

COURRIER DES LECTEURS

L'ORDINATEUR EN PAPIER : SIMULATION INFORMATIQUE A L'ECOLE

Nous avons reçu de Monsieur Jean DANIAU, IDEN à Poitiers, un compte rendu de stage d'instituteurs, dont l'adaptation à une classe est aisée . . . c'est pourquoi nous le publions ici.



L'idée de chercher à simuler le fonctionnement d'un ordinateur est apparue, en septembre, octobre et novembre 1984 à l'occasion de stages d'initiation à l'informatique regroupant des instituteurs du département de la Vienne (Académie de Poitiers) enseignant dans des écoles dotées de micro-ordinateurs de type TO7-70 ou sur le point d'être équipées.

Elle a été inspirée d'abord par le visionnement d'une bande vidéo du C.R.D.P. de Grenoble présentant une activité de simulation conduite avec des élèves de CM*. Elle est née surtout de la nécessité de lever une difficulté qu'éprouvaient nombre de stagiaires à situer très précisément le statut de l'instruction "INPUT" en Basic (version TO7). A partir de petits programmes construits par les stagiaires, une analyse du rôle de cette instruction avait permis de mettre à jour sa complexité cachée derrière une fonction apparemment simple, celle de prendre en compte une donnée numérique ou non introduite par l'utilisateur avec le clavier de la machine. L'effet induit par cette instruction est, on le sait, quadruple :

- 1) Ouvrir une "case-mémoire" au nom de la variable qui suit l'instruction (ex. : "INPUT A\$ ")
- 2) Faire apparaître sur l'écran le signe "?" qui appelle une réaction de l'utilisateur.
- 3) Attendre de l'utilisateur la donnée correspondant au type de variable déclaré (donnée affichée à l'écran après ?).
- 4) Placer dans la "case-mémoire" la valeur de la variable donnée par l'utilisateur (ou faire afficher le message "? Redo" pour le cas où la donnée proposée ne correspond pas au type de variable).

Or cette complexité, bien comprise à première vue à la lecture d'un listing avec les explications de l'animateur, ne passait plus (ou avait du mal à passer), dès qu'on se donnait

**"L'informatique ; pourquoi faire ?" série France Face à l'Avenir 83/84 film vidéo en prêt ; dossier de 20 pages et série de 5 diapos. en vente.*

un petit projet simple à réaliser tel que, par exemple, un programme de calcul de l'aire et du périmètre d'un rectangle à partir de ses dimensions données par l'utilisateur :

```

5  CLS
10 PRINT "AIRE ET PERIMETRE D'UN RECTANGLE"
20 PRINT "DONNE LA LARGEUR : LA"
30 INPUT LA
40 PRINT "DONNE LA LONGUEUR : LO"
50 INPUT LO
60 A = LA * LO
70 P = (LA + LO) * 2
80 PRINT "L'AIRE EST A = " ; A
90 PRINT "LE PERIMETRE EST P = " ; P
100 END

