des autres, sans espace ni ponctuation. Les deux élèves vérifient si la transmission de l'information s'est bien passée. Si ce n'est pas le cas, le binôme peut identifier le problème et changer ou améliorer sa stratégie.

Notes pédagogiques

- Le texte à encoder contient uniquement des lettres majuscules sans accents, des points et des espaces. De ce fait, il est probable que la plupart des propositions des élèves consistent à associer chaque lettre de l'alphabet à un nombre, conformément à l'ordre alphabétique (1 pour coder A, 2 pour coder B, etc. jusqu'à 26 pour coder Z) et à ajouter une correspondance entre le point, l'espace et deux autres nombres (par exemple 27 et 28 ou bien 27 et 0).
- Toutefois, certains groupes imaginent de coder par des nombres différents les lettres majuscules (de 1 à 26 par exemple) et minuscules (de 27 à 52 par exemple).

La question des lettres accentuées peut également émerger, ainsi que celle des caractères de ponctuation autres que le point. Bien que le message à coder ne comporte pas de chiffres, certains groupes peuvent chercher à les prendre en compte : ils peuvent alors associer les chiffres 0 à 9 aux nombres 0 à 9, et le codage des lettres ne commence qu'à partir de 10.

- Toutes ces propositions doivent être valorisées, même si on effectue ensuite, pour des raisons de simplicité, le choix de ne pas coder les chiffres et d'en rester à un codage identique pour toutes les variantes d'une lettre (A, a, â, à tous codés par 01, par exemple).
- Enfin, certains élèves risquent d'anticiper sur la question du chiffrage du message en proposant une correspondance peu intuitive entre caractères textuels et nombres, voire en proposant une correspondance instable (qui change tous les 10 caractères par exemple). On proposera à ces élèves de retenir leurs idées pour une autre séance (Séance 4 de la Séquence 3, page 322).

Mise en commun

Lors de la mise en commun, l'enseignant demande à un premier binôme de présenter sa proposition. Celle-ci est confrontée aux propositions des autres groupes (voir notes pédagogiques ci-dessus), en pesant les avantages et les inconvénients de chacune. Puis la classe s'arrête sur un choix de consensus : celui d'une table de correspondance biunivoque (aucune ambiguïté, ni à l'encodage ni au décodage) entre les caractères textuels que l'on souhaite utiliser dans les messages et des nombres.

La table de correspondance retenue pour la suite est la suivante (noter que la table véritablement utilisée peut varier d'une classe à l'autre, notamment pour l'espace et le point) :

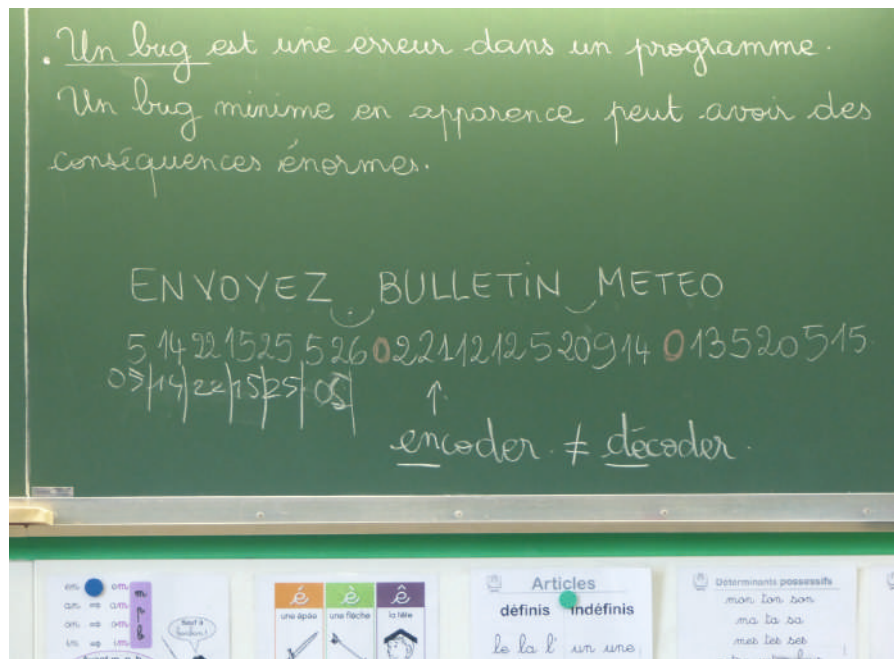
Caractère textuel	A	B	C	D	E	F	G	H
Nombre	01	02	03	04	05	06	07	08

