Chapitre 1

Applications didactiques

Ce chapitre est une aide pour étudier Dr. Geo à partir d'exemples. Contrairement aux chapitres précédents, l'approche est plus concrète par rapport à des situations précises. Le contenu de ce chapitre s'est constitué à partir de contributions diverses.

1.1 Aire et Périmètre

Une des possibilités d'utilisation didactique de Dr. Geo consiste dans l'utilisation des scripts Smalltalk pour résoudre des exercices de géométrie.

Comme exemple, nous allons montrer la solution d'un problème classique mettant en oeuvre le théorème de Pythagore dont le texte est le suivant :

> *Soit un trapèze rectangle ABCD où sont connues les bases et la hauteur. Calculer le périmètre et l'aire du trapèze.*

Solution :

Commençons par construire la figure dans Dr. Geo qui doit être comme ci-dessous :

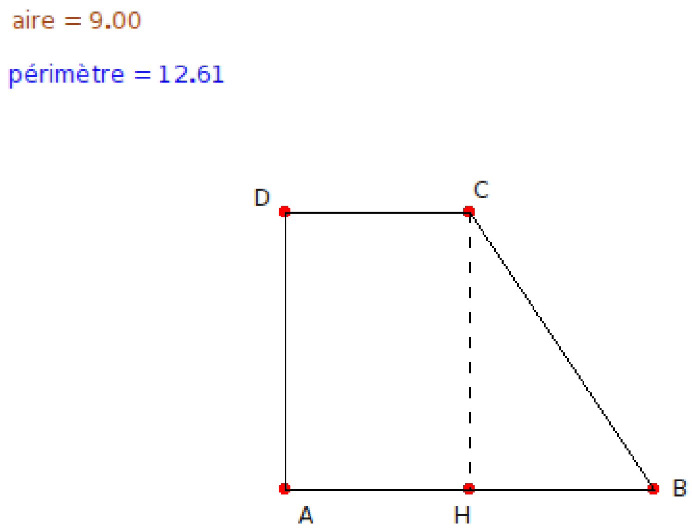

Figure 1.1 – Trapèze rectangle

La figure comprend les données à partir desquelles nous pouvons résoudre le problème. D'abord nous pouvons répondre à la première question de l'aire, pour cela nous pouvons écrire le script Smalltalk suivant ayant comme entrée les deux bases et la hauteur du trapèze :

```
aireTrapezeBase1: b1 base2: b2 height: h
"Calcule l'aire d'un trapèze rectangle connaissant
ses deux segments de base et une hauteur"
   ^ h length * (b1 length + b2 length) / 2
```

Pour calculer le périmètre, nous écrivons un script dans lequel nous calculons la longueur BC à l'aide du théorème de Pythagore :

```
perimetreTrapezeBase1: b1 base2: b2 hauteur: h
"Calculer l'aire d'un trapèze rectangle connaissant
ses deux bases et une hauteur"
| hb bc |
   hb := (b2 length - b1 length) abs.
   bc := (hb squared + h length squared) sqrt.
   ^ b1 length + b2 length + h length + bc
```

Il n'est pas difficile, si vous suivez le même modèle, de développer d'autres exemples similaires.

1.2 Théorème et conjectures

Les scripts Smalltalk permettent de résoudre des exercices mais aussi de comprendre de façon plus approfondie l'énonciation des théorèmes et de vérifier des conjectures. Dans cette section nous commençons par analyser le théorème de Tolomeo :

> *Étant donné un quadrilatère inscrit dans un cercle la somme du produit des côtés opposés est égale au produit des diagonales.*

Nous pouvons construire la figure avec Dr. Geo comme ci-dessous où nous avons implémenté deux scripts qui calculent respectivement la somme du produit des côtés opposés et le produit des diagonales.