```

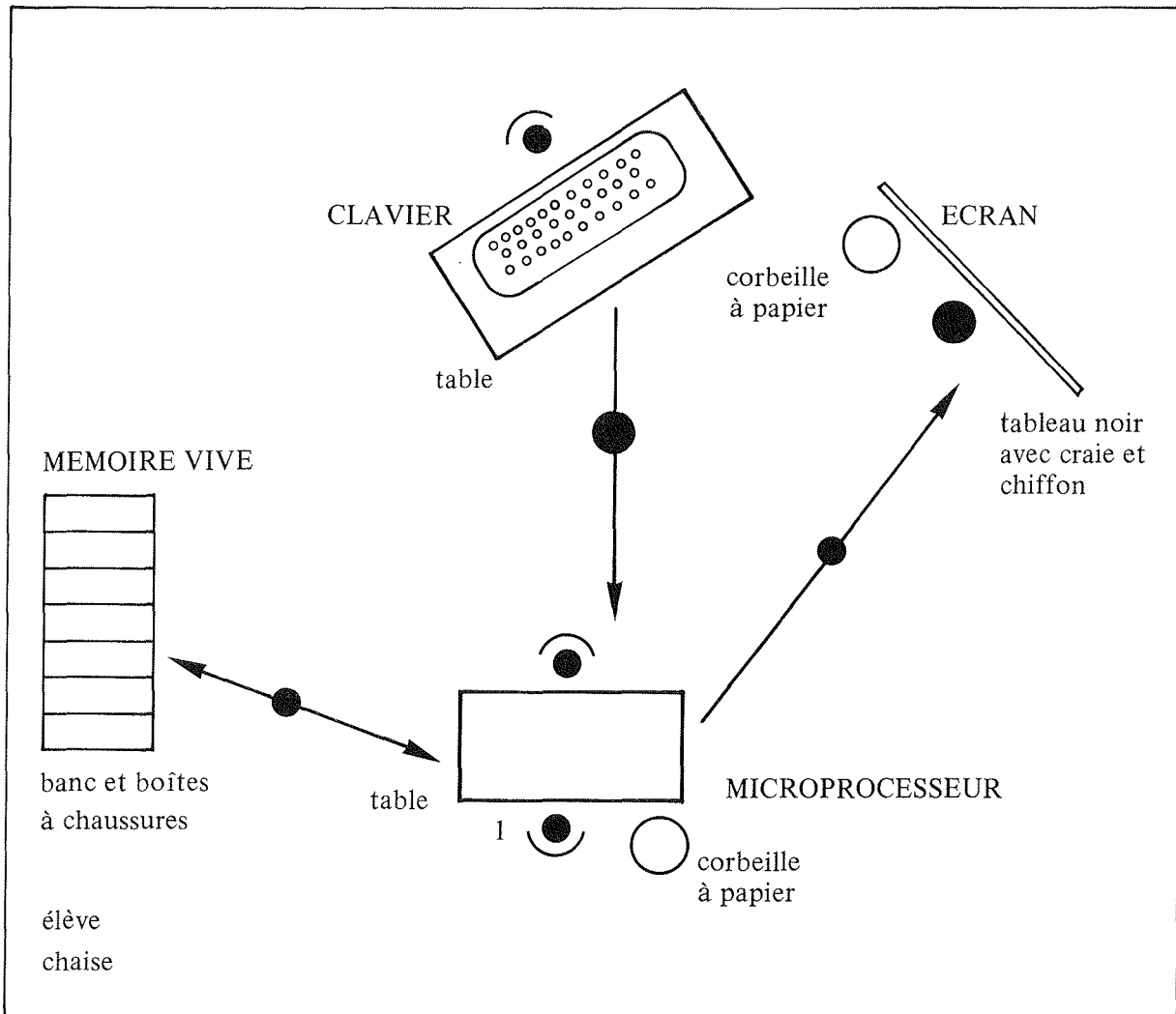
Une autre difficulté devait aussi être surmontée, celle attachée au statut du signe " = " en BASIC. Il s'agit, comme dans les lignes 60 et 70 ci-dessus, d'un signe **d'affectation** d'une valeur écrite à droite de " = " à une variable dont le nom est écrit à gauche et non d'un signe de relation entre deux signifiants renvoyant au même signifié comme c'est le cas en mathématique. En outre le signe " = " a aussi parfois le statut d'un caractère dans une chaîne (lignes 80 et 90). A cette ambiguïté s'ajoute celle liée aux symboles du type LO ou LA qui tantôt sont pris comme des **chaînes de caractères** (lignes 20 et 40), tantôt comme des **noms de variables** (lignes 30 et 50) étiquetant des "cases mémoires", tantôt enfin comme des **valeurs de variables** traitées à l'aide de fonctions (lignes 60 et 70) ou simplement affichées à l'écran (variables A et P aux lignes 80 et 90).

Bref, il fallait trouver le moyen d'y voir plus clair en analysant précisément **ce que fait la machine quand elle exécute un programme**. Il nous a semblé qu'un **jeu de rôles** mettant en évidence pas à pas les actions de la machine, serait de nature à mieux faire comprendre le sens du code utilisé, de la syntaxe et des fonctions mises en œuvre.

* * *

Le matériel nécessaire à cette simulation est simple : des chaises, deux tables, un tableau noir et un chiffon, du papier de grand format (papier journal blanc), des marqueurs, des boîtes à chaussures, un banc, un rouleau de "scotch", du carton de grand format, trois corbeilles à papier, une paire de ciseaux.

