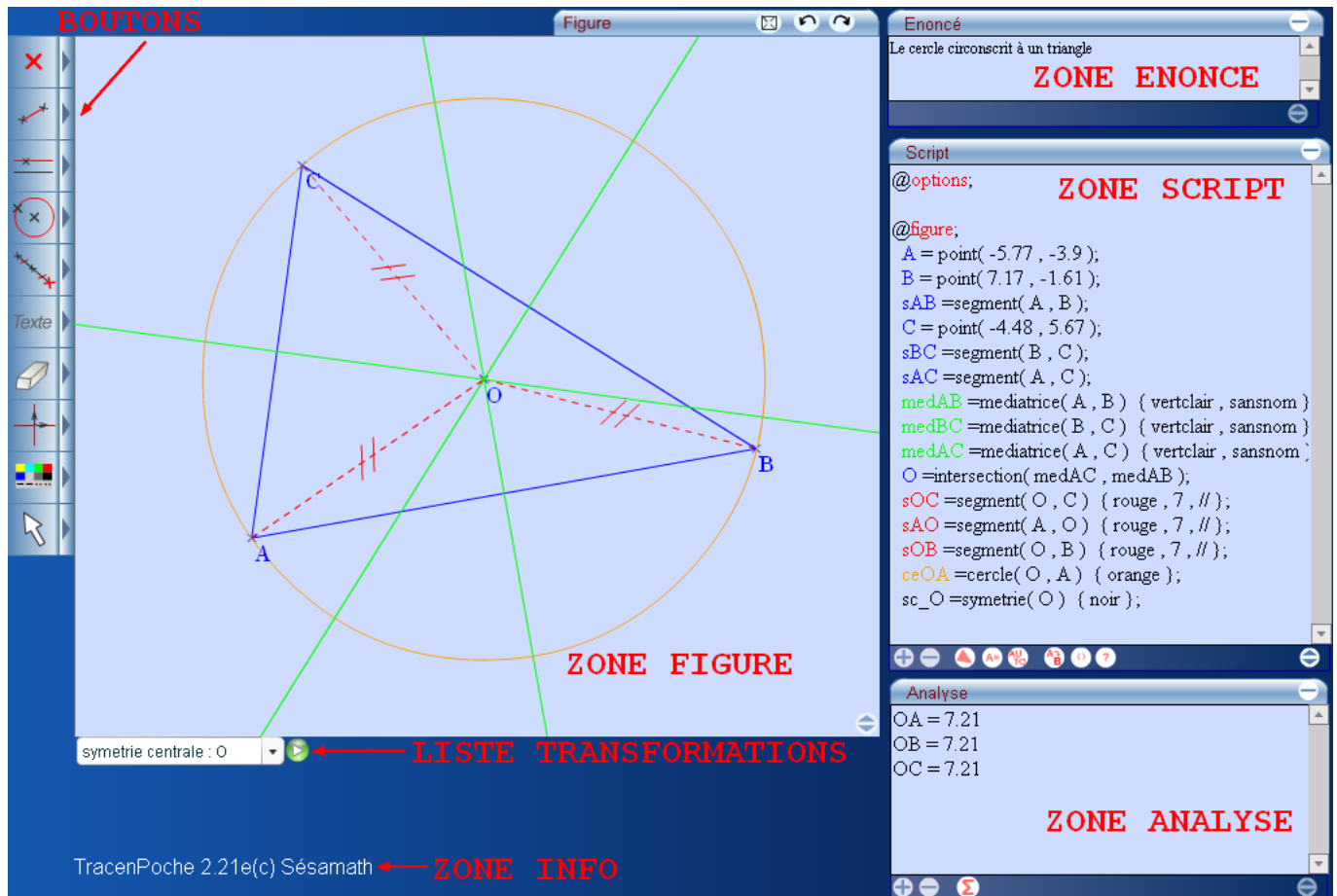


VOS PREMIERS PAS AVEC TRACENPOCHE

Un coup d'oeil sur l'interface de TracenPoche :



La **zone de travail** comporte un **script**, une **figure**, un **énoncé**, une **zone d'analyse**, une **boîte liste** permettant d'utiliser des transformations, un **texte d'informations** et enfin des **boutons**.

La **zone info** affiche un message qui varie selon l'action en cours. Cette zone constitue une aide appréciable quand on découvre le logiciel.

La zone **énoncé** n'est pas modifiable. Elle est remplie au moment du chargement du fichier.