Caractère textuel	I	J	K	L	M	N	O	P
Nombre	09	10	11	12	13	14	15	16

Caractère textuel	Q	R	S	T	U	V	W	X
Nombre	17	18	19	20	21	22	23	24

Caractère textuel	Y	Z	.	Espace
Nombre	25	26	27	28

Noter que les nombres 1 à 9 (correspondant aux lettres A à I) ont été notés 01 à 09, de façon que tous les nombres utilisés pour coder les caractères textuels soient écrits avec autant de chiffres (2 en l'occurrence). Ainsi, 0221 se lit 02 21 et se décode en BU, et 2201 se lit 22 01 et se décode en VA. Alors que si les lettres A à I étaient codées par les nombres 1 à 9, le texte encodé 221 pourrait être lu soit 2 21, soit 22 1 et être décodé indifféremment en BU ou en VA.



Discussion sur l'importance d'un encodage sur 2 chiffres pour chaque caractère.
Classe de CM1 de Carole Vinel (Paris)

Exercice – Encoder un message (par groupes)

L'enseignant distribue aux élèves le haut de la Fiche 29 (table de correspondance et consigne 1) et leur demande d'effectuer l'encodage du message destiné au poste de contrôle (consigne 1), en utilisant la table de correspondance choisie par la classe.

Une fois les messages encodés, l'enseignant affiche les résultats et annonce que le message vient d'être envoyé au poste de contrôle. Il insiste sur le fait que le message de départ et le message encodé contiennent la même information sous deux formes différentes.

Défi – Décoder un message (par groupes)

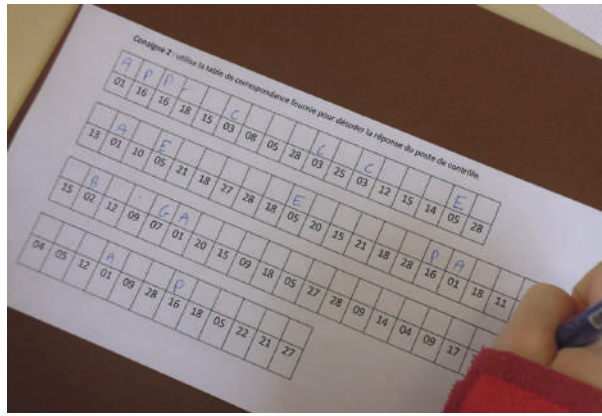
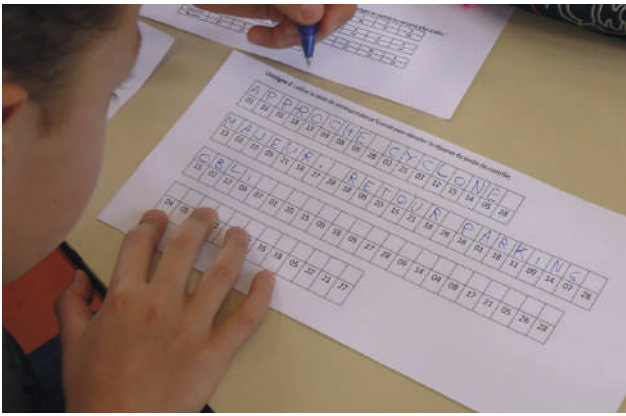
L'enseignant annonce que la classe vient de recevoir la réponse du poste de contrôle, sous forme encodée. Il va maintenant s'agir de décoder cette réponse.

Il distribue la partie basse de la Fiche 29 (consigne 2) et laisse aux élèves le temps de mener à bien le décodage.

La classe partage le message décodé :

APPROCHE CYCLONE MAJEUR. RETOUR PARKING OBLIGATOIRE. INDIQUEZ DELAI PREVU.

La mission ne peut donc pas être prolongée, le rover doit rentrer immédiatement à la base.



Deux stratégies de décodage, au fil du texte (à gauche) ou pour toutes les occurrences d'un caractère dans le texte (à droite).
Classe de CM2 d'Anne-Marie Lebrun (Bourg-la-Reine)

Note pédagogique

L'exercice d'encodage et le défi de décodage peuvent être menés de façon collaborative : un élève donné encode ou décode seulement une ligne du message, et les résultats sont mis en commun. Nous proposons toutefois de distribuer les messages en entier aux élèves, ce qui simplifie la logistique de la séance, permet aux élèves les plus rapides d'avoir constamment du travail et à tous de garder en mémoire le travail collectif accompli.

