

KIM

Kit Informatique et Mathématique



Introduction

KIM, acronyme de Kit Informatique et Mathématique, est un ensemble de composants logiciels basés sur un thème commun. Les informaticiens appellent cela une bibliothèque de procédures.

Celui qui est décrit dans ce document se rapporte à la géométrie plane et à la géométrie analytique. D'autres KIM sont en cours de développement. Je citerai en particulier KIM Graphe qui permet de représenter graphiquement des fonctions et qui présente l'intérêt d'avoir un environnement graphique proche de celui de KIM Géométrie plane.

KIM est destiné à une approche de la programmation structurée à travers des activités mathématiques du niveau du collège. KIM présente une analogie certaine avec les micro-mondes développés en Logo sous l'impulsion de S. Papert. Bien que le langage informatique utilisé soit différent, leurs objectifs sont communs.

KIM et les micro-mondes Logo sont élaborés dans une perspective constructiviste. Leurs utilisateurs évoluent dans un cadre structuré mais non fermé. Ils peuvent le faire évoluer, l'étendre. Ils peuvent repousser les murs. Savoir réaliser cette extension, est l'un des objectifs cognitifs principaux de leur pratique. Les activités proposées se déroulant dans le champ de l'informatique et des mathématiques, les acteurs mettent en œuvre des outils spécialisés dans ces deux domaines. La maîtrise de ces outils constitue un autre volet des objectifs à atteindre. L'intérêt est de mettre en parallèle les outils utilisés dans ces deux domaines et, de par leur interaction, de mettre en exergue des compétences transversales et transférables. Cette capacité de transfert cognitif n'est-elle pas l'une des composantes de l'intelligence ?

Sans souci d'exhaustivité, le tableau suivant permet de préciser ces notions :

Compétences		
en informatique	en mathématique	transversales/cognitives
<ul style="list-style-type: none"> • Décomposer un programme en procédures • Décomposer une procédure en instructions simples 	<ul style="list-style-type: none"> • Analyser une figure géométrique • Savoir la décrire avec le vocabulaire approprié 	<ul style="list-style-type: none"> • Analyser une situation • Savoir la décrire
<ul style="list-style-type: none"> • (Re)construire un programme par une succession de procédures. • (Re)construire une procédure par une succession d'instructions. 	<ul style="list-style-type: none"> • (Re)construire une figure géométrique • Mise en équation d'un problème • Utilisation de variables, d'inconnues 	<ul style="list-style-type: none"> • (Re)construire un objet géométrique
<ul style="list-style-type: none"> • Savoir élaborer une nouvelle procédure à partir de cas particuliers 	<ul style="list-style-type: none"> • S'appropriier les propriétés importantes d'une figure. 	<ul style="list-style-type: none"> • Savoir généraliser

A la lumière des résultats des expérimentations Logo, je ne laisse pas l'élève se « débrouiller » sans lui avoir donné au préalable, par une pédagogie structurée, les connaissances nécessaires à son autonomie (c.f. le scénario pédagogique

proposé ultérieurement). Je ne perds pas de vue que les objectifs de la pratique de KIM sont la découverte de la programmation (structurée) en informatique, le renforcement des connaissances en mathématique et l'appropriation de méthodologies relevant du transfert cognitif. Cette pédagogie est un cadre et n'exclut pas, bien au contraire, les moments où l'élève peut exprimer sa créativité.

Je sais aussi que les résultats mesurés ou constatés à la suite de la pratique des ateliers Logo sont décevants par rapport aux espoirs qu'ils avaient faits naître. Se pourrait-il que Logo ne soit pas le langage informatique le plus à même de développer les capacités cognitives de l'enfant ? Sa conception relève de circonstances historiques : Il dérive du langage informatique (Lisp) le plus en vogue, à l'époque, dans les laboratoires d'**intelligence** artificielle que fréquentait S. Papert. De là à l'associer au développement de l'intelligence, en l'adaptant aux possibilités de l'enfant, il n'y avait qu'un pas que S. Papert a franchi avec un enthousiasme communicatif. Le problème est que Logo a ses spécificités, en particulier, la programmation récursive, qui ont été mises en exergue par S. Papert. Ces préoccupations sont éloignées des pratiques méthodologiques de l'élève qui ne les rencontre que tardivement dans son cursus scolaire (fin de l'enseignement secondaire).

Les objectifs de KIM étant plus spécifiques, j'ai choisi

- D'une part, un langage informatique accusé de bien des défauts : le Basic. Pas le Basic des débuts, celui tant décrié - en particulier, par S. Papert - car il ne permettait pas la programmation procédurale, et par conséquent, le développement de compétences cognitives associées, mais une version récente qui ne souffre pas de ces défauts. Le choix aurait pu se porter sur un autre langage informatique de même nature, tel le Pascal²² ou le C, mais il se trouve que QBasic est disponible à un coût beaucoup plus faible et que, bien qu'il ne dispose pas de toutes les fonctionnalités de ces deux langages, celles qu'il propose, suffisent à la réalisation des objectifs fixés.

L'organisation générale de chaque KIM (environnement graphique et pédagogique associé à un ensemble de procédures de base) permet d'ailleurs une traduction dans les autres langages précités, même dans leur forme les plus récentes relevant de la programmation objet, de la programmation événementielle ou de la programmation visuelle²³.

- Et d'autre part, des applications mathématiques plus proches du vécu d'un élève de collège, telle la géométrie euclidienne ou la géométrie associée à un repère cartésien (analytique). Il n'a que faire de la géométrie auto-référentielle de Logo, beaucoup plus proche des préoccupations d'un enfant plus jeune (découverte de l'espace environnant).

Après cet exposé introductif et justificatif, je propose au lecteur de partir à la découverte de KIM Géométrie plane.

²² Ouvrage d'initiation à la programmation en Pascal, inspiré par Logo : Cousot Laurent, Patrick, Radhia et Thibault, *Premières leçons de programmation en Turbo Pascal*, MacGraw-Hill, Paris, 1991

²³ La programmation visuelle prit son essor grâce au cheval de Troie que fut Visual Basic (Microsoft).

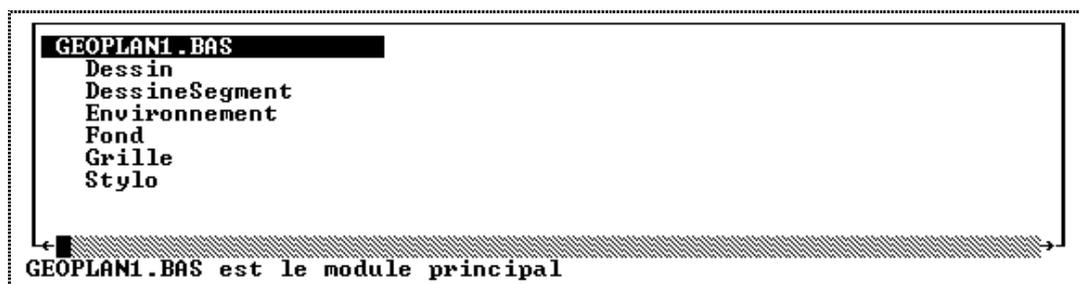
Les composants logiciels de KIM Géométrie plane

Les composants de KIM Géométrie plane sont répartis dans trois programmes gigognes :

- **GEOPLAN 1** contient le minimum de procédures. Son usage le destine plutôt à la découverte et à la prise en main. Le scénario pédagogique proposé ultérieurement n'utilise que cette bibliothèque logicielle.
- **GEOPLAN 2** contient un plus grand nombre de procédures utiles à la construction de figures de géométrie plane.
- **GEOPLAN 3** complète GEOPLAN 2 par un ensemble de nouvelles procédures construites à partir de celles définies dans les deux premiers modules. La construction de ces procédures peut faire l'objet de recherche et donc servir d'objectifs pédagogiques ou servir de modèle à la construction d'autres similaires.

Les composants de KIM GEOPLAN 1

La liste des procédures disponibles apparaît à l'utilisateur sous la forme :



L'accès à cette liste se fait par l'intermédiaire des commandes du menu supérieur :

Affichage/Subs...

La procédure Environnement

La procédure Environnement définit l'environnement graphique. Elle affiche à l'écran une fenêtre graphique accompagnée d'un système de coordonnées.