La **zone script** contient le script de la figure.

Un script est divisé en deux parties :

```
@options;
@figure;
```

Ces mots clefs sont obligatoires (avec le caractère @).

Le paragraphe **@options** permet de préciser certaines options qui concernent le repérage : nature et format du repère, trame et aimantage.

Le paragraphe `@figure;` contient la description textuelle de la figure, la syntaxe de base est la suivante :
`Nom_objet = nature_objet(paramètre1, paramètre2,...) { option1, option2,...} ;`

Ce paragraphe est complété au fur et à mesure de la création des objets à l'aide des **boutons**.
 Vous pouvez ensuite en modifier le contenu manuellement afin par exemple d'ajouter des options à un objet pour en modifier son aspect ou son comportement.

L'activité suivante va vous permettre de vous familiariser avec les fonctions de base de TracenPoche.

Activité : Mon premier carré avec TracenPoche

Il existe plusieurs manières pour construire un carré en géométrie dynamique :

- soit à partir de son centre et un de ses sommets puis en utilisant ensuite des rotations,
- soit à partir d'une diagonale puis en utilisant ensuite une médiatrice et un cercle,
- soit à partir d'un de ses côtés puis en utilisant 2 rotations

Nous allons utiliser ici une 4ème méthode qui vous permettra de découvrir un maximum des capacités de TracenPoche.

1ère étape : Construire le côté [AB].


La construction du côté [AB] nécessite la création de 3 objets :

- le point A,
- le point B,
- le segment [AB].

Pour construire le point A nous allons utiliser la souris en cliquant sur le bouton "Créer un point"



Le fond du bouton change alors de couleur , ceci indique qu'une construction est en cours.

Pour annuler une construction en cours il suffit d'appuyer sur le bouton "Escape" .

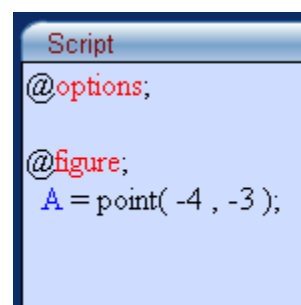
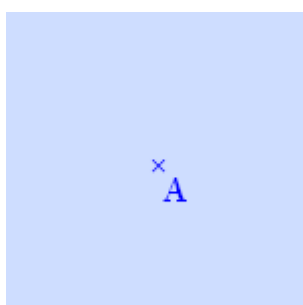
Après avoir appuyé sur le bouton "Créer un point", un message apparaît dans la **zone info** :

Cliquer sur la figure pour créer le point

Après avoir cliqué dans la **zone figure** le programme affiche alors une zone de texte et propose un nom par défaut, ici A.

Comme le nom convient, appuyez sur la touche **ENTREE** (ou cliquer avec la souris sur la zone de dessin). Si le nom ne convenait pas, on aurait pu entrer un autre nom et valider par **ENTREE**.

Dans la **zone figure** apparaît maintenant le point A ainsi que la commande le définissant dans la zone script :



Pour construire un nouveau point dès maintenant, il y aurait 3 possibilités :

- soit on clique à nouveau sur le bouton,
- soit on utilise la combinaison de touches **CTRL + SHIFT** pour réutiliser le même bouton que précédemment,
- soit on le crée directement dans le script.

Mais continuons notre étude du point a : les coordonnées qui s'affichent dans le script sont celles du point A dans un repère qui, par défaut, est orthonormé mais invisible.

Pour le rendre visible, il vous suffit d'appuyer sur le bouton "Afficher le repère"

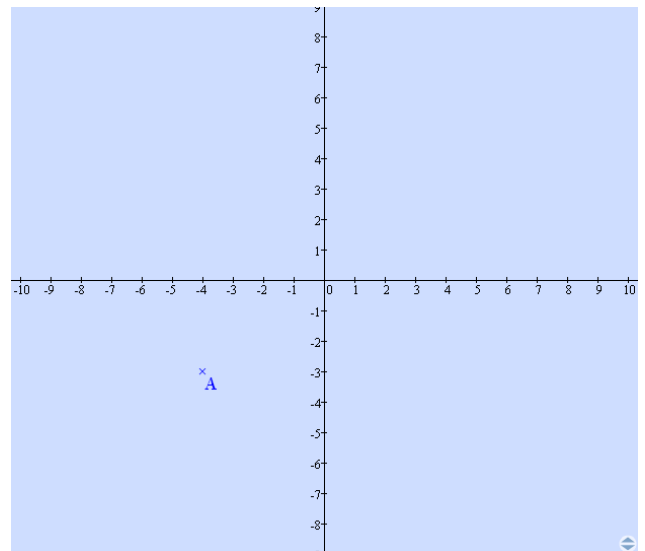


En affichant le repère une première fois le paragraphe `@options;` de la **zone script** est modifié :

```
@options;
  repereortho(310,270,30,1,1){ 0 ,
moyen , noir , num1 };
```