Conclusion et traces écrites

La classe synthétise collectivement ce qui a été appris au cours de cette séance :

- *Un caractère peut être représenté par un nombre.*
- *Un texte, qui est une succession de caractères, peut donc être représenté par une succession de nombres.*
- *L'encodage du texte est le remplacement de ses caractères par les nombres qui leur correspondent. Le décodage est l'opération inverse.*

Les élèves notent ces conclusions dans leur cahier de sciences. L'enseignant, quant à lui, complète l'affiche « Qu'est-ce que l'informatique ? » avec ce que la classe a appris au cours de cette séance sur les notions d'information et d'algorithme.

Prolongement

Avec les plus grands, le message à décoder peut être un peu plus long. Par exemple, le message ci-dessous a la propriété amusante d'être un pangramme (toutes les lettres de l'alphabet y sont présentes au moins une fois) :

WARNING. APPROCHE CYCLONE MAJEUR. VENTS EXCEPTIONNELLEMENT FORTS ATTENDUS. RETOUR PARKING OBLIGATOIRE. INDIQUEZ DELAI PREVU.

FICHE 29

Encoder et décoder un message

Table de correspondance entre caractères utilisés dans les messages et nombres servant à les coder :

Caractère	A	B	C	D	E	F	G	H
Nombre	01	02	03	04	05	06	07	08
Caractère	I	J	K	L	M	N	O	P
Nombre	09	10	11	12	13	14	15	16
Caractère	Q	R	S	T	U	V	W	X
Nombre	17	18	19	20	21	22	23	24
Caractère	Y	Z	.	Espace				
Nombre	25	26	27	28				

Consigne 1 : utilise la table de correspondance fournie pour encoder le message destiné au poste de contrôle.

E	N	V	O	Y	E	Z		B	U	L	L	E	T	I	N		M	E	T	E	O	.	
P	R	O	L	O	N	G	A	T	I	O	N		S	O	R	T	I	E		S	A	U	F
	R	I	S	Q	U	E		T	E	M	P	E	T	E	.		M	E	R	C	I	.	



Consigne 2 : utilise la table de correspondance fournie pour décoder la réponse du poste de contrôle.

01	16	16	18	15	03	08	05	28	03	25	03	12	15	14	05	28							
13	01	10	05	21	18	27	28	18	05	20	15	21	18	28	16	01	18	11	09	14	07	28	
15	02	12	09	07	01	20	15	09	18	05	27	28	09	14	04	09	17	21	05	26	28		
04	05	12	01	09	28	16	18	05	22	21	27												



Séance 3 – Comment coder des informations en binaire ?

Résumé	Les élèves doivent maintenant se contenter de deux symboles (0 et 1) pour transmettre des messages. Ils explorent les moyens d'encoder différentes informations (les 4 points cardinaux nord/sud/est/ouest, les 7 jours de la semaine, etc.) en associant des 0 et des 1, et découvrent ainsi le codage binaire de l'information.
Notions (cf. scénario conceptuel, page 215)	« Information » <ul style="list-style-type: none"> • Un ordinateur représente n'importe quelle information par un code qui n'utilise que 2 symboles, 0 et 1, appelés « bits » : c'est le code binaire. • Le codage binaire permet de représenter toutes sortes de données, notamment des nombres ou des caractères textuels. • Plus on juxtapose de bits, plus on peut représenter d'éléments différents.
Matériel	Pour chaque élève <ul style="list-style-type: none"> • Fiche 30, page 232
Lexique	Liste d'éléments, bit, codage binaire
Durée	1 h 30

Situation déclenchante

L'enseignant explique à la classe qu'en réalité, les instruments électroniques ne peuvent pas transmettre directement les nombres : ils transmettent des flux de signaux lumineux ou électriques. Ces signaux n'ont que deux états : NON (pas de signal)/OUI (signal) aussi appelés 0 et 1. Les élèves doivent donc repenser leur système de codage avec cette nouvelle contrainte.

La question est donc la suivante : comment encoder des informations en utilisant seulement des 0 et des 1 ?