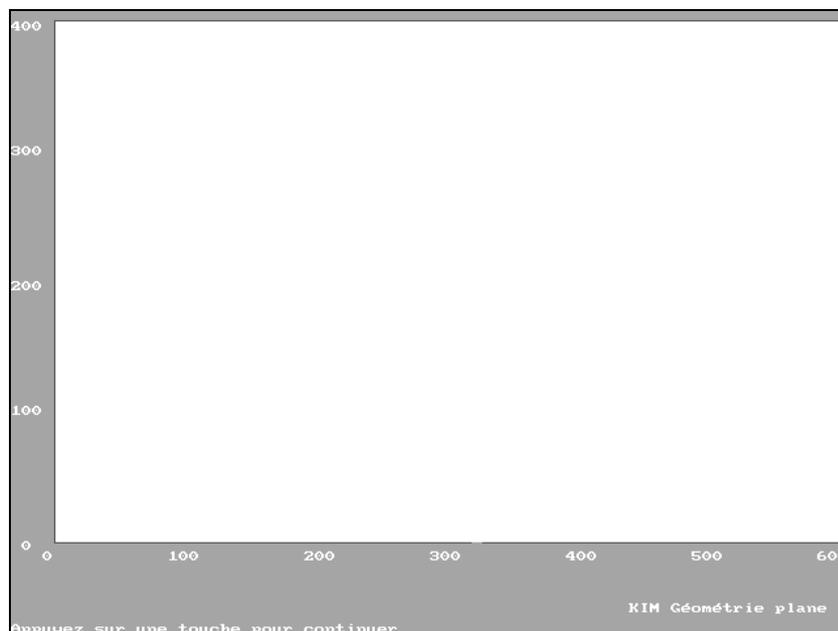
La fenêtre graphique (de fond blanc) est destinée à recevoir les figures géométriques réalisées par l'utilisateur.

Le système de coordonnées permet de définir la position des points importants de la figure. La position horizontale (abscisse) d'un point doit être un nombre entier compris entre 0 et 600, sa position verticale (ordonnée) un nombre entier compris entre 0 et 400. Deux points contigus de l'écran sont séparés par la valeur 1.

Le mode graphique a été choisi de manière :

- à obtenir le plus de précision possible

- à conserver l'isotropie de la fenêtre graphique. Par exemple, cela permet de dessiner un cercle dont l'aspect à l'écran soit bien celui d'un cercle et pas celui d'une ellipse.

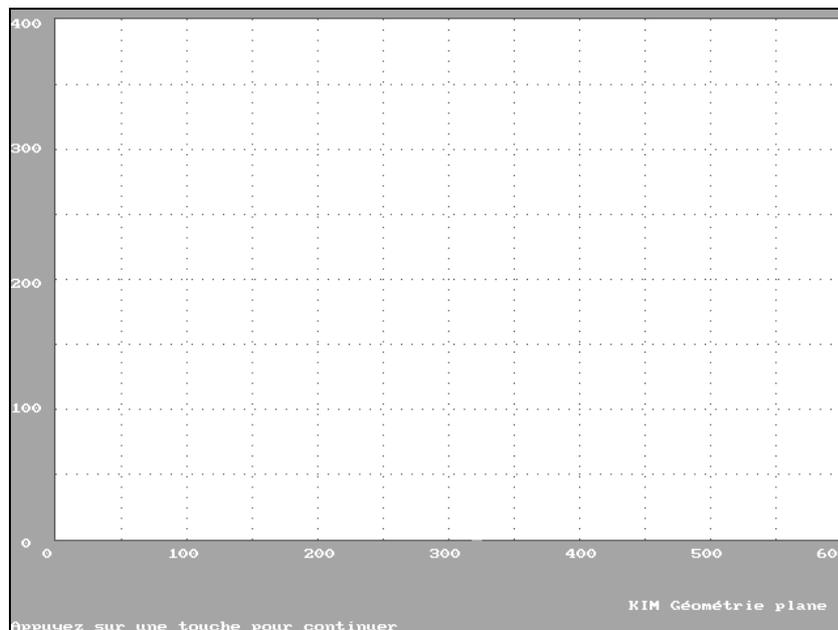


Ce choix a entraîné des contraintes au niveau des choix de couleur pour le fond d'écran, qui n'ont pu être levées, en partie, que par l'usage d'une routine écrite en assembleur (appelée par la procédure Fond).

La procédure Grille

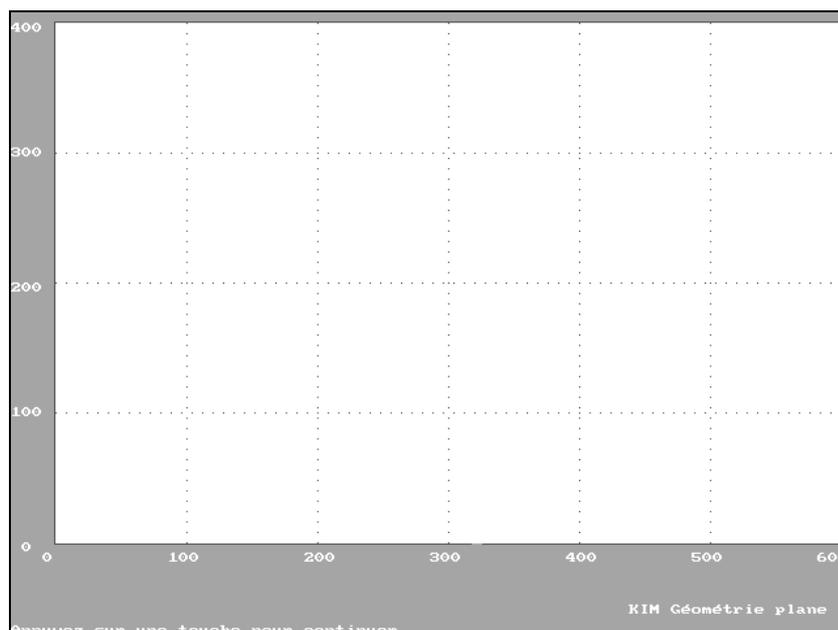
La procédure Grille permet de superposer à la fenêtre graphique un quadrillage. L'intervalle entre deux lignes est défini comme paramètre de la procédure.

La réalisation de cette procédure relève d'une volonté d'aide pédagogique : il est beaucoup plus facile de repérer la position d'un point avec un quadrillage que sans et ainsi, de pouvoir corriger soi-même d'éventuelles erreurs.



Résultat obtenu avec Grille 50





Résultat obtenu avec Grille 100



La procédure Stylo

La procédure Stylo définit la couleur du prochain élément géométrique à dessiner (point, segment, droite, cercle et autres), ceci tant qu'une nouvelle couleur ne sera pas choisie.

Après l'instruction

```
Stylo Bleu
```

l'élément géométrique suivant sera dessiné en bleu.

Les couleurs disponibles sont définies dans le programme principal :

```
CONST Cyan      = 1      ' codes des couleurs
CONST Mauve     = 2
CONST Gris      = 3
CONST Carmin    = 4
CONST Noir      = 8
CONST Bleu      = 9
CONST Vert      = 10
CONST Rouge     = 12
CONST Blanc     = 15
```

A chaque couleur correspond un code numérique. De fait, les deux instructions

```
Stylo Bleu
```

```
Stylo 9
```

sont équivalentes et donnent le même résultat.

La procédure Dessin

La procédure Dessin est la seule procédure qu'il soit nécessaire de modifier. C'est elle qui contient la suite d'instructions nécessaires à la réalisation du dessin.

Pour la modifier efficacement, il faut, dans l'ordre :

- Déclarer les points nécessaires à la construction de la figure

exemple

```
' Déclarations
DIM A AS Pt          'A est un Pt
DIM B AS Pt          'B est un Pt
DIM C AS Pt          'C est un Pt
```

Les éléments de texte suivant une apostrophe (') sont des commentaires et sont facultatifs. Ils ne sont pas indispensables au bon fonctionnement du programme mais permettent d'informer le lecteur en explicitant certaines parties de celui-ci.

- Définir la position de ces points, en précisant pour chacun d'entre eux, son abscisse puis son ordonnée

exemple

```
' Définir les coordonnées des points A, B et C
A.X = 100: A.Y = 100      'A.X : abscisse de A ; A.Y : ordonnée de A
B.X = 100: B.Y = 300      'B.X : abscisse de B ; B.Y : ordonnée de B
C.X = 400: C.Y = 300      'C.X : abscisse de C ; C.Y : ordonnée de C
```

et seulement ensuite,

- écrire la suite d'instructions nécessaires à l'exécution de la figure.

La procédure DessineSegment

La procédure DessineSegment dessine un segment de droite entre deux points. Elle nécessite deux paramètres : le nom de chacun des points aux extrémités du segment.

Exemple d'utilisation (extrait de la procédure Dessin)

```
' Déclarations
    DIM A AS Pt          'A est un Pt
    DIM B AS Pt          'B est un Pt

' Actions
' Définir les coordonnées des points A et B
    A.X = 100: A.Y = 50   'A.X : abscisse de A ; A.Y : ordonnée de A
    B.X = 500: B.Y = 300 'B.X : abscisse de B ; B.Y : ordonnée de B

    Grille 100
    Stylo Bleu
    DessineSegment A, B   'Dessin du segment [AB]
```


La procédure DessinePoint

J'attire l'attention du lecteur sur le fait que cette procédure se nomme bien DessinePoint et non pas DessinePt.