Cette commande définit les caractéristiques du repère que sont : la position de l'origine en pixels, l'unité du repère en pixels et la graduation.

Pour le masquer il suffit d'appuyer à nouveau sur le bouton



La commande définissant le repère ne disparaît pas mais elle est modifiée :

```
@options;
  repereortho(310,270,30,1,1){ 0 , moyen , noir , num1 ,i}
```

Dans les accolades (qui définissent les options d'affichage du repère et non ses caractéristiques) est apparue la lettre "i", i comme invisible.

Vous pouvez modifier la position du point A à l'aide la souris en cliquant dessus avec le bouton gauche de la souris et en le maintenant enfoncé.

Une fois le point A positionné à votre convenance, relâchez le bouton de la souris. Le script est alors **actualisé**, ce qui signifie que ce sont les nouvelles coordonnées du point A qui s'affichent :

```
Script
@options;
repereortho(310,270,30,1,1){
@figure;
A = point(-2.73 , -4.77 );
```


On va maintenant créer le point B.

Pour ceci, on ne va pas utiliser le bouton mais directement le déclarer dans le script.
 Il suffit de taper : `B=point()` ; dans la **zone script** sous la commande qui définit le point A.

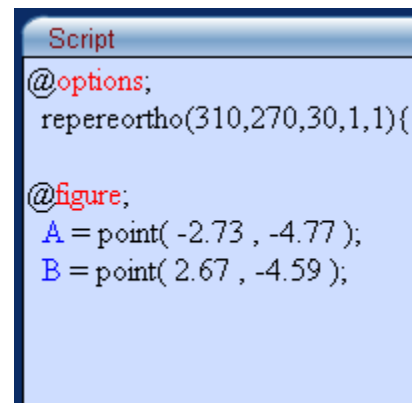
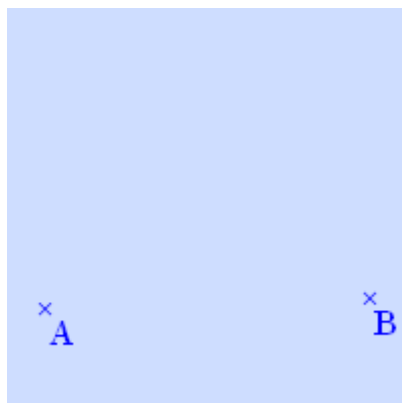
```
@figure;
A = point( -2.73 , -4.77 );
B=point();
```

Il ne faut surtout pas oublier de terminer la ligne par point virgule ";" .
 Ne rien mettre dans les parenthèses en lieu et place des coordonnées indique à TracenPoche de placer B de manière aléatoire.
 Si vous voulez le placer de manière précise, il suffit d'indiquer ses coordonnées séparées par une virgule.


Le réflexe, pour beaucoup, est d'appuyer ensuite sur la touche **ENTREE**. Si vous le faites rien ne se passe si ce n'est que le curseur dans la zone script est passé à la ligne suivante.
 Par contre, toujours aucune trace du point B dans la zone figure.

Pour que la figure soit **actualisée** par rapport au contenu du script, il suffit d'appuyer sur la touche **F9** du clavier ou d'appuyer sur le bouton "Actualiser la figure"  qui se situe en bas de la **zone script**.

Le point B apparaît alors dans la **zone figure** et le script affiche les coordonnées du point B :



Reste à construire maintenant le segment [AB].

On utilise pour cela le bouton  et il reste à suivre les consignes affichées dans la **zone info** :

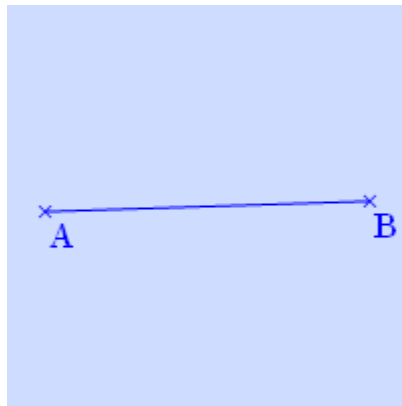
Première extrémité ? Cliquer sur un premier point ou cliquer sur la figure pour le créer

Cliquer sur le point A. La **zone info** devient :

cliquer sur la seconde extrémité du segment

Cliquer sur le point B.