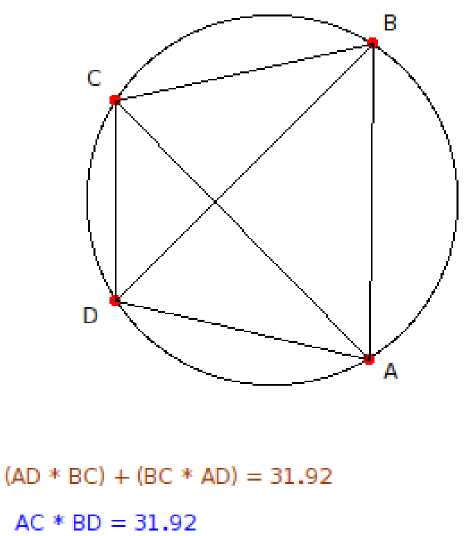

Figure 1.2 – Théorème de Tolomeo : quadrilatère convexe

Le premier script est le suivant :

```
tolomeoSommeS1: ab s2: bc s3: cd s4: ad
"Choisir quatre côtés consécutifs du quadrilatère"
^ (ad length * bc length) + (ab length * cd length)
```

Le second script est :

```
tolomeoProduitD1: ac d2: bd
"Choisir les deux diagonale du quadrilatère"
^ ac length * bd length
```

Comme nous pouvons le voir les valeurs retournées par les deux scripts, en accord avec le théorème de Tolomeo, sont les mêmes [1]. Lorsque nous modifions dynamiquement la figure, les valeurs des scripts sont toujours identiques, sauf dans la situation suivante où le quadrilatère perd sa convexité :

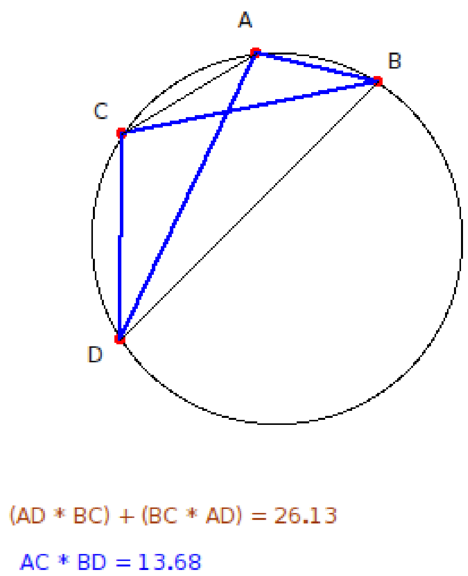

FIGURE 1.3 – Théorème de Tolomeo : quadrilatère non convexe

Dans ce cas le théorème n'est pas vrai et l'énoncé précédent n'est pas bien énoncé, il doit donc être reformulé comme ci-dessous :

> *Étant donné un quadrilatère CONVEXE inscrit dans un cercle la somme du produit des côtés opposés est égale au produit des diagonales.*

À ce stade les conjectures apparaîssent naturellement : est-ce que le théorème de Tolomeo est valide pour un quadrilatère convexe **non inscrit dans un cercle** ?

Avec DR. GEO nous pouvons vérifier que cette conjecture est fausse comme le montre la figure suivante, où il nous a suffit de détacher du cercle le point B en l'attrapant avec la touche $\boxed{\text{SHIFT}}$ enfoncée.

Le lecteur n'aura pas de difficulté à utiliser DR. GEO dans la construction d'exemples didactiques, probablement plus connus, relativement aux théorèmes de Pythagore et d'Euclide.

1. Il ne s'agit que d'une vérification numérique.

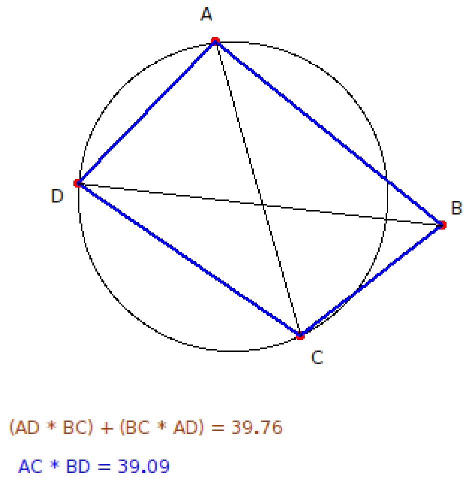

FIGURE 1.4 – Réfutation de la conjecture

1.3 Nombre irrationnel

Une construction classique, relative au nombre irrationnel, connue sous le nom de spirale de Teodoro, permet de construire géométriquement la racine carrée de nombres entiers à partir d'un triangle rectangle isocèle.