Cette procédure matérialise un point sous forme d'une croix (représentation usuelle en géométrie). Elle nécessite un seul paramètre à sa suite : le nom du point.

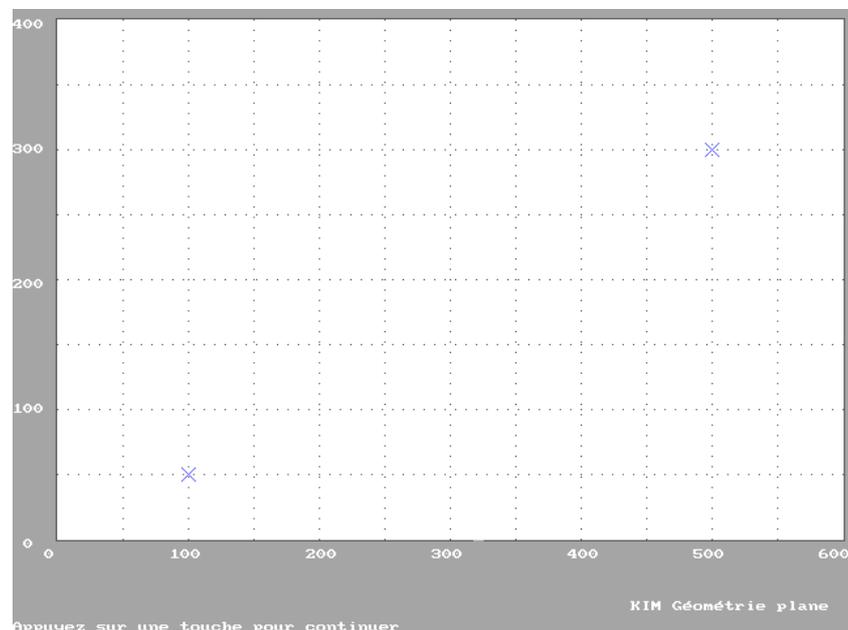
Exemple d'utilisation (extrait de la procédure Dessin)

```
' Déclarations
  DIM A AS Pt   'A est un Pt
  DIM B AS Pt   'B est un Pt

' Définir les coordonnées des points A et B
  A.X = 100: A.Y = 50
  B.X = 500: B.Y = 300

\ Actions
  Grille 50
  Stylo Bleu
  DessinePoint A
  DessinePoint B
```

Résultat



La procédure DessineCercle

La procédure DessineCercle nécessite à sa suite deux paramètres : le nom du point central et la valeur numérique du rayon de ce cercle.

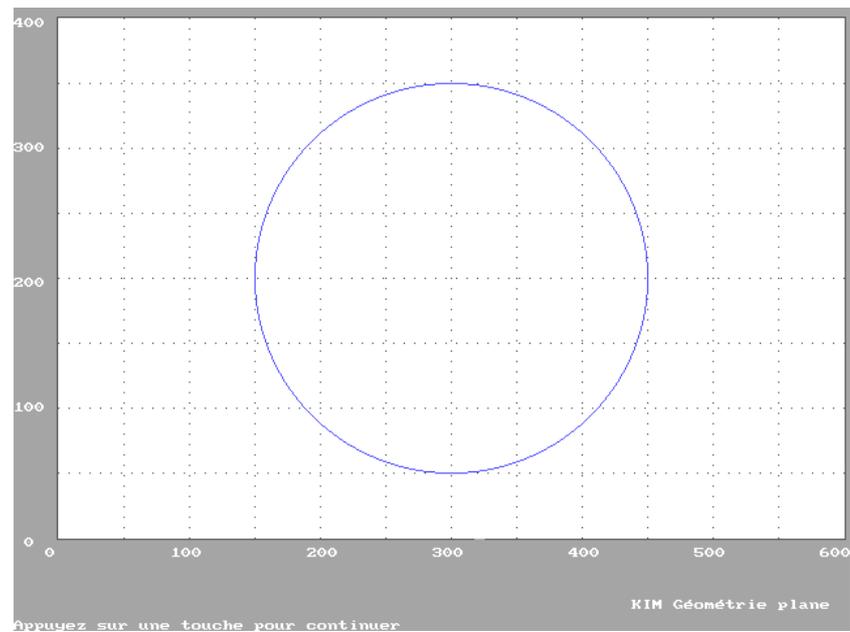
Exemple d'utilisation (extrait de la procédure Dessin)

```
' Déclarations
  DIM A AS Pt          'A est un Pt

' Définir les coordonnées du point A
  A.X = 300: A.Y = 200

' Actions
  Grille 50
  Stylo Bleu
  DessineCercle A, 150
```

Résultat



La procédure DessineDroite

La procédure DessineDroite est analogue à la procédure DessineSegment. La différence visible est que le trait dessiné s'étend jusqu'aux limites de la fenêtre graphique. Elle nécessite deux paramètres : les noms des deux points par lesquels passe la droite.

Exemple d'utilisation (extrait de la procédure Dessin)

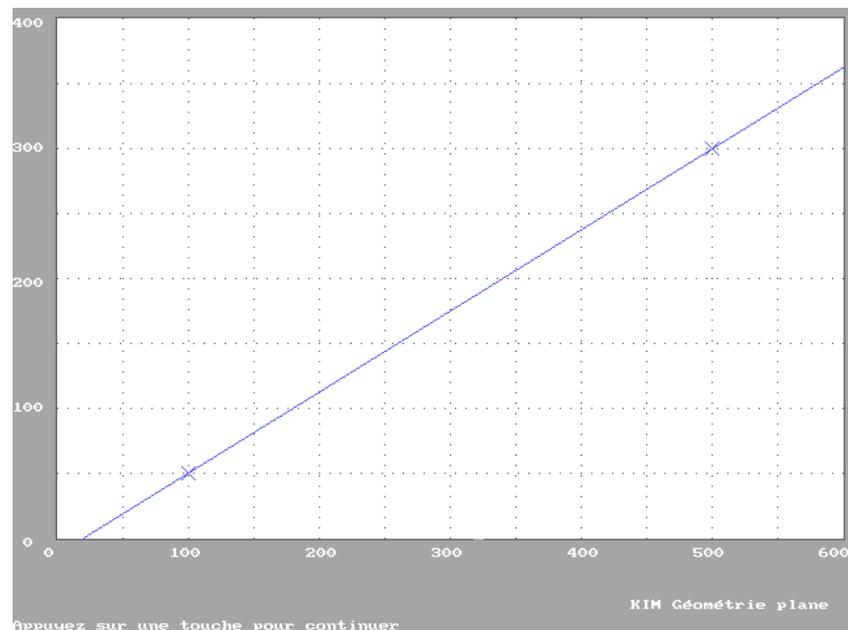
```
' Déclarations
  DIM A AS Pt   'A est un Pt
  DIM B AS Pt   'B est un Pt

' Définir les coordonnées des points A et B
  A.X = 100: A.Y = 50
  B.X = 500: B.Y = 300

\ Actions
  Grille 50
  Stylo Bleu
  DessineDroite A, B
  DessinePoint A
  DessinePoint B
```

Dans cet exemple, les instructions DessinePoint sont utilisées pour matérialiser la position des points A et B.

Résultat



La procédure DessineParallele

La procédure DessineParallele (sans accent sur l'avant-dernier « e » !) dessine une droite parallèle à une droite donnée. Cette procédure nécessite trois paramètres : les noms de trois points.

Exemple : L'instruction **DessineParallele A, B, C** dessine une droite qui passe par A parallèlement à la droite (BC).

Exemple d'utilisation (extrait de la procédure Dessin)

```
' Déclarations
  DIM A AS Pt   'A est un Pt
  DIM B AS Pt   'B est un Pt
  DIM C AS Pt   'C est un Pt

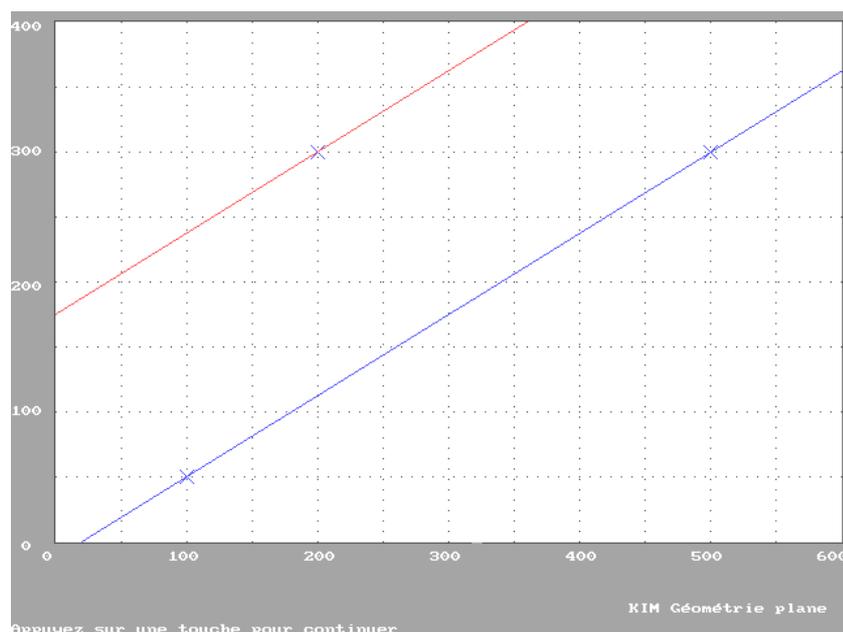
' Définir les coordonnées des points A et B
  A.X = 100: A.Y = 50
  B.X = 500: B.Y = 300
  C.X = 200: C.Y = 300

' Actions
  Grille 50

  Stylo Bleu
  DessinePoint A
  DessinePoint B
  DessinePoint C
  DessineDroite A, B

  Stylo Rouge
  DessineParallele C, A, B   ' Parallèle à (AB) passant par C.
```