Voici le résultat obtenu dans la **zone figure** et dans la **zone script** :



```
Script
@options;
repereortho(310,270,30,1,1){

@figure;
A = point( -2.73 , -4.77 );
B = point( 2.67 , -4.59 );
sAB =segment( A , B );
```


Il faut noter que l'on aurait pu commencer par créer le segment, les points auraient été créés à l'endroit où on aurait cliqué pour indiquer la première puis la deuxième extrémité.

2ème étape : Construire le sommet C.

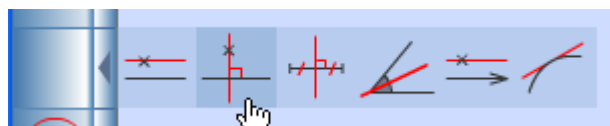
Le sommet C doit vérifier plusieurs conditions :

- les droites (AB) et (BC) doivent être perpendiculaires,
- les longueurs AB et BC doivent être égales.

Commençons par tracer la perpendiculaire au segment [AB] passant par le point B.

Nous allons utiliser le bouton "Droite perpendiculaire"  qui se trouve dans le 3ème menu.

Pour avoir accès à ce bouton, il suffit d'appuyer sur la flèche située à droite du 3ème bouton pour développer le menu :

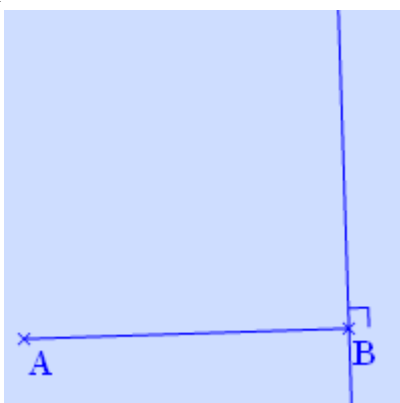


La zone info affiche :

Droite passant par le point ? et perpendiculaire à ? Cliquer sur un point ou sur une direction

Cliquez alors sur le point B puis sur le segment [AB] (Remarque : TracenPoche autorise la saisie inverse pour perpendiculaire et parallèle).



La perpendiculaire est alors tracée et nommée automatiquement :



```
Script
@options;
repereortho(310,270,30,1,1){ 0 , moyen

@figure;
A = point( -2.73 , -4.77 );
B = point( 2.67 , -4.59 );
sAB =segment( A , B );
perpBsAB =perpendiculaire( B , sAB );
```

Pour renommer la droite (si le nom ne nous convient pas) il existe 3 méthodes :

- on peut utiliser le bouton "Renommer un objet"  situé dans le 7ème menu (il permet de renommer les objets en cliquant sur ceux-ci),
- on peut utiliser le bouton "Renommer un objet"  situé en bas de la zone script (une fenêtre s'ouvre demandant l'ancien nom et le nouveau nom de l'objet),
- on peut également modifier le nom de la perpendiculaire directement dans la zone script puis actualiser ensuite la figure.


L'objet "perpBsAB" n'apparaissant qu'une seule fois dans le script, on peut utiliser ici cette dernière méthode.

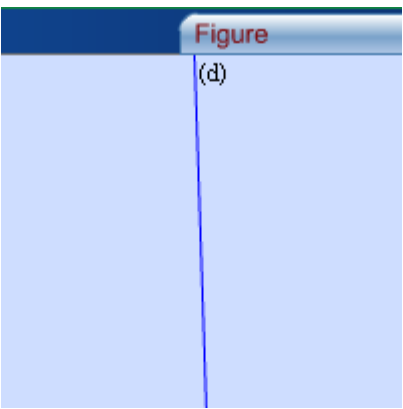
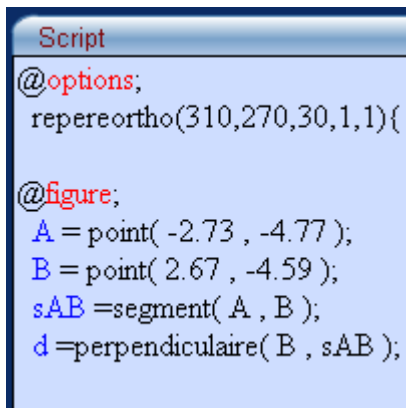
On commence par effacer le nom de la droite directement dans la zone script :

```
@figure;
A = point( -2.73 , -4.77 );
B = point( 2.67 , -4.59 );
sAB =segment( A , B );
=perpendiculaire( B , sAB );
```