Considérons le triangle OAB où $OA = 1$:

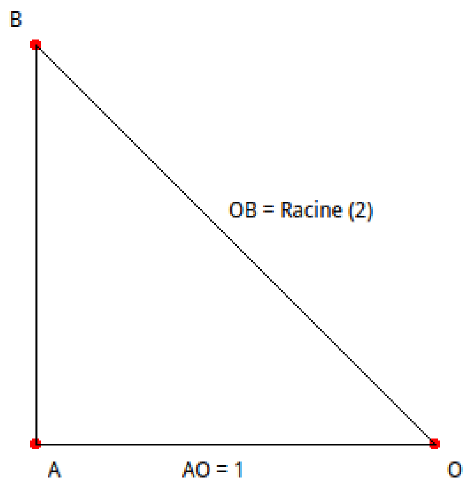

FIGURE 1.5 – Construction de la racine de 2

Par le théorème de Pythagore nous avons OB égale à la racine carrée de 2. Si maintenant, avec la figure, nous construisons un nouveau triangle rectangle en B, avec les côtés OB et BC tel que $BC = 1$.

Toujours par le théorème de Pythagore, il est clair que l'hypothénuse OC de OBC a pour longueur la racine carrée de 3. En itérant le processus précédent à l'infini nous obtenons toutes les racines carrées des nombres naturels.

La nature itérative de la construction s'adapte parfaitement à l'utilisation des figures Smalltalk.

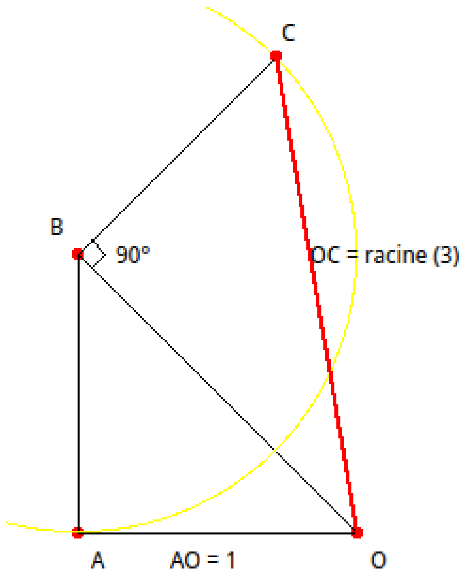

FIGURE 1.6 – Construction de la racine 3

Considérons alors le code suivant :

```
| figure triangle |
figure := DrGeoCanvas new fullscreen.
triangle := [:p1 :p2 :p3  :n | |s1 s2 s3 perp cercle p4 |
   s1 := figure segment: p1 to: p2.
   s2 := (figure segment: p2 to: p3) color: Color red.
   s3 := figure segment: p3 to: p1.
   perp := (figure perpendicular: s3 at: p3) hide.
   cercle := (figure circleCenter: p3 to: p2) hide.
   p4 := (figure altIntersectionOf: cercle and:  perp) hide.
   n > 0 ifTrue: [triangle value: p1 value: p3 value: p4 value: n -1]].

triangle
   value: 0@0
   value: -1@0
   value: -1@1
   value: 50
```

Le triangle du début est défini à travers les coordonnées seulement par commodité. Le code est la transcription littérale de la procédure itérative décrite précédement. Une fois exécuté par DR. GEO le code donne la figure ci-dessous.

Les hypothénuses de chaque triangle ont pour longueur les racines carrées des nombres entiers naturels compris entre 2 et 52.