Résultat



Commentaire mathématique

Dans le code de cette procédure, les élèves des classes de troisième retrouveront l'utilisation de la propriété : deux droites sont parallèles si et seulement si elles ont le même coefficient directeur. Avec la notation des coordonnées d'un point utilisée dans ce programme informatique, cette propriété se traduit par une égalité de la forme $\frac{P.Y - C.Y}{P.X - C.X} = \frac{B.Y - A.Y}{C.X - A.X}$ où P désigne un point quelconque de la droite passant par C parallèlement à la droite (AB).

La procédure DessinePerpendiculaire

La procédure DessinePerpendiculaire est conçue de manière analogue à la précédente DessineParallele et nécessite trois paramètres, les noms de trois points.

Exemple : DessinePerpendiculaire A, B, C dessine une droite passant par A perpendiculairement à la droite (BC).

Commentaire mathématique

Cette fois encore, le lecteur trouvera dans le code de cette procédure l'utilisation d'une autre propriété connue : deux droites sont perpendiculaires si et seulement si le produit de leur coefficient directeur est égal à -1. Avec la notation des coordonnées d'un point utilisée dans ce programme informatique, cette propriété se traduit par une égalité de la forme $\frac{P.Y - C.Y}{P.X - C.X} \times \frac{B.Y - A.Y}{C.X - A.X} = -1$ où P désigne un point quelconque de la droite passant par C perpendiculairement à la droite (AB).

Exemple d'utilisation (extrait de la procédure Dessin)

```
' Déclarations
  DIM A AS Pt    'A est un Pt
  DIM B AS Pt    'B est un Pt
  DIM C AS Pt    'C est un Pt

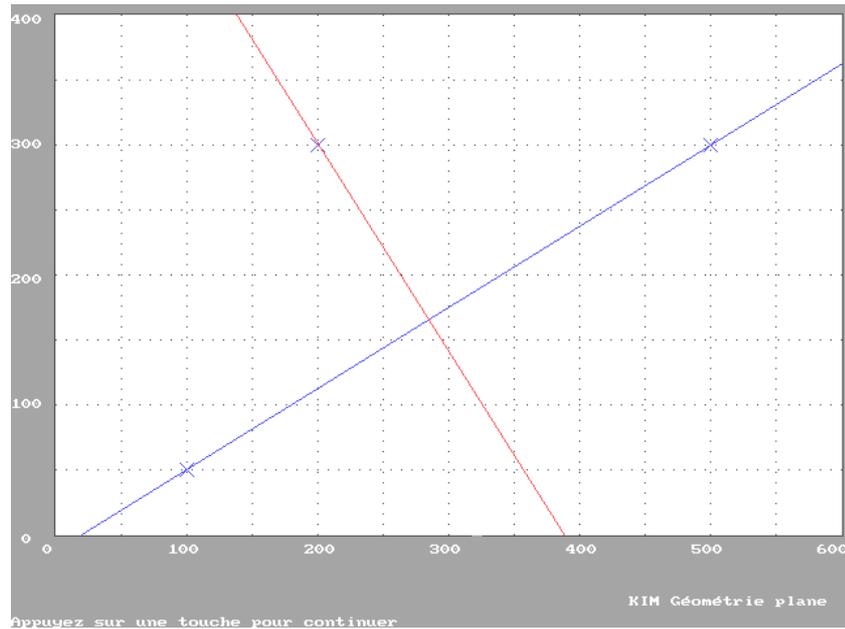
' Définir les coordonnées des points A et B
  A.X = 100: A.Y = 50
  B.X = 500: B.Y = 300
  C.X = 200: C.Y = 300

\ Actions
  Grille 50

  Stylo Bleu
  DessinePoint A
  DessinePoint B
  DessinePoint C
  DessineDroite A, B

  Stylo Rouge
  DessinePerpendiculaire C, A, B ' perpendiculaire à (AB) passant par C
```

Résultat



La procédure Milieu

La procédure Milieu définit la position d'un point, milieu de deux autres. Cette procédure nécessite trois paramètres qui désignent trois points.

Exemple

Milieu A, B, I attribue au point I les coordonnées du milieu de [AB].

Exemple d'utilisation (extrait de la procédure Dessin)

```
' Déclarations
  DIM A AS Pt    'A est un Pt
  DIM B AS Pt    'B est un Pt
  DIM I AS Pt    'I est un Pt

' Corps de la procédure

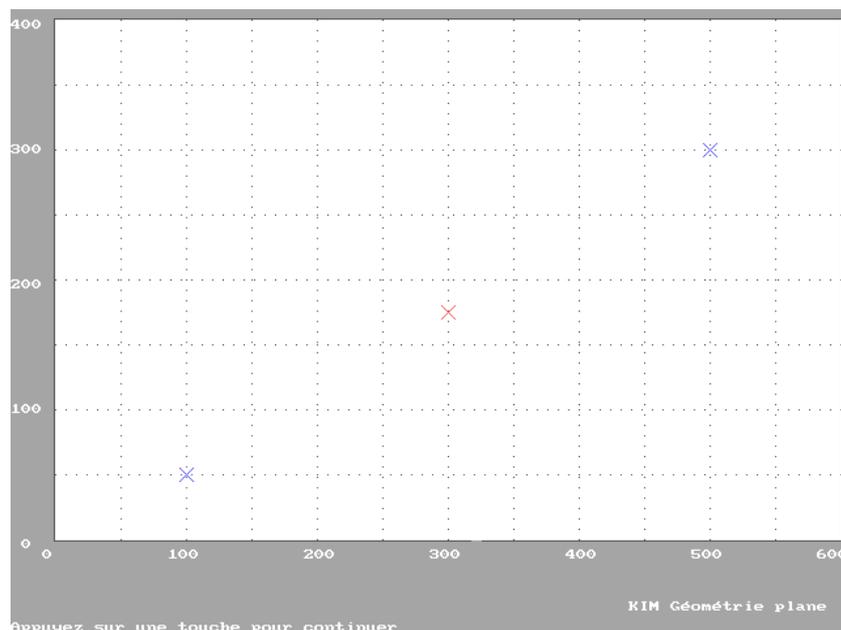
' Définir les coordonnées des points A et B
  A.X = 100: A.Y = 50
  B.X = 500: B.Y = 300

  Grille 50
  Milieu A, B, I      ' I milieu de [AB] est défini, mais non dessiné

  Stylo Bleu
  DessinePoint A
  DessinePoint B

  Stylo Rouge
  DessinePoint I      ' I est dessiné
```

Résultat



En disposant de l'ensemble des procédures incluses dans KIM GEOPLAN 2, l'utilisateur est mieux armé pour construire de nouvelles figures géométriques. La description de leur processus de construction, sous la forme d'une procédure, transforme le statut de ces figures. L'outil procédure permet de construire un objet géométrique, qui devient lui-même outil et permet de reproduire le processus. Avec la construction de composants logiciels, nous nous trouvons en présence d'une forme d'« implémentation », dans le cadre informatique, de la dialectique outil-objet²⁴ chère à de nombreux didacticiens en mathématique. Dans cette théorie du développement cognitif, les outils qui servent à agir sur les connaissances, deviennent eux-mêmes, une fois exhibés et nommés, objets de connaissance.

C'est dans cette optique qu'ont été élaborées les nouvelles procédures disponibles dans KIM GEOPLAN 3.

Les composants de KIM GEOPLAN 3

La liste des procédures proposées apparaît à l'utilisateur sous la forme suivante :

```

GEOPLAN3.BAS
Dessin
DessineCercle
DessineDroite
DessineHauteurs
DessineMedianes
DessineMediatrice
DessineMediatrices
DessineParallele
Dessine...iculaire
DessinePoint
DessineSegment
DessineTriangle
Environnement
Fond
Grille
Milieu
Stylo
  
```

← **GEOPLAN3.BAS est le module principal** →

²⁴ Douady Régine, *Jeux de cadre et dialectique outil-objet dans l'enseignement des mathématiques*, thèse d'état, Paris VII, 1984

Les nouvelles procédures sont des exemples de ce qu'il est possible de réaliser à partir de celles disponibles dans KIM GEOPLAN 2.