On tape ensuite le nouveau nom souhaité, ici "d" :

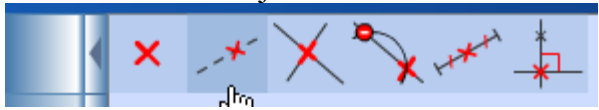
```
@figure;
A = point( -2.73 , -4.77 );
B = point( 2.67 , -4.59 );
sAB =segment( A , B );
d =perpendiculaire( B , sAB );
```

Il suffit ensuite d'actualiser la figure ( ou F9) pour que les modifications soient prises en compte tant dans le script que dans la figure :



Placer le point C sur la droite (d).

Nous allons utiliser le bouton "Point sur un objet" situé dans le 1er menu :

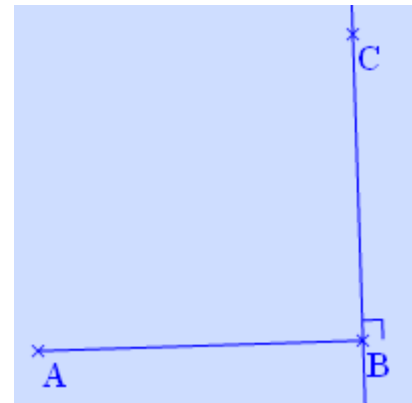
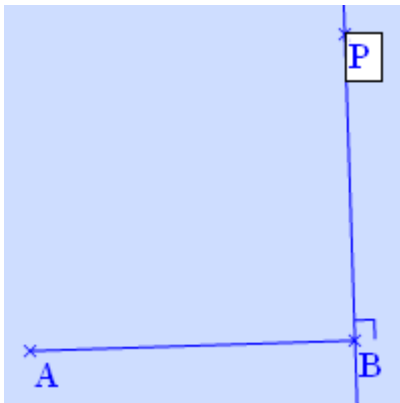


Après avoir cliqué sur ce bouton la zone info affiche :

cliquer sur un objet ou sur l'origine d'un axe

Il suffit de cliquer alors sur la droite (d) et un point apparaît, par défaut nommé P. Il suffit d'effacer

la lettre P puis de la remplacer par C. En appuyant sur la touche **ENTREE** ou en cliquant ailleurs dans la figure la construction sera validée :



Dans le script le point C est maintenant défini :

```
@figure;
A = point( -2.73 , -4.77 );
B = point( 2.67 , -4.59 );
sAB =segment( A , B );
d =perpendiculaire( B , sAB );
C =pointsur( d , 5.09 );
```

Le **5.09** correspond ici à la distance entre les points B et C. Vous n'aurez probablement pas le même nombre car cette mesure dépend de l'endroit où il a été cliqué sur (d) pour créer le point. Ce point est mobile sur la droite (d) : il suffit de le bouger à la souris en maintenant le bouton souris enfoncé.

Comme nous voulons construire un carré, il faut que cette distance soit égale à la longueur du segment [AB].

Pour cela nous allons définir dans le script une variable x égale à la longueur AB.

Nous définissons manuellement cette variable **x** dans le script avant la commande qui définit le point C (ce n'est pas obligatoire mais plus logique pour la suite), puis nous actualisons la figure :

```
@figure;
A = point( -2.73 , -4.77 );
B = point( 2.67 , -4.59 );
sAB =segment( A , B );
d =perpendiculaire( B , sAB );
var x = AB;
C =pointsur( d , 5.09 );
```

```
@figure;
A = point( -2.73 , -4.77 );
B = point( 2.67 , -4.59 );
sAB =segment( A , B );
d =perpendiculaire( B , sAB );
var x =AB { 5.40299916712931 };
C =pointsur( d , 5.09 );
```

Entre les accolades est apparue la valeur actuelle de **x**, c'est à dire la longueur AB.

Pour que le point C soit toujours placé tel que les longueurs AB et BC soient égales, il suffit de remplacer le **5.09** par **x** dans le script et d'actualiser.

