



FÉDÉRATION DE L'ENSEIGNEMENT SECONDAIRE CATHOLIQUE

Sciences - Informatique

Outil d'accompagnement du programme

3e degré - Humanités générales et technologiques



VERSION PROVISOIRE

Table des Matières

Préface	5
Représentation et traitement de l'information	7
Le texte vu par l'homme et par la machine	7
L'image vue par l'homme et par la machine	12
Les nombres vus par l'homme et par la machine	13
L'avenir du goût et des odeurs ?	20
Le matériel informatique	23
Les unités de mesure de capacité	23
Le système informatique	26
Traitement de texte	31
Travail d'interdisciplinarité – création d'un modèle pour les dissertations	31
Le tableur	33
L'adressage absolu, l'adressage relatif	33
Découverte du tableur	38
Encodage de formules et recopiage - Mise en page	39
Encodage de formules (le si... alors ...), format des nombres	44
Gestionnaire de bases de données	49
Méthode d'analyse d'une base de données	49
Programmation procédurale	59
Introduction	59
Compétences	60
Séquence et affectation	61
Lecture	67
Ecriture	71
Conditionnelle et alternative	75
Répétitions	79
Et si vous ne souhaitez pas utiliser un ordinateur lors de l'élaboration d'algorithmes	83
Bibliographie	93
Livres	93
Sites	94

Préface

Ce document d'accompagnement du programme de l'option « Sciences-Informatique » a été rédigé par deux enseignantes, Michèle Sermeus et Christel Depotte, de l'Institut Saint-François d'Ath, à la demande du secteur informatique de la FESeC. Les membres du secteur informatique sont conscients de la pertinence mais aussi des limites de certaines situations proposées. Cependant, toutes les situations d'apprentissage présentées ont été expérimentées et validées en classe par les auteurs avec des élèves; elles sont présentées à titre d'exemples et non comme un ensemble de solutions définitives pour résoudre les nombreux problèmes pédagogiques, didactiques et méthodologiques posés par l'enseignement des technologies informatiques dans l'enseignement secondaire. Elles devraient servir avant tout de source d'inspiration pour les enseignants chargés des différents cours de cette option.

Le cours de programmation est probablement le cours dont l'enseignement est le plus complexe, d'autant plus qu'il s'agit d'aborder non seulement les concepts de la programmation procédurale, mais aussi ceux de la programmation orientée objets (POO). L'enseignement de la programmation procédurale a déjà fait l'objet de publications variées depuis de nombreuses années, et parmi les différentes méthodes pédagogiques proposées, l'utilisation de robots considérés comme exécutants, auxquels il faut faire faire des actions, se retrouve sous différentes formes : Karel le robot, les robots virtuels de Meurice R., Logo et sa tortue, Images pour programmer de Duchâteau Charles ... Une nouvelle exploitation de ce concept de « robot » est proposée dans le document pour parcourir l'ensemble des actions de base et l'ensemble des structures de contrôle; à chaque enseignant de l'adapter au contexte de sa classe s'il souhaite l'utiliser.

En ce qui concerne la POO, il faut reconnaître que les diverses expériences menées dans les écoles doivent encore être analysées et étudiées pour en extraire quelques suggestions pédagogiques. Il serait présomptueux de proposer des situations d'apprentissage actuellement. Durant l'année scolaire 2002-2003, un séminaire consacré à ce sujet a été organisé par le Cefis en collaboration avec le secteur informatique de la FESeC. Les documents rédigés à l'occasion de ce séminaire pourront certainement alimenter la réflexion (ces documents n'étaient pas encore disponibles à la date de rédaction de ce texte).

Par abus de langage, le mot « ordinateur » est utilisé pour représenter un système informatique formé de l'ordinateur muni d'un système d'exploitation et d'un logiciel particulier en fonction du contexte d'utilisation. Derrière ce système, il faut percevoir une équipe de développeurs qui ont conçu le produit avec toutes les contraintes liées à l'informatique : traitement formel de l'information, limites des langages utilisés, limites du matériel, les considérations (en constante évolution et adaptation) de l'ergonomie ...

Le temps imparti pour mener à bien la rédaction de ce document n'a pas été suffisant pour aborder tous les domaines du programme ainsi que les problèmes liés à l'évaluation. Ce document d'accompagnement s'inspire déjà des travaux en cours dans les différents groupes de réflexion à propos de l'évaluation dans les options des humanités générales et technologiques. D'autres exemples de situations et des indications sur l'évaluation seront proposés dans un avenir proche.

Représentation et traitement de l'information

Le texte vu par l'homme et par la machine

Description de la situation d'apprentissage

Les élèves reçoivent les instructions suivantes : « Tu as à ta disposition deux signes □ et ✕. Tu dois transmettre un message à un copain en n'utilisant que les deux signes à ta disposition mis les uns derrière les autres. Ton copain recevra le message formé de □ et ✕ ainsi que les informations nécessaires à la compréhension du message. Tu dois rédiger très clairement le document pour ton copain qui va devoir comprendre le message. »

Consignes aux élèves

Par groupes de deux, les élèves reçoivent un texte à coder. Par exemple « Qui es-tu ? » ; « D'où viens-tu ? » ; ou autre (il y a toujours des espaces, une apostrophe ...)

Les élèves doivent réfléchir, inventer un code, rédiger un mode d'emploi du code inventé avec rigueur :

- Le message doit être transformé en □ et ✕.
- Les signes □ et ✕ doivent se suivre sans aucun espace
- Le message doit être accompagné d'un mode d'emploi pour permettre à un autre groupe de le déchiffrer.
- Le mode d'emploi doit être très clair et bien rédigé.

Compétences terminales et transversales

- Compétences relationnelles et de communication. - Les élèves travaillent en groupes.
- Construire un message cohérent et rigoureux. - Les élèves écrivent un "mode d'emploi" sur la façon dont ils ont organisé le codage du message. Ce mode d'emploi doit absolument être très rigoureux pour qu'il soit compris par les copains.
- Produire un document à l'aide d'un logiciel. - Les élèves peuvent rédiger leur mode d'emploi à l'aide d'un logiciel de traitement de texte.

Tâches du professeur

Le professeur groupe les élèves par deux et distribue à chaque groupe un message différent.

Déroulement de la leçon

Très vite, les élèves se posent des questions et interpellent le professeur.

- Peut-on utiliser des couleurs ? Il est bien évident que non.
- Peut-on superposer les signes ? La réponse est encore non.
- Faut-il indiquer les espaces ? Faut-il mettre le point d'interrogation ? Faut-il distinguer les majuscules des minuscules ?

Le professeur doit les convaincre de tout transmettre à leur copain. Les espaces, les signes de ponctuation font partie du message et doivent être transmis. Celui qui recevra le message ne pourra rien inventer; il devra suivre le mode d'emploi de façon rigoureuse.

La même difficulté apparaît dans tous les groupes : faire comprendre le passage d'une lettre à l'autre à celui qui recevra le message.

- Les élèves trouvent des solutions. Ils réalisent leur mode d'emploi.
- Les messages sont échangés.
- Les différents groupes commencent alors le travail de décodage des messages.

Certains groupes n'arrivent pas à décoder le message reçu. Il est intéressant d'analyser avec toute la classe la cause du problème : dans certains travaux, le mode d'emploi a été mal réalisé, dans d'autres, c'est la façon de coder qui est inexacte.

Voici quelques exemples de travaux d'élèves

Exemple 1

Voici le message et le mode d'emploi écrits par un groupe d'élèves.

Message :

□ □ × × × □ × □ × □ × □ □ □ □ × □ × □ × × × × × □ × □ □ □ × ×
 × × □ × □ × □ □ × × × □ □ □ × □ × □ □ □ □ × □ × × × × × □ × □
 × × × × □ × □ □ □ × □ × □ □ × × × ×

Mode d'emploi :

× = a × × = b × × × = c × × × × = d
 □ = e □ × = f □ = g □ × × × = h
 × □ × = espace et changement de lettre
 × □ □ × = tiret

Le groupe qui a reçu ce message n'a pas pu le décoder. S'il est assez facile d'y repérer les changements de lettres (ils sont soulignés dans le message ci-dessous) et le tiret (il est double-souligné), il n'est pas facile de comprendre le codage des lettres.

C'est là que va naître l'idée de coder toutes les lettres avec des suites de signes □ et × toujours de la même longueur. Le problème de choisir la bonne longueur subsiste cependant.

- Une suite de 1 seul caractère n'offre que 2 possibilités □, ×
- Une suite de longueur deux n'offre que 4 possibilités □ ×, × □, □ □, × ×
- Une suite de longueur trois offre 8 possibilités □ □ □, × □ □, □ × □, □ □ ×, □ × ×, × □ ×, × × □, × × ×
- Une suite de longueur quatre offre 16 possibilités □ □ □ □ ...

Pour les plus observateurs, le nombre de possibilités peut être calculé à l'aide des puissances de deux.

Il faut alors essayer d'évaluer le nombre de possibilités dont on aura besoin. Il faut rechercher les caractères à coder : les lettres en majuscules, les lettres en minuscules, les chiffres, les signes de ponctuation, les caractères accentués, les signes d'opérations.

Au total, il y a environ 120 caractères à coder. Une suite de 7 signes conviendra puisqu'elle nous offre 128 possibilités.

On peut alors se demander si un mot souligné peut être transmis dans le message. Avec un code au début et à la fin du mot à souligner, ce problème est facilement résolu.

En informatique, le fait qu'un courant passe ou ne passe pas dans un câble, qu'un emplacement sur un disque soit magnétisé ou non oblige à travailler à l'aide de deux états. Ces deux états sont symbolisés par les deux chiffres 0 et 1. Les signes □ et × peuvent devenir 0 et 1. 0 et 1 sont appelés des chiffres binaires.

Conclusions

Un message sous forme de texte est numérisable. La quantité de caractères qui peuvent être codés est limitée par la longueur de la suite formée des chiffres binaires 0 et 1.

Extensions

A partir de cette situation, les notions de bit, byte et octet peuvent être expliquées. Le binaire et l'hexadécimal ont des raisons d'être abordés. Il est intéressant de parler de l'évolution du codage des caractères en ASCII, en UNICODE.

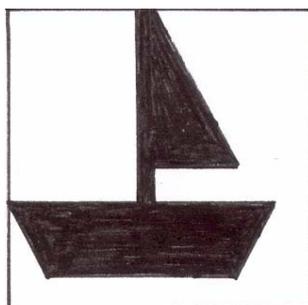
Cette situation montre aussi que l'ordinateur pourra traiter les informations « sans les comprendre », de façon formelle. Pour lui, l'information est dépourvue de sens. Si un connaisseur de la langue française peut donner des sens différents à des mots identiques, l'ordinateur ne pourra faire aucune différence.

Dans une phrase comme « Tu bois dans le bois », une personne de langue française donne des sens différents aux deux mots identiques. La machine va coder les deux mots de façon identique. Il est intéressant de faire réfléchir les élèves sur les limites du correcteur orthographique et celles des programmes de traduction.

L'image vue par l'homme et par la machine

Description de la situation d'apprentissage

Avec les mêmes consignes que celles de la situation précédente, transmettre à un copain le message suivant

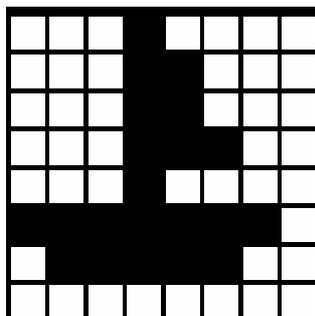


Déroulement de la leçon

Notre intention était d'amener les élèves à construire une grille qui divise le dessin en petits carrés.

S'ils n'y pensent pas, il est souhaitable d'agrandir une image au format bmp dans un logiciel d'image pour faire apparaître les carrés.

Le professeur peut proposer aux élèves de construire une grille de huit carrés de côté.



Les élèves proposent un repérage des cases comme au combat naval. La case A1 sera blanche, la case A2 sera blanche, la case B1 sera blanche, la case B4 sera noire ...

Avec l'acquis de la première situation, on se rend compte qu'il faudra trois signes pour coder la lettre, trois signes pour coder le chiffre et un signe pour coder la couleur. Un ensemble de sept signes par case permet de coder l'image.

Il est bon de faire remarquer qu'un plus grand nombre de carrés donnerait une image plus précise. Il faudrait aussi un plus grand nombre de signes pour la coder.

Si le dessin est en couleur, on peut remplacer le seul signe donnant la couleur (noir / blanc) par un nombre de signes augmentant en fonction de l'augmentation du nombre de couleurs.

Conclusions

Les images sont numérisables si elles sont décomposées en petits carrés. Le nombre de carrés limitera la qualité de l'image. Plus il y a de points, plus le code sera long. Plus il y a de couleurs utilisées, plus le code sera long.

Extensions

A partir de cette situation, les notions de pixel, résolution d'image, codification des couleurs peuvent être expliquées.

Les nombres vus par l'homme et par la machine

Introduction

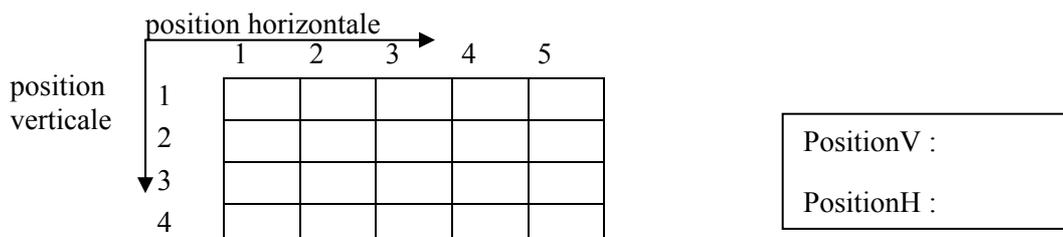
Cette situation d'apprentissage lie le cours de programmation et son robot, les notions de logique vues au cours de mathématique et l'utilisation d'une calculatrice scientifique disponible avec la plupart des systèmes d'exploitation. Elle permet de découvrir comment un ordinateur code les nombres entiers négatifs sur un octet. Les élèves ont vu au cours de programmation que les nombres entiers codés sur un octet sont compris entre -128 et 127 . Le cours d'informatique peut leur apporter une explication de ces valeurs limites.

Savoirs requis

Les notions de bit, d'octet, de système binaire.

Description de la situation d'apprentissage

Le cours de programmation travaille avec un robot qui se positionne dans un domaine rectangulaire (de 4 lignes et 5 colonnes) accompagné de deux compteurs indiquant la position verticale (PositionV) et la position horizontale (PositionH).



Voici une proposition : « Le robot est dans la première ligne ou dans la première colonne. »

- Tu dois écrire cette proposition sous forme logique en utilisant des propositions élémentaires.
- Tu dois écrire cette proposition sous forme logique en utilisant les notations vues au cours de programmation.

Tu sais que l'ordinateur travaille en binaire et qu'il est capable de travailler uniquement avec deux positions 1 et 0. Conventionnellement, on symbolise Vrai par 1 et Faux par 0. Utilise ces notations pour résoudre les exercices qui suivent.

- Je place le robot en position verticale 1 et en position horizontale 3. La proposition est-elle vraie ou fausse ?
- Je place le robot en position verticale 2 et en position horizontale 3. La proposition est-elle vraie ou fausse ?
- Je place le robot en position verticale 4 et en position horizontale 1. La proposition est-elle vraie ou fausse ?
- Je place le robot en position verticale 1 et en position horizontale 1. La proposition est-elle vraie ou fausse ?

Ouvre la calculatrice en mode d'affichage scientifique. Impose à la calculatrice de travailler en binaire et vérifie la table de vérité du « ou » logique. Quelle conclusion en tirer quant au mode de fonctionnement de la calculatrice ?

Recherche

- non (0)
- non (1)

Comment l'ordinateur a-t-il travaillé ?

Continuons notre exploration de la calculatrice. Recherche

- 111 ou 100
- 1001 ou 1110
- 1010 ou 1000
- 100 ou 11

Comment l'ordinateur a-t-il travaillé ?

La calculatrice va devenir infernale. Travaille maintenant en système décimal, tu remarques que les opérateurs logiques sont toujours accessibles. Que vont fournir ces expressions insensées ?

- 8 ou 2
- 4 ou 5
- 3 ou 2
- 9 ou 2

Comment l'ordinateur a-t-il travaillé ?

Recherche

- non(5)
- non(10)
- non(7)
- non(100)

Comment l'ordinateur a-t-il travaillé ?

Compétences terminales et transversales

- Analyser des informations. L'élève doit réfléchir à la signification des résultats que lui communique la calculatrice.
- Synthétiser des informations. L'élève doit réunir les éléments qu'il découvre petit à petit dans la situation pour les utiliser toutes dans la dernière partie de la recherche.

Savoirs disciplinaires

Notion de codage de l'information

Tâches du professeur

Le professeur doit faire prendre conscience aux élèves que les cours ont des liens entre eux. Il suit l'avancement des élèves dans leurs recherches et répond aux questions qui se posent ; il doit diriger la fin de la leçon pour aider les élèves dans la découverte de la codification des nombres entiers négatifs en complément à 2.

Consignes aux élèves

Les élèves doivent travailler d'abord par écrit ; ils doivent utiliser leur connaissance d'autres cours pour résoudre les problèmes. Ils travaillent ensuite avec l'ordinateur ; ils analysent les informations données par la calculatrice et synthétisent leurs résultats.

Déroulement de la leçon

Le début de la leçon ne pose pas de problème ; la proposition sous forme logique devient « La position horizontale du robot est égale à 1 ou la position verticale du robot est égale à 1 ». En utilisant les notations du cours de programmation, la proposition devient « $(\text{PositionH} == 1)$ ou $(\text{PositionV} == 1)$ »

En continuant les recherches, les élèves découvrent que

- 111 ou 100 donne 111
- 1001 ou 1110 donne 1111
- 1010 ou 1000 donne 1010
- 100 ou 11 donne 111

L'ordinateur applique la règle du « ou » chiffre par chiffre de droite à gauche.

100 ou 11 sera interprété comme 100 ou 011

Tout ce qui précède se déroule assez rapidement avec les élèves ; le travail qui suit apporte plus d'étonnement et demande plus de réflexion.

Lorsque la calculatrice devient infernale, les élèves sont amenés à réfléchir et à imaginer des solutions.

En système décimal, les fonctions logiques restent accessibles.

- 8 ou 4 donne 12
- 4 ou 5 donne 5
- 3 ou 2 donne 3
- 9 ou 2 donne 11

Devant ces valeurs qui étonnent, ils essaient de trouver une explication. Vu ce qu'ils viennent de découvrir sur la fonction « ou » et leur connaissance en binaire, beaucoup d'entre eux ont pensé à convertir les nombres en binaire et se rendent compte qu'en utilisant ce qu'ils ont découvert ci-dessus, la calculatrice a travaillé avec la fonction « ou » et les valeurs des nombres en binaire.

En effet,

- 8 en base 10 vaut 1000 en binaire
- 4 en base 10 vaut 100 en binaire
- 1000 ou 100 donne 1100
- 1100 en binaire vaut 12 en base 10

Les autres expressions peuvent se vérifier de la même manière. Les élèves ont découvert comment l'ordinateur pouvait leur afficher des résultats aussi étranges pour le calcul d'expressions plutôt insensées.