La procédure DessineTriangle

La procédure DessineTriangle nécessite trois paramètres : les sommets du triangle.

Code de la procédure

```
SUB DessineTriangle (P1 AS Pt, P2
AS Pt, P3 AS Pt)
    DessineSegment P1, P2
    DessineSegment P1, P3
    DessineSegment P2, P3
END SUB
```

Le triangle est composé de trois segments.

Exemple d'utilisation (extrait de la procédure Dessin)

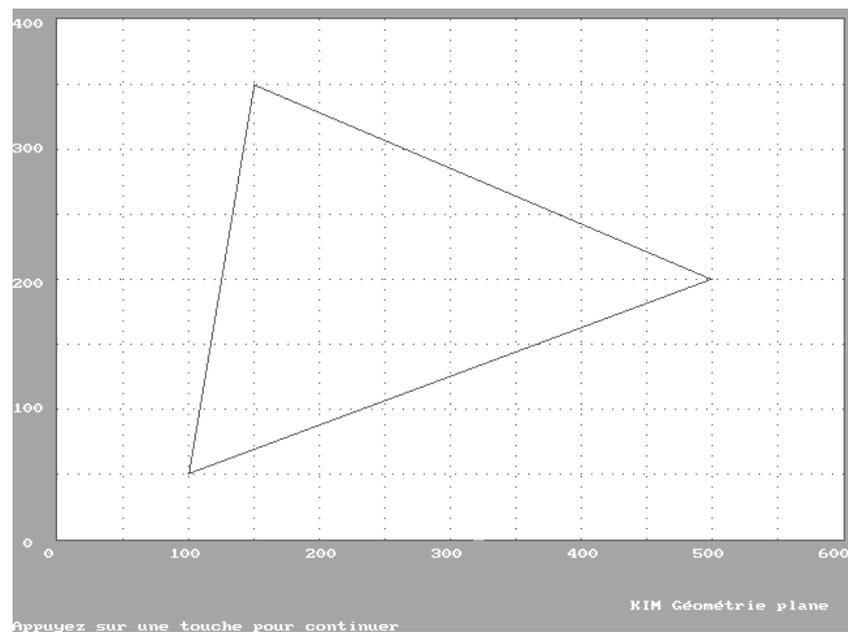
```
' Déclarations
    DIM A AS Pt    'A est un Pt
    DIM B AS Pt    'B est un Pt
    DIM C AS Pt    'C est un Pt

' Définir les coordonnées des
points A,B et C
    A.X = 100: A.Y = 50
    B.X = 500: B.Y = 200
    C.X = 150: C.Y = 350

' Actions
    Grille 50

    Stylo Noir
    DessineTriangle A, B, C
```

Résultat



La procédure DessineMediatrice

La procédure DessineMediatrice (sans accent sur l'avant-dernier « e » !) nécessite deux paramètres : les points qui désignent les extrémités du segment de droite dont on veut tracer la médiatrice.

Code de la procédure

```
SUB DessineMediatrice (P1 AS Pt,
P2 AS Pt)
' Déclaration
  DIM M AS Pt
' Action
  Milieu P1, P2, M
  DessinePerpendiculaire M, P1,
P2
END SUB
```

La procédure trace la perpendiculaire au segment passant par son milieu.

Exemple d'utilisation (extrait de la procédure Dessin)

```
' Déclarations
  DIM A AS Pt 'A est un Pt
  DIM B AS Pt 'B est un Pt

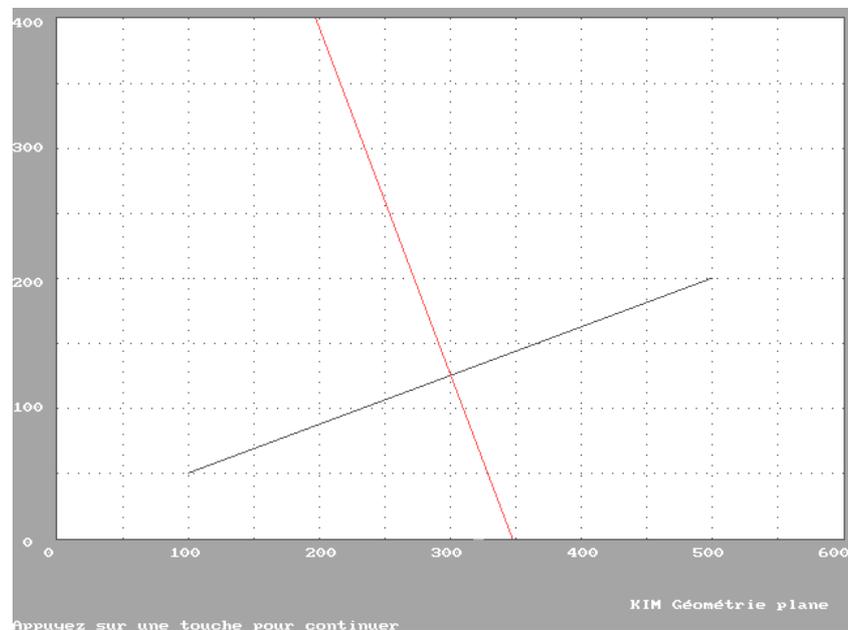
' Définir les coordonnées des
points A et B
  A.X = 100: A.Y = 50
  B.X = 500: B.Y = 200

' Actions
  Grille 50

  Stylo Noir
  DessineSegment A, B

  Stylo Rouge
  DessineMediatrice A, B
```

Résultat



La procédure DessineMediatrices

La procédure DessineMediatrices (avec un s final pour la distinguer de la procédure précédemment décrite) trace les médiatrices des trois côtés d'un triangle. Elle nécessite trois paramètres : les points désignant les sommets du triangle.

Code de la procédure

```
SUB DessineMediatrices (P1 AS Pt,
P2 AS Pt, P3 AS Pt)
    DessineMediatrice P1, P2
    DessineMediatrice P2, P3
    DessineMediatrice P1, P3
END SUB
```

La programmation fait appel à la procédure DessineMediatrice.

Exemple d'utilisation (extrait de la procédure Dessin)

```
' Déclarations
    DIM A AS Pt    'A est un Pt
    DIM B AS Pt    'B est un Pt
    DIM C AS Pt    'C est un Pt

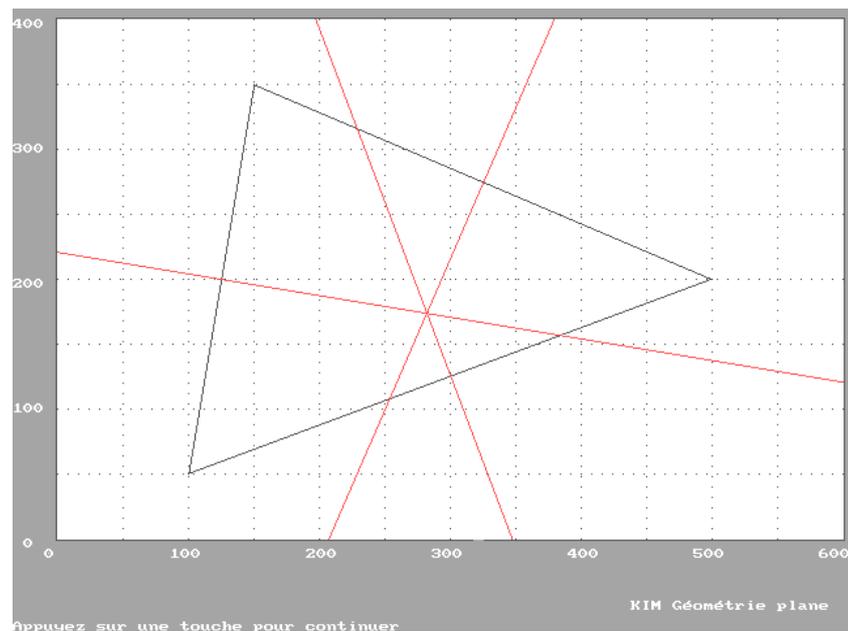
' Définir les coordonnées des
points A et B
    A.X = 100: A.Y = 50
    B.X = 500: B.Y = 200
    C.X = 150: C.Y = 350

' Actions
    Grille 50

    Stylo Noir
    DessineTriangle A, B, C

    Stylo Rouge
    DessineMediatrices A, B, C
```

Résultat



Le lecteur trouvera d'autres procédures telles DessineHauteurs et DessineMedianes élaborées suivant le modèle de DessineMediatrices dans KIM GEOPLAN 3.