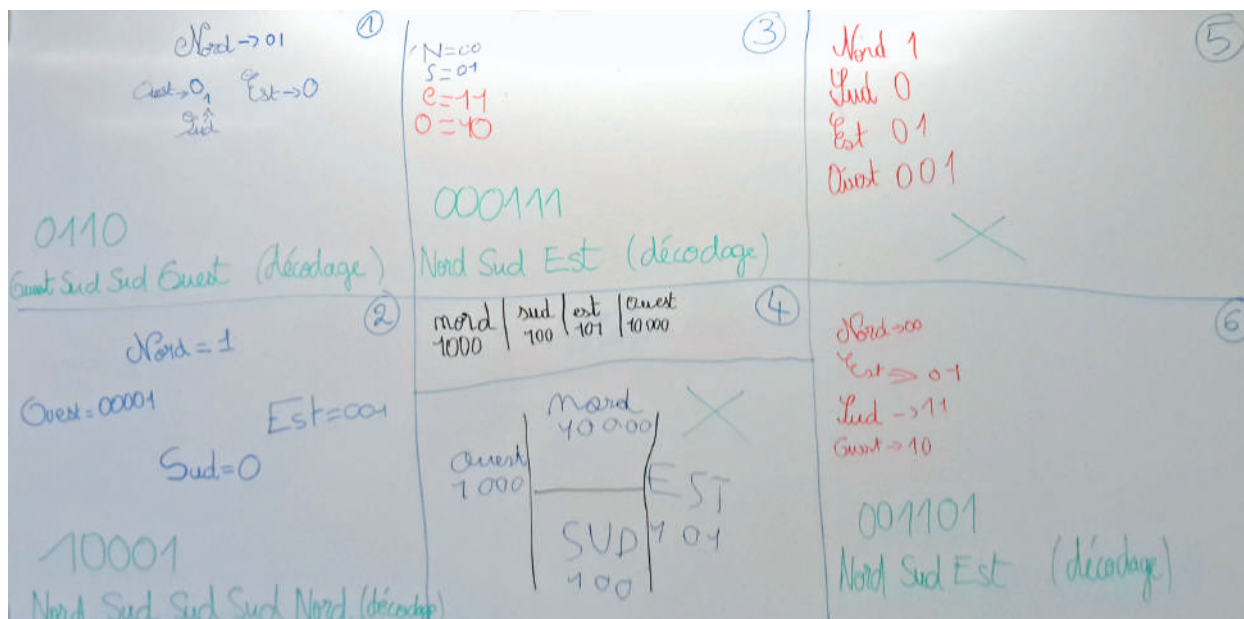
Recherche : stratégie d'encodage utilisant seulement des 0 et des 1 (par binômes et collectivement)

L'enseignant rappelle que le rover peut être dirigé depuis le poste de contrôle en utilisant les 4 mots Nord/Sud/Est/Ouest. Il demande aux élèves de réfléchir en binômes à un moyen de transmettre ces 4 mots en utilisant seulement des 0 et des 1.



Propositions de quelques élèves. Classe de CM1-CM2 de Thyphaine Collignon (Vernaison)

Après quelques minutes, les groupes partagent leurs propositions (voir les notes pédagogiques ci-dessous). La classe discute les propositions et constate que celles qui fonctionnent consistent toutes à assembler plusieurs 0 et 1.



Mise en commun et discussion sur l'importance d'encoder chacune des informations sur un même nombre de chiffres, en vue du décodage. Classe de CM2 de Christelle Crusberg (Champigny-sur-Marne)

Notes pédagogiques

• Une première proposition peut consister à remplacer :

- « Nord » par « 00 » ;
- « Sud » par « 11 » ;
- « Est » par « 01 » ;
- « Ouest » par « 10 ».

Dans ce type de proposition, chaque point cardinal est codé par une succession de deux « 0 ou 1 » (un doublet). Toutes les combinaisons de deux « 0 ou 1 » sont utilisées. Il est important de faire remarquer aux élèves que selon les groupes ce n'est pas forcément le même point cardinal qui est associé à un doublet donné : toutes les correspondances point cardinal/doublet se valent.

• Une deuxième proposition peut consister à remplacer :

- « Nord » par « 1000 » ;
- « Sud » par « 0100 » ;
- « Est » par « 0010 » ;
- « Ouest » par « 0001 ».

Dans ce type de proposition, chaque point cardinal est codé par une succession de quatre « 0 ou 1 ». Seules quelques-unes des combinaisons de quatre « 0 ou 1 » sont utilisées (par exemple, la combinaison 0000 ne l'est pas).