La première phase de l'activité consiste à matérialiser les principaux organes d'un micro-ordinateur en utilisant le matériel rassemblé et en donnant aux élèves un rôle à jouer. Le schéma d'organisation est le suivant. Il s'agit bien évidemment d'une représentation volontairement **très simplifiée du dispositif réel** : il n'est pas question avec des enfants d'entrer dans le détail des organes constitutifs d'un micro-ordinateur ni des liaisons entre les éléments qui le composent.



* L'organe central, le **micro-processeur** est représenté par deux élèves se faisant face et assis à une table ; ils possèdent des gros crayons feutres, une paire de ciseaux et des carrés de papier pour les échanges d'informations entre les divers organes de la machine. Un des élèves sera chargé de donner les ordres tandis que l'autre traitera l'information reçue (faire les calculs par exemple). Une corbeille recevra les papiers portant les informations non mémorisées.

* La **mémoire vive** est représentée par quatre ou cinq élèves assis sur des chaises ayant devant eux un banc sur lequel sont disposées une dizaine de boîtes à chaussures. Ces élèves peuvent écrire sur des papiers avec des marqueurs ou fixer des étiquettes sur les boîtes (avec du papier collant). Une corbeille à papier est à côté d'eux.

* L'**écran** est le tableau noir (avec des craies de plusieurs couleurs éventuellement et un chiffon). Un élève, debout, est chargé d'écrire au tableau. Il a à côté de lui une corbeille à papier.

* enfin le **clavier** est symbolisé par une feuille de carton posée sur une table sur laquelle sont dessinées les touches A, Z, E, R, T, Y, . . . etc. Un élève assis devant le clavier est l'utilisateur de l'ordinateur.

* entre le micro-processeur et les autres organes des liaisons électriques sont établies (symbolisées sur le schéma par des flèches simples ou doubles) qu'on appelle en terme technique les "BUS". On pourra les matérialiser sur le sol à la craie ou avec des cartons découpés. Pour chaque "BUS" un élève servira de messenger.

Il convient d'insister à ce stade sur le fait que **tout passe par le micro-processeur** : il n'y a aucune liaison directe entre le clavier et l'écran par exemple comme on pourrait le croire à première vue.

Détail pratique : on peut travailler à cette simulation avec une classe entière ; une demi-classe joue les rôles tandis que l'autre demi-classe observe et éventuellement participe à la discussion qui ne manque pas de s'instaurer dès que le jeu commence. On procèdera, bien entendu, à des permutations d'élèves au cours du jeu.

* * *

La seconde phase de la simulation consiste à mimer l'exécution d'un programme, tel que celui proposé ci-dessus. Mais au préalable il aura fallu l'élaborer et le transcrire sur des bandes de papier (ou de carton) suffisamment grandes pour être lues par tous ; chaque ligne du programme sera transcrite sur une bande. On supposera dès le départ que le programme est "entré" et stocké dans la mémoire vive. Les élèves qui représentent ce dernier organe placent les bandes en évidence devant eux les unes sur les autres dans l'ordre des adresses (numéros de lignes).

Le fonctionnement de l'ordinateur sera mimé essentiellement par des échanges d'informations sur des **rectangles de papier** de bonnes dimensions (15 x 20) par exemple pour que les inscriptions qui y seront transcrites avec de gros marqueurs puissent être lues par tous. (D'où le titre du présent article : "l'ordinateur en papier"). Plusieurs sortes de papiers seront utilisées que nous avons désignées dans la suite de ce texte par des mots-clés :

– "**bandes-lignes**" : bandes de papier manipulées par les élèves de la mémoire-vive sur lesquelles figurent chaque ligne du programme exécuté avec son adresse et ses instructions.

– "**papier-messages**" ou "**messages**" : destinés à faire passer une information d'un organe à l'autre. Ils sont détruits à mesure de leur utilisation (corbeille à papier).

– "**étiquette**" : papier sur lequel est écrit le nom d'une variable et qui sert à marquer une case-mémoire (boîte à chaussure).

– "**papier-mémoire**" : papier sur lequel est inscrite la valeur d'une variable (le contenu d'une case-mémoire) introduit dans une boîte à chaussure. Il est détruit quand on le remplace par un autre papier-mémoire portant une nouvelle valeur.