Exemple d'utilisation (extrait de la procédure Dessin de KIM GEOPLAN 3)

```
' Déclarations
DIM A AS Pt          'A est un Pt
DIM B AS Pt          'B est un Pt
DIM C AS Pt          'C est un Pt
DIM O AS Pt          'O Est un Pt

' Définir les coordonnées des points A, B et C
A.X = 100: A.Y = 100
B.X = 500: B.Y = 50
C.X = 400: C.Y = 300

' Actions
Grille 100

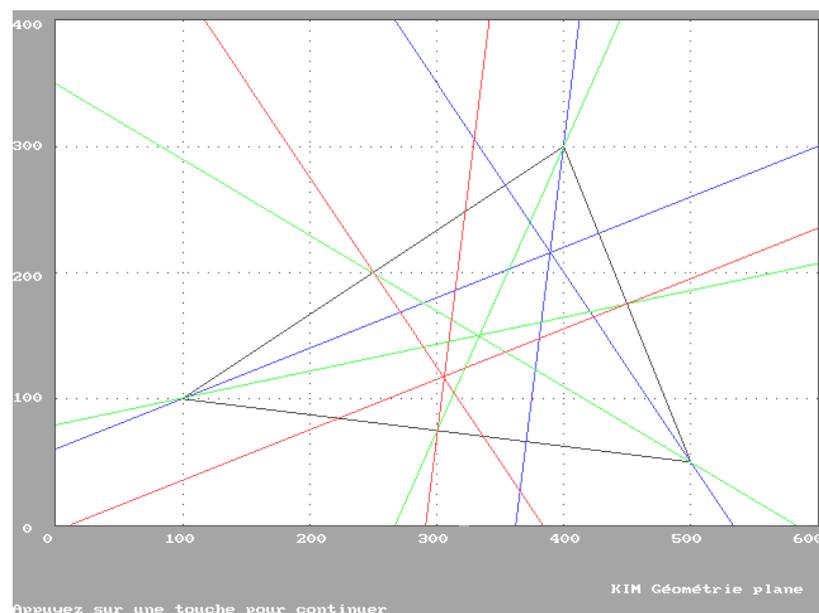
Stylo Noir
DessineTriangle A, B, C

Stylo Bleu
DessineHauteurs A, B, C

Stylo Vert
DessineMedianes A, B, C

Stylo Rouge
DessineMediatrices A, B, C
```

Résultat



Scénario pédagogique

Le scénario est découpé en quatre séquences :

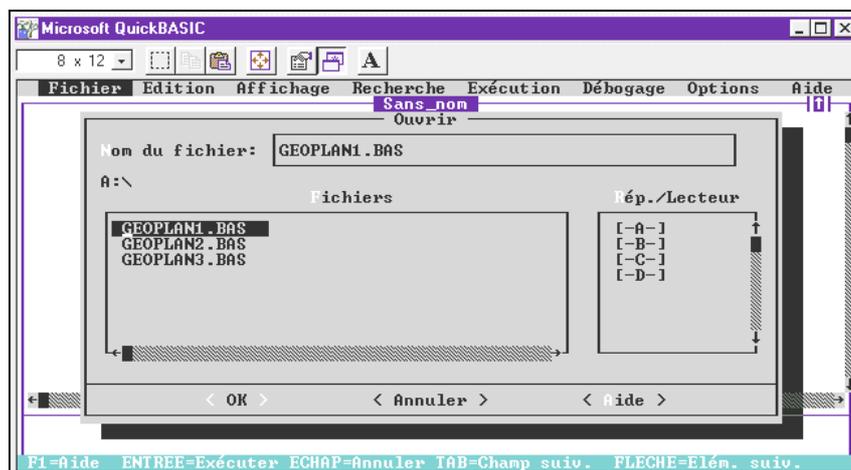
- **L'enveloppe** : prise en main de l'éditeur de Qbasic suivi d'une première réalisation (dessin de l'enveloppe).
- **Le carré** : après la réalisation d'une première procédure, modification du code de manière à utiliser une variable représentant la longueur d'un côté du carré.
- **La procédure DessineCarre** : transformation de la procédure précédente en une nouvelle procédure avec paramètre : DessineCarre avec utilisation du copier/coller, puis utilisation de cette procédure pour dessiner une tête stylisée.
- **Les carrés emboîtés** : après la réalisation d'un premier programme, modification de son code de manière à utiliser le concept d'itération en programmation : la boucle POUR (FOR...NEXT pour les anglophones).

L'enveloppe

Charger KIM GEOPLAN 1

Effectuer successivement les actions suivantes :

1. Mettre en service QBASIC.EXE.
2. Appuyer sur la touche [ECHAP] pour éviter la fenêtre d'aide.
3. Dans le menu supérieur, choisir les options Fichier, puis Ouvrir...
4. Dans le cadre Rép./Lecteur, sélectionner le lecteur A, cliquer sur <OK>, en bas de la fenêtre.
5. Dans le cadre Fichiers, sélectionner le fichier GEOPLAN1, cliquer sur <OK>, en bas de la fenêtre.



Si tout se passe comme prévu, le programme GEOPLAN1 est chargé dans la mémoire de l'ordinateur. Il est composé de différentes parties accessibles par l'option Affichage du menu supérieur.

Sélectionner la procédure Dessin

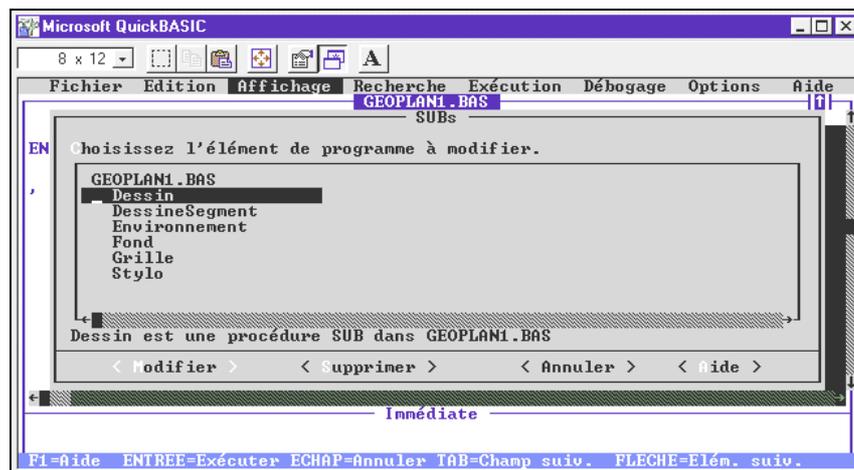
La seule procédure dont il est nécessaire de modifier le code pour obtenir un nouveau dessin, est la procédure Dessin. Le code des autres procédures peut

être consulté à loisir, mais il faut prendre garde de ne pas les modifier, sous peine de dysfonctionnement.

Pour modifier la procédure Dessin :

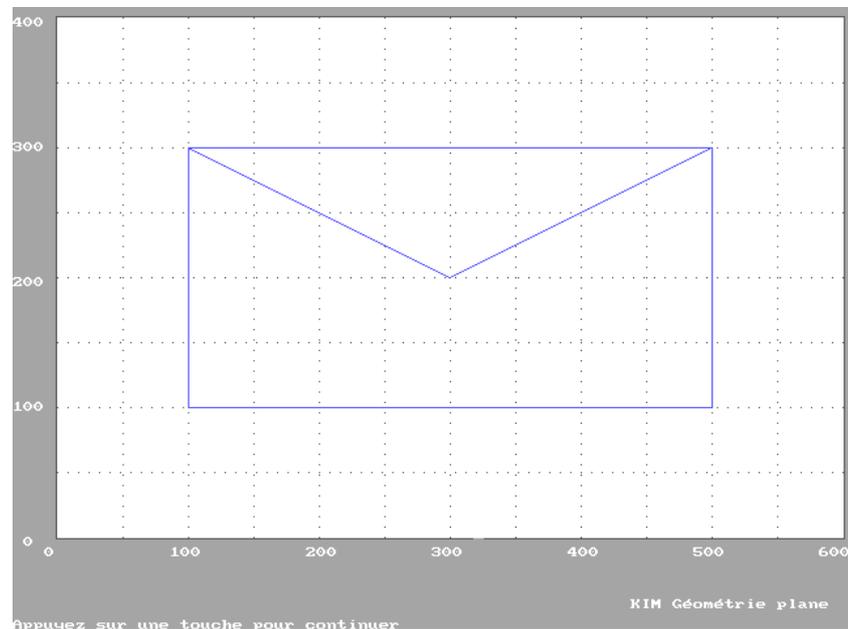
Dans le menu supérieur, choisir les options Affichage puis SUBs...

Dans le cadre SUBs, sélectionner la procédure Dessin, cliquer sur <Modifier>, en bas de la fenêtre.



Modifier la procédure Dessin

Il s'agit de modifier la procédure, pour obtenir le résultat suivant.



Pour cela, il faut :

- Réfléchir à la manière dont on procéderait avec une règle et un crayon et indiquer à l'ordinateur toutes les actions à effectuer
- Repérer la position de tous les points importants de la figure, les déclarer
exemple pour le point A :

```
DIM A AS Pt
```

- Déterminer leurs coordonnées (abscisse et ordonnée), les recopier.