• Une troisième proposition peut consister à remplacer :

« Nord » par :	« Sud » par :	« Est » par :	« Ouest » par :
1	0	0	0
0 0	0 0	0 1	1 0
0	1	0	0

Face à ce type de proposition, l'enseignant précise que l'idée est bonne, mais que les 0 et les 1 ne peuvent être transmis que les uns à la suite des autres, donc sans positionnement dans l'espace. Cette troisième proposition devient alors équivalente à la proposition 2 : pour chaque point cardinal, on transmet quatre « 0 ou 1 » dont un seul est un « 1 ». Il y a quatre positions possibles pour le « 1 », une par point cardinal.

Exercice : choix du nombre de 0 et de 1 à assembler pour coder les jours de la semaine (par groupes)

L'enseignant indique que tous les messages échangés entre la base et le rover sont datés. Il faut donc être capable d'indiquer avec des 0 et des 1, entre autres choses, le jour de la semaine. L'enseignant demande aux élèves de proposer une façon de coder les 7 jours de la semaine en alignant le moins possible de « 0 ou 1 ».



Classe de CM2 de Christelle Crusberg (Champigny-sur-Marne)

Mise en commun

La mise en commun révèle qu'il suffit d'aligner trois « 0 ou 1 », par exemple comme ceci (mais il y a une multitude de possibilités équivalentes) :

- 000 pour lundi ;
- 001 pour mardi ;
- 010 pour mercredi ;
- 011 pour jeudi ;
- 100 pour vendredi ;
- 101 pour samedi ;
- 110 pour dimanche.

Il reste même une combinaison (ici 111) qui n'est pas utilisée.

L'enseignant guide la formulation orale d'une conclusion avec pour support la Fiche 30 projetée et distribuée à chacun :

- On peut coder les éléments d'une liste par une succession de « 0 ou 1 » aussi appelés « bits » (contraction de *binary digit*, soit « nombre binaire »).

- Avec un seul bit, on peut coder tous les éléments d'une liste qui n'a pas plus de 2 éléments (blanc/noir, éteint/allumé sont des exemples de listes à deux éléments): on associe le «0» à un élément de la liste, et le «1» à l'autre élément de la liste (la classe donne des exemples de listes à deux éléments et chacun note un exemple sur la Fiche 30).
- Avec deux bits juxtaposés, on peut coder tous les éléments d'une liste qui n'a pas plus de 4 éléments (comme Nord/Sud/Est/Ouest), car il existe 4 façons différentes d'agencer 2 chiffres valant 0 ou 1: «00», «01», «10» et «11» (les élèves notent chacun sur la Fiche 30 un exemple de liste codable sur 2 bits, mais pas sur 1 bit: ces listes ont 3 ou 4 éléments).
- Avec trois bits juxtaposés, on peut coder tous les éléments d'une liste qui n'a pas plus de 8 éléments (par exemple les 7 jours de la semaine), car les 8 seules combinaisons de 0 et de 1 possibles sont: «000», «001», «010», «011», «100», «101», «110» et «111» (les élèves notent chacun sur la Fiche 30 un exemple de liste codable sur 3 bits, mais pas sur 2 bit – ces listes ont 5, 6, 7 ou 8 éléments).
- Plus on juxtapose de bits, plus on peut représenter d'éléments: 16 éléments au maximum en juxtaposant 4 bits, 32 éléments au maximum en juxtaposant 5 bits, 64 éléments au maximum en juxtaposant 6 bits, etc.

Chaque élève propose sur la Fiche 30 une liste libre codable sur 4 bits mais pas sur 3 bits (une telle liste doit comporter entre 9 et 16 éléments inclus): les élèves proposent par exemple les 12 mois de l'année, les 10 chiffres 0 à 9, les 12 heures du cadran des montres, les 10 doigts des mains, les prénoms de leurs 9 cousins germains, 16 noms d'animaux de la ferme, etc.

Défi (par binômes)

L'enseignant demande aux élèves de trouver combien de bits il faut juxtaposer pour pouvoir coder les 26 lettres de l'alphabet.

Après quelques minutes, la mise en commun permet de conclure que 4 bits ne suffisent pas pour coder les 26 lettres de l'alphabet (codage de 16 éléments au maximum: $2 \times 2 \times 2 \times 2 = 16$), mais que 5 bits sont suffisants (codage de 32 éléments au maximum: $2 \times 2 \times 2 \times 2 \times 2 = 32$).