Pour lancer l'exécution du programme l'utilisateur frappe au clavier la commande "RUN". Une simulation très proche de la réalité devrait, pour chacun des caractères frappés

transmettre l'information au micro-processeur qui la transmettrait à l'écran. Pour aller plus vite l'élève-bus établissant la liaison clavier-micro-processeur, transcrit l'ordre dans sa totalité (soit ici les 3 lettres du mot "RUN") sur un papier qu'il porte au micro-processeur. Celui-ci confie le message à l'élève-bus-micro-processeur-écran.

L'élève-écran reçoit le message qu'il écrit au tableau en majuscules.

Dès qu'au clavier la touche "ENTREE" est enfoncée, l'élève-bus-clavier apporte au micro-processeur un carton tout prêt portant la mention "entrée" ou tout autre symbole conventionnel signifiant que ce qui vient d'être frappé au clavier est validé et exécutable.

En recevant cette dernière information le micro-processeur interroge à voix haute la mémoire-vive : "quelle est la première instruction du programme ? ". Les élèves de la mémoire vive prennent la bande-ligne "5 CLS", recopient l'instruction utile (soit ici "CLS") sur une feuille confiée à l'élève-bus-mémoire qui la transmet au micro-processeur ; la bande-ligne "5 CLS" est placée sous le paquet de bandes-lignes par les élèves de la mémoire-vive. Une manière plus rapide de procéder consiste à transmettre directement la bande-ligne au micro-processeur ; mais elle est moins "fidèle" car elle suggère, à tort, que dans cette opération la mémoire se vide, alors qu'elle est seulement "lue" tout en conservant les éléments du programme réutilisables indéfiniment.

Le message "CLS" est envoyé à l'écran. L'élève-écran efface le tableau et jette le message dans la corbeille à papiers (l'ordre étant exécuté cette information n'est pas conservée).

Le micro-processeur demande à la mémoire-vive "quelles sont les instructions de la ligne suivante ?".

Le micro-processeur reçoit le message : PRINT "AIRE ET PERIMETRE D'UN RECTANGLE" qu'il ampute avec une paire de ciseaux du mot PRINT (jeté au panier) le reste du texte étant transcrit à l'écran. L'élève-écran écrit en haut et à gauche AIRE ET PERIMETRE DU RECTANGLE et jette le papier au panier. On s'aperçoit ici que l'élève-micro-processeur doit, au passage, supprimer les guillemets, signes qui annoncent une chaîne de caractères et qui ne sont pas reproduits à l'écran.

L'instruction de la ligne 20 se traite comme celle de la ligne 10. Au tableau, l'élève-écran écrit en allant à la ligne : DONNE LA LARGEUR : LA.

Les choses deviennent plus délicates quand le micro-processeur reçoit le contenu de la ligne 30 : INPUT LA car il a ici quatre opérations à faire comme nous l'avons dit plus haut :

1ère opération : donner l'ordre à la mémoire-vive d'ouvrir une "case-mémoire" portant l'étiquette LA (nom de la variable). Les élèves de la mémoire-vive fixent sur une boîte à chaussure vide une étiquette portant l'indication "LA".

2ème opération : envoyer à l'écran un message portant le signe "?" affiché sur une nouvelle ligne.

3ème opération : l'utilisateur frappe au clavier un nombre (par exemple 24) qui est transcrit sur un papier-message transmis au micro-processeur puis à l'écran qui affiche 24 après le ? (ce qui donne : ? 24).

4ème opération : si l'élève-micro-processeur estime que la donnée est en accord avec le type de variable déclaré, il transcrit cette donnée sur un message transmis à la mémoire-vive. Le papier portant la mention "24" est logé dans la case-mémoire ouverte au nom de LA.

Il pourrait être intéressant au cours d'une manipulation ultérieure de frapper une donnée non conforme au type de variable attendu (par exemple "24" ou "TRUC" avec guillemets ou 24 CM ou TRUC sans guillemets) pour tester la présence d'esprit de l'élève-micro-processeur qui devra alors faire afficher le message : ? Redo (qui veut dire "refaire") et attendre une nouvelle donnée de l'utilisateur.

La ligne 40 produira les mêmes effets que la ligne 20 et la ligne 50 (INPUT LO) que la ligne 30. Une seconde case-mémoire a été ouverte au nom de LO (longueur) et remplie par un papier portant une valeur numérique (37 par exemple).