Continuons nos recherches.

- Non (5) donne -6
- Non (10) donne -11
- Non (7) donne -8
- Non (100) donne -101

Les élèves restent ici beaucoup plus perplexes. Certains n'y voient que des choses étranges, d'autres, plus observateurs, se rendent compte que le « non » transforme le nombre en lui ajoutant 1 puis en prenant son opposé.

L'objectif étant de découvrir la codification des nombres entiers sur un octet, la tâche du professeur est de faire rassembler aux élèves tous les éléments dont ils disposent :

- Le travail se fait sur un octet, c'est un choix pour éviter les écritures trop longues.
- La fonction « non » travaille sur un octet, un mot ou un double mot.

- La codification d'un nombre entier positif sur un octet est simple ; la conversion en binaire est connue, il suffit d'ajouter des « 0 » devant la valeur binaire du nombre pour former le code du nombre sur 8 bits.

Revenons au premier énoncé : non (5) =

Avec tous les éléments connus, les élèves peuvent rechercher la codification de 5 sur un octet :

5 vaut 101 en binaire ; sur un octet 5 sera codé 00000101.

Vu le travail réalisé par la fonction « non », non (5) = non (00000101) = 11111010.

La recherche de non (5) a donné -6 sur l'ordinateur ; les élèves en concluent que le code de -6 sera 11111010.

Suivant les élèves, un deuxième ou même un troisième calcul peut être nécessaire à la compréhension.

Pour les aider, on peut faire le tableau

nombre	Code du nombre sur un octet	Non (code)	Nombre négatif correspondant
4	00000100	11111011	-5
5	00000101	11111010	-6
6	00000110	11111001	-7
7	00000111	11111000	-8
8	00001000	11110111	-9

Le professeur peut alors, en guidant les élèves, arriver à leur faire découvrir la méthode de codification des nombres négatifs dites « en complément à 2 » qu'utilisent nos ordinateurs.

Le professeur peut poser la question « Comment coder le nombre -6 sur un octet en partant du nombre 6 ? »

- 6 se convertit en binaire 110
- 6 a pour code sur un octet 00000110
- en appliquant le « non » bit par bit, cela donne 11111001 qui est le code de -7
- en ajoutant 1 au code de -7, on aura le code de -6 ; cela donne 11111001 + 1 d'où 11111010

Un élève a trouvé une autre solution

- 6 se convertit en binaire 110
- 6 a pour code sur un octet 00000110
- en retirant 1 au code de 6, on a 00000110 - 1 = 00000101 qui est le code de 5
- en appliquant le « non » bit par bit à 00000101, cela donne 11111010 qui est le code de -6

Il reste à travailler avec les élèves les limites de cette codification des nombres entiers sur un octet. Plusieurs possibilités peuvent être envisagées.

Possibilité 1

Sur un octet, je code 256 valeurs, le nombre 0 possède un code, il reste 255 codes possibles. Il est donc impossible de coder des entiers allant de -128 à 128 . Coder des entiers allant de -127 à 127 laisserait un code inutilisé. Il y a deux solutions possibles : coder les nombres de -128 à 127 ou de -127 à 128 . Si 128 est codé, son code vaut 10000000 . Si -128 est codé, son code vaut 01111111 et en ajoutant 1, on obtient 10000000). Il est dès lors impossible de coder 128 sinon -128 serait codé aussi et deux nombres auraient le même code.

Possibilité 2

Des élèves s'y retrouvent mieux en complétant le tableau jusqu'à ses limites

Nombre	Code du nombre sur un octet	Non (code)	Nombre négatif correspondant
0	00000000	11111111	-1
1	00000001	11111110	-2
2	00000010	11111101	-3
3	00000011	11111100	-4
4	00000100	11111011	-5
5	00000101	11111010	-6
6	00000110	11111001	-7
7	00000111	11111000	-8
8	00001000	11110111	-9
126	01111101	10000001	-127
127	01111111	10000000	-128

En observant le tableau, on voit apparaître les limites de nombres entiers codés sur un seul octet

Conclusions

Après avoir découvert qu'un ordinateur peut coder des caractères à l'aide de 0 et de 1, qu'un ordinateur peut coder des images à l'aide de 0 et de 1, les élèves ont découvert comment un ordinateur peut coder des nombres négatifs sans se servir du signe – par la méthode du complément à 2 ; ils ont aussi pris conscience des limites de la codification des nombres entiers sur un octet. Ils sont ainsi conscients qu'un ordinateur ne pourra pas coder n'importe quel nombre et que le nombre de bits sur lesquels un nombre est codé impose des limites à la valeur des nombres.

Extensions

Si le temps le permet, il est possible de montrer aux élèves comment l'ordinateur pourra calculer une différence de deux nombres entiers en n'utilisant que l'addition.

L'avenir du goût et des odeurs ?

Introduction

Si la numérisation du texte, des images ... est acquise, qu'en est-il de celle du goût et des odeurs qui a un avenir prometteur dans notre société. La situation d'apprentissage ci-dessous va proposer un travail d'interdisciplinarité entre les cours d'informatique (partie sur le traitement de l'information et partie sur l'informatique et la société), de logiciels et de français. Elle mènera à la réalisation d'une dissertation sur l'apport à notre société de la codification des odeurs et du goût.

Description de la situation d'apprentissage

Si le son et les images numérisés font aujourd'hui partie du quotidien et si l'utilité de cette numérisation ne fait de doute pour personne, on peut se demander si les saveurs et les odeurs sont aujourd'hui numérisables et si leur numérisation a une quelconque utilité.

Déroulement de la leçon

La question laisse les élèves très perplexes. Les avis sont partagés. Un groupe d'élèves pense qu'il est impossible de numériser l'odorat et le goût car ils relèvent de l'appréciation de chacun. L'argument ne tient pas ; les images et le son peuvent aussi plaire ou ne pas plaire suivant la personne qui regarde ou écoute. Un groupe d'élèves pense que « oui » mais ne sait pas trop pourquoi et beaucoup sont sans avis.

Pour les aider à répondre à la question, voici des adresses de sites Internet intéressants.

<http://www.alpha-mos.com/> parcourir : Société, Technologie, Applications, Produits (odeurs et goût)

www.acfas.ca/congres/congres67/s1176.htm

www.museesdegrasse.com parcourir : Français, Musée international de la parfumerie, Entrez, La création en parfumerie.

Ce site parle des nez humains utilisés en parfumerie. Tout un métier ! L'informatique pourrait-elle leur venir en aide ?

<http://www-drt.cea.fr/pages/se2i/pages/nez.htm>

<http://www.cybersciences.com/cyber/3.0/n926.asp>

<http://www.forumlabo.com/2002/abstracts/2002/26langue.htm>

Pour d'autres recherches sur Internet, utiliser les mots clés « Nez électronique » et « Langue électronique ».

Sans s'attarder à la partie technique de la codification, le professeur amène les élèves à relever sur les sites Internet les domaines d'application dans la société des odeurs et des goûts numérisés : la parfumerie, la détection des mines, la conservation des aliments, le goût à donner aux médicaments pour faire disparaître l'amertume d'un produit...

Pour le cours de français

Le professeur de français communique aux élèves le sujet de dissertation précis à réaliser sur base des documents récoltés sur Internet ou dans la presse. Voici l'énoncé du sujet proposé :

« Les progrès technologiques et la recherche informatique mettent sur pied un nez électronique. Quelles peuvent être les applications concrètes d'une telle découverte ? Quelles seront les conséquences positives ou négatives sur le plan humain ? »

Pour le cours de logiciel

L'élève doit réaliser la dissertation au traitement de texte en utilisant le modèle qui a été conçu précédemment en interdisciplinarité (cours de logiciels et de français).

Compétences terminales et transversales

- Produire un document comprenant du texte et répondant aux critères de qualité éditoriale à l'aide d'un traitement de texte.
- Prendre conscience des effets des choix technologiques sur l'environnement, de leurs incidences sur le mode de vie ; présenter et argumenter les conséquences d'un choix technologique en fonction de ses impacts sociaux, économiques, environnementaux, éthiques et culturels, analyser ceux-ci.
- Recueillir et traiter les informations en fonctions d'un problème.
- Analyser et synthétiser des informations.

Tâches du professeur

Le professeur d'informatique présente la situation et aide à la recherche sur Internet. Il donne aux élèves les mots clés pour la recherche. Le professeur de français réalise le sujet de

dissertation et prend en charge l'évaluation du contenu de la dissertation. Le professeur de logiciels prend en charge et évalue l'utilisation du traitement de texte.

Consignes aux élèves

Les élèves doivent réaliser des recherches sur Internet. Ils rassemblent des informations. Ils écrivent ensuite leur dissertation et la transcrivent en utilisant le traitement de texte.

Conclusions

Ce travail a permis la découverte d'un enjeu informatique dans la société, il a éveillé la curiosité des élèves dans un domaine qui leur était inconnu.

Le matériel informatique

Les unités de mesure de capacité

Description de la situation d'apprentissage

Partie 1

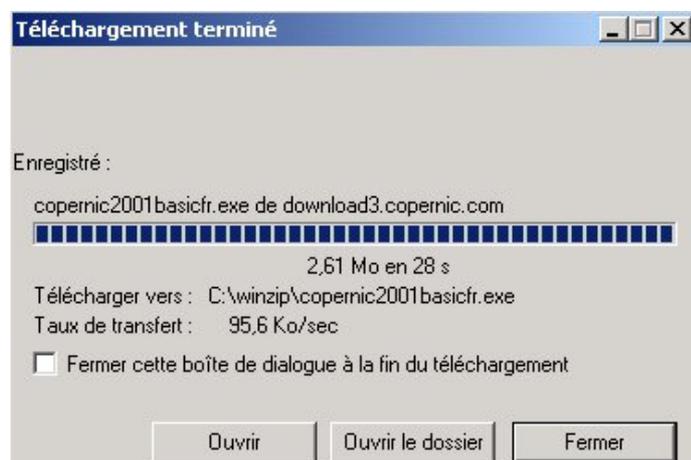
Observe la copie d'écran ci-dessous. Elle te donne des détails au sujet d'un document de traitement de texte. En analysant les informations du document, recherche avec précision combien d'octets contient 1 Kilo-octet (1 Ko).



Partie 2

Observe la copie d'écran ci-dessous. Elle te donne des renseignements au sujet d'un téléchargement. Lorsque tu télécharges un fichier, les informations se déplacent d'un endroit à un autre. Que t'indique le « taux de transfert » ?

En analysant les données du document, recherche avec précision combien de Ko contient 1 Méga-octet (1 Mo).



Partie 3

Observe la copie d'écran ci-dessous. Elle te donne des renseignements sur un disque dur. En analysant les informations du document, recherche avec précision combien d'octets contient 1 Giga-octet (1 Go).



Déroulement de la leçon

Les élèves travaillent par groupes de deux. Ils doivent écrire leurs calculs clairement.

Compétences terminales

- Compétences relationnelles et de communication. - Les élèves travaillent en groupes.
- Utiliser un langage technique correct et rigoureux. - Les élèves doivent formuler leurs calculs correctement, ils devront maîtriser les unités de capacité à la fin de la leçon.

Savoirs disciplinaires

Les unités de mesure de capacités.

Déroulement de la leçon (suite).

Partie 1

Les élèves tiennent compte des informations qui donnent la taille du document et écrivent leur calcul : $34\,816 : 36 = 1024$

1 kilo-octet vaudrait donc 1024 octets.

Certains qui s'attendaient à 1000 comme réponse prennent la peine de vérifier la valeur trouvée avec les informations qui concernent la taille occupée par le document sur le disque. Ils écrivent le calcul : $36\ 864 : 36 = 1024$.

La conclusion que 1 Ko vaut 1024 octets est alors unanime.

Il faut faire remarquer que 1024 correspond à la 10^e puissance de deux et que c'est la puissance de deux qui a la valeur la plus proche de la valeur 1000.

1 Kilo octets = 1024 octets = 2^{10} octets

Des élèves s'interrogent aussi sur la différence entre la taille d'un fichier et la taille occupée par ce fichier sur le disque. C'est l'occasion de parler de l'organisation d'un disque dur.

Partie 2

Le taux de transfert indique la quantité d'informations qui arrivent à la machine en une seconde.

On retrouve ici une notion de vitesse ; il serait peut-être intéressant de travailler avec le professeur de mathématique lorsqu'il aborde la matière des dérivées.

Les élèves ont raisonné de façon logique. La machine transfère 95,6 Ko en 1 seconde. En 28 secondes, elle aura transféré $95,6 \times 28 = 2676,8$ Ko. Pour trouver la quantité de Kilo octets contenue dans un Méga octet, ils écrivent le calcul : $2676,8 : 2,61 = 1025,6$

La valeur trouvée n'est malheureusement pas exactement 1024. Elle en est cependant assez proche. La différence vient des erreurs d'arrondis dans les données fournies par l'ordinateur. Les élèves s'en rendent compte rapidement. Pour rester cohérent avec les découvertes du numéro 1, on en déduit que 1 Mo = 1024 Ko.

L'important est d'avoir découvert qu'en informatique, les unités de mesure correspondent à des multiples de puissances de deux et non à des multiples de puissances de dix.

1 Mo = 1024 Ko = 2^{10} Ko = 2^{20} octets

Partie 3

Si les calculatrices ne permettent pas le travail avec d'aussi grands nombres, il est possible de travailler avec un tableur.

Les élèves écrivent le calcul : $4327174144 : 4,02 = 1076411478,61$

Vu l'acquis des parties précédentes, les élèves se rendent compte qu'il faut trouver la puissance de deux correspondante à la valeur trouvée.

Suivant leur connaissance en mathématique, ils peuvent travailler par tâtonnement ou passer au logarithme en base 2.

Ceux qui travaillent par tâtonnement approchent la valeur ci-dessus quand ils calculent 2^{30}

Le $\log_2 1076411478,61$ vaut 30,00358253. En arrondissant, 1 Go = 2^{30} octets.

Encore une fois, il faut tenir compte du fait que l'ordinateur a arrondi les nombres.

Conclusion

La **taille** d'un fichier est la quantité d'octets nécessaires pour coder ce fichier.

La **capacité** d'une unité permettant de stocker des informations (disque dur, cd-rom, mémoire interne ...) correspond à la quantité d'octets qui peuvent être mémorisés par cette unité.

L'unité de base en informatique est l'**octet** ou **byte**. A partir de l'unité de base, sont définis :

1 K	Kilo	=	2^{10} octets	=	1 024 octets
1 M	Méga	=	2^{20} octets	=	1 048 576 octets
1 G	Giga	=	2^{30} octets	=	1 073 741 824 octets
1 T	Tera	=	2^{40} octets	=	1 099 511 627 776 octets

Rem. : 1K = 1Ko = 1Kb, 1M = 1 Mo = 1 Mb

Le **taux de transfert** correspond à un débit, il mesure la quantité d'informations qui se déplacent d'un endroit à un autre en une unité de temps.

Extension

A partir de publicités, rechercher les capacités et les taux de transfert des éléments d'un ordinateur familial. A partir d'articles, rechercher les capacités et les taux de transfert des éléments d'ordinateurs utilisés dans la recherche ou dans l'industrie.

Il est aussi utile de résoudre des exercices qui feront manipuler les unités de capacité. Ces exercices permettront d'avoir une idée plus parlante du nombre de pages qui peuvent être mémorisées sur une unité ...

Le système informatique

Introduction

Avant de travailler cette situation, les élèves ont eu l'occasion de découvrir l'intérieur d'un PC et de visualiser les différents composants et leurs connecteurs.

Description de la situation d'apprentissage

Un club de tennis souhaite acheter un ordinateur pour gérer le fichier des membres, réaliser sa revue mensuelle, réaliser ses affiches pour les activités organisées par le club et envoyer ses résultats à la fédération. Le responsable du club s'adresse à votre classe pour obtenir des offres. Il devra présenter, lors d'une réunion de comité, les différentes propositions que vous aurez retenues et que vous aurez argumentées pour justifier les dépenses.

Déroulement de la leçon

Partie 1

La situation est présentée aux élèves (groupe de 12 élèves). Un budget est fixé ainsi qu'un timing. Des revues sont mises à leur disposition et des PC leur permettent de consulter des sites internet. Une liste de sites expliquant le matériel informatique leur est fournie (Commentcamarche.com, choixpc.com, ...). Ils doivent coopérer pour décider du matériel à acheter.

Un chef de groupe est nommé, qui coordonnera les recherches et se chargera de réunir les travaux des membres du groupe en un seul document réalisé au traitement de texte.

Un éventail du matériel indispensable est dressé et des idées de matériels supplémentaires qui pourraient apporter un "plus" au club de tennis sont envisagées. Un cahier des charges est établi.

Les élèves décident de travailler seuls ou par groupes de deux; ils reçoivent la responsabilité d'étudier une unité précise. Dans cette première partie, ils lisent des articles et consultent les sites proposés.

Pour les uns, cela ne pose aucun problème, mais pour beaucoup d'élèves, l'effort de lire et de comprendre un article est déjà un travail très contraignant. Le professeur doit intervenir pour les guider dans leur lecture et leur permettre de retirer les informations essentielles et de trouver la signification de certains termes.

Partie 2

L'énoncé de la situation demande de motiver le choix du matériel. Les élèves ont tendance à regarder uniquement le prix du matériel et à constituer un ordinateur qui respecte le budget sans se préoccuper des caractéristiques. Un schéma de travail leur est alors proposé pour les aider.

- rechercher ce qui caractérise le matériel et en donner une définition.

exemples:

- Le disque dur est caractérisé par sa capacité, sa vitesse de rotation
- La capacité d'un disque dur est la quantité d'informations qu'il pourra stocker. Elle se mesure en Go.

- rechercher deux ou trois possibilités d'achat pour ce type de matériel qui conviendront au club de tennis.

exemples:

- 40 Go Seagate 5400 t/min UDMA/100 80 Euros
- 40 Go IBM DTLA 120GXP 7200 t/min, UDMA/100 90 Euros
- 40 Go Seagate Barracuda IV 7200 t/min, UDMA/100 85 Euros

- Justifier la sélection du matériel.

Un disque dur de 40 Go suffit pour la réalisation des travaux prévus par votre club et pour une utilisation occasionnelle d'internet. La vitesse de rotation de 5400 tours par minute est suffisante pour de petites applications, mais une vitesse de 7200 est mieux, et l'interface UDMA 100 est en concordance avec la carte mère choisie.

- Donner un avis sur le choix à effectuer et motiver le choix

Si le budget le permet, le choix se portera sur le disque IBM, qui, de l'avis de nombreux internautes, est un disque dur performant, fiable et plus silencieux que le Seagate. Il apportera un meilleur confort de travail.

Cette partie pose problème à beaucoup d'élèves : la rédaction des textes est laborieuse et le respect du schéma est difficile à obtenir.

Une fois le texte rédigé, les élèves doivent le transmettre tapé au kilomètre au chef de groupe.