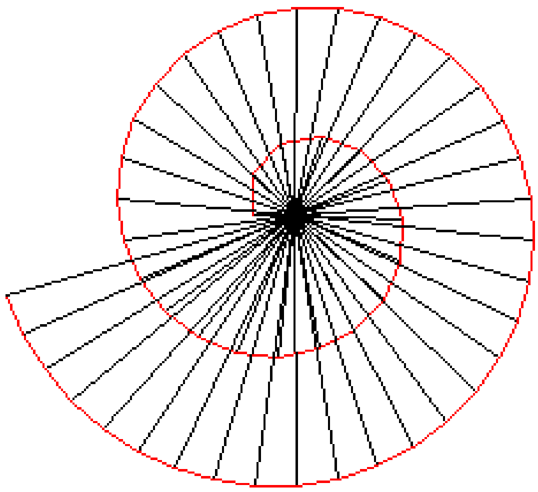

FIGURE 1.7 – Spirale de Teodoro

La même spirale avec les éléments intermédiaires de construction montre combien il serait difficile de construire *à la main* une telle figure :

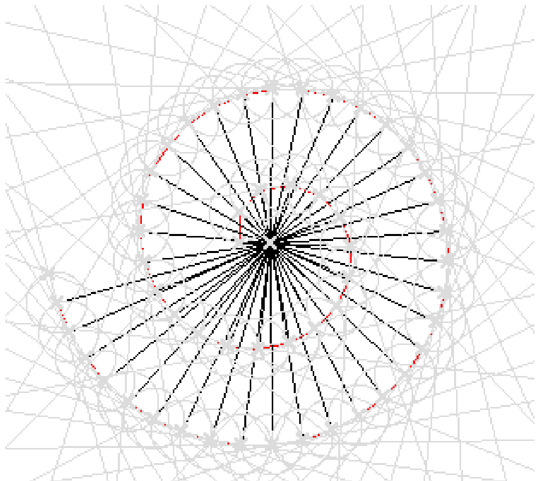

FIGURE 1.8 – Spirale de Teodoro avec les éléments cachés révélés

Comme bonus, nous vous proposons ci-dessous une version animée de la construction de la spirale. Pour cela nous envoyons au canevas les messages `#do:` et `#update`. Ils sont documentés dans le chapitre **??**, p. **??**. Noter l'utilisation de la classe `Delay` pour ralentir la construction :

```
| figure triangle delay|
figure := DrGeoCanvas new fullscreen.
triangle := [:p1 :p2 :p3  :n | |s1 s2 s3 perp cercle p4 |
   s1 := figure segment: p1 to: p2.
   s2 := (figure segment: p2 to: p3) color: Color red.
   s3 := figure segment: p3 to: p1.
   perp := figure perpendicular: s3 at: p3.
   cercle := figure circleCenter: p3 to: p2.
   p4 := figure altIntersectionOf: cercle and:  perp.
```

```
   figure update.
   (Delay forMilliseconds: 200) wait.
   perp hide. cercle hide. p4 hide.
   n > 0 ifTrue: [triangle value: p1 value: p3 value: p4 value: n -1]].

figure do: [triangle value: 0@0 value: -1@0 value: -1@1 value: 50]
```

1.4 Spirale de Baravelle

Comme nous l'avons vu précédement, à l'aide de figure Smalltalk il est possible de construire de façon intuitive et simple des figures permettant de *visionner* des situations qui en programmation sont récursives – ou cycliques.

Nous pouvons un peu approfondir cet aspect, en modifiant le code Smalltalk utilisé pour la construction des nombres irrationnels, afin d'obtenir une figure fameuse de la littérature des mathématiques, à savoir la spirale de Baravelle.

Le code définissant la spirale est le suivant :

```
| figure triangle |
figure := DrGeoCanvas new fullscreen.
triangle := [:p1 :p2 :p3  :n | |s1 s2 s3 m  perp cercle p4 |
   s1 := figure segment: p1 to: p2.
   s2 := figure segment: p2 to: p3.
   s3 := figure segment: p3 to: p1.
   m := (figure middleOf: p1 and: p3) hide.
   perp := (figure perpendicular: s3 at: p3) hide.
   cercle := (figure circleCenter: p3 radius: (figure distance: m to:  p3) hide) hide.
   p4 := (figure altIntersectionOf: cercle and:  perp) hide.
   n > 0 ifTrue: [triangle value: m value: p3 value: p4 value: n -1]].

triangle
   value: (figure point: 0@5)
   value: (figure point: 5@5)
   value: (figure point: 5@0)
   value: 9.
triangle
   value: (figure point: 0@ -5)
   value: (figure point: -5@ -5)
   value: (figure point: -5@0)
   value: 9
```

À partir de la figure et du code Smalltalk correspondant nous percevons bien la nature itérative du mécanisme de construction de la figure. Un problème intéressant que nous laissons au lecteur, consiste à établir à quel moment les deux rameaux de la spirale convergent.