Quand le contenu de la ligne 60 est lu par l'élève-micro-processeur, un certain nombre de difficultés sont à lever et notamment la tentation de faire afficher "A = LA * LO" ou le résultat du calcul ($37 * 24 = 888$), alors que rien n'est frappé au clavier et que l'ordre PRINT (afficher) ne figure pas dans la ligne. Or, il faut comprendre ici que l'on aborde une phase de traitement de l'information reçue sans que rien ne soit ni demandé à l'utilisateur, ni porté à sa connaissance : **tout se fait à l'intérieur de la machine sans qu'on le voie**. Il sera donc intéressant de détailler les opérations effectuées :

1) L'élève-micro-processeur donne l'ordre à la mémoire-vive d'ouvrir une "case-mémoire" au nom de la variable numérique A (une étiquette avec la mention "A" est fixée sur une boîte à chaussure vide) ;

2) L'élève-micro-processeur interroge la mémoire-vive "quelle est la valeur de la variable LA ?". Un des élèves-mémoire-vive cherche le contenu de la case-mémoire qui porte l'étiquette LA et le recopie sur un papier-message (ici 24) transmis par l'élève-bus au micro-processeur. La même opération est effectuée pour la variable LO et l'élève-micro-processeur est alors en possession des deux valeurs de LA et LO soit 24 et 37.

Noter qu'il peut être tentant, pour aller plus vite, de transmettre au micro-processeur les papiers-mémoires placés dans les boîtes LO et LA ; **mais alors après lecture, ces deux cases-mémoires seraient vides ce qui ne traduirait pas la réalité** du fonctionnement de la machine puisque la lecture de la mémoire-vive n'a pas pour effet d'effacer les données enregistrées.

3) L'élève-micro-processeur n° 1 prépare un papier-message portant l'information : $37 * 24$ qu'il transmet à l'élève-micro-processeur n° 2 placé devant lui et chargé du traitement des informations. Ce dernier fait le calcul au brouillon ou à la calculatrice et écrit la réponse sur un **autre papier-mémoire** (soit ici 888) transmis à la mémoire-vive qui le place, sur ordre du micro-

processeur, dans la case-mémoire portant l'étiquette A. Il nous **semble ici nécessaire de ne pas utiliser le signe =** transcrit sur un papier portant l'écriture : $37 * 24 = 888$ car, comme nous l'avons indiqué le signe = a, dans la ligne 60, non pas son statut mathématique habituel, mais le sens de : "affecter à la variable A la valeur 888".

La ligne 70 donne lieu aux mêmes opérations qui aboutissent à placer dans une boîte à chaussure vide portant l'étiquette P (périmètre) la valeur calculée du périmètre soit 122 transcrite sur un papier-mémoire.

Quand le micro-processeur lit la ligne 80, il demande à la mémoire-vive le contenu de la case-mémoire A. Cette information lui est transmise après avoir été reproduite sur un message (soit 888). Le micro-processeur donne l'ordre à l'écran d'afficher : l'AIRE EST A = 888.

Avec la ligne 90 les mêmes opérations sont effectuées mais, cette fois, l'écran affiche LE PERIMETRE EST P = 122.

En recevant le contenu de la ligne 100 (END) l'élève-micro-processeur doit comprendre que l'exécution du programme est terminée et il écrit sur un papier-message "OK" envoyé à l'écran, puis un second papier-message portant la mention "-" (curseur clignotant) . L'utilisateur est ainsi averti que sa commande initiale (RUN) a été complètement exécutée et que l'ordinateur attend de nouvelles instructions.

* * *

Une fois cette première simulation faite, il ne faut pas manquer de la reprendre une seconde fois non seulement pour que les rôles soient tenus par d'autres élèves, jusqu'ici spectateurs, mais surtout parce que cette reprise est l'occasion de percevoir le rôle de certaines commandes et instructions et d'approcher le fonctionnement de la machine. Dans cette perspective on pourra par exemple simuler les effets des commandes : "LIST" (affichage du listing du programme), "CONTROLE C" (interruption d'un programme cours), "NEW" (effaçage de la mémoire-vive), etc . . . Avec les instructions PSET, LINE, BOX (fonctions graphiques), on sera conduit à utiliser un quadrillage. Ainsi sera créée une situation nouvelle et motivante pour travailler sur le thème du repérage dans le plan. Enfin en introduisant volontairement des erreurs au clavier, on percevra mieux certaines règles régissant le langage utilisé (adéquation des données entrées aux types de variables déclarés ; statut du signe = ; . . etc).