Partie 3

Il reste au chef de groupe à rassembler les documents des différents groupes et à tout mettre en forme. Il doit aussi réaliser un tableau récapitulatif justifiant le respect du budget et remettre le document imprimé à transmettre au club de tennis.

Partie 4

Le travail fait l'objet d'une présentation orale menée par le chef de groupe, qui donne une vue d'ensemble du travail rédigé et invite les autres à présenter les différents matériels qu'ils ont étudiés.

Tâche du professeur

Le professeur doit présenter la situation aux élèves, fournir le matériel nécessaire à la réalisation du projet en collaboration avec les élèves (revues, adresses de site, ordinateurs connectés à Internet ...).

Il doit aider les élèves à retrouver les caractéristiques des différents matériels et veiller à ce que les caractéristiques essentielles apparaissent dans le travail. Il doit faire respecter le plan de travail et le timing. Il doit veiller à ce que le contenu des textes écrits par les élèves soient corrects.

Il doit, enfin, pendant ou après la présentation, répondre aux questions des élèves sur les parties travaillées par leurs camarades de classe.

Consignes aux élèves

Les élèves doivent s'organiser, se répartir le travail, communiquer entre eux pour choisir du matériel compatible et respecter le budget. Ils doivent rechercher des informations, trouver les caractéristiques du matériel avec leur signification et rédiger un texte clair en respectant les différents points du schéma. Ils doivent collaborer pour finaliser la rédaction du document. La présentation orale les oblige à parler devant les autres.

Compétences terminales et transversales

- Compétences relationnelles et de communication. - Les élèves organisent un travail en se répartissant des tâches et en se tenant au courant de l'avancement de chacun. Ils présentent leur travail oralement.

- Construire un message cohérent et rigoureux. - Les élèves décrivent le matériel, motivent leur choix.
- Produire un document à l'aide d'un logiciel. - Les élèves rédigent leur travail à l'aide d'un logiciel de traitement de texte.
- Analyser des informations. Les élèves doivent trouver les caractéristiques des différents matériels et les expliquer.
- Synthétiser des informations. Les élèves doivent réunir les informations des différentes sources et les synthétiser.
- Recueillir et traiter les informations en fonction d'un problème.

Savoirs disciplinaires

- Architecture générale d'un ordinateur
- Le système informatique : machine, systèmes d'exploitation, logiciels
- Critères de choix d'un ordinateur.

Traitement de texte

Travail d'interdisciplinarité – création d'un modèle pour les dissertations

Introduction

La leçon se déroule lorsque les bases du traitement de texte (notion de format de police, de format de paragraphe, notions de zone de travail et de marges, notions d'en-tête et pied de page) sont acquises. Elle a été réalisée en début d'année, les élèves ont eu peu de notions de mise en page, ils ont besoin de consignes précises.

Description de la situation d'apprentissage

Pendant l'année, tu vas devoir rédiger plusieurs dissertations pour le cours de français. Tu devras réaliser tes dissertations à l'aide d'un traitement de texte en respectant toujours les mêmes paramètres de mise en page. Ces paramètres sont les suivants :

- Choisir le format du papier : A4, orientation paysage.
- Réduire la zone de travail en imposant une marge de gauche de 8,5 cm, une marge de droite de 2 cm et une marge du haut de 3,5 cm.
- Placer le nom et le prénom en en-tête à gauche, la date du jour à droite sur la première ligne.
- Placer la classe sous le nom dans l'en-tête ; la classe est espacée de 6 points du nom.
- Choisir la police Arial taille 12 pour le texte de la dissertation.
- Indiquer le mot « titre » à l'emplacement du titre des dissertations ; le centrer, le souligner, mettre la police en taille 18, l'espacer de 14 points du paragraphe qui le suit.
- Espacer les paragraphes du texte de 10 points les uns des autres et les justifier.

Comment vas-tu t'y prendre ? Donne l'idée générale. Réalise ensuite le travail sur ordinateur.

Déroulement de la leçon

Les élèves avancent des idées : « il faudrait faire comme du papier à en-tête ». Ce sera l'occasion de faire remarquer que de nombreuses informations apparaissent, en effet, sur toutes les dissertations et qu'il serait dommage de ne pas en profiter pour automatiser le travail. Une liste de ces informations est établie. Il faut les convaincre de réaliser une fois un travail qui pourra être réutilisé pour toutes les dissertations et d'automatiser leur travail. Ils proposent de

réaliser un document et d'en faire des copies pour les dissertations futures. Il est alors temps de leur expliquer la notion de modèle.

Compétences terminales

Utiliser les concepts qui s'imposent pour une tâche en étant conscient de leurs possibilités et de leurs limites.

Tâches du professeur

Les professeurs de logiciels et de français se mettent d'accord sur la présentation à imposer aux élèves suivant leur niveau de connaissance au moment du travail en traitement de texte. Le professeur de français a imposé ici une mise en page qui lui facilite l'écriture de remarques dans la marge et laisse la possibilité d'une correction à côté de l'erreur (la marge de gauche est très large).

Le professeur de logiciel va amener les élèves à découvrir tous les éléments « invariants » dans leur dissertation, il va expliquer comment réaliser un nouveau modèle et comment utiliser ce modèle comme base d'un nouveau document.

Il est important d'expliquer le chemin menant au dossier qui va contenir les modèles sur l'ordinateur.

A chaque dissertation réalisée au cours de l'année, le professeur de français impose l'utilisation du modèle et la remise d'un document imprimé ; le professeur d'informatique contrôle l'utilisation correcte du modèle et du traitement de texte en général en imposant la remise d'une disquette ou d'un cd contenant le document réalisé au traitement de texte ou encore l'envoi par mail de la dissertation.

Consignes aux élèves

L'élève doit réaliser le modèle en respectant les consignes. Il apprend à créer et à utiliser des modèles. Il emporte son modèle sur disquette pour pouvoir travailler à la maison ou se l'envoie par mail.

Conclusions

Les élèves ont découvert les modèles et sont amenés à les utiliser tout au long de l'année. C'est l'occasion d'entretenir la matière. Plus tard dans l'année, l'utilisation de styles pourra venir compléter le travail.

Le tableur

L'adressage absolu, l'adressage relatif

Description de la situation d'apprentissage

Partie 1

Voici un tableau dans lequel les cases sont repérées par leur numéro de ligne et leur numéro de colonne. La case qui se trouve à l'intersection de la ligne n et de la colonne m a pour référence Ln Cm.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													

Dans ce tableau, placer un pion vert dans une case quelconque. Placer un pion jaune à droite du pion vert, deux cases plus loin. Placer, enfin, un pion rouge ailleurs dans le tableau. A tout moment, le pion vert peut bouger. Le pion jaune doit toujours être 2 cases à droite du pion vert. Le pion rouge ne bouge pas.

Dans le tableau suivant, en respectant la notation imposée, écrire la référence des cases dans lesquelles se trouvent les trois pions au départ, faire bouger le pion vert 4 fois et noter la référence des cases dans lesquelles se trouvent les trois pions après chaque mouvement. Deux pions ne peuvent pas se trouver dans la même case.

Compléter la dernière ligne du tableau en supposant que le pion vert se trouve dans une case quelconque référencée Lx Cy.

	Pion vert	Pion jaune	Pion rouge
Départ			
Après le premier déplacement			
Après le deuxième déplacement			
Après le troisième déplacement			
Après le quatrième déplacement			
En général	Lx Cy		

Partie 2

Voici le même tableau. Ecrire dans la case L4 C2 une formule qui calcule la somme des deux nombres se trouvant dans le tableau.

	1	2	3	4	5	6	7	8	
1									
2		7							
3		4							
4									
5									
6									
7									
8									
9									

Dans le même tableau, placer le nombre 10 dans la case L2 C4 et le nombre 7 dans la case L3 C4. Ecrire la formule qui calcule la somme des deux derniers nombres dans la case L4 C4.

Dans le même tableau, placer le nombre 8 dans la case L4 C7 et le nombre 9 dans la case L5 C7. Ecrire la formule qui calcule la somme des deux derniers nombres dans la cellule L6 C7.

Ecrire une seule formule qui pourra être utilisée pour les trois calculs en supposant que la réponse est dans la case Lx Cy :

Partie 3

En travaillant toujours dans le même tableau, écrire 15 dans la case L1 C4. Ecrire la suite des nombres 5, 9, 11, 18 dans les cases L3 C1, L4 C1, L5 C1, L6 C1. Dans les cases L3 C2, L4 C2, L5 C2, L6 C2 écrire les formules donnant le produit de chacun des nombres de la suite par le nombre qui se trouve dans L1 C4.

	1	2	3	4	5	6	7	8	
1									
2									
3									
4									
5									
6									
7									
8									
9									

Ecrire une seule formule qui pourra être utilisée pour les quatre calculs en supposant que la réponse est dans la cellule Lx Cy :

Déroulement de la leçon

Partie 1

La situation est donnée aux élèves, qui doivent se débrouiller pour la résoudre sans aide. Elle ne contient pas de difficulté majeure. Elle requiert simplement des compétences en compréhension à la lecture. Chacun a le choix de la position des pions et des déplacements. Seule la formule générale doit être la même pour tous. En supposant que le pion rouge soit à l'intersection de la ligne 7 et de la colonne 3, la solution est :

	Pion vert	Pion jaune	Pion rouge
Départ			
En général	$Lx Cy$	$Lx Cy+2$	$L7 C3$

Le professeur doit veiller à ce que chaque élève respecte les notations imposées. Il peut à ce moment introduire le nouveau vocabulaire. Le pion jaune occupe une position relative à la position du pion vert. La position du pion jaune peut s'exprimer en fonction de la position du pion vert. Le pion rouge occupe une position absolue par rapport au pion vert. Il est fixe, sa position s'exprime par la référence de la case dans laquelle il se trouve, cette position est indépendante de la position du pion vert.

Partie 2

Les élèves continuent leur travail. Ils résolvent cette seconde partie.

	1	2	3	4	5	6	7	8
1								
2		7		10				
3		4		7				
4		$L2 C2 + L3 C2$		$L2 C4 + L3 C4$			8	
5							9	
6							$L4 C7 + L5 C7$	
7								
8								
9								

La seule formule qui pourra être utilisée pour les trois calculs en supposant que la réponse est dans la case $Lx Cy$ est $Lx-2 Cy + Lx-1 Cy$.

Si chaque calcul peut être réalisé à l'aide de références absolues, il faut remarquer que ce calcul peut être écrit une seule fois en travaillant avec des références relatives. Il faut faire prendre conscience aux élèves que les formules qu'ils écriront dans un tableur utiliseront par défaut les références relatives. Lorsqu'elles seront copiées d'une cellule vers une autre, elles réaliseront toujours le même « schéma » de calcul ; dans cette situation la formule permet d'« additionner le contenu de la case qui se trouve dans la même colonne deux lignes plus haut et le contenu de la case qui se trouve dans la même colonne une ligne plus haut ».

Partie 3

Le travail continué

	1	2	3	4	5	6	7	8
1				15				
2								
3	5	L3 C1 * L1 C4						
4	9	L4 C1 * L1 C4						
5	11	L5 C1 * L1 C4						
6	18	L6 C1 * L1 C4						
7								
8								
9								

La formule qui pourra être utilisée pour les quatre calculs en supposant que la réponse est dans la cellule Lx Cy est $Lx Cy-1 * L1 C4$.

Si chaque calcul peut être réalisé à l'aide de références absolues, il faut remarquer que ce calcul peut être écrit une seule fois en travaillant avec une combinaison de références relatives et absolues. Dans cette situation la formule permet de « multiplier le contenu de la case qui se trouve dans la colonne précédente sur la même ligne par le contenu de la case qui se trouve à la ligne 1 colonne 4 ». L'élève retrouve des éléments similaires à ceux de la première partie : pion vert, pion jaune, pion rouge.

Compétences terminales

- Traiter des informations.
- Construire un message cohérent et rigoureux. Les élèves doivent construire des formules correctes qui résolvent le problème et respectent les règles de notation.
- Utiliser les concepts, les modèles qui s'imposent pour une tâche donnée en les maîtrisant, en comprenant leur emploi. Les élèves doivent découvrir les notions de référence relative et de référence absolue à une cellule.
- Analyser des problèmes liés aux principes, aux théories.
- Analyser des informations.

Tâches du professeur

Aider les élèves dans leur recherche, faire apparaître le nouveau vocabulaire « relatif » et « absolu » au moment où les élèves ont perçu la différence entre les deux.

Faire respecter les notations.

Consignes aux élèves

Les élèves doivent construire des formules en respectant des règles strictes de notation et en passant à l'abstraction. Ils doivent, à la fin de la leçon, pouvoir associer le nouveau vocabulaire aux nouveaux concepts découverts.

Conclusions

Le travail sur le tableur permet l'utilisation de références absolues et de références relatives dans les formules. Par défaut, quelle que soit la notation utilisée pour référencer les cellules, un tableur travaille avec des références relatives. Pour que le tableur considère une référence comme absolue, l'utilisateur doit l'indiquer dans la formule.

Les élèves ont un soutien visuel (pion vert – pion jaune – pion rouge) de la notion de référence relative et absolue auquel un rattachement pourra être fait en cas de nécessité dans les exercices.

Il est intéressant de montrer aux élèves qu'un même tableur peut référencer les cellules avec des notations différentes et que, derrière les formules, il y a une programmation stricte.

Découverte du tableur

Description de la situation d'apprentissage

Ouvrir le tableur. Une grille apparaît composée de lignes et de colonnes. Rechercher les limites de ce tableau, déterminer à l'aide du tableur le nombre maximum de cellules que peut contenir la feuille de calcul. Présenter les résultats de manière lisible en indiquant dans le tableur les libellés des valeurs qui y apparaissent.

Déroulement de la leçon

Les élèves reçoivent l'énoncé et cherchent à se déplacer dans le tableur pour trouver le nombre de lignes et le nombre de colonnes de la feuille. Ces valeurs peuvent changer d'un logiciel à l'autre. C'est le moment pour eux de découvrir les possibilités de déplacement dans la feuille de leur tableur pour trouver au plus vite les valeurs recherchées. Une fois ces valeurs trouvées, une simple formule qui utilise la multiplication peut résoudre le problème. Le professeur peut aussi en profiter pour faire découvrir les fonctions qui calculent le nombre de lignes et de colonnes d'une plage de cellule ; cela évite les déplacements dans la feuille et pourra

éventuellement être utilisé dans d'autres exercices. Il reste à veiller à ce que la présentation des résultats soit correcte.

Compétences terminales

- Recueillir et traiter des informations
- Analyser des informations

Tâches du professeur

- Expliquer la notion de cellule et le système de repérage dans le logiciel choisi.
- Présenter les différents moyens de se déplacer dans la feuille de calculs.
- Faire comprendre le principe de fonctionnement d'un tableur.

Consignes aux élèves

Les élèves doivent découvrir les déplacements dans une feuille de calculs et rechercher le nombre de lignes et le nombre de colonnes de la feuille ; ils vont apprendre à encoder une formule ; ils vont présenter leurs résultats. Ils devront observer que l'alignement dans une cellule est lié par défaut à la nature de l'information qui s'y trouve.

Conclusions

Les notions de base du tableur sont mises en place. Il faut proposer des exercices où interviendra la mise au point de formules simples pour fixer les notions apprises dans les deux premières situations.

Encodage de formules et recopiage - Mise en page

Description de la situation d'apprentissage Partie 1

Un voyageur de commerce se déplace pour son travail dans l'Euroland, aux Etats-Unis et en Suisse. Il souhaite disposer d'un tableau de conversion des monnaies de ces pays pour ses dépenses personnelles. Le tableau doit convertir une vingtaine de valeurs à savoir 5 €, 10 €, 15 € ... Il doit être de petite taille pour être rangé dans son portefeuille.

Le voyageur désire disposer d'un même tableau de conversion pour discuter de ses contrats. Les valeurs de 1000 €, 1500 €, 2000 €, ... doivent être converties.

Il souhaite, le cas échéant, pouvoir imprimer un tableau similaire commençant à une valeur de son choix avec un incrément de son choix.

Les tableaux doivent être imprimés avec une présentation du type :

Euros	USD	CHF
5		
10		
15		
...		

Taux USD	0,93
Taux CHF	0,68

Partie 2

Les taux de conversion sont en évolution permanente. Transforme le travail pour avoir une mise à jour des taux via un site Internet.

Déroulement de la leçon

Partie 1

La situation est présentée aux élèves accompagnée d'une feuille vierge du tableur qu'ils devront compléter avant de travailler sur ordinateur en mentionnant par des couleurs différentes les cellules encodées et les cellules calculées et en indiquant les formules à utiliser. Les élèves travaillent en utilisant les notations de leur tableur. Pour garder ces notes les plus générales possible, ce sont les notations de la première situation qui sont reprises. Des formules indiquées s'avèrent inutiles ; les élèves utiliseront la poignée de recopie de leur tableur. C'est le moment de la faire découvrir. Ainsi, la suite arithmétique des valeurs des monnaies en euros peut être générée par les valeurs de départs 5 et 10 en Excel, par exemple, en utilisant simplement la poignée de recopie.

	Cellules encodées					
	Cellules calculées					
	1	2	3	4	5	6
1	Euros	USD	CHF		Taux USD	0,93
2					Taux CHF	0,68
3	5	Lx Cy-1 * L1 C6	Lx Cy-2 * L2 C6			
4	Lx-1 C1 + 5	Lx Cy-1 * L1 C6	Lx Cy-2 * L2 C6			
5	Lx-1 C1 + 5	Lx Cy-1 * L1 C6	Lx Cy-2 * L2 C6			
	...					

Les élèves doivent bien utiliser les références relatives et absolues dans les formules ; s'ils éprouvent des difficultés, le rappel des pions vert, jaune et rouge les met très vite sur le bon chemin.

Une fois la solution bien mise en place sur papier, les élèves travaillent sur ordinateur et doivent faire des recherches pour répondre à la contrainte de mise en page du format porte-feuille. Le tableau des dépenses personnelles est ainsi terminé et peut être imprimé.

Le tableau des valeurs permettant de discuter des contrats s'obtient simplement en modifiant les valeurs de la première colonne du travail précédent.

La troisième partie du problème doit permettre de commencer à une valeur au choix du voyageur de commerce avec un incrément de son choix. La solution la plus intéressante est d'introduire deux données séparées de façon à ce que l'utilisateur du tableau ne doive pas modifier les formules. Le travail sur papier donne :

	1	2	3	4	5	6
1	Euros	USD	CHF		Taux USD	0,93
2					Taux CHF	0,68
3	L3 C6	Lx Cy-1 * L1 C6	Lx Cy-2 * L2 C6		Valeur de départ	
4	Lx-1 C1 + L4 C6	Lx Cy-1 * L1 C6	Lx Cy-2 * L2 C6		Incrément	
5	Lx-1 C1 + L4 C6	Lx Cy-1 * L1 C6	Lx Cy-2 * L2 C6			
	...					

Seules les cellules L3 C6 et L4 C6 doivent pouvoir être modifiées par l'utilisateur, les autres cellules peuvent être protégées contre l'écriture. La méthode de protection d'une feuille de calcul peut être vue ici.