Conclusion et traces écrites

La classe synthétise collectivement ce qui a été appris au cours de cette séance:

- *On peut encoder les éléments d'une liste par une succession de «0 ou 1» aussi appelés «bits». On parle alors d'un codage «binaire».*
- *Plus on juxtapose de bits, plus on peut représenter d'éléments: 2 éléments au maximum avec un seul bit, $2 \times 2 = 4$ éléments au maximum en juxtaposant 2 bits, $2 \times 2 \times 2 = 8$ éléments au maximum en juxtaposant 3 bits, $2 \times 2 \times 2 \times 2 = 16$ éléments au maximum en juxtaposant 4 bits, $2 \times 2 \times 2 \times 2 \times 2 = 32$ bits en juxtaposant 5 bits, ... $2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 256$ éléments en juxtaposant 8 bits, etc.*
- *Le codage binaire permet de représenter toutes sortes de données, notamment des nombres ou des caractères textuels.*
- *Un ordinateur représente n'importe quelle information en utilisant un codage binaire.*

Les élèves notent ces conclusions dans leur cahier de sciences tandis que l'enseignant complète l'affiche: «*Qu'est-ce que l'informatique?*»

Exercice en ligne

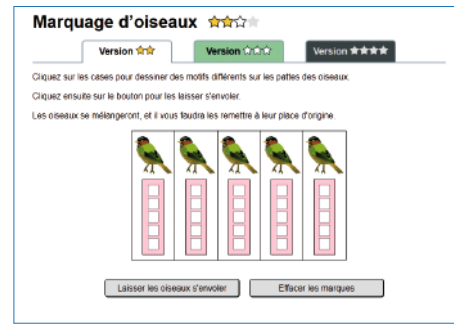
Un exercice en ligne sur le site Web du projet (voir page 349) permet de s'exercer à l'utilisation du codage binaire.

Prolongement (pour le cycle 4)

En cycle 4, on peut insister davantage sur une façon commode de retrouver toutes les combinaisons de 0 et de 1, par exemple, sur 5 bits: si on demande aux élèves de retrouver les 32 combinaisons de 5 bits possibles, ils s'aperçoivent que c'est très difficile de tous les trouver et de ne pas proposer les mêmes plusieurs fois.

On peut alors suggérer une approche récursive:

- Avec seulement 1 bit, on a la liste très facilement: 0, 1.
- Pour 2 bits, il suffit de recopier deux fois la liste sur 1 bit: une fois en ajoutant un 0 devant, une fois en ajoutant un 1 devant: on obtient 00, 01, 10, 11
- Pour 3 bits, on procède de même: on recopie deux fois la liste obtenue pour 2 bits, une fois en ajoutant un 0 devant, une fois en ajoutant un 1 devant: 000, 001, 010, 011, 100, 101, 110, 111. Et ainsi de suite pour 4 bits, 5 bits, etc., de proche en proche.



FICHE 30
Coder les éléments d'une liste en binaire

Consigne : dans chaque bulle vide, écris un exemple de liste dont tous les éléments peuvent être codés sur 1 bit, 2 bits, 3 bits ou 4 bits.

1 bit :

0

1

2 bits :

00

01

10

11

3 bits :

000

001

010

011

100

101

110

111

4 bits :

0000

0001

0010

0011

0100

0101

0110

0111

1000

1001

1010

1011

1100

1101

1110

1111

Défi : détermine combien de bits il faut juxtaposer au minimum pour pouvoir coder chacune des 26 lettres de l'alphabet.



Séance 4 – Comment encoder et décoder un message en binaire ?

Résumé	Dans la continuité de la séance précédente, les élèves appliquent leurs découvertes à l'encodage d'un court message textuel en binaire, puis au décodage d'un message en binaire qu'ils reçoivent.
Notions (cf. scénario conceptuel, page 213)	« Information » • Le codage binaire permet de représenter toutes sortes de données, notamment des nombres ou des caractères textuels.
Matériel	Pour chaque groupe • Fiche 31, page 236
Lexique	Liste d'éléments, bit, codage binaire
Durée	1 h

Situation déclenchante

L'enseignant rappelle aux élèves qu'ils ont reçu de la base l'annonce qu'une tempête se prépare. L'équipe de la base a demandé une information : le temps que mettra le rover à rentrer à la base. Aujourd'hui, les élèves vont encoder leur réponse en binaire.

Recherche : encoder un message à destination de la base en binaire (collectivement puis par groupes)

L'enseignant précise que le message que les élèves devront envoyer à la base comporte seulement des lettres majuscules, des espaces et des points (soit 28 sortes de caractères textuels). Il demande aux élèves de décider de combien de bits ils ont absolument besoin pour encoder chaque lettre et leur suggère, en cas de besoin, de se référer à leur travail de la séance précédente (Fiche 30). La classe s'accorde sur la contrainte d'un codage sur 5 bits par caractère.