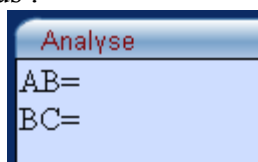
```
@figure;
A = point( -2.73 , -4.77 );
B = point( 2.67 , -4.59 );
sAB =segment( A , B );
d =perpendiculaire( B , sAB );
var x =AB { 5.40299916712931 };
C =pointsur( d , x );
```


Les distances AB et BC sont égales, ce qui signifie que le point C n'est plus libre (essayez de le déplacer à la souris, c'est maintenant impossible) sur la droite (d), mais que sa position dépend des points A et B.

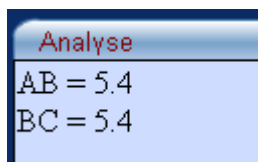
Visualiser cette "dépendance".

Nous allons pour cela utiliser la **zone analyse** située sous la **zone script**.

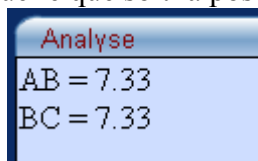
La **zone analyse** va permettre d'évaluer les longueurs AB et BC. Pour cela, il suffit de la compléter comme sur l'image ci-dessous :



Pour que ces 2 longueurs soient évaluées il faut appuyer sur le bouton . Les longueurs seront alors affichées :



Si vous bougez le point B par exemple, les 2 longueurs sont évaluées en temps réel et vous vous rendez compte qu'elles restent égales quelle que soit la position du point B :

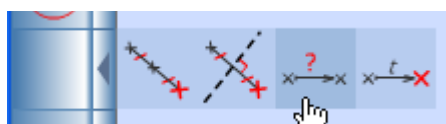


3ème étape : Construire le sommet D.

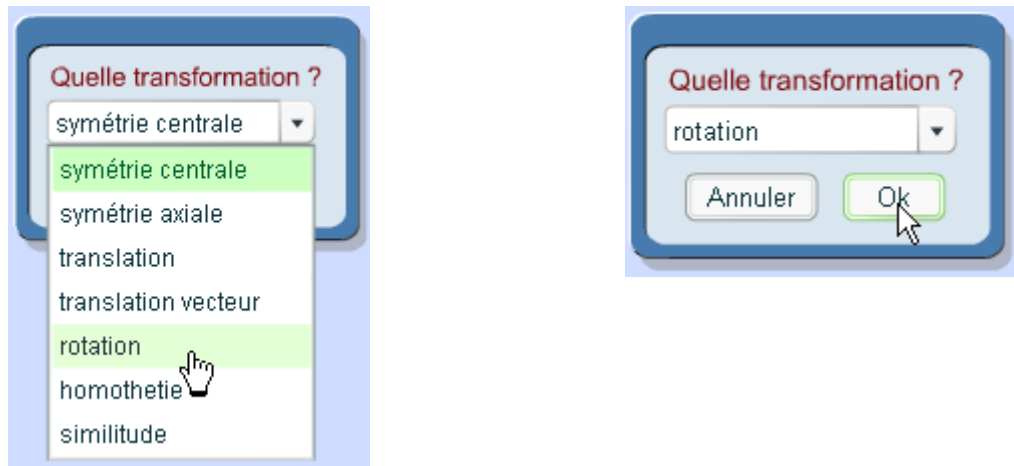
Le but de cette activité étant de vous montrer un maximum des possibilités de TracenPoche, nous allons construire le sommet D en utilisant une rotation : la rotation de centre A et d'angle 90° dans le sens direct.

Avant de pouvoir utiliser cette transformation, il faut la définir dans TracenPoche.

Pour le faire nous allons utiliser le bouton "Déclarer une transformation" qui se situe dans le 5ème menu :



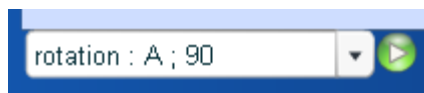
Le fait de cliquer sur ce bouton ouvre une fenêtre qui permet de choisir la nature de la transformation que l'on veut déclarer, ici une rotation :



La **zone info** nous indique la marche à suivre, il faut donc cliquer sur le point A puis renseigner la mesure de l'angle de la rotation, ici 90 (nous aurions pu mettre le nom d'une variable).