Exemple pour le point A :

```
A.X = 100 : A.Y = 50
```

- Choisir une ou plusieurs couleurs et tracer tous les segments de droites nécessaires.

Une des nombreuses solutions possibles pour la procédure Dessin, est :

```
SUB Dessin
' Déclarations
  DIM A AS Pt 'A est un Pt
  DIM B AS Pt 'B est un Pt
  DIM C AS Pt 'C est un Pt
  DIM D AS Pt 'D est un Pt
  DIM E AS Pt 'E est un Pt

' Définir les coordonnées des points A, B, C, D et E.
  A.X = 100: A.Y = 100
  B.X = 100: B.Y = 300
  C.X = 500: C.Y = 300
  D.X = 500: D.Y = 100
  E.X = 300: E.Y = 200

' Actions
  Grille 50
  Stylo Bleu
  DessineSegment A, B
  DessineSegment B, C
  DessineSegment C, D
  DessineSegment D, A
  DessineSegment B, E
  DessineSegment E, C

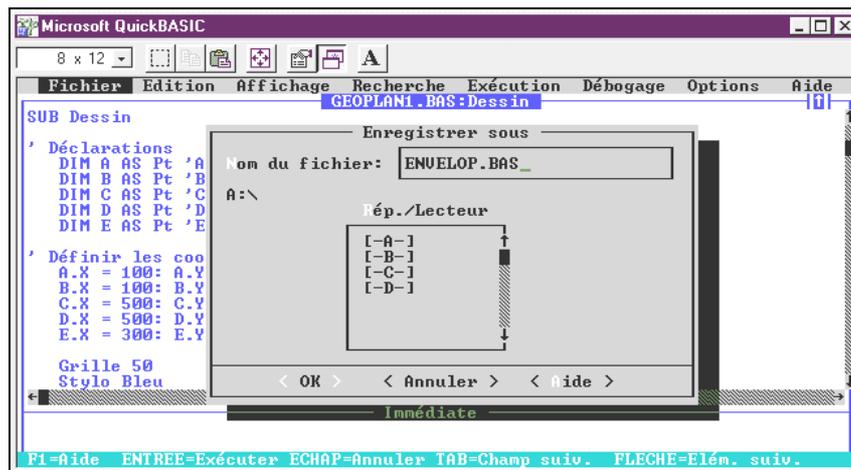
END SUB
```

Enregistrer le dessin

Même si le dessin n'est pas terminé, le programme informatique doit être enregistré pour pouvoir être terminé ou consulté lors d'une prochaine séance.

Pour cela, effectuer les actions suivantes :

- Dans le menu supérieur, choisir les options Fichier, puis Enregistrer sous...
- Dans le cadre Nom du fichier, remplacer GEOPLAN1.BAS par un autre nom, par exemple : ENVELOP.BAS. Cliquer sur <OK> en bas de la fenêtre.

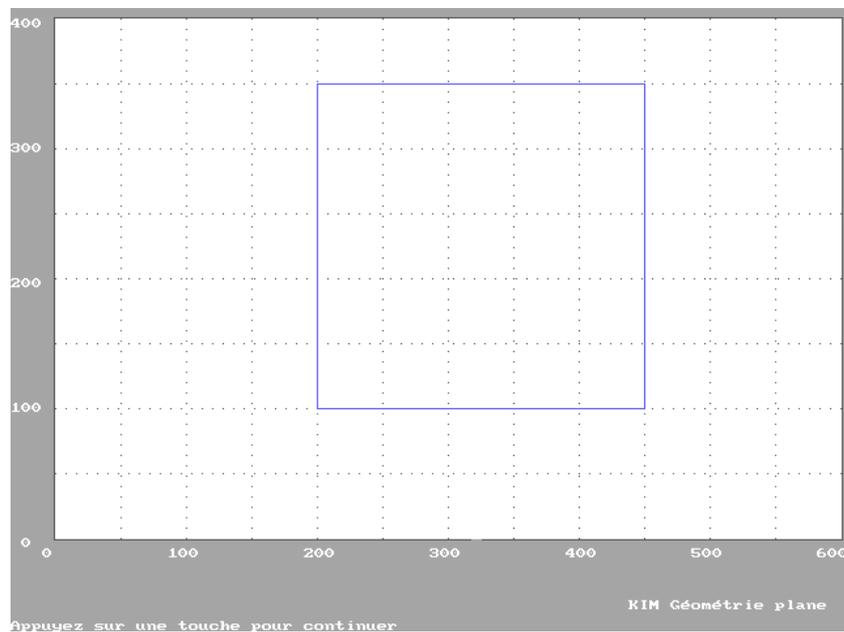


Le carré

Programmation libre²⁵

Après avoir chargé GEOPLAN 1, modifier la procédure Dessin de manière à obtenir le carré représenté ci-dessous en respectant les deux contraintes suivantes :

1. Le sommet supérieur gauche du carré est le point A.
2. La programmation du carré doit commencer par DessineSegment A,...



Pour réaliser le carré, l'une des solutions possibles est :

```
SUB Dessin
' Déclarations
```

²⁵ Enfin, presque libre...

```

DIM A AS Pt 'A est un Pt
DIM B AS Pt 'B est un Pt
DIM C AS Pt 'C est un Pt
DIM D AS Pt 'D est un Pt

' Définir les coordonnées des points A, B, C, D et E.

A.X = 200: A.Y = 350
B.X = 450: B.Y = 350
C.X = 450: C.Y = 100
D.X = 200: D.Y = 100

' Actions
Grille 50
Stylo Bleu
DessineSegment A, B
DessineSegment B, C
DessineSegment C, D
DessineSegment D, A

END SUB

```

Figure paramétrée

Il est bien connu qu'un carré a quatre côtés égaux. Nous pouvons profiter de cette propriété pour définir les coordonnées de trois sommets du carré, par exemple B, C et D, à partir d'un quatrième A en utilisant une variable qui représente la longueur d'un côté. La procédure Dessin peut être transformée comme suit :

```

SUB Dessin
' Déclarations
DIM A AS Pt 'A est un Pt
DIM B AS Pt 'B est un Pt
DIM C AS Pt 'C est un Pt
DIM D AS Pt 'D est un Pt

DIM Cote AS INTEGER

' Corps de la procédure
' Fixer la valeur de Cote
Cote = 250
' Coordonnées des points A, B,
C, D et E.

```

```

' Exercice de mise en équations !
A.X = 200: A.Y = 350
' Seule la position de A est fixée
B.X = A.X + Cote: B.Y = A.Y
' B et A ont la même ordonnée
C.X = B.X: C.Y = A.Y - Cote
' C et B ont la même abscisse
D.X = A.X: D.Y = C.Y
' D et A ont la même abscisse

' Actions
Grille 50
Stylo Bleu
DessineSegment A, B
DessineSegment B, C
DessineSegment C, D
DessineSegment D, A

END SUB

```

suite ↗

Cette transformation de la procédure Dessin est un exercice de mise en équation et nécessite un accompagnement approprié de la part de l'enseignant.

Son intérêt réside dans le fait que l'on peut modifier facilement la taille du carré en changeant la valeur attribuée à la variable Cote. En outre, la position du carré à l'intérieur de la fenêtre graphique peut évoluer en fonction des coordonnées attribuées au sommet supérieur gauche (le point A).

La procédure DessineCarre

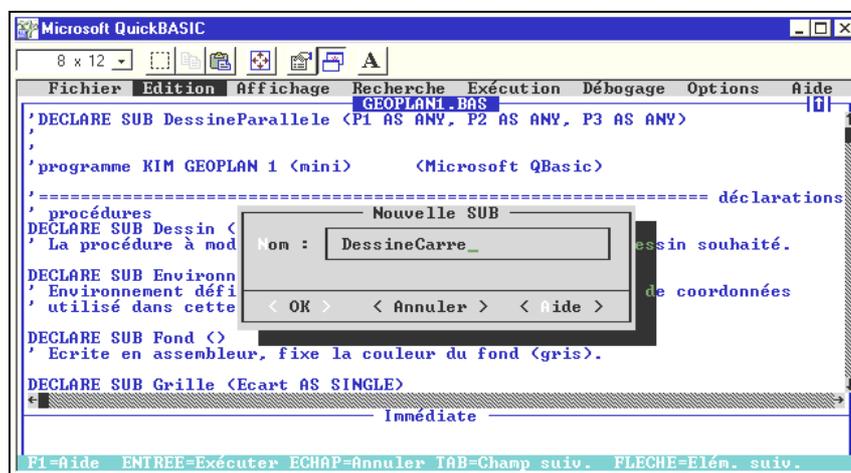
Création d'une nouvelle procédure

Le code de la procédure précédente va être utilisé en grande partie dans la création d'une nouvelle procédure DessineCarre indépendante de la procédure Dessin. Tout comme la procédure DessineSegment, la procédure DessineCarre pourra être utilisée pour réaliser de nouveaux dessins.