L'enseignant distribue alors le haut de la Fiche 31. Il donne aux élèves 5 minutes pour décider d'une correspondance entre caractères et groupes de 5 bits. Lors de la mise en commun, la classe s'entend par exemple pour la table de correspondance page suivante :

5 bits	00000	00001	00010	00011	00100	00101	00110	00111
Caractère	A	B	C	D	E	F	G	H
5 bits	01000	01001	01010	01011	01100	01101	01110	01111
Caractère	I	J	K	L	M	N	O	P
5 bits	10000	10001	10010	10011	10100	10101	10110	10111
Caractère	Q	R	S	T	U	V	W	X
5 bits	11000	11001	11010	11011	11100	11101	11110	11111
Caractère	Y	Z	.	(espace)	<i>aucune signification (on peut, si on le souhaite, utiliser ces groupes de 5 bits pour coder d'autres signes de ponctuation)</i>			

L'enseignant donne alors aux groupes d'élèves la mission d'utiliser cette table de correspondance pour encoder en binaire le texte suivant :

DIX MINUTES

Il distribue pour cela le milieu de la Fiche 31. La classe obtient :

Lettre	D	I	X		M	I	N	U	T	E	S
Groupe de 5 bits	00011	01000	10111	11011	01100	01000	01101	10100	10011	00100	10010

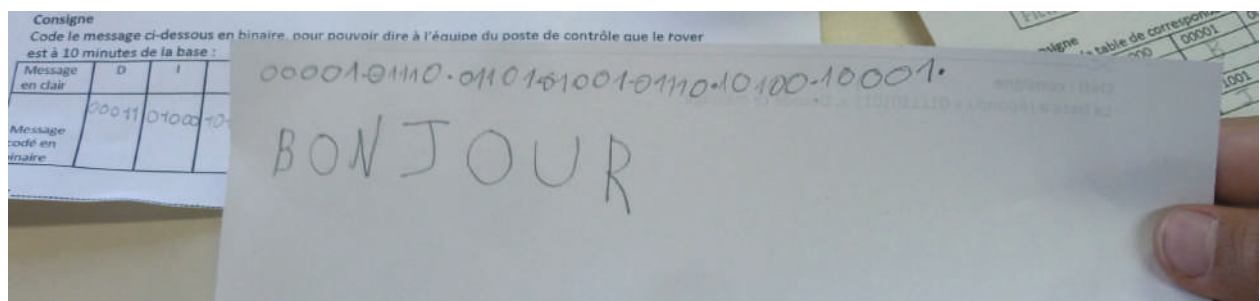
Défi : décoder un message envoyé par la base (par binômes)

L'enseignant fournit aux élèves la réponse ultime de la base (Fiche 31), que les élèves doivent décoder : 0111001010

Le découpage en groupes de 5 bits (quintuplets) donne 01110 et 01010, ce qui d'après la table de correspondance correspond aux lettres O et K. Le message reçu de la base est donc «OK».

Exercice libre : encoder et décoder des messages en binaire.

L'enseignant laisse aux élèves une dizaine de minutes pour encoder de courts messages et les distribuer à des camarades qui les décodent. Cette activité est très motivante pour les élèves et leur permet de consolider l'apprentissage en cours.



Classe de CM1 de Carole Vinel (Paris)

Conclusion

La classe revient sur la conclusion de la séance précédente, en particulier la notion suivante : *le codage binaire permet de représenter toutes sortes de données, notamment des textes.*

Prolongement débranché

Afin que les élèves comprennent mieux pourquoi l'électronique impose souvent une représentation binaire des données, il est possible de faire une analogie avec des circuits électriques comprenant autant d'ampoules disposées en parallèle que de bits utilisés pour le codage. Chaque ampoule est en série avec un interrupteur. On peut placer chaque interrupteur en position ouverte/fermée (soit 0/1, OFF/ON ou encore ampoule éteinte/allumée). Ce sont les deux seuls états possibles pour un interrupteur. Dans les appareils électroniques, les composants électroniques ne sont pas des ampoules et des interrupteurs mais, de la même façon, ils peuvent être alimentés en courant, ou non : il est pratique alors de distinguer ces 2 états et de coder l'information en binaire.