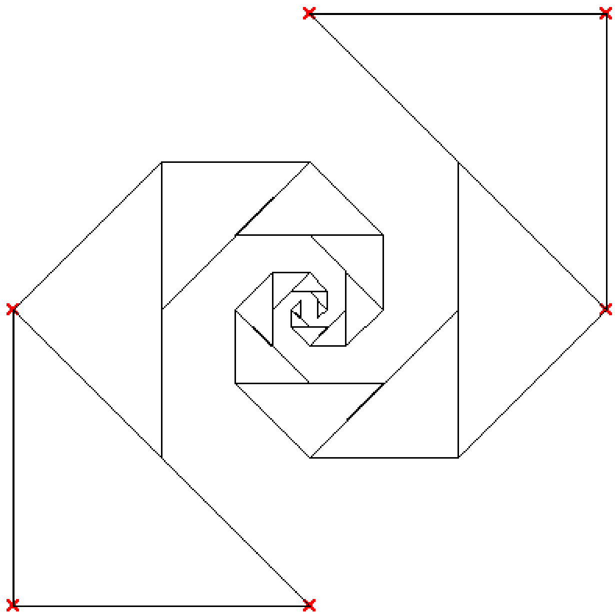

FIGURE 1.9 – La spirale de Baravelle suite à l'exécution du code Smalltalk

1.5 Catena di Pappo

Une utilisation de base de Figure Smalltalk de DR. GEO consiste en la reproduction de figure dont nous connaissons les caractéristiques analytiques.

L'exemple de construction que nous proposons est représenté par la fameuse "Catena di Pappo".

Les centres et rayons successifs des cercles qui la constituent ont une expression analytique connue, il est donc aisé de reproduire la figure par programmation.

```
| figure circle a o m|
figure := DrGeoCanvas new fullscreen.
circle := [:n | |r c p |
   r := (figure freeValue: 15 / (n squared + 6)) hide.
   c := figure point: (15 / (n squared + 6) * 5) @ (15 / (n squared + 6) * n * 2).
   c small; round.
   p := figure circleCenter: c radius: r.
   n > 0 ifTrue: [circle value: n - 1]].

circle value: 10 .
a := (figure point: 5@0) name: 'A'.
o := (figure point: 0@0) name: 'O'.
m := figure
   middleOf: o
   and: ((figure point: 15@0) name: 'B').
m name: 'M'.
figure
   circleCenter: m to: o;
   circleCenter: a to: o;
   line: a to: o.
```

le code de la figure est relativement intuitif et ne nécessite pas de commentaire.

Un exercice non trivial, que nous laissons au lecteur, consiste à déterminer une construction à la règle et au compas conduisant à une implémentation itérative.