En simulant plusieurs fois de suite l'exécution du même programme les élèves seront à même de mieux comprendre le système de "remplissage" des variables. Si par exemple on reprend le programme ci-dessus avec des valeurs nouvelles pour LA et LO (par exemple 14,5 et 30), les élèves de la mémoire-vive devront enlever de la case-mémoire étiquetée "LA", le papier-mémoire portant la valeur précédente et y mettre à la place la nouvelle valeur : un papier-mémoire portant l'indication "14,5". En même temps devra être déchiré et jeté au panier le papier-mémoire portant l'indication 37 : on traduit ainsi le fait que, dans une variable, la nouvelle valeur "écrase" la précédente complètement oubliée. Le principe du "compteur" ($K = K + 1$) pourra aussi être mis en évidence grâce à un jeu de rôles : on retire de la boîte étiquetée "K" la valeur de K

précédemment entrée (par exemple 10) et on la remplace par cette même valeur "incrémentée" (soit 11) . . . etc.

On peut imaginer aussi que l'on décide de représenter le lecteur-enregistreur de cassettes relié au micro-processeur comme les autres périphériques. On pourra alors simuler les effets de la commande "SAVE" (enregistrer un programme stocké en mémoire-vive) et ceux de la commande "LOAD" (charger en mémoire-vive un programme enregistré sur cassette).

Toutes ces activités (et bien d'autres que nous n'avons pas décrites) pourront être conduites, on le voit, sans utiliser directement les machines. Il nous semble qu'elles s'adresseront à des élèves de cours moyen, une fois qu'un premier contact aura été pris avec les ordinateurs. Bien entendu on recherchera d'abord des situations simples et peut-être sera-t-il nécessaire, en **un premier temps, de procéder à une simulation en mode direct** : instruction PRINT, calculs à l'aide des fonctions numériques (usage des parenthèses, propriétés des opérations), activités simples sur les chaînes de caractères. L'exécution simulée d'un programme court (plus court que celui que nous avons proposé ci-dessus) viendra ensuite.

On gardera évidemment à l'esprit qu'en mettant en place un jeu de rôles inspiré de l'expérience décrite ici, il ne s'agira nullement de viser à une mémorisation du langage BASIC, ni à un entraînement systématique à la programmation : **notre ambition n'est pas de préparer des informaticiens dès l'école élémentaire**. Nous pensons cependant que des activités de ce type sont conformes aux directives de la circulaire ministérielle du 24 Mars 1983 qui recommande aux maîtres de proposer aux élèves des activités **d'éveil logistique** (initiation à la pensée algorithmique, entraînement à l'analyse des problèmes, au codage des actions élémentaires . . .) et des **activités d'éveil technologique** (éveil à la structure et au fonctionnement des ordinateurs, approche du cheminement et du traitement de l'information dans la machine). Il nous semble aussi que la "simulation informatique" organisée dans une classe est de nature à donner à des activités relevant d'autres disciplines (français et calcul) une tonalité nouvelle puissamment motivante : conjugaison, orthographe, calcul mental, usage des parenthèses en calcul, propriétés des opérations arithmétiques, repérage sur quadrillage, propriétés des figures géométriques, . . . etc.

L'exemple que nous avons traité ici utilise le langage BASIC. Il est souhaitable qu'un travail de simulation puisse être fait en LOGO afin de faire percevoir aux élèves que le principe du fonctionnement d'un ordinateur est indépendant du code. Disons seulement que dans ce cas l'importance de l'écran sera accrue et que l'accent sera davantage mis sur les déplacements de la tortue représentés au tableau (ou sous le préau, ou sur le sol de la salle polyvalente) et prévus par l'ensemble de la classe à partir de petits programmes simples. Le passage devant les machines permettra de vérifier si la réflexion collective menée sans ordinateurs a été pertinente ou pas. On disposera ainsi d'une situation très motivante pour étudier les angles, les rapports de longueur, les figures géométriques simples et aider les élèves à mieux structurer l'espace.

A chacun d'explorer cette voie dont nous n'avons ici donné qu'un aperçu de la richesse qu'elle recèle.