Remarques

Le travail multi-feuilles est une matière facultative du programme. Cette situation est cependant une bonne occasion pour en parler. Pour garder des traces des trois parties de l'exercice, il suffit de réaliser la première partie en sélectionnant trois feuilles. Tout le travail réalisé sur une feuille se répercute sur les deux autres et l'élève peut alors modifier la deuxième feuille pour réaliser la deuxième partie de la situation, puis la troisième feuille pour la troisième partie.

La situation peut aussi servir à l'apprentissage de l'utilisation du nom des cellules. Il est possible, par exemple, de nommer les cellules contenant les taux de conversion, la valeur de départ et l'incrément et d'utiliser leur nom dans les formules.

Partie 2

Cette situation a été réalisée dans des conditions bien précises :

- le tableur utilisé était EXCEL 2000,
- le site utilisé était « <http://bourse.tfl.fr/devises.phtml> » en février 2003.

L'utilisation d'un autre tableur demandera une adaptation des notes qui vont suivre.

Tout changement survenu dans la structure du site après cette date peut rendre la suite de ces notes inutilisables.

Sur une feuille du classeur, il faut importer les données venant du Web en prenant dans la barre des menus **Données/Données externes/Nouvelle requête sur le Web**. Dans la fenêtre qui s'ouvre, il faut donner l'adresse du site Web sur lequel il faut aller chercher les données (<http://bourse.tfl.fr/devises.phtml>), prendre uniquement les tables au format HTML complet. Le tableau des taux de conversion se charge alors dans la feuille. Les cellules du tableau du voyageur de commerce contenant la valeur des taux de conversion sont alors modifiées. Leur contenu sera le résultat d'une formule simple : « = référence de la cellule contenant le taux de change approprié dans la feuille de données importée du Web ». Des mises à jour des taux et,

dès lors, des tableaux pour le voyageur de commerce peuvent ainsi être effectuées régulièrement.

Pour que tout fonctionne bien, il faut veiller à ce que le format des nombres importés du Web corresponde au format des nombres sur l'ordinateur de l'élève.

Compétences terminales

- Recueillir et traiter des informations.
- Analyser des informations pour résoudre un problème.
- Mobiliser ses connaissances dans diverses disciplines pour résoudre un problème réel.
- Maîtriser des compétences de communication.

Tâches du professeur

- Présenter le problème et s'assurer que les élèves connaissent les formules de conversion d'une monnaie à une autre.
- Obliger les élèves à prendre de bonnes habitudes en différenciant les cellules encodées des cellules calculées, en utilisant des couleurs par exemple, ou en établissant une liste des cellules utilisées dans le travail, en précisant si celles-ci sont calculées ou encodées.
- Convaincre les élèves que c'est le logiciel qui doit « travailler » et pas eux en utilisant un maximum de cellules avec des formules. L'idée d'une valeur de départ et d'un incrément dans des cellules annexes doit être développée.
- S'assurer que les notions de références absolues et relatives sont bien utilisées.

Pour la deuxième partie, faire chercher par les élèves les plus avancés dans le travail la marche à suivre pour récupérer des données d'une page Web dans le tableur ; proposer ensuite à ces élèves d'expliquer aux autres la façon de procéder. Le professeur peut laisser rechercher un site donnant les taux de change ou en proposer aux élèves. Tous les sites ne permettent pas l'importation de données. En voici deux qui fonctionnent bien en date du 15/02/2003 :

<http://www.boursorama.com/devises/devises.phtml>

<http://bourse.tfl.fr/devises.phtml>

Il est important de vérifier avant le cours que la requête Web fonctionne pour avoir un site sûr à proposer aux élèves. En effet, les sites Web évoluent et des données accessibles un jour peuvent ne plus l'être le lendemain.

Ces sites travaillent avec le « point » comme séparateur décimal ; il faudra attirer l'attention des élèves sur le fait d'avoir un paramétrage correspondant dans leur logiciel.

Guider ensuite les élèves pour faire importer les données dans une feuille du tableur et faire modifier la cellule contenant les taux pour assurer sa mise à jour via les données récupérées sur Internet.

Consignes aux élèves

- Identifier les cellules qui contiennent des valeurs constantes, et les cellules dont les valeurs sont calculables par une formule. Ajouter au tableau des cellules pour la valeur de départ et pour l'incrément.
- Trouver les formules à recopier.
- Encoder les différentes valeurs et formules en « copiant » au maximum.
- Mettre en forme le tableau pour l'imprimer en taille réduite.
- Pour la deuxième partie, rechercher un site Internet donnant les taux de change.
- Découvrir la création de requête Web en vue d'utiliser des données externes au tableur et assurer l'utilisation de ces données comme taux de change.

Conclusions

Les notions de recopie de formules et de mise en forme sont mises en place. La notion de nom de cellule peut aussi être abordée. Il faut proposer des exercices pour fixer les notions apprises.

Encodage de formules (le si...alors ...), format des nombres

Introduction

Cette situation a été réalisée au moment du Mondial de football en 2002. Elle peut facilement s'adapter à d'autres événements sportifs plus récents. Les élèves connaissent la structure alternative par le cours de programmation.

Description de la situation d'apprentissage : la Coupe du monde 2002

La Belgique fait partie du groupe H au Mondial. Elle rencontrera le Japon, la Tunisie et la Russie. Un match gagné rapporte 3 points, un match nul rapporte 1 point et une défaite 0. Alors qu'aucun match n'a été joué, il est possible de préparer tous les calculs à réaliser pour comptabiliser les points des différentes équipes. En encodant les résultats des matches, le calcul des points et la comptabilisation des buts doivent se faire automatiquement. Ton travail doit respecter la mise en forme suivante :

GROUPE H			Les points	Belgique	Tunisie	Russie	Japon
mardi 4 juin	Japon- Belgique	2 - 2		1			1
mercredi 5 juin	Russie- Tunisie	2 - 0			0	3	
dimanche 9 juin	Japon- Russie	1 - 0				0	3
Lundi 10 juin	Tunisie- Belgique	1 - 1		1	1		
vendredi 14 juin	Belgique- Russie	3 - 2		3		0	

vendredi 14 juin	Tunisie- Japon	0 - 2		0	3
total des points				5	7

Les buts	Belgique	Tunisie	Russie	Japon
total buts marqués	6	1	4	5
total buts encaissés	5	5	4	2
différences	+1	-4	0	+3

Déroulement de la leçon

Les élèves doivent d'abord travailler sur papier afin de se concentrer sur le problème à résoudre. Ils utilisent la même méthode de travail que dans la situation précédente.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	GROUPE H								Les points	Belgique	Tunisie	Russie	Japon
2	mardi 4 juin	Japon - Belgique			2 - 2					(1)			(2)
3	mercredi 5 juin	Russie - Tunisie			2 - 0						(3)	(4)	
4	dimanche 9 juin	Japon - Russie			1 - 0							(5)	(6)
5	lundi 10 juin	Tunisie - Belgique			1 - 1					(7)	(8)		
6	vendredi 14 juin	Belgique - Russie			3 - 2					(9)		(10)	
7	vendredi 14 juin	Tunisie - Japon			0 - 2						(11)		(12)
8									total des points	(13)	(14)	(15)	(16)
9													
10									Les buts	Belgique	Tunisie	Russie	Japon
11									total buts marqués				
12									total buts encaissés				
13									différences				

Les élèves utilisent les notations de leur tableur, nous utilisons dans ces notes des notations plus générales vues précédemment.

Pour connaître les points obtenus par la Belgique au premier match, il faut comparer le nombre de buts marqués par les deux équipes. Il faut encoder les résultats du match en trois cellules.

La valeur de la cellule L2 C10 (1) répondra à des conditions. Le professeur invite les élèves à réaliser un algorithme comme ils le font au cours de programmation :

```

Si (L2 C5 = L2 C7)
Alors
L2 C10 ← 1
Sinon
Si (L2 C5 < L2 C7)
    Alors    L2 C10 ← 0
    Sinon    L2 C10 ← 3
Finsi
Finsi

```

Cet algorithme doit être traduit sous forme de formule dans le tableur utilisé.

Les cellules L2 C13 (2), L3 C11 (3),... , L7 C13 se travaillent de la même façon.

La formule à placer dans la cellule L8 C10 (13) n'est qu'une simple addition $L2 C10 + L5 C10 + L6 C10$. Des formules semblables rempliront les cellules (14) (15) et (16)

Le calcul des buts marqués et des buts encaissés se fait aussi par une simple addition. La différence entre les buts marqués et les buts encaissés est facile, elle est la seule formule qui peut être recopiée.

Sur ordinateur, il reste à guider les élèves pour la modification du format des nombres représentant la différence des buts marqués et des buts encaissés pour faire apparaître les signes.

Compétences terminales

- Recueillir et traiter des informations.
- Analyser des informations pour résoudre un problème.
- Mobiliser ses connaissances dans diverses disciplines pour résoudre un problème réel.

Tâches du professeur

- Présenter le problème et s'assurer que tous les élèves ont compris le système d'attribution des points à un match de foot.
- Convaincre les élèves que c'est le logiciel qui doit « travailler » et pas eux en utilisant un maximum de cellules avec des formules. L'idée d'utiliser une fonction si ... alors ... pour attribuer les points à l'équipe doit apparaître.
- Expliquer les formats des nombres pour permettre l'apparition des signes « + » dans les différences de but.
- Vérifier le respect de l'alignement des données.
- Proposer l'idée de verrouiller un maximum de cellules pour ne permettre que la modification des résultats des matches.

Consignes aux élèves

- Identifier les cellules qui contiennent des valeurs constantes, et les cellules dont les valeurs sont calculables par une formule.
- Trouver les formules à recopier.
- Mettre en forme le tableau en respectant les formats des nombres imposés et les alignements imposés.

Conclusions

La notion de format d'affichage des nombres a trouvé son utilité. La fonction si...alors.... a été utilisée. Ces notions doivent faire l'objet d'exercices.

Gestionnaire de bases de données

Méthode d'analyse d'une base de données

Introduction

Avant de se lancer dans l'étude d'un logiciel de bases de données, il est bon de faire réfléchir les élèves à la façon d'organiser les données d'un problème. Le travail qui va suivre est basé sur le modèle conceptuel des données. Des élèves de 6^e, débutants en Système de Gestion de Bases de Données Relationnelles, ont travaillé avec cette méthode qui a permis une bonne compréhension de la matière sur les index et sur les relations dans un Système de Gestion de Bases de Données Relationnelles.

Description de la situation d'apprentissage

Un informaticien indépendant réalise des programmes informatiques pour des clients. Il veut gérer son travail à l'aide d'une base de données. Tu dois réfléchir aux données à organiser et à la façon de les organiser.

Tâches du professeur

La classe travaille sous la direction du professeur, qui va essayer de développer chez les élèves une méthode d'analyse d'un problème de base de données.

Compétences terminales

- Recueillir et traiter des informations.
- Analyser des informations pour résoudre un problème.

Étape n°1 : les données

Comment les obtenir ? Dans la pratique, c'est le demandeur de la base de données qui fournira les données à y incorporer. Dans la situation donnée, la classe va essayer de les imaginer.

Les élèves ont réfléchi de la façon suivante : le problème parle de clients, les données concernant les clients seront les

- Nom du client,
- Prénom du client,
- Adresse du client,
- Code Postal du client,
- Localité du client,
- Date De Naissance du client,
- Téléphone Eventuel du client,
- Gsm Eventuel du client,
- N°De Tva Eventuel du client,
- L'objet Du Programme Commandé par le client,
- Le Langage Utilisé pour le programme,
- La Durée De Conception du programme,
- La Date De La Commande du programme

Cette liste peut changer en fonction des idées des élèves. Elle aurait pu être au départ imposée dans la situation, mais nous pensons qu'il est intéressant de faire rechercher par les élèves les données intéressantes pour ce problème.

Etape n°2 : l'organisation

Une fois cette liste établie, les élèves sont amenés à organiser leurs données. La classe a proposé d'établir une fiche par client, qui contiendrait toutes les données énumérées plus haut. Le professeur est amené à soulever le problème suivant :

Un même client peut souhaiter la réalisation de plusieurs programmes. Comment travailler ?

Les élèves ont proposé deux solutions :

La première consiste à allonger la liste des données en gardant les renseignements sur le client et en augmentant la quantité de données concernant les programmes pour différencier les programmes demandés par le client.

- Nom,
- Prénom,
- Adresse,
- Code postal,
- Localité,
- Date de naissance,
- Téléphone éventuel,
- Gsm éventuel,
- N°de TVA éventuel,
- L'objet du programme 1,
- Le langage utilisé dans le programme 1,
- La durée de conception du programme 1,
- La date de la commande du programme 1,
- L'objet du programme 2,

- Le langage utilisé dans le programme 2,
- La durée de conception du programme 2,
- La date de la commande du programme 2,
- L'objet du programme 3,
- Le langage utilisé dans le programme 3,
- La durée de conception du programme 3,
- La date de la commande du programme 3,
- ...

La deuxième consiste à établir une fiche par programme.

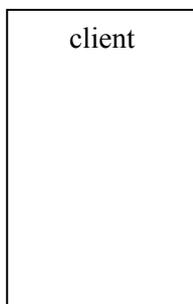
- | | | |
|--|--|---|
| <ul style="list-style-type: none"> ▪ Nom, ▪ Prénom, ▪ Adresse, ▪ Code postal, ▪ Localité, ▪ Date de naissance, ▪ Téléphone éventuel, ▪ Gsm éventuel, ▪ N°de TVA éventuel, ▪ L'objet du programme 1, ▪ Le langage utilisé dans le programme 1, ▪ La durée de conception du programme 1, ▪ La date de la commande du programme 1, | <ul style="list-style-type: none"> ▪ Nom, ▪ Prénom, ▪ Adresse, ▪ Code postal, ▪ Localité, ▪ Date de naissance, ▪ Téléphone éventuel, ▪ Gsm éventuel, ▪ N°de TVA éventuel, ▪ L'objet du programme 2, ▪ Le langage utilisé dans le programme 2, ▪ La durée de conception du programme 2, ▪ La date de la commande du programme 2, | <ul style="list-style-type: none"> ▪ Nom, ▪ Prénom, ▪ Adresse, ▪ Code postal, ▪ Localité, ▪ Date de naissance, ▪ Téléphone éventuel, ▪ Gsm éventuel, ▪ N°de TVA éventuel, ▪ L'objet du programme 3, ▪ Le langage utilisé dans le programme 3, ▪ La durée de conception du programme 3, ▪ La date de la commande du programme 3 ... |
|--|--|---|

Dans les deux cas, les données concernant les clients se mélangent aux données concernant les programmes et ces deux organisations amènent des redondances.

Une bonne organisation d'une base de données doit éviter les redondances. Les élèves sont amenés à décomposer les données en « objets élémentaires ». Un objet est un ensemble d'éléments de même nature. Les objets sont les composants de base du modèle conceptuel de données.

Dans notre problème, deux objets apparaissent, qui seront symbolisés par un rectangle : l'objet client et l'objet programme.

Objet client



Objet programme



Ces deux objets vont se décomposer en un certain nombre de propriétés. Ces propriétés sont les données élémentaires qui vont apporter une connaissance de chaque élément de l'objet.

Notre objet client va ainsi se décomposer en 10 propriétés :

- Nom,
- Prénom,
- Adresse,
- Code postal,
- Localité,
- Date de naissance,
- Téléphone éventuel,
- Gsm éventuel,
- N°de TVA éventuel

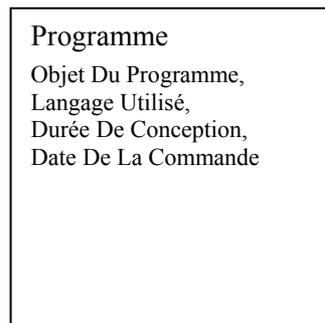
Notre objet programme va se décomposer en 4 propriétés :

- L'objet du programme,
- Le langage utilisé,
- La durée de conception,
- La date de la commande

Les deux objets seront symbolisés par deux rectangles contenant les propriétés



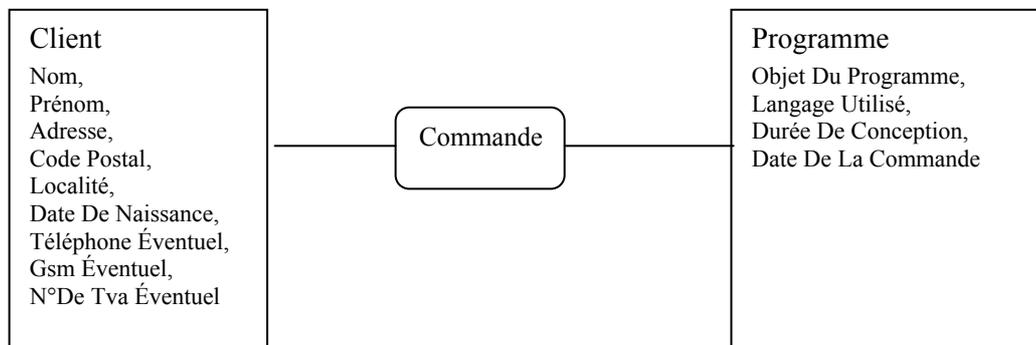
Objet client avec ses propriétés



Objet programme avec ses propriétés

Les propriétés des objets n'apparaissent ainsi qu'une seule fois, il n'y a plus aucune redondance. C'est une des conditions d'un bon travail.

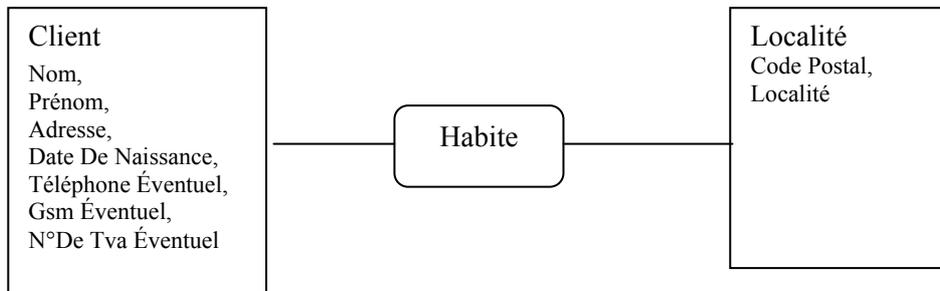
Il est cependant évident de se dire que ces deux objets ont un lien entre eux. Le client « commande » un programme. Ce lien va être symbolisé de la façon suivante :



Le travail sur les bases de données se fait dans un univers relationnel. Un lien entre deux objets sera appelé une relation.

Etape n°3 : un peu plus sur les relations

La réflexion peut être poussée plus loin. Par souci d'automatisation, on peut se dire qu'un même code postal amenant toujours une même localité, il serait intéressant d'encoder une seule fois les localités à associer aux codes postaux. L'objet « client » peut éclater en deux objets dépendants l'objet « client » et l'objet « localité ». Ces deux objets seront liés par la relation « habite ».



Etape n°4 : l'identification des données

Le professeur peut soulever un nouveau problème. Dans le problème traité, des cas ennuyeux pourraient se produire. Deux clients peuvent éventuellement avoir les mêmes noms et prénoms. Comment les distinguer ?