FICHE 31
Encoder/décoder un message en binaire

Consigne

Propose une table de correspondance: remplis le tableau ci-dessous en associant chaque groupe de 5 bits à un caractère textuel.

5 bits	00000	00001	00010	00011	00100	00101	00110	00111
Caractère								
5 bits	01000	01001	01010	01011	01100	01101	01110	01111
Caractère								
5 bits	10000	10001	10010	10011	10100	10101	10110	10111
Caractère								
5 bits	11000	11001	11010	11011	11100	11101	11110	11111
Caractère								



Voici la table de correspondance que nous allons tous utiliser par la suite:

5 bits	00000	00001	00010	00011	00100	00101	00110	00111
Caractère	A	B	C	D	E	F	G	H
5 bits	01000	01001	01010	01011	01100	01101	01110	01111
Caractère	I	J	K	L	M	N	O	P
5 bits	10000	10001	10010	10011	10100	10101	10110	10111
Caractère	Q	R	S	T	U	V	W	X
5 bits	11000	11001	11010	11011	11100	11101	11110	11111
Caractère	Y	Z	.	(espace)	aucune signification (on peut, si on le souhaite, s'en servir pour encoder d'autres signes de ponctuation)			



Consigne

Encode le message ci-dessous en binaire pour pouvoir dire à l'équipe du poste de contrôle que le rover est à 10 minutes de la base:

Message en clair	D	I	X		M	I	N	U	T	E	S
Message codé en binaire											



Consigne

La base a répondu «0111001010». Décode ce message.

Cette ressource est issue du projet thématique **1,2,3... CODEZ !**, paru aux Éditions Le Pommier.

Claire Calmet, Mathieu Hirtzig et David Wilgenbus

1,2,3... CODEZ !

Enseigner l'informatique à l'école et au collège
(cycles 1, 2 et 3)

FONDATION La main à la pâte
POUR L'ÉDUCATION À LA SCIENCE

Qu'il s'agisse de préparer les enfants aux métiers de demain, de les aider à comprendre le monde numérique dans lequel ils vivent – afin qu'ils soient en mesure d'agir sur lui et non de le subir –, de les sensibiliser aux enjeux de citoyenneté, ou encore de favoriser la coopération ou développer leur créativité... l'informatique doit être enseignée à tous, dès le plus jeune âge.

Le projet « 1, 2, 3... codez ! » développé par la Fondation *La main à la pâte*, Inria et France 101 vise à initier les élèves et leurs enseignants à la science informatique, de la maternelle à la classe de 6^e. Il propose à la fois des activités branchées (nécessitant un ordinateur, une tablette ou un robot) permettant d'introduire les bases de la programmation et des activités débranchées (informatique sans ordinateur) permettant d'aborder des concepts de base de la science informatique (algorithme, langage, information...).

Un outil clés en main
Ce guide pédagogique comporte :

- 3 progressions pour la classe (cycles 1, 2 et 3)
 - Une approche pluridisciplinaire associant démarche d'investigation et pédagogie de projet ;
 - Des séances clés en main, testées en classe, organisées en séquences thématiques et scénarisées pour chaque cycle ;
 - Des fiches documentaires à photocopier ;
- Des éclairages pédagogiques et scientifiques pour guider l'enseignant dans la mise en œuvre du projet.

Les auteurs
Claire Calmet est formatrice et responsable des liens avec le monde de l'entreprise et de la recherche à la Fondation *La main à la pâte*.
Mathieu Hirtzig est webmestre et médiateur scientifique à la Fondation *La main à la pâte*.
David Wilgenbus est formateur et responsable de la production de ressources à la fondation *La main à la pâte*. Il coordonne le projet « 1, 2, 3... codez ! ».

FONDATION La main à la pâte
Lancée en 1996 par Georges Charpak, prix Nobel de physique, avec le soutien de l'Académie des sciences et du ministère de l'Éducation nationale, *La main à la pâte* vise à promouvoir à l'école primaire un enseignement de science et de technologie de qualité : <http://www.fondation-lamap.org>

Avec le soutien de :

Illustration : Catherine Zimmmermann

Éditions Le Pommier

74651106
21 €
Diffusion Bélin

Retrouvez l'intégralité de ce projet sur : <https://www.fondation-lamap.org/projets-thematiques>.

Fondation *La main à la pâte*

43 rue de Rennes
75006 Paris
01 85 08 71 79
contact@fondation-lamap.org

Site : www.fondation-lamap.org

FONDATION
La main à la pâte
POUR L'ÉDUCATION À LA SCIENCE