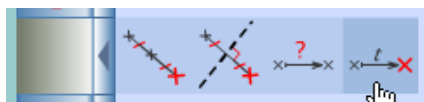
Une fois validée, la rotation de centre A et d'angle 90° dans le sens direct est déclarée, elle apparaît alors dans la **liste des transformations** en dessous de la **zone figure** ainsi que dans le script :



```
C = pointsur( d , x );
r_A = rotation( A , 90 ) { noir };
```

Il reste donc à construire le point D image du point B par cette rotation.

Pour cela il faut utiliser le bouton "Construire une image par une transformation" qui se trouve dans le 5ème menu également :



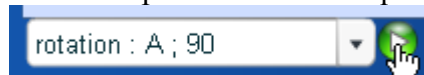
La **zone info** nous guide pour la construction du point D :

Image de ? Cliquer sur un objet à transformer.

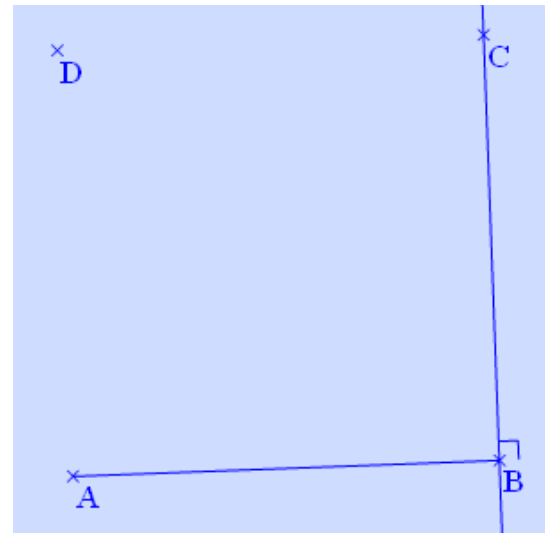
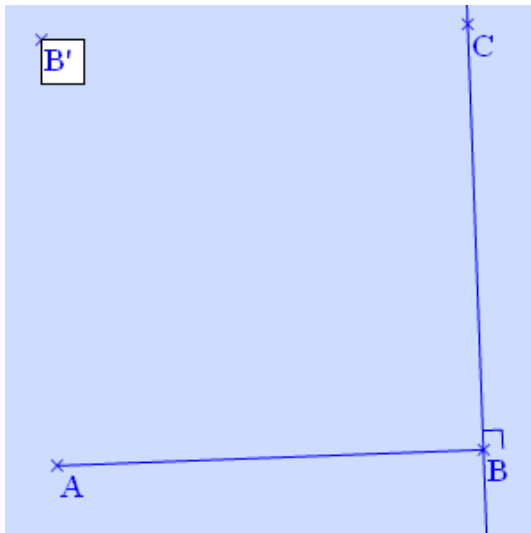
Cliquez sur le point B.

Image de B par ? Choisir une transformation dans la liste puis OK

Une seule transformation étant déclarée on peut valider en cliquant sur le bouton vert :



Par défaut, le point est nommé B', il suffit de le renommer D, puis de valider :



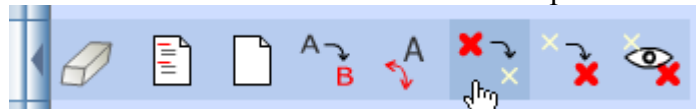
Le point D est maintenant défini dans le script :

```
r_A=rotation( A , 90 ) { noir };
D=image( r_A , B );
```

4ème étape : Construction du carré.

Pour finir la construction du carré, nous allons commencer par masquer la droite (d).

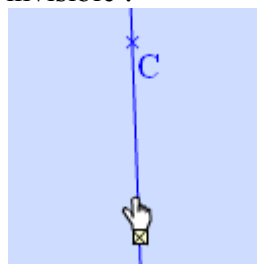
Pour le faire nous allons utiliser le bouton "Rendre invisible" disponible dans le 7ème menu :



La zone info nous indique qu'il suffit de cliquer sur les objets que l'on souhaite rendre invisible :

Cliquer sur les objets à rendre invisible

Cliquez sur la droite (d) pour la rendre invisible :



La droite (d) est maintenant invisible, ce qui se traduit par l'apparition de { i } dans le script comme pour le repère :