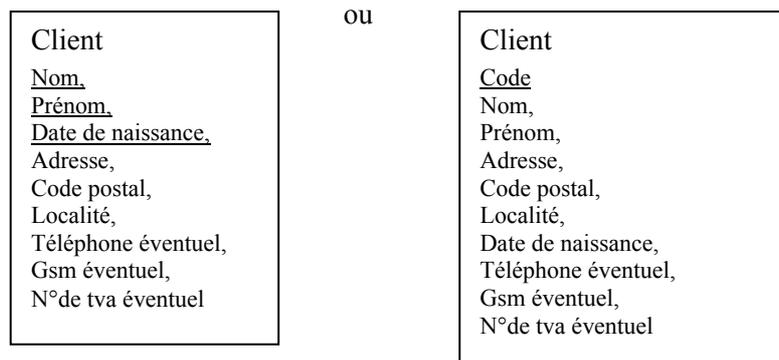
Il est indispensable que chaque élément d'un objet puisse être identifié de façon unique, que chaque élément puisse se distinguer des autres.

Des idées de solutions sont proposées. En voici deux qui sont acceptables.

- Chaque client peut être identifié par son nom et son prénom, en tenant compte en plus de sa date de naissance. On peut supposer que ces trois éléments combinés peuvent identifier les clients de façon unique.
- Chaque client peut recevoir un code qui lui sera attribué par le programmeur lors de sa première commande.

Dans les deux cas, la propriété ou les propriétés qui permettront l'identification du client seront appelées l'identifiant. Dans la première solution, l'identifiant est dit composé, dans l'autre, l'identifiant est dit élémentaire.

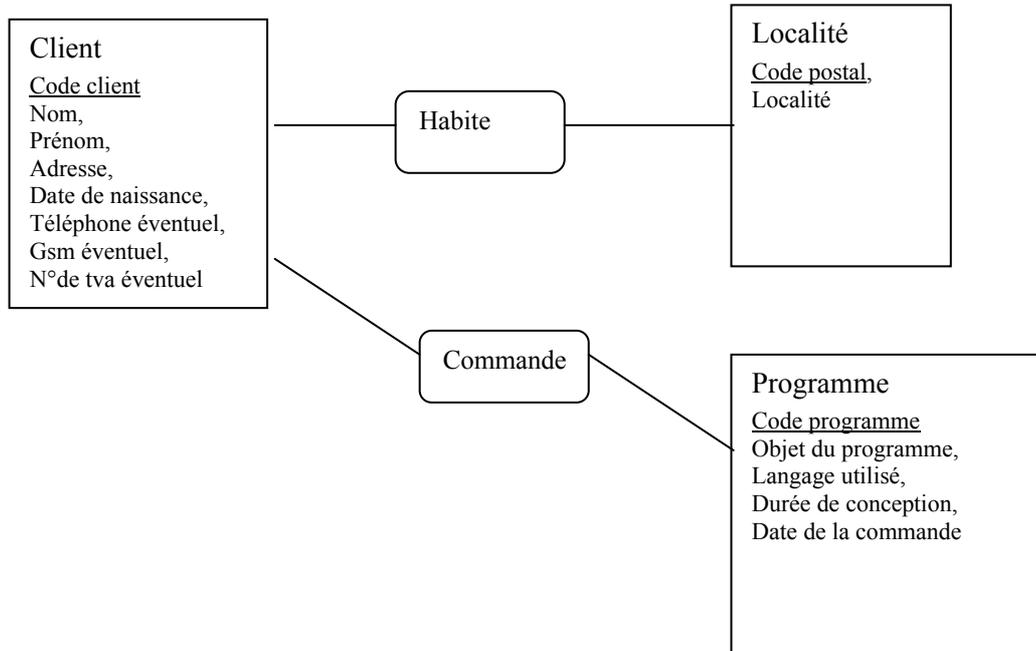
Dans le schéma, l'identifiant sera souligné. On aura :



Si les clients du programmeur sont des indépendants ou des entreprises et ont tous un numéro de TVA, ce numéro pourrait aussi servir d'identifiant. On parle alors d'identifiant externe, en opposition au code attribué au client par le programmeur qui sera appelé un identifiant interne.

L'objet programme doit lui aussi avoir un identifiant. Après la réflexion sur l'objet client, la solution qui paraît la plus adéquate est d'attribuer un code à chaque programme.

Tout le cheminement qui précède nous mène au schéma :



On attirera l'attention des élèves sur le choix de la clé primaire dans le cas des localités : le code postal est-il une bonne clé primaire ? A quelles conditions ? Quelles alternatives ?

Tous ensemble, élèves et professeur résumant alors les règles à respecter dans l'analyse d'un problème de bases de données.

Règles à respecter dans l'analyse d'une base de données

- Les objets sont des ensembles d'éléments de même nature.
- Chaque objet a des propriétés. Ces propriétés sont non redondantes.
- Un objet possède un identifiant qui permet de distinguer de façon unique chaque élément. L'identifiant peut être élémentaire ou composé, interne ou externe.
- Deux objets peuvent être reliés entre eux.
- Un objet peut être scindé en deux objets dépendants.

Etape n°5 : les cardinalités

Dans une relation, un élément d'un objet peut être relié à un élément d'un autre objet. Différents cas peuvent se produire.

Dans le cas de la relation « habite », un client habite une seule localité. Cependant, une localité peut être répertoriée dans l'objet sans qu'un client y habite ou, au contraire, une même localité

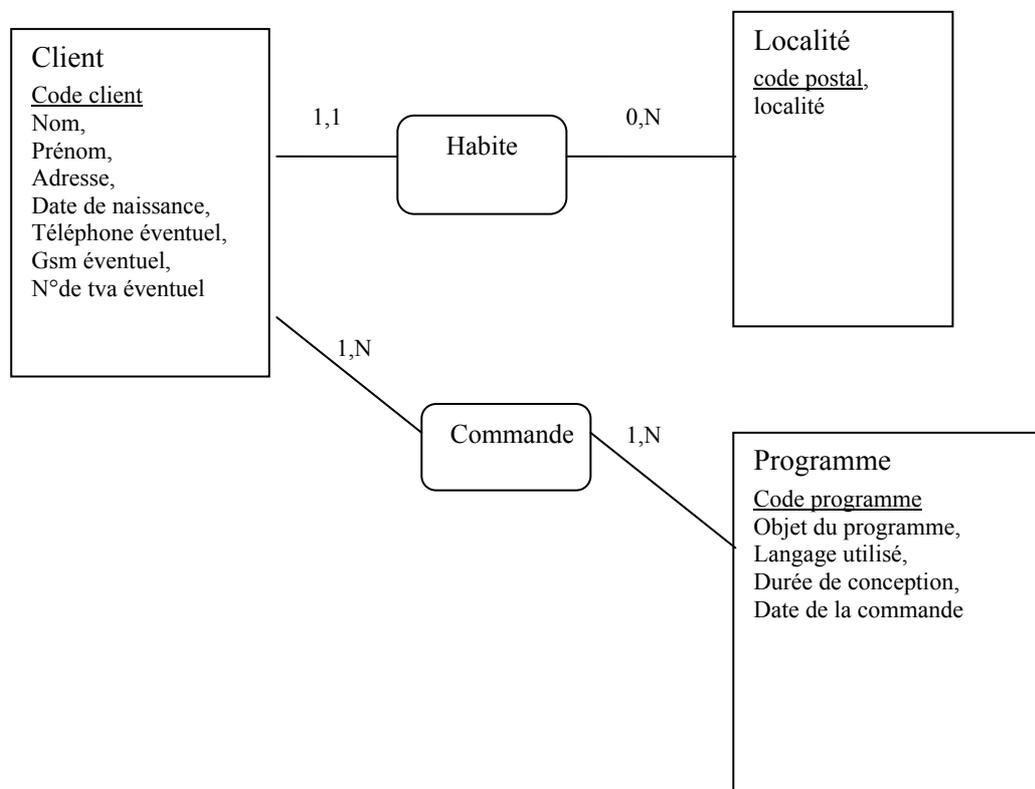
peut être le lieu de résidence de plusieurs clients. Toutes les relations ne seront donc pas du même type.

Chaque relation a une cardinalité qui exprime le nombre minimum et maximum de fois qu'un élément d'un objet est en relation avec des éléments de l'autre objet.

La relation « habite » est une relation 1,1 si elle est considérée du point de vue du client. Cette relation, considérée du point de vue des localités, est une relation 0,N. Une localité peut n'être habitée par aucun client ou, au contraire, être habitée par un nombre N de clients.

Dans le cas de la relation « commande », un client peut commander plusieurs programmes, c'est une relation 1,N du point de vue du client. Un programme a été commandé par un client, éventuellement un même programme peut être commandé par plusieurs clients, c'est une relation 1,N du point de vue de l'objet « programme ».

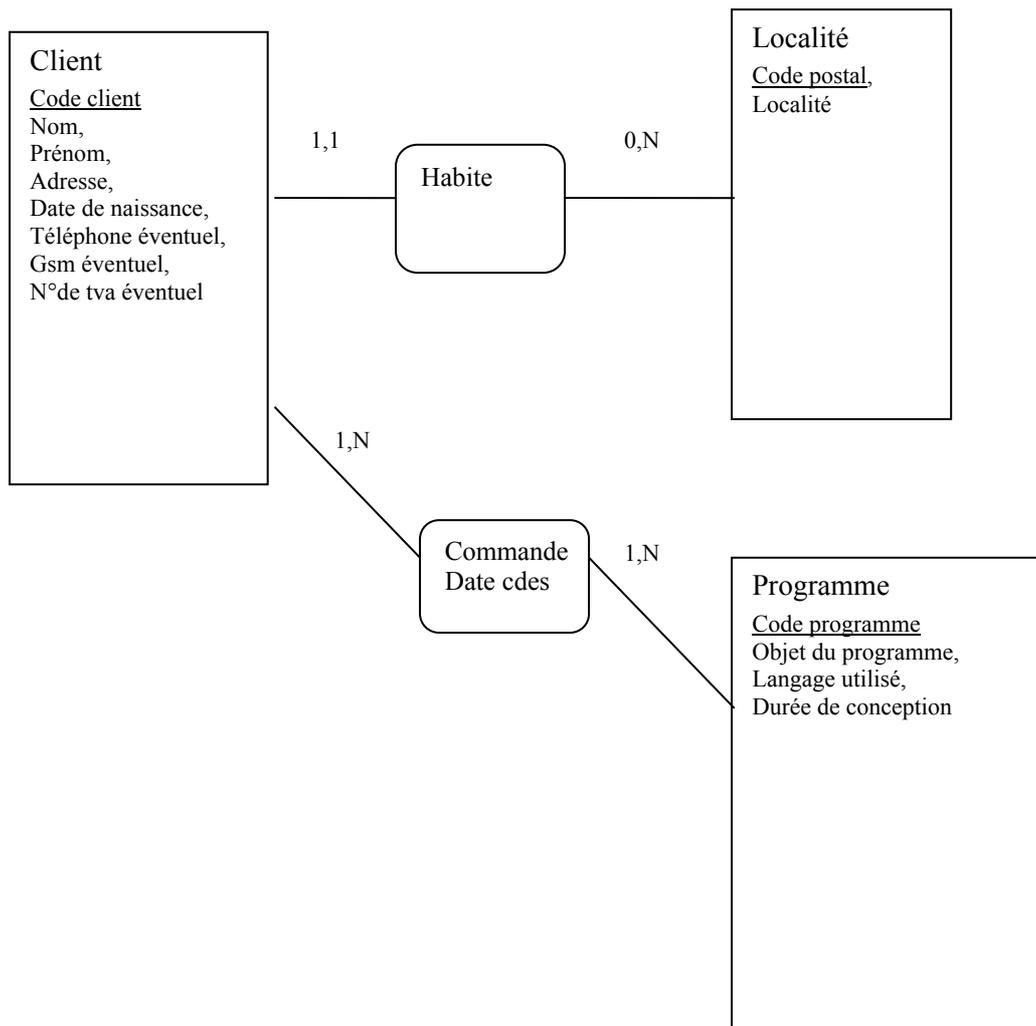
Ces valeurs peuvent compléter le schéma.



Règles à respecter dans l'analyse d'une base de données (suite)

- Le nombre minimum et maximum de fois qu'un élément d'un objet est en relation avec des éléments de l'autre objet

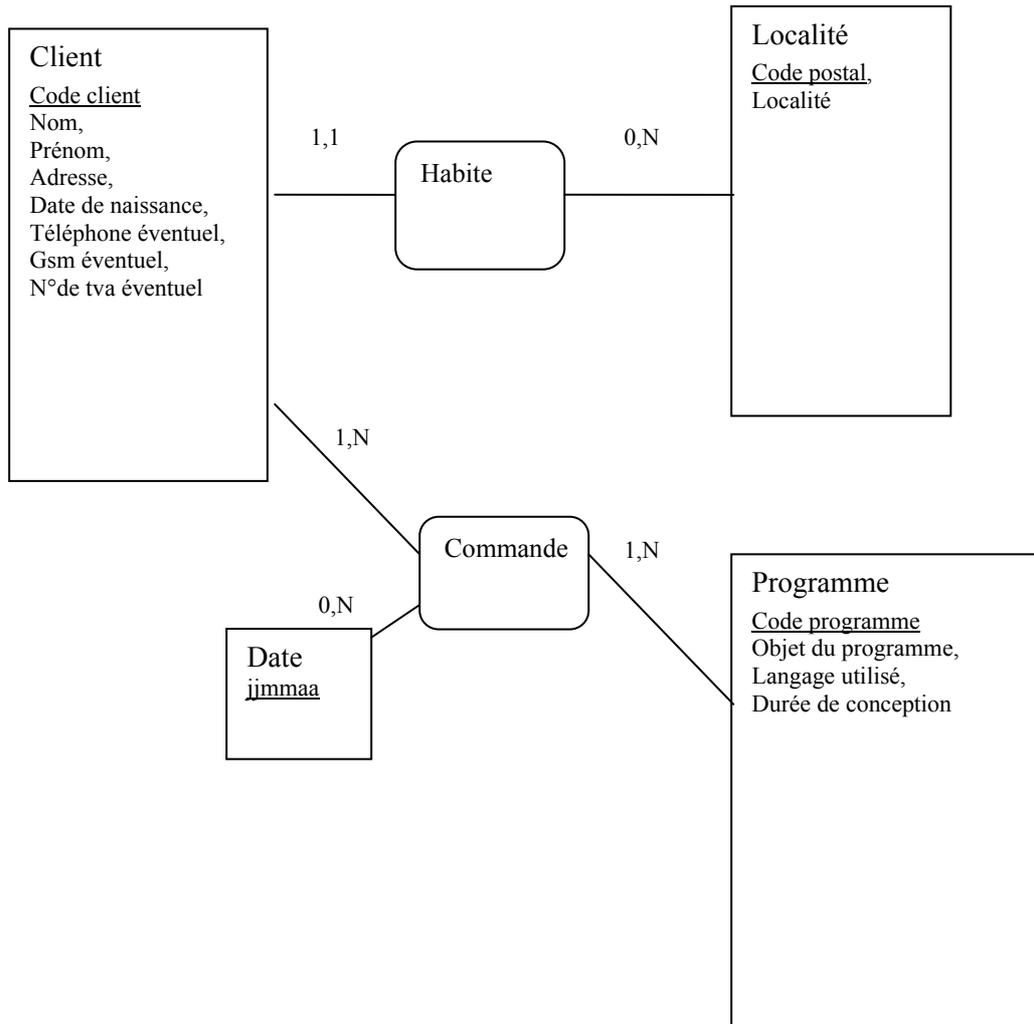
Il reste un problème à résoudre. Ci-dessus, dans les relations, on a supposé qu'un même programme allait pouvoir servir à plusieurs clients. La propriété « date de la commande » pose alors un problème de redondance dans l'objet programme, puisque un programme peut avoir été commandé à plusieurs dates différentes. Cette date est en réalité un complément à l'action de « commander ». Nous l'écrivons dans la relation. Ce qui donnera le schéma suivant :



Il est aussi possible d'envisager un nouvel objet « date ». Cet objet sera lié au client et au programme. Un client commandera un programme à une certaine date. La nouvelle relation est appelée une relation « ternaire » car elle fait intervenir trois objets.

A une certaine date, il est possible qu'il n'y ait aucune commande ou, au contraire, qu'il y ait N commandes.

Le schéma « final » (mais qui pourrait encore évoluer) devient :



Conclusions

Cette méthode d'analyse a permis aux élèves d'établir des règles précises à respecter lors de l'analyse d'un problème de base de données. Elle les a beaucoup aidés dans les exercices qui ont suivi et leurs a permis d'associer les outils d'un logiciel de gestion de bases de données à des notions connues.

Programmation procédurale

Introduction

Lors de nos quelques années d'enseignement, nous avons pu constater que les élèves ne parviennent pas à assimiler toutes les notions algorithmiques ensemble. Nous avons donc élaboré un document dans lequel une approche progressive des notions de base de la programmation procédurale est favorisée, afin de faciliter la compréhension. Chaque notion ainsi introduite doit être prolongée par de nombreux exercices et doit être complétée par un cours qui structure les notions abordées et leurs conditions d'existence. La matière qui se trouve dans ce document couvre 1 année de cours à 2 périodes par semaine en 5^e Sciences-Informatique. En procédant de la sorte, nous avons pu constater une diminution importante du nombre d'échecs. Nous avons refusé la fatalité ; nous avons refusé d'accepter le fait que "le cours de programmation est très dur et donc, il y a beaucoup d'échecs, c'est normal". Cette manière de travailler a permis à beaucoup d'élèves d'entrer progressivement dans les processus de réflexion liés à la programmation et de parvenir à résoudre de nombreux problèmes (relativement complexes) en fin de cinquième.

Nous avons observé le jeu du "Pacman", ce petit personnage jaune qui se déplace dans un labyrinthe pour manger des boules blanches, où le joueur gagne la partie quand toutes les boules ont été mangées. Nous avons pensé qu'il était possible de travailler dans le même sens, et nous avons choisi de déplacer une croix sur une grille afin de faire disparaître des boules présentes aléatoirement sur cette grille. La situation de départ est simple et permet l'introduction d'une instruction algorithmique de base (affectation). Cette situation est améliorée au fur et à mesure pour permettre, à chaque fois, l'introduction d'une nouvelle notion. Chaque étape de la conception sera accompagnée d'un petit programme que les élèves pourront aisément faire tourner sur leurs machines. La première situation est accompagnée du programme "Pacman01.exe", la deuxième situation du programme "Pacman02.exe" ...

Le défi que nous proposons aux élèves est d'analyser la programmation de ce petit jeu simplifié en n'utilisant que les notions de base de la programmation (affectation, écriture, lecture, alternative, répétition). Ils vont être amenés, notamment, à manipuler la croix et à dialoguer avec l'ordinateur. C'est ce dialogue qui sera traduit en pseudo-langage.

L'avantage de ce type d'exercice est qu'il demande une participation active de l'élève, c'est lui qui va réaliser les actions. L'élève s'implique "physiquement" dans l'exercice, ce qui peut l'aider dans la compréhension des processus de la programmation.