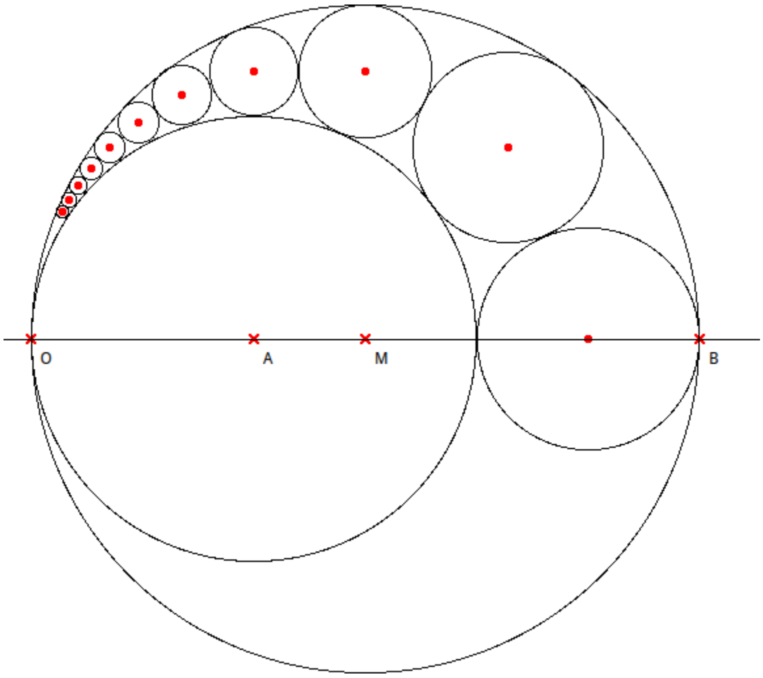

FIGURE 1.10 – Catena di Pappo

1.6 Calcul de π

Le calcul approximatif de π a joué un rôle important dans l'histoire des Mathématiques. Les méthodes pour ce type de calcul sont diverses et contiennent souvent des améliorations d'une méthode à l'autre. Nous vous proposons une approche du problème très simplifiée que nous appelons – même si ce n'est pas tout à fait exacte – **Méthode par exhaustion**. Cette approche a toutefois l'avantage de montrer l'essence même de la méthodologie.

Nous commençons avec la construction d'un hexagone régulier inscrit dans un cercle.

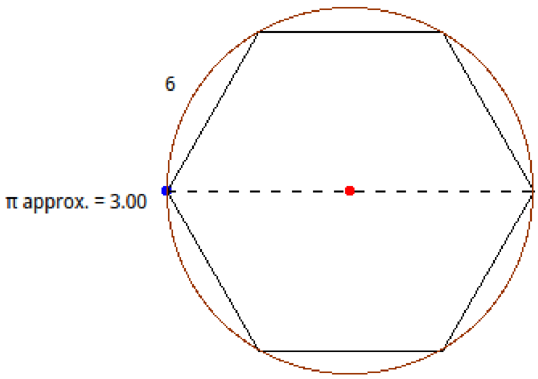

FIGURE 1.11 – Hexagone régulier inscrit

L'idée de la méthode par exhaustion consiste dans un premier temps à approximer la longueur du cercle avec le périmètre P_0 de l'hexagone et de calculer une approximation de

π quotient de P_0 par le diamètre du cercle. Clairement l'approximation de π obtenue sera de 3.

Lors d'une deuxième étape nous pouvons, en utilisant Dr. Geo, construire un côté du dodécagone inscrit dans le cercle. Nous calculons son périmètre P_1 ainsi qu'une approximation successive de π comme quotient de P_1 par le diamètre.

Un petit script est écrit pour calculer ce quotient à partir du polygone et d'un diamètre du cercle :

```
approxPIpolygon: poly diameter: d
"Approximation de PI à partir d'un polygone régulier dans un cercle.
Sélectionner: le polygone et un diamètre du cercle"
   ^ poly length / d length
```

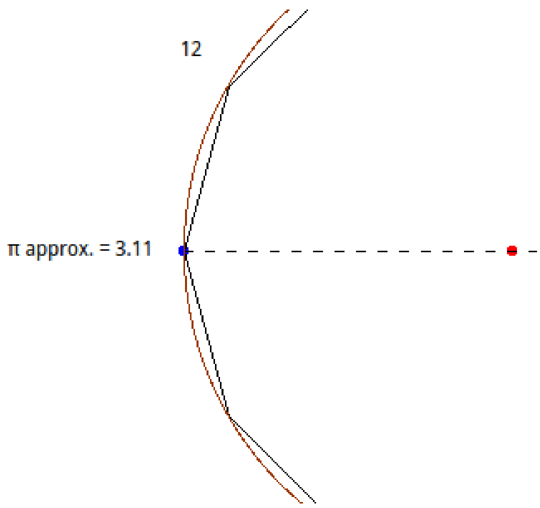

Figure 1.12 – Approximation de π

En augmentant le nombre de côtés du polygone régulier inscrit nous obtenons de meilleures approximations.