Pour créer la nouvelle procédure DessineCarre :

Choisir les options Edition, puis Nouvelle SUB... du menu supérieur.

Compléter le cadre Nom par DessineCarre puis cliquer sur <OK>



Dans la nouvelle fenêtre qui apparaît, compléter le code de la procédure de manière à obtenir :

```
SUB DessineCarre (A AS Pt, Cote AS
INTEGER)
' Déclarations
    DIM B AS Pt 'B est un Pt
    DIM C AS Pt 'C est un Pt
    DIM D AS Pt 'D est un Pt
' Définir les coordonnées des
points B, C et D
    B.X = A.X + Cote: B.Y = A.Y
' B et A ont la même ordonnée
    C.X = B.X: C.Y = A.Y - Cote
' C et B ont la même abscisse
```

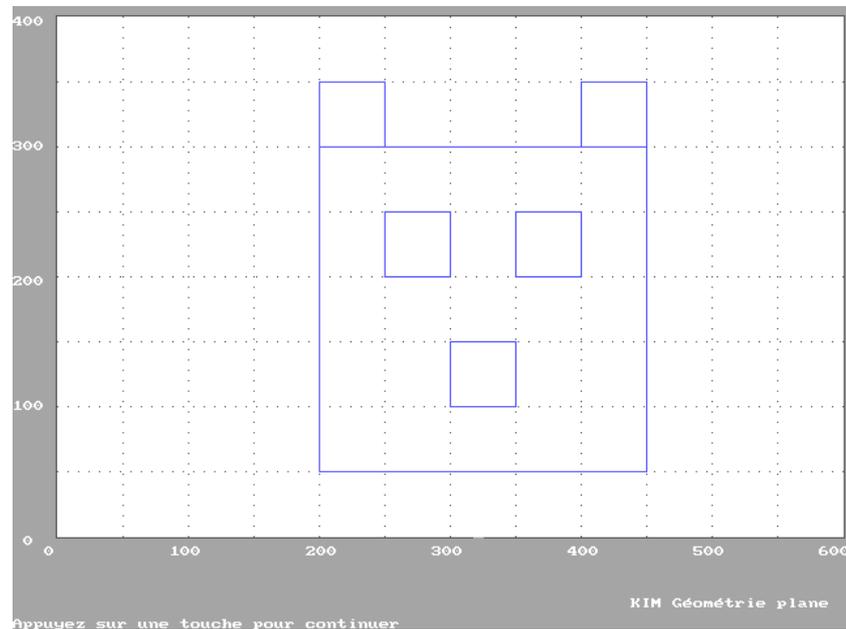
```
    D.X = A.X: D.Y = C.Y
' D et A ont la même abscisse
' Actions
    DessineSegment A, B
    DessineSegment B, C
    DessineSegment C, D
    DessineSegment D, A
END SUB
```

suite ↗

Les virtuoses du « copier-coller » pourront, avec profit, réutiliser le code de la procédure Dessin et l'adapter pour obtenir le même résultat.

La tête stylisée

Utiliser la nouvelle procédure DessineCarre pour construire le dessin suivant.



Le listing suivant donne, en partie, la solution.

```

SUB Dessin

' Déclarations

  DIM A AS Pt 'A est un Pt
  DIM B AS Pt 'B est un Pt
  DIM C AS Pt 'C est un Pt
  DIM D AS Pt 'D est un Pt
  DIM E AS Pt 'E est un Pt
  DIM F AS Pt 'F est un Pt

' Coordonnées des points
A,B,C,D,E et F
  A.X = 200: A.Y = 300
  B.X = 250: B.Y = 250
  C.X = 350: C.Y = B.Y
  D.X =   : D.Y =
  E.X =   : E.Y =
  F.X =   : F.Y =

' Actions

  Grille 50
  Stylo Bleu

' Tête
  DessineCarre A, 250

' Yeux
  DessineCarre B, 50
  DessineCarre C,

' Museau
  DessineCarre D,

' Oreilles
  DessineCarre E,
  DessineCarre F,

END SUB

```

suite ↗

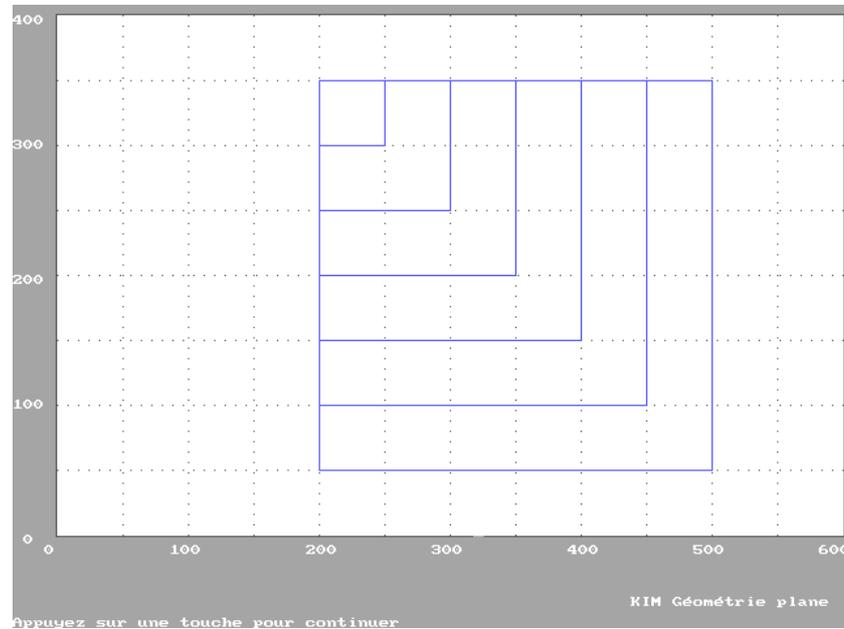
Il est tout à fait possible de changer les couleurs ou d'ajouter d'autres éléments (exemple : poils et moustaches).

Les carrés emboîtés

Recherche personnelle

Programmer le dessin suivant

Aide : Utiliser la procédure DessineCarre



Utilisation de l'itération

Il est possible d'éviter la répétition fastidieuse de la procédure DessineCarre en utilisant la forme itérative :

```

SUB Dessin
' Déclarations
  DIM A AS Pt 'A est un Pt
  DIM Cote AS INTEGER

' Coordonnées du point A
A.X = 200: A.Y = 350

' Actions
Grille 50
Stylo Bleu

FOR Cote = 50 TO 300 STEP 50
  DessineCarre A, Cote
NEXT Cote
END SUB

```

La variable **Cote** varie de la valeur 50 à la valeur 300...

...de 50 en 50.

Cote prend donc les valeurs successives : 50, 100, 150, 200, 250 et 300.

Conclusion

L'expérimentation de KIM, par l'application du scénario pédagogique précédemment exposé, s'est faite avec un groupe d'élèves de quatrième en fin d'année scolaire pendant quatre séances d'une heure. La brièveté de cette expérimentation, ne me permet pas d'apporter des conclusions étayées quant à ses apports. J'ai, cependant observé, plus que dans d'autres activités informatiques, le plaisir pris par les élèves à pratiquer la programmation, surtout quand les résultats obtenus à l'écran ne correspondent pas aux objectifs initiaux du programmeur en herbe. Pouvoir se tromper sans crainte, puis savoir corriger ses erreurs, avec ou sans aide, font partie de l'intérêt que beaucoup portent à la programmation.

La reconstruction des procédures définies dans KIM GEOPLAN 3 me paraît être un objectif raisonnable pour des élèves de collège. Elle met en jeu - au sens propre - des connaissances en mathématique et en informatique et nécessite des transferts cognitifs réciproques entre ces deux matières.

A ceux qui désireraient poursuivre ma démarche et enrichir KIM Géométrie plane de composants logiciels, je proposerais l'élaboration de procédures donnant la distance entre deux points, la position du point d'intersection de deux droites ou construisant une demi-droite, un secteur angulaire, des polygones de toute nature et, plus difficile, la bissectrice d'un angle.

Je propose un autre axe de développement plus ambitieux. Il existe des logiciels de dessin géométrique permettant leur réalisation par l'usage de la souris et de barres d'outils²⁶. L'utilisateur doit se contenter des fonctionnalités proposées car, n'étant pas associés à un langage de programmation procédural, ils ne permettent pas le développement de nouveaux composants.

Dans cette optique, KIM peut servir de base de réflexion à la réalisation d'un logiciel qui ferait appel aux techniques de programmation les plus récentes dans le domaine de la programmation objet et visuelle et qui permettrait, à la fois, une certaine aisance dans la réalisation du dessin et des possibilités d'évolution au gré des besoins de l'utilisateur.

²⁶ En particulier, l'Atelier de Géométrie (Edusoft)