Compétences

Les **compétences terminales** qui sont particulièrement développées sont :

- Analyser les schémas fonctionnels, structurels d'un système ou d'un objet technique.
- Analyser et résoudre des problèmes.
- Décrire à partir des spécifications du cahier des charges, le comportement d'une fonction.
- Vérifier par simulation et / ou par des essais et des mesures, la conformité d'une fonction.

Les **compétences transversales** qui sont particulièrement développées sont :

- Définir et formuler une difficulté technique à résoudre afin d'y apporter une solution.
- Analyser des informations.
- Utiliser les concepts en les maîtrisant, en comprenant leur emploi et en étant conscient de leurs possibilités et de leurs limitations.
- Elaborer des concepts pour une tâche technique à effectuer.
- Choisir parmi des concepts le plus adéquat pour une tâche technique à effectuer et pouvoir justifier ce choix.
- Exploiter et s'appropriier un message.

Séquence et affectation

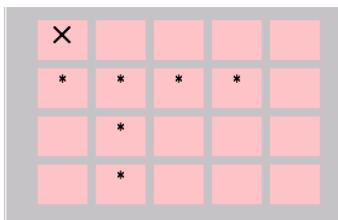
Nous nous sommes rendu compte que l'affectation n'est pas si simple à comprendre. Par exemple, l'instruction $A \leftarrow 1$ est évidente, mais l'instruction $A \leftarrow A + 1$ l'est beaucoup moins. La situation proposée permet à l'élève de rencontrer "naturellement" cette affectation grâce aux modifications de positions horizontale et verticale.

L'élève va déplacer la croix (à l'aide des boutons présents dans le programme) et constater immédiatement la modification dans les positions. Il suffira ensuite de traduire cela en pseudo-code (avec des notations imposées par le professeur). Cette manière de procéder donne à l'élève une représentation "visuelle" de la notion d'affectation.

Savoirs requis

Nous considérons que les élèves connaissent les variables et les types de données.

Description de la situation d'apprentissage



La croix se déplace sur une grille rectangulaire de dimensions finies (4 lignes et 5 colonnes) et doit manger les points noirs qui se situent sur des cases précises (obtenues aléatoirement) en effectuant le moins de déplacements possible. Il suffit que la croix passe sur une case contenant un point noir pour que celui-ci soit mangé. Dans cette situation, la croix démarrera toujours de la 1^{ère} case en haut à gauche.

- Vous devez décrire vos actions lors du jeu, afin qu'un copain de classe puisse comprendre ce que vous avez fait.
- Chaque élève doit alors échanger ce qu'il a écrit avec un copain de classe et celui-ci doit comprendre, sans ambiguïté, les actions effectuées par le premier élève.

Cette situation s'accompagne du petit programme Pacman01.exe. On y trouve la grille sur laquelle se positionnent aléatoirement 6 points et la croix dans le coin supérieur gauche. De plus, deux zones montrent les valeurs des deux variables. Quatre boutons permettent de manipuler les déplacements de la croix. Un bouton permet de lancer le jeu autant de fois que souhaité et un autre de quitter définitivement le jeu.

Remarque : au départ, on peut simuler ce programme par une grille sur papier et faire déplacer la croix sur cette grille, car le petit programme que nous proposons indique déjà les actions élémentaires et les variables à utiliser, ce qui doit, bien sûr, être découvert par les élèves.

Proposition de solution

Il faut amener les élèves à découvrir que chaque case de la grille va être repérée par deux variables que nous noterons PositionV (pour la position en hauteur - ligne) et PositionH (pour la position en largeur - colonne).

PositionV :

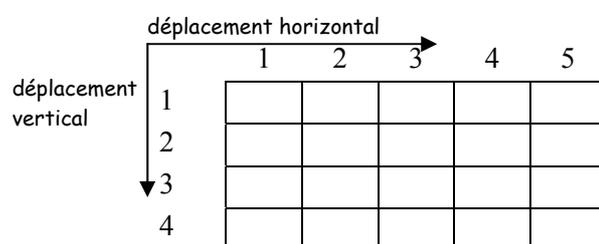
PositionH :

Ensuite, les élèves doivent se mettre d'accord sur les termes à utiliser et sur les déplacements que peut effectuer la croix, par exemple, la croix ne peut pas se déplacer en oblique.

On envisage, par exemple, 4 actions élémentaires :

- **Cliquer sur AvancerH** que nous noterons **AvancerH** = la croix avance d'une case horizontalement;
- **Cliquer sur ReculerH** que nous noterons **ReculerH** = la croix recule d'une case horizontalement;
- **Cliquer sur AvancerV** que nous noterons **AvancerV** = la croix avance d'une case verticalement;
- **Cliquer sur ReculerV** que nous noterons **ReculerV** = la croix recule d'une case verticalement.

Avancer horizontalement s'effectue de gauche à droite et avancer verticalement s'effectue de haut en bas (comme indiqué par le sens des flèches ci-dessous).

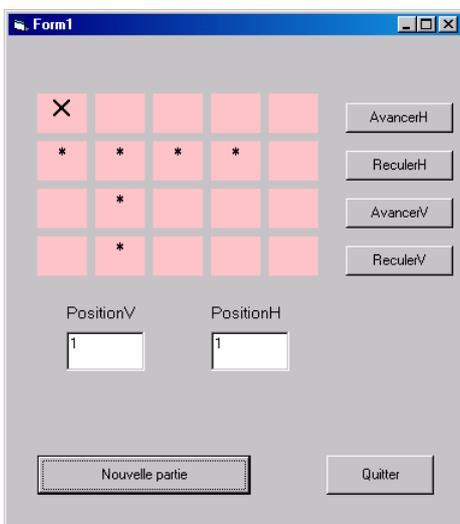


De plus, il faut trouver une manière de préciser la position de la croix au départ. Nous pouvons convenir de traduire la phrase "prend la valeur" en utilisant une flèche \leftarrow . Donc PositionV prend la valeur 1 sera écrit : PositionV \leftarrow 1

Il sera aussi convenu que chaque action sera écrite sur une ligne et que le lancement du jeu commencera par le mot "Début" et se terminera par le mot "Fin".

Pour terminer, les élèves échangent les solutions obtenues, afin de vérifier si le voisin comprend ce qui est écrit.

Simulation



Les élèves peuvent alors jouer et noter sur papier la "programmation" demandée.

Remarque : dans le programme Pacman01.exe, vous pouvez voir une représentation "physique" des cases mémoires, ce qui permet de suivre l'évolution des valeurs des variables en mémoire.

Voici, par exemple, une situation que pourrait rencontrer un élève :

	H1	H2	H3	H4	H5
V1	X	•	•	•	
V2				•	
V3		•		•	
V4					

L'élève doit donc noter la position de départ de la croix et noter, ensuite, les déplacements à effectuer pour manger les points noirs.

Début

```

PositionV ← 1
PositionH ← 1
AvancerH
AvancerH
AvancerH
AvancerV
AvancerV
ReculerH
ReculerH

```

Fin

Cette série d'instructions sera à échanger entre élèves pour qu'ils puissent analyser le travail réalisé par leurs voisins et déceler les problèmes éventuels.

Prolongement 1

- Nous pouvons aussi essayer d'exprimer les déplacements en utilisant les compteurs.

Regardons l'évolution des compteurs des positions au fur et à mesure des déplacements.

Déplacement n°	0	1	2	3	4	5	6	7
PositionV	1	1	1	1	2	3	3	3
PositionH	1	2	3	4	4	4	3	2

Ce qui pourrait s'écrire :

Début

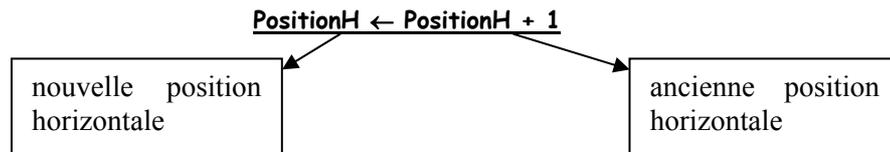
```

PositionV ← 1
PositionH ← 1
PositionH + 1
PositionH + 1
PositionH + 1
PositionV + 1
PositionV + 1
PositionH - 1
PositionH - 1

```

Fin

Mais alors, on remarque que la ligne "PositionH + 1" ne modifie pas la valeur de PositionH. Par contre, il faut augmenter de 1 la valeur de PositionH pour obtenir une nouvelle valeur de PositionH, ce que nous convenons d'écrire :



Les instructions deviennent, sous cette convention :

```

Début
  PositionV ← 1
  PositionH ← 1
  PositionH ← PositionH + 1
  PositionH ← PositionH + 1
  PositionH ← PositionH + 1
  PositionV ← PositionV + 1
  PositionV ← PositionV + 1
  PositionH ← PositionH - 1
  PositionH ← PositionH - 1
Fin
  
```

Prolongement 2

- Nous allons compter le nombre de points noirs que la croix mange sur son chemin. Ce comptage doit se faire au fur et à mesure que la croix mange les points noirs.

Il va falloir stocker quelque part le nombre de points noirs mangés, c'est-à-dire qu'il va falloir créer une nouvelle variable (que nous appellerons, par exemple, NombreNoirs). Cette nouvelle variable va accumuler le nombre de points noirs mangés, c'est-à-dire qu'on ajoutera 1 à la variable à chaque fois qu'un point noir sera mangé.

```

Début
  PositionV ← 1
  PositionH ← 1
  NombreNoirs ← 0
  PositionH ← PositionH + 1
  NombreNoirs ← NombreNoirs + 1
  PositionH ← PositionH + 1
  NombreNoirs ← NombreNoirs + 1
  PositionH ← PositionH + 1
  NombreNoirs ← NombreNoirs + 1
  PositionV ← PositionV + 1
  NombreNoirs ← NombreNoirs + 1
  PositionV ← PositionV + 1
  NombreNoirs ← NombreNoirs + 1
  PositionH ← PositionH - 1
  PositionH ← PositionH - 1
  NombreNoirs ← NombreNoirs + 1
Fin
  
```

Tâches du professeur

- Installer les petits programmes sur les machines des élèves avant le cours.

- Accompagner les élèves dans leur cheminement intellectuel en les dirigeant, mais sans rien imposer, ce qui leur permettra de découvrir les actions élémentaires ...
- Imposer des notations strictes, claires et précises.
- Veiller au respect des notations.
- Veiller à l'exactitude des algorithmes proposés par les élèves lors du jeu.
- Proposer aux élèves des échanges d'algorithmes afin de déceler des erreurs ou des imprécisions éventuelles.

Consignes aux élèves

- S'approprier l'énoncé.
- Discuter avec ses camarades pour se mettre d'accord sur les actions élémentaires à envisager.
- Jouer quelques parties en notant scrupuleusement le déroulement du jeu.
- Proposer son algorithme à son voisin pour en vérifier l'exactitude et la rigueur.

Déroulement de la leçon

Pour que la croix avance de 3 cases, certains respectent les actions élémentaires et écrivent la solution correcte :

AvancerH
AvancerH
AvancerH

D'autres écrivent l'une des possibilités suivantes :

AvancerH : 3
AvancerH 3
AvancerH + 3

Il faut alors rappeler que l'action effectuée dans Pacman01.exe est "AvancerH" et que rien d'autre n'est permis à l'heure actuelle.

Le professeur présente alors la notation choisie pour l'affectation (\leftarrow), ce qui ne leur pose pas de problème. De plus, l'instruction $\text{PositionH} \leftarrow \text{PositionH} + 1$ est découverte rapidement. Ils acceptent bien que la croix ait une position horizontale appelée PositionH et une position verticale appelée PositionV , quelle que soit la case de la grille dans laquelle elle se trouve.

En fin de cours, les élèves parviennent à dégager les notions qui ont été abordées et peuvent exprimer les éléments indispensables à l'existence d'une affectation. Il faut les aider à formaliser ce qu'ils sentent (et comprennent ?), car, par exemple, « $\text{PositionH} + 1$ » s'appelle une expression pour le professeur, mais pas toujours pour l'élève.

Savoirs acquis

- Mettre une valeur dans PositionV et / ou PositionH (affectation d'une valeur à une variable).
- Calculer une expression et la mettre dans PositionV et / ou PositionH (affectation d'une expression à une variable).
- Réaliser des actions dans un ordre précis - l'ordre ne peut pas être changé sans y réfléchir (séquence).
- Réaliser une accumulation.

Les règles à respecter pour l'instruction d'affectation

- Toutes les variables utilisées doivent être correctement déclarées.
- La partie de droite (du signe \leftarrow) est une constante, une variable ou une expression (par exemple : $A \leftarrow 10$; $A \leftarrow B$; $A \leftarrow 2*B+15$ peuvent exister).
- La partie de gauche (du signe \leftarrow) est toujours une variable.
- La partie de droite doit fournir une valeur (par exemple : $A \leftarrow 2*B+15$ cette expression n'est pas correcte si B n'a pas de valeur au moment d'appliquer l'affectation ou si B est une variable de type CARACTERE).
- Le type de la valeur obtenue à droite du signe \leftarrow est compatible avec le type de la variable qui est à gauche du signe \leftarrow .

Lecture

Savoirs requis

Les notions de la situation précédente :

- Variables
- Types de données
- Affectation
- Séquence

Description de la situation d'apprentissage

Comment améliorer le jeu que nous avons introduit dans la première situation ?

- L'élève (joueur) devrait donner son nom (ou son pseudo) au départ pour être identifié par la machine.
- Le joueur devrait pouvoir choisir sa position de départ.

Cette situation s'accompagne du petit programme **Pacman02.exe**. Nous y trouvons les mêmes éléments que dans le programme **Pacman01.exe**, sauf que la croix n'occupe pas la case supérieure gauche; elle n'occupe aucune case par défaut. Dès que le programme est lancé, une boîte de dialogue vide apparaît, dans laquelle le joueur (élève) doit entrer son nom. De plus, à chaque début de nouvelle partie, deux boîtes de dialogue apparaissent, dans lesquelles le joueur doit encoder les coordonnées de la case de départ de la croix.

Proposition de solution

En fait, des données devront être fournies à la machine par le biais du clavier. Nous avons donc besoin d'une nouvelle action élémentaire :

- **LireEtMettreDans** = lis ce qui a été tapé au clavier et mets-le dans une variable (une case mémoire particulière). Cette instruction sera donc suivie par le nom d'une variable.

Simulation

En reprenant le problème cité dans la situation 1, on ne modifie que légèrement ce qui a déjà été construit.



Le programme demande à l'élève de fournir son nom et de choisir sa position de départ.

En réalité, le programme ne "demande" rien à l'élève, une boîte de dialogue vide s'ouvre et l'élève doit savoir qu'il doit d'abord encoder son nom, et ensuite, quand il lance une nouvelle partie, encoder sa position horizontale et, enfin, sa position verticale.

	H1	H2	H3	H4	H5
V1		•	•	•	
V2				•	
V3		•		•	
V4					

Imaginons que l'élève décide de partir de la PositionH valant 2 et la PositionV valant 1. Il gagne un déplacement pour manger tous les points noirs.

Cet élève doit écrire précisément ce qui est fait et ce que ça entraînera pour le jeu. Par exemple, quand il dit qu'il "tape son nom au clavier", la tâche allouée au programme est donc de "LireEtMettreDans" une variable qu'il est raisonnable d'appeler "nom". Une nouvelle zone apparaît alors dans le programme Pacman02.exe qui représente le contenu de la variable "nom".

Début

```

LireEtMettreDans nom
LireEtMettreDans PositionH
LireEtMettreDans PositionV
AvancerH
AvancerH
AvancerV
AvancerV
ReculerH
ReculerH

```

Fin

Remarque : l'utilisation d'un petit programme pour simuler ce jeu facilite cette étape, car le joueur voit des boîtes de dialogue apparaître, la première pour encoder son nom, la deuxième la position horizontale et la troisième la position verticale. Il y aurait donc un encodage réel par le joueur et une meilleure visualisation de l'instruction LireEtMettreDans. De plus, après l'encodage des données, l'élève verra apparaître le contenu des zones mémoires sur la feuille de jeu.

Tâches du professeur

- Rappeler les notions vues précédemment.
- Proposer une nouvelle action élémentaire.
- Préciser ce que doit encoder l'élève dans chaque boîte de dialogue.
- Veiller à l'exactitude des algorithmes proposés par les élèves.
- Demander aux élèves d'expliquer oralement leurs algorithmes, afin de vérifier la bonne compréhension.

Consignes aux élèves

- Respecter scrupuleusement les instructions du professeur.
- Réaliser le jeu calmement et noter avec précision.
- Échanger les algorithmes pour vérifier si la résolution proposée est correcte et suffisamment précise.
- Analyser le fonctionnement du programme PACMAN02.

Déroulement de la leçon

Cette manière d'introduire les notions facilite l'arrivée des parties théoriques, que les élèves sont assez réticents à étudier dans l'enseignement technique.

De plus, l'instruction "lire" est difficile à comprendre, car elle "signifie" que l'ordinateur fait deux actions. Après une première leçon sur l'instruction "lire", un peu plus de la moitié des élèves abandonnent la transformation "LireEtMettreDans", les autres continuent à l'utiliser.

Savoirs acquis

- Utiliser l'instruction de lecture.

Les règles à respecter pour l'instruction de lecture

- Le nom qui suit l'instruction **lire** doit toujours être une variable ou plusieurs variables séparées par des virgules.
- Toutes les variables utilisées doivent être correctement déclarées.

Ecriture

Bien que cette instruction soit d'un abord plus aisé, les élèves rencontrent quand même la troisième notion importante et, pour éviter tout problème, il nous paraît indispensable d'y consacrer un peu de temps. Nous avons d'abord associé le mot "Ecrire" à cette instruction, mais, par la suite, le mot "Afficher" nous a paru plus adéquat.

Savoirs requis

Les notions de la situation précédente :

- Variables et types de données
- Affectation et séquence
- Lecture

Description de la situation d'apprentissage

Pour que le programme soit plus pratique à utiliser, il faudrait afficher un message à l'écran, qui préciserait quels sont les éléments qui doivent être encodés. L'utilisateur est capable d'encoder son nom, ensuite la PositionH et, pour finir, la PositionV, car le professeur lui a dit que c'était dans cet ordre que les éléments devaient être fournis. Il serait donc bon que le programme puisse afficher lui-même un message pour indiquer ce qu'il attend de l'utilisateur (l'élève).

Cette situation s'accompagne du petit programme **Pacman03.exe**. Dans ce petit jeu, on retrouve tous les éléments du **Pacman02.exe**, mais en plus, dans les boîtes de dialogue, le joueur trouve des informations sur la donnée qu'il doit encoder.

Proposition de solution

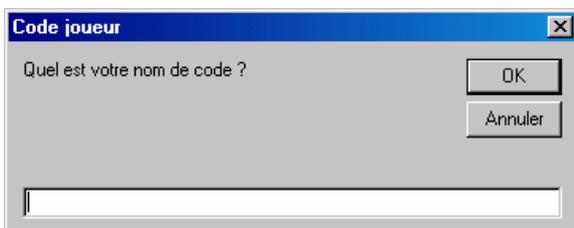
Nous devons donc ajouter une action élémentaire :

- **Afficher** = ce qui suit ce mot est affiché à l'écran, afin que l'utilisateur du programme puisse le voir.

Simulation

Remarque : si le professeur n'utilise pas un petit programme pour simuler cette situation, il devra indiquer des messages sur le tableau (en guise d'écran), afin que les élèves puissent s'y conformer.

En faisant fonctionner le petit programme qui accompagne cette situation (Pacman03.exe), l'élève pourra modifier l'algorithme écrit auparavant en tenant compte des affichages successifs :





Supposons que la grille de départ est la même que pour la situation précédente :

Début

Afficher "Quel est votre nom de code ?"
LireEtMettreDans nom
Afficher "choisissez la PositionH (entre 1 et 5)"
LireEtMettreDans PositionH
Afficher "choisissez la PositionV (entre 1 et 4)"
LireEtMettreDans PositionV
 AvancerH
 AvancerH
 AvancerV
 AvancerV
 ReculerH
 ReculerH

Fin

Prolongement

- Le jeu est modifié, la croix doit toujours manger tous les points noirs, mais en effectuant le moins de déplacements possibles. Les points noirs étant mis de manière aléatoire sur la grille, il faut compter le nombre total de déplacements effectués par la croix et afficher ce résultat à l'écran pour savoir quel élève a gagné. Il est évident que l'élève doit choisir intelligemment sa position de départ.

Imaginons que la grille de départ soit la suivante :

	1	2	3	4	5
1		•		•	
2	•		•		
3			•		
4					

L'élève doit fournir son nom et choisir sa position de départ, imaginons qu'il décide de partir de la PositionH valant 1 et la PositionV valant 2, car il estime que cette position va lui permettre d'effectuer moins de déplacements.

Le programme doit utiliser une variable supplémentaire afin de comptabiliser les déplacements et, en fin de parcours, il devra afficher à l'écran le contenu de cette variable (dans le programme, un bouton permet d'afficher le nombre de déplacements effectués).

Voici, idéalement, ce que cet élève devrait noter :

Début

```

Afficher "Quel est votre nom de code ?"
LireEtMettreDans nom
Afficher "choisissez la PositionH (entre 1 et 5)"
LireEtMettreDans PositionH
Afficher "choisissez la PositionV (entre 1 et 4)"
LireEtMettreDans PositionV
NombreDepl ← 0
ReculerV
NombreDepl ← NombreDepl + 1
AvancerH
NombreDepl ← NombreDepl + 1
AvancerH
NombreDepl ← NombreDepl + 1
AvancerH
NombreDepl ← NombreDepl + 1
AvancerV
NombreDepl ← NombreDepl + 1
ReculerH
NombreDepl ← NombreDepl + 1
AvancerV
NombreDepl ← NombreDepl + 1
Afficher NombreDepl

```

Fin

Tâches du professeur

- Rappeler les notions vues précédemment.
- Proposer une nouvelle action élémentaire.
- Faire évoluer la situation au fil du temps.

Consignes aux élèves

Les mêmes que pour l'instruction de lecture.

Savoirs acquis

- Afficher un message à l'écran.
- Afficher le contenu d'une variable.

Déroulement de la leçon

Le mot "Afficher" a été suggéré par les élèves. Nous avons pensé au mot "Ecrire", plus courant en programmation, mais les élèves étaient perturbés par ce terme.

Les règles à respecter pour l'instruction d'écriture

- Ce qui suit l'instruction **Afficher** doit toujours être une chaîne de caractères ou une combinaison de chaînes de caractères et de variables séparées par une virgule.
- Toutes les variables utilisées doivent être correctement déclarées.

Conditionnelle et alternative

Bien que les mots "Si, Alors, Sinon" fassent partie du vocabulaire couramment utilisé par les élèves, la structure d'une alternative ne paraît pas acquise. Nous avons donc souhaité mettre l'accent sur le fait qu'une condition est évaluée et qu'en fonction du résultat (vrai ou faux) obtenu, certaines actions sont exécutées. Nous avons donc remplacé :

- Le mot "Si" par "Evaluer la valeur de la condition" pour indiquer qu'une condition doit être évaluée.
- Le mot "Alors" par "Instructions à effectuer lorsque la valeur obtenue est Vrai" pour indiquer que les instructions qui suivent doivent être exécutées lorsque l'évaluation a fourni la valeur "vrai".
- Le mot "Sinon" par " Instructions à effectuer lorsque la valeur obtenue est Faux " pour indiquer que les instructions qui suivent doivent être exécutées lorsque l'évaluation a fourni la valeur "faux".

Cette approche sera progressivement abandonnée et remplacée par la structure Si...Alors...Sinon, au fur et à mesure que les élèves auront bien intégré l'emploi de cette structure de contrôle. On pourra toujours faire référence à cette approche en cas de difficultés de compréhension.

Savoirs requis

- Toutes les notions des situations précédentes.
 - La notion de valeur logique ou booléenne.
 - Les opérateurs de comparaison.
 - Les opérateurs logiques
- Ces opérateurs devraient être rencontrés dans le cours de mathématique(logique). Les élèves devraient pouvoir écrire une condition et l'évaluer, ce qui faciliterait le passage à l'alternative.

Description de la situation d'apprentissage

Maintenant que nous pouvons déplacer la croix dans la grille et que les déplacements sont comptés, nous rencontrons un nouveau problème : que se passe-t-il quand l'utilisateur fait une erreur de déplacements de la croix ?

Cette situation s'accompagne du petit programme Pacman04.exe. Dans les Pacman01 à Pacman03, quand le joueur cliquait sur le bouton "AvancerH" une fois de trop, le programme l'éjectait tout de suite. Maintenant (Pacman04), le joueur reçoit un message disant qu'il ne peut plus avancer ou plus reculer. Mais comment ces messages ont-ils été programmés ? Comment faire pour que la croix ne sorte pas de la grille ?

Proposition de solution

Nous allons devoir tester les valeurs de PositionH et PositionV afin de fixer les déplacements qui sont possibles ou pas. Voici donc une nouvelle action élémentaire :

- Evaluer la valeur de (condition) = le robot évalue la valeur de la condition qui se trouve entre les parenthèses. Le résultat obtenu est VRAI ou FAUX.

En fonction du résultat de cette évaluation, le programme devra effectuer un traitement ou un autre. Nous allons associer une série d'instructions dans le cas où la valeur est VRAI et une autre série d'instructions dans le cas où la valeur est FAUX.

On écrira:

Evaluer la valeur de (.....condition)

Instructions à effectuer lorsque la valeur obtenue est Vrai

... }
 ... } série d'instructions à exécuter si la valeur du test est VRAI
 ...

Instructions à effectuer lorsque la valeur obtenue est Faux

... }
 ... } série d'instructions à exécuter si la valeur du test est FAUX
 ...

Fin Evaluer

Simulation

En essayant de se déplacer sur les bords de la grille, l'élève va se rendre compte des limites imposées par les dimensions de celle-ci.

En effet, quand l'élève essaie AvancerH ou AvancerV une fois de trop, le message "Impossible d'avancer" apparaît. Il en va de même pour ReculerH ou ReculerV, pour lesquels le message "Impossible de reculer" apparaît.



Les élèves doivent associer ces messages d'erreurs aux PositionV et PositionH occupées par la croix. Ainsi donc, à chaque déplacement de la croix, le programme doit faire une vérification pour voir si ce déplacement est possible ou pas. Quand la croix occupe la PositionH égale à 5 et que le joueur choisi d'avancer horizontalement d'une case supplémentaire (AvancerH), il n'est plus possible d'avancer et le message d'erreur doit s'inscrire, mais, si l'une des conditions n'est pas respectée, il n'y a pas de problème et le déplacement horizontal peut s'effectuer.

Si on ne rentre pas dans les problèmes de l'implémentation de la vérification des actions comme AvancerH ..., les instructions liées à l'envoi des messages sont :

Evaluer la valeur de ((PositionH == 5) et (AvancerH))

Instructions à effectuer lorsque la valeur obtenue est Vrai

Afficher "Impossible d'avancer"

Instructions à effectuer lorsque la valeur obtenue est Faux

positionH \leftarrow positionH + 1

Fin Evaluer

Evaluer la valeur de ((PositionH == 1) et (demande de ReculerH))

Instructions à effectuer lorsque la valeur obtenue est Vrai

Afficher "Impossible de reculer"

Instructions à effectuer lorsque la valeur obtenue est Faux

positionH \leftarrow positionH - 1

Fin Evaluer

Evaluer la valeur de ((PositionV == 4) et (demande de AvancerV))

Instructions à effectuer lorsque la valeur obtenue est Vrai

Afficher "Impossible d'avancer"

Instructions à effectuer lorsque la valeur obtenue est Faux

positionV \leftarrow positionV + 1

Fin Evaluer

Evaluer la valeur de ((PositionV == 1) et (demande de ReculerV))

Instructions à effectuer lorsque la valeur obtenue est Vrai

Afficher "Impossible de reculer"

Instructions à effectuer lorsque la valeur obtenue est Faux

positionV \leftarrow positionV - 1

Fin Test

Tâches du professeur

- Insister sur la cohérence et le résultat de conditions.
- Exprimer en français le test qui doit être effectué.
- Faire comprendre qu'une condition est un booléen.
- Insister sur le fait que la structure alternative n'est parcourue qu'une seule fois (pour préparer l'arrivée des boucles).

Consignes aux élèves

- Respecter les notations.
- Réfléchir au résultat d'une condition.
- Combiner deux conditions.

Savoirs acquis

- Écrire une condition en utilisant des opérateurs de comparaison (==, <, >, <=, >=, <>).
- Combiner les conditions en utilisant des opérateurs logiques (et, ou, ou bien, non).
- Réaliser un traitement selon une condition (Evaluer la valeur de Instructions à effectuer lorsque la valeur obtenue est Vrai Instructions à effectuer lorsque la valeur obtenue est FauxFin Evaluer).

Déroulement de la leçon

Pour commencer, la partie « Evaluer », c'est-à-dire l'écriture de conditions, devrait être vue séparément afin de fixer cette notion avec les élèves. Le passage à l'alternative s'avère alors plus facile.

Ensuite, lorsque nous avons introduit la notion « Evaluer », nous pensions que les élèves allaient rapidement l'abandonner pour n'utiliser que le « Si », ce qui était une erreur. Les élèves ont longtemps utilisé la structure « Evaluer la valeur de Instructions à effectuer lorsque la valeur obtenue est Vrai Instructions à effectuer lorsque la valeur obtenue est FauxFin Evaluer ». Dans le langage courant, ils disent d'ailleurs « je dois évaluer... » ou « je dois tester... » (« je dois regarder la variable A » par exemple). Il a fallu 10 périodes de cours pour que tous les élèves passent à l'écriture du « Si Alors Sinon Fin si » classique et, régulièrement, certains continuent à se référer à l'écriture de départ (qui leur semble beaucoup plus claire).

Les règles à respecter pour l'écriture d'une condition

- Une condition s'écrit entre parenthèses.
- Une condition est composée de deux expressions séparées par un opérateur de comparaison (<, >, <=, >=, ==, ><).
- Une condition est évaluée par l'ordinateur et fournit comme résultat VRAI ou FAUX.
- Plusieurs conditions peuvent être combinées par des opérateurs logiques (et, ou, non) pour toujours fournir comme résultat VRAI ou FAUX.

Les règles à respecter pour la conditionnelle et l'alternative

- Après le mot clé **Si**, il doit y avoir une condition ou plusieurs conditions séparées par des opérateurs de comparaison.
- La structure **Si** comporte toujours une partie **Alors**.
- La structure **Si** comporte parfois une partie **Sinon**.
- La fin d'une structure **Si** est indiquée par le mot clé **Fin Si**.

Répétitions

Nous sommes maintenant dans les derniers mois de l'année scolaire et les élèves ont déjà acquis quelques notions importantes de la programmation. La structure répétitive est souvent arrivée dans les conversations, surtout pour la vérification de l'encodage de données correctes. Après la structure alternative, qui ne peut être parcourue qu'une seule fois, les élèves attendent une structure qui peut être parcourue plusieurs fois. Il suffit donc de préciser les notations et les exigences (règles) de cette structure qui est acquise sans difficulté.

Savoirs requis

- Toutes les notions des situations précédentes.

Description de la situation d'apprentissage

Pour mettre le dernier point à notre petit programme, il faut remarquer qu'au début de la partie, le programme demande d'encoder la position horizontale et la position verticale. La position horizontale doit être un nombre entre 1 et 5 et la position verticale, un nombre entre 1 et 4. Nous souhaitons que le programme vérifie que le nombre encodé est correct et qu'il repose la question jusqu'à l'obtention d'un nombre correct. Comment faire ?

Cette situation s'accompagne du petit programme **Pacman05.exe**, dans lequel nous reprenons la totalité du Pacman04, mais, en plus, l'encodage des positions horizontale et verticale est recommencé jusqu'à l'obtention de valeurs correctes. Comment le programme peut-il vérifier que l'encodage est correct ?

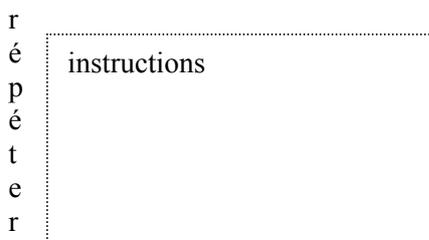
Proposition de solution

Nous voyons apparaître la structure de boucle, car les mêmes instructions peuvent être répétées plusieurs fois de suite. Ici, on ne sait pas combien de tours seront faits, c'est une condition qui servira d'arrêt.

Il existe deux structures de boucles :

- Graphiquement, la première structure pourrait être :

Tant que (condition) == Vrai

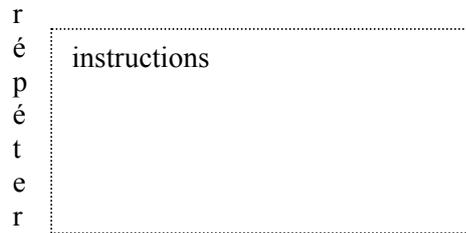


Fin tant que

- La condition aura une valeur au départ et à chaque tour de boucle. Cette valeur doit être Vrai pour que le corps de la boucle soit exécuté au moins 1 fois.
- La boucle s'arrêtera dès que la condition aura la valeur Faux.

- Graphiquement, la deuxième structure pourrait être :

Répéter



jusqu'à (condition) == Vrai

- La boucle est parcourue une première fois quelle que soit la valeur de la condition.
- La boucle s'arrêtera dès que la condition aura la valeur Vrai.
- La condition aura la valeur Faux à chaque tour de boucle.

Simulation

Les élèves ont le choix entre les deux types de boucle. Au vu du fonctionnement du programme, ils peuvent proposer la résolution suivante :

Répéter

Afficher "choisissez la PositionH (entre 1 et 5)"

LireEtMettreDans PositionH

jusqu'à (PositionH >= 1) et (PositionH <= 5)

Répéter

Afficher "choisissez la PositionV (entre 1 et 4)"

LireEtMettreDans PositionV

jusqu'à (PositionV >= 1) et (PositionV <= 4)

Mais ils pourraient aussi proposer d'utiliser l'autre solution qui les oblige à initialiser les variables pour pouvoir entrer dans la boucle au moins une fois :

Afficher "choisissez la PositionH (entre 1 et 5)"

LireEtMettreDans PositionH

Tant que (PositionH < 1) ou (PositionH > 5)

Afficher "choisissez la PositionH (entre 1 et 5)"

LireEtMettreDans PositionH

Fin tant que

PositionV ← 0

Tant que (PositionV < 1) ou (PositionV > 4)

Afficher "choisissez la PositionV (entre 1 et 4)"

LireEtMettreDans PositionV

Fin tant que

Tâches du professeur

- Amener les élèves à exprimer les mots "tant que" et "jusqu'à".
- Faire percevoir la différence entre ces deux notions (en utilisant des exemples évidents où un type de structure doit être utilisé et pas l'autre).

Consignes aux élèves

- Tester la différence entre les programmes Pacman04 et Pacman05.
- Utiliser le programme sur des jeux de données différents et choisis intelligemment.
- Analyser le fonctionnement du programme Pacman05.

Savoirs acquis

- L'utilité d'une boucle.
- Les problèmes liés à l'utilisation d'une boucle.
- Les deux types de boucles.

Déroulement de la leçon

La notion de répétition est assez facile à comprendre pour les élèves, mais sa mise en pratique l'est beaucoup moins. Après 5 à 6 périodes de cours, ils repèrent bien les instructions qui doivent apparaître dans les boucles (celles qui seront répétées). Le professeur doit alors insister sur les différentes règles qui doivent être vérifiées lors de la construction d'une boucle (voir ci-dessous). Il faut aussi beaucoup insister sur le « nombre de fois où la boucle va tourner », c'est-à-dire qu'il est indispensable que l'élève vérifie quand et avec quelle(s) valeur(s) commence et se termine la boucle.

Dans les séances de cours que nous avons données, nous n'acceptons de corriger un exercice que lorsque l'élève avait vérifié toutes les règles imposées. Il a fallu 4 périodes de cours pour que les résolutions se fassent complètement, avec toutes les vérifications nécessaires.

Les règles à respecter pour la répétition "tant que"

- les variables intervenant dans le test de la condition sont définies;
- ce qui suit **Tant que** est bien une condition ou une combinaison de conditions qui fournit comme valeur Vrai ou Faux;
- à un moment on va entrer, peut-être, dans la boucle;
- les instructions à l'intérieur de la boucle modifient la valeur des variables de la condition pour qu'il soit possible de sortir de la boucle.

Les règles à respecter pour la répétition "jusqu'à"

- les variables intervenant dans le test de la condition sont définies;
- ce qui suit **Jusqu'à** est bien une condition ou une combinaison de conditions qui fournit comme valeur Vrai ou Faux;
- les instructions à l'intérieur de la boucle modifient la valeur des variables de la condition pour qu'il soit possible de sortir de la boucle.

Et si vous ne souhaitez pas utiliser un ordinateur lors de l'élaboration d'algorithmes ...

Nous avons aussi testé une autre manière d'introduire les concepts de base de la programmation procédurale. Cette façon de faire est assez classique, mais nous utilisons, comme précédemment, une autre manière de nommer les instructions de base. De plus, nous donnons, à nouveau, un support "visuel" aux concepts introduits, ce qui ne peut être que bénéfique à certains élèves.

Nous utilisons un robot qui se déplace dans une grille rectangulaire. Le grand avantage de cette situation est que l'élève peut "se mettre à la place" du robot. Il peut s'imaginer lui-même en train de réaliser les opérations commandées au robot. Durant les cours, on peut même réaliser véritablement cette mise en situation. On veillera dans ce cas à éventuellement modifier les actions primitives (AvancerH, AvancerV, ...) pour les adapter à une situation dans laquelle les notions d'horizontale et de verticale propres à un écran ne sont plus d'application.

De plus, le professeur peut proposer une représentation du robot et, chaque variable peut être matérialisée par une poche du robot. En effet, le robot pourrait être représenté avec deux jambes, un seul bras et quelques poches et pourrait réaliser certaines actions spécifiques.

Par exemple :

- pour avancer d'une case, on peut "voir" le robot avancer d'une case;
- pour mettre un nombre dans une variable, on peut "voir" le robot prendre le nombre situé dans la case et le mettre dans une de ses poches (qui est sa mémoire) ;
- pour permuter le contenu de deux variables (de deux poches), on peut "voir" le robot réaliser les actions avec son unique bras et une poche supplémentaire ;
- etc.

Nous vous exposons ci-dessous, en vrac, les situations proposées aux élèves, accompagnées de leurs divers prolongements. Ces situations permettent d'introduire beaucoup de notions importantes en programmation (affectation, lecture, écriture, alternative, répétitions, somme, comptage, maximum, minimum ...).

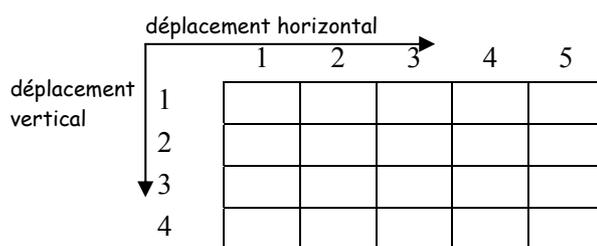
Description de la situation d'apprentissage 1

Un robot se déplace dans un domaine rectangulaire de dimensions finies (4 lignes et 5 colonnes), divisé en cases. Ce robot ne peut effectuer que 5 actions élémentaires :

- **SeMettreEnPosition (X,Y)** = le robot se place à une position particulière où X est la position verticale et Y la position horizontale;
- **AvancerH** = le robot avance d'une case horizontalement;
- **ReculerH** = le robot recule d'une case horizontalement;
- **AvancerV** = le robot avance d'une case verticalement;
- **ReculerV** = le robot recule d'une case verticalement.

Avancer horizontalement s'effectue de gauche à droite et avancer verticalement s'effectue de haut en bas (comme indiqué par le sens des flèches ci-après).

Ce domaine rectangulaire s'accompagne de deux compteurs qui indiquent la position verticale (PositionV) et la position horizontale (PositionH) du robot.



PositionV :
PositionH :

Dans chacun des cas proposés ci-dessous, l'élève doit écrire :

- un "algorithme" qui précise la position de départ du robot et les déplacements que celui-ci va effectuer;
- un "algorithme" qui analyse l'évolution du compteur entre le début et la fin du déplacement.

Prolongement n°1

Du point de vue du robot : il est en PositionV qui vaut 1 et en PositionH qui vaut 1. Il doit avancer de 3 cases horizontalement.

Du point de vue du compteur : en regardant le compteur au début (position de départ) et à la fin (position d'arrivée), comment exprimer ces résultats ?

Prolongement n°2

Du point de vue du robot : il est en PositionV qui vaut 3 et en PositionH qui vaut 1. Il doit reculer de 2 cases verticalement.

Du point de vue du compteur : même chose que pour le prolongement n°1.

Prolongement n°3

Du point de vue du robot : il démarre de la case (2,5) et va jusqu'à la case (4,2).

Du point de vue du compteur : même chose que pour le prolongement n°1.

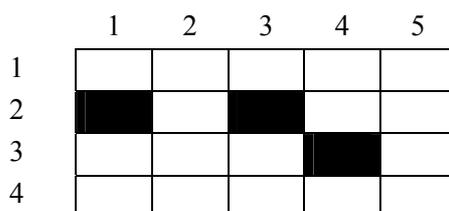
Prolongement n°4

Du point de vue du robot : il est en PositionV qui vaut 1 et en PositionH qui vaut 1 et il veut atteindre la case qui a comme coordonnées PositionV est 2 et PositionH est 4.

Du point de vue du compteur : même chose que pour le prolongement n°1.

Prolongement n°5

L'énoncé va être légèrement modifié et certaines cases vont être noircies pour indiquer qu'on ne peut pas y passer.



PositionV :
PositionH :

Du point de vue du robot : il doit aller de la case (1,1) à la case (2,4).

Et pour le compteur, quelles sont les étapes qui doivent apparaître ?

Prolongement n°6

Nous revenons à la grille vierge : le robot doit parcourir toutes les cases de la grille, mais ne jamais traverser deux fois la même case.

Et pour le compteur, quelles sont les étapes qui doivent apparaître ?

Description de la situation d'apprentissage 2

Le robot se trouve devant une grille dans laquelle chaque case contient un nombre.

	1	2	3	4	5
1	7	10	2	4	9
2	6	2	1	10	4
3	8	4	2	3	0
4	2	5	5	0	7

PositionV :

PositionH :

Le robot ne voit pas tous les nombres, il est seulement capable de voir le nombre qui est dans la case où il se trouve. Mais le robot est capable d'effectuer une nouvelle action élémentaire :

- **LireEtMettreDans** = prend le nombre qui est dans la case où il se trouve et le met dans une variable.

Il faut donc choisir le nom d'une variable dans laquelle le robot pourra mettre le nombre. Par exemple :

LireEtMettreDans Résultat = le robot prend le nombre qui est dans la case où il se trouve et met ce nombre dans la variable Résultat.

LireEtMettreDans Coupe = le robot prend le nombre qui est dans la case où il se trouve et le met dans la variable Coupe.

De plus, on considère maintenant que le robot est capable :

- de reconnaître, nommer et utiliser une **variable**;
- d'utiliser correctement **l'affectation**.

Prolongement n°1

Le robot est dans la case (1,1) et doit placer un à un tous les nombres qui se trouvent sur la première ligne dans une variable qui s'appellera Valeur.

Prolongement n°2

Le robot est dans la case (1,1) et parcourt la première ligne. Il voudrait pouvoir faire la somme de tous les nombres qu'il trouve sur son chemin (pour ne pas surcharger la mémoire, on utilise toujours le moins de variables possible !).

Prolongement n°3

Le robot est dans la case (2,2) et parcourt la ligne jusqu'au bout, il voudrait calculer à chaque fois le carré du nombre qu'il rencontre (en utilisant le moins de variables possible).

Prolongement n°4

Le robot est dans la case (1,1) et parcourt la première colonne. Il doit calculer le carré de chacun des nombres rencontrés et en faire la somme (en utilisant le moins de variables possible).

Prolongement n°5

On imagine maintenant que le robot trouve sur son parcours les lettres suivantes :

1	2	3	4	5	6	7
B	O	N	J	O	U	R

Le robot doit lire ces lettres une à une. Le but est de former à la fin le mot complet (BONJOUR). Comment va-t-il faire ?

Description de la situation d'apprentissage 3

On ajoute une nouvelle action élémentaire à notre robot :

- **Afficher** = le robot affiche (sur un support quelconque) ce que contient une variable ou affiche un message, afin que l'utilisateur du programme puisse le voir.

Il faut à nouveau choisir le nom d'une variable dont le robot pourra afficher le contenu. Par exemple :

Afficher poche1 = le robot montre ce que contient la variable poche1.

Prolongement n°1

Le robot part de la case (1,1) et va montrer à l'utilisateur le nombre qui est dans case (1,3).

Prolongement n°2

Le robot part de la case(1,1) et montre tous les nombres jusqu'à la case (1,3).

Prolongement n°3

Je souhaite que le robot m'envoie un message « Bonjour et bienvenue » au départ.

Énoncé de la situation d'apprentissage 4

Le robot est à nouveau capable d'une nouvelle action élémentaire :

- **Evaluer la valeur de ()** = le robot évalue la condition qui se trouve entre les parenthèses. Le résultat obtenu est VRAI ou FAUX.

Par exemple :

Evaluer la valeur de $(10 > 5)$ fournit la valeur VRAI car 10 est plus grand que 5.

Evaluer la valeur de $(10 < 5)$ fournit la valeur FAUX car 10 n'est pas plus petit que 5.

Evaluer la valeur de $(10 == 5)$ fournit la valeur FAUX car 10 n'est pas égal à 5.

Evaluer la valeur de $(10 >> 5)$ fournit la valeur VRAI car 10 est différent de 5.

Evaluer la valeur de $(10 \geq 5)$ fournit la valeur VRAI car 10 est plus grand ou égal à 5.
Evaluer la valeur de $(10 \leq 5)$ fournit la valeur FAUX car 10 n'est pas plus petit ou égal à 5.

Prolongement n°1

Le robot est-il dans la première ligne ?

Prolongement n°2

Le robot peut-il avancer d'une case horizontalement ?

Prolongement n°3

Le robot peut-il reculer d'une case verticalement ?

Prolongement n°4

Le robot est-il dans la case (3,1) ?

Prolongement n°5

Le robot est-il dans la première ligne ou dans la première colonne ?

Prolongement n°6

Le robot ne doit pas se trouver dans la troisième colonne.

Prolongement n°7

Le robot est-il sur un bord ?

Prolongement n°8

Le robot ne doit pas se trouver sur un bord.

Prolongement n°9

Le robot doit se trouver dans la colonne 4 ou la colonne 5 et doit pouvoir avancer d'une case verticalement.

Description de la situation d'apprentissage 5

Nous allons utiliser l'action élémentaire (Evaluer la valeur de), qui a été vue dans la partie précédente (Situation 4), pour réaliser soit une série d'instructions, soit une autre série d'instructions. Pour rappel, vous savez que le robot est capable de Evaluer la valeur de (condition), c'est-à-dire qu'il est capable de voir si la valeur de la condition est VRAI ou FAUX. Nous allons associer une série d'instructions dans le cas où la valeur est VRAI et une autre série d'instructions dans le cas où la valeur est FAUX.

Cette nouvelle action élémentaire s'écrira :

Evaluer la valeur de (.....condition

Instructions à effectuer lorsque la valeur obtenue est Vrai

... }
 ... } série d'instructions à exécuter si la valeur du test est VRAI
 ...

Instructions à effectuer lorsque la valeur obtenue est Faux

... }
 ... } série d'instructions à exécuter si la valeur du test est FAUX
 ...

Fin Evaluer

Pour rappel, dans les prolongements qui suivent, les cases noires représentent des cases où le robot ne peut pas passer.

Prolongement n°1

On repart dans cet exercice d'une grille vierge :

	1	2	3	4	5
1					
2					
3					
4					

PositionV :
PositionH :

Le robot est quelque part dans la première ligne. Il doit avancer d'une case horizontalement si c'est possible, sinon, il descend d'une ligne.

Prolongement n°2

La grille est modifiée :

	1	2	3	4	5
1					
2					
3					
4					

PositionV :
PositionH :

Le robot est quelque part dans la première ligne et veut aller dans la troisième ligne.

Prolongement n°3

Voici la grille :

	1	2	3	4	5
1					
2	■		■		
3					
4					

PositionV :

PositionH :

Le robot est quelque part dans la première ligne et veut aller dans la troisième ligne.

Prolongement n°4

Et une nouvelle grille :

	1	2	3	4	5
1					
2	■		■		■
3					
4					

PositionV :

PositionH :

Le robot est quelque part dans la première ligne et veut aller dans la troisième ligne.

Prolongement n°5

Toujours une nouvelle grille :

	1	2	3	4	5
1					
2					
3			■		
4					

PositionV :

PositionH :

Sachant que le robot n'est pas sur un bord, il doit avancer d'une case horizontalement, si c'est possible, sinon, il avance d'une case verticalement.

Prolongement n°6

De nouveau la grille vierge :

	1	2	3	4	5
1					
2					
3					
4					

PositionV :

PositionH :

Le robot est n'importe où et doit avancer d'une case horizontalement si c'est possible, sinon, il avance d'une case verticalement si c'est possible, sinon, il ne fait rien.

Prolongement n°7

Reprenons la grille remplie de nombres :

	1	2	3	4	5
1	7	10	2	4	9
2	6	2	1	10	4
3	8	4	2	3	0
4	2	5	5	0	7

PositionV :

PositionH :

Le robot part de la case (1,1), doit trouver le plus petit élément se trouvant sur la première ligne et montrer à l'utilisateur le minimum obtenu.

Prolongement n°8

En reprenant la grille :

	1	2	3	4	5
1	7	10	2	4	9
2	6	2	1	10	4
3	8	4	2	3	0
4	2	5	5	0	7

PositionV :

PositionH :

Sachant que le robot part de la case (1,1), il doit trouver le plus grand élément placé sur la première ligne et montrer à l'utilisateur le maximum obtenu.

Description de la situation d'apprentissage 6

Remarque

Nous allons maintenant introduire le concept de "boucle". Nous n'avons pas trouvé de nouveaux termes pour remplacer le "tant que" et le "jusqu'à". Une petite enquête a montré que les élèves utilisent facilement (dans des phrases courantes) le "jusqu'à". Par contre, le "tant que" ne fait pas partie de leur vocabulaire. Ces deux concepts vont être expliqués par des exemples (qui utilisent encore le robot). Cela permettra de montrer progressivement aux élèves :

- ce qu'est une boucle;
- pourquoi utiliser une boucle;
- qu'il existe deux types de boucles;
- que ces deux types de boucles entraînent un traitement différent de la part de l'ordinateur et, donc, qu'il est parfois préférable d'utiliser une boucle plutôt qu'une autre;
- qu'il y a des règles à respecter dans la construction d'une boucle;

Prolongement n°1

Le robot est quelque part dans la première ligne et doit aller jusqu'au bout de la ligne.

	1	2	3	4	5
1					
2					
3					
4					

PositionV :

PositionH :

Prolongement n°2

Le robot est en position (1,1), il fait la somme des éléments qu'il rencontre sur son chemin et il s'arrête quand la valeur de la somme dépasse 60 (pour commencer, on suppose que cette somme sera atteinte à un moment donné). Le robot indique alors quelle est la position horizontale où la somme a été atteinte.

	1	2	3	4	5
1	20	50	40	30	10
2					
3					
4					

PositionV :

PositionH :

Quelles notions avons-nous rencontrées dans ces études de cas ?

Situation 1

- Mettre une valeur dans une variable.
- Calculer une expression et la mettre dans une variable.
- Réaliser une séquence d'instructions.

Situation 2

- Lire une valeur et la mettre dans une variable.
- Utiliser une variable supplémentaire pour résoudre un problème.
- Faire un calcul à l'aide de variables.
- Faire la somme de plusieurs nombres (accumulation).

Situation 3

- Afficher le contenu d'une variable.
- Afficher un message.

Situation 4

- Ecrire une condition en utilisant des opérateurs de comparaison (==, <, >, <=, >=, <>).
- Combiner les conditions en utilisant des opérateurs logiques (et, ou, ou bien, non).

Situation 5

- Réaliser un traitement selon une condition.
- Imbriquer plusieurs structures "Evaluer".
- Rechercher un maximum et un minimum.

Situation 6

- Utiliser une boucle (d'un type ou d'un autre).
- Vérifier les conditions d'existence d'une boucle.

Bibliographie

La bibliographie proposée ci-dessous reprend une liste de références dont les contenus ne sont pas toujours directement liés au texte de ce document d'accompagnement. Ces références peuvent être des sources d'informations complémentaires pour les différents cours de l'option. D'autres ressources sont proposées sur le site du secteur informatique de la Fesec à l'adresse www.segec.be/fesec/secteurs/informatique.

Livres

- DEITEL ET DEITEL, (2000). *Comment Programmer en Java (troisième édition)*. Québec : Les Editions Reynald Goulet Inc.
- DEITEL, DEITEL & NIETO, (2000). *Internet & World Wide Web, How to program*. New Jersey: Prentice-Hall
- DEITEL, LIPERI & WIEDERMAN (2002). *Python how to program*. Prentice Hall.
- FLORIO E. (2000). *Guide de la couleur et de la page imprimée*. Atelier Perrousseaux; Reillanne;
- FOWLER M. (2001). *UML, le Tout en Poche*. Paris : Campus Press;;
- GAULD A. (2001). *Learn to program using Python*. Addison-Wesley
- HAMMOND M., ROBINSON A. (2000). *Python programming on Win32*. Sebastopol (USA): O'Reilly Editions.
- HORSTMANN C. S. ET CORNELL G. (2000). *Au coeur de Java2 Vol 1 Notions fondamentales*. Paris : CampusPress France.
- HORSTMANN C. S. ET CORNELL G. (2000). *Au coeur de Java2 Vol 2 Fonctions avancées*. Paris : CampusPress France.
- KRUG S. (2001). *Zéro prise de tête*. Paris : Dunod Internet professionnel.
- LAI M. (2000). *Penser objet avec UML et Java*. Paris : Dunod.
- LUTZ M. (2001). *Programming Python (2nd edition)*. O'Reilly.
- LYNCH P., HORTON S. (1999). *Web Style Guide. Basic Design and Principles for Creating Web Sites*. University Press Yale
- PERROUSSEAU Y. (1995, 2001). *Manuel de typographie française élémentaire*. Reillanne : Atelier Perrousseaux.
- PERROUSSEAU Y. (1996, 2001). *Mise en page et impression*. Reillanne : Atelier Perrousseaux.
- SASS F. (1998). *Mise en page*. Bruxelles : Licap.
- VANDEPUT E. (2002). *Programmer avec des objets, Notes de cours provisoires*. Namur : CeFIS FUNDP.
- VANDEPUT E. (1997). *Pour une utilisation efficace des technologies de l'information*. Namur : CeFIS FUNDP.

Sites

Les nez et les langues électroniques : <http://www.alpha-mos.com/>

DUROCHER, R., NOËL, P., BORDELEAU, C. Odorat électronique : application à la détection de mines et d'explosifs. www.acfas.ca/congres/congres67/s1176.htm

Musées de Grasse. www.museesdegrasse.com

Une langue électronique. <http://www.cybersciences.com/cyber/3.0/n926.asp>

MIFSUD, J. ISZ, S. TSUNG, T. la langue électronique : amertume, goûts et arômes des médicaments. <http://www.forumlabo.com/2002/abstracts/2002/26langue.htm>

MERCER, R. Object-Oriented Programming: Helping Beginning Programmers See the Forest and the Trees. <http://www.fbeedle.com/technote/01-01/oop/oop.html>

BERGIN, J. Why Procedural is the Wrong First Paradigm if OOP is the Goal. <http://csis.pace.edu/~Ebergin/papers/Whynotproceduralfirst.html>

BERGIN, J. Teaching Objects with Elementary Patterns. <http://csis.pace.edu/~bergin/patterns/TeachingObjectsElemPat.html>

ECOOP '99 Workshop. Introducing OO Design and Programming - with Special Emphasis on Concrete Examples. <http://www.gdf.hu/angster/Ecoop.htm>

HONG, J. The Use of Java as an Introductory Programming Language. <http://www.acm.org/crossroads/xrds4-4/introjv.html>

BERGIN, J. STEHLIK, M. ROBERTS, J. PATTIS R. Karel J. Robot A Gentle Introduction to the Art of Object-Oriented Programming in Java. <http://csis.pace.edu/~bergin/KarelJava2ed/Karel++JavaEdition.html>



FÉDÉRATION DE L'ENSEIGNEMENT SECONDAIRE CATHOLIQUE
Rue Guimard, 1 - 1040 Bruxelles
Tél: 02/507.07.55 - Fax: 02/507.08.54