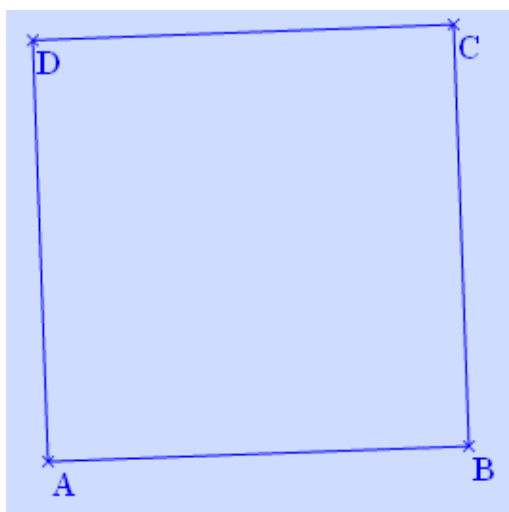


```
sAB =segment( A , B );
d=perpendiculaire( B , sAB ) { i };
var x =AB { 7.33497102925431 };
```

Le bouton "Rendre invisible" est toujours actif (cela se remarque à sa couleur), il faut cliquer à

nouveau dessus pour le désactiver.

Reste à tracer les 3 côtés manquants : [BC], [CD] et [DA] en utilisant le bouton "Segment".



Le script ressemble maintenant à celui-ci :

```
@options;
  repereortho(310,270,30,1,1){ 0 , moyen ,
  noir , num1 ,i};

@figure;
  A = point( -2.73 , -4.77 );
  B = point( 4.6 , -4.5 );
  sAB =segment( A , B );
  d =perpendiculaire( B , sAB ) { i };
  var x =AB { 7.33497102925431 };
  C =pointsur( d , x );
  r_A =rotation( A , 90 ) { noir };
  D =image( r_A , B );
  sBC =segment( B , C );
  sCD =segment( C , D );
  sDA =segment( D , A );
```

Vous venez donc de réaliser votre premier carré avec TracenPoche.

Ne nous arrêtons pas ici, et découvrons d'autres fonctions de TracenPoche.

5ème étape : *Embellissements.*

Pour modifier l'aspect d'un objet on utilise des options qui s'ajoutent avant le point virgule ; en fin de déclaration de construction, entre accolades { }.

Vous connaissez déjà une option pour rendre invisible un objet : { i }.

Certaines options se modifient à la souris, d'autres uniquement en modifiant directement le contenu du script.

A la souris, on peut par exemple modifier la couleur d'un objet, le style ou l'épaisseur d'un trait, l'aspect d'un point, le codage ou les caractères.

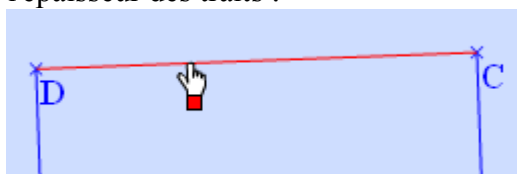
Pour cela il faut cliquer sur le bouton "Couleurs et styles de traits d'un objet" .

Une palette de couleurs apparaît alors :

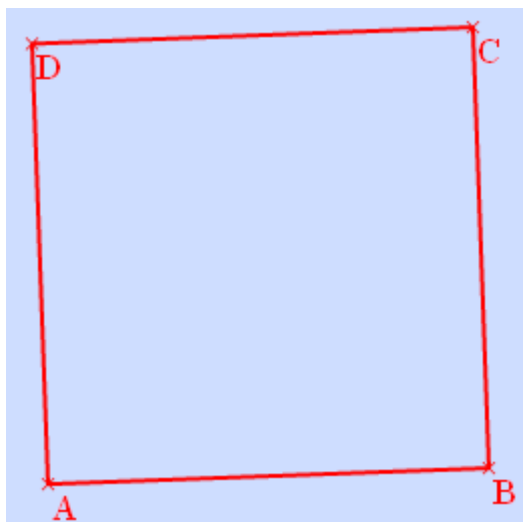


Colorier les objets en rouge.

Il faut pour cela cliquer sur la couleur rouge puis sur chacun des objets que l'on souhaite colorier, de même si on veut modifier l'épaisseur des traits :



Une fois modifiés la couleur des objets ainsi que l'épaisseur des traits, voilà ce que vous devriez obtenir :



```
@figure;
A = point( -2.73 , -4.77 ) { rouge };
B = point( 4.6 , -4.5 ) { rouge };
sAB = segment( A , B ) { rouge , 2 };
d =perpendiculaire( B , sAB ) { i };
var x =AB { 7.33497102925431 };
C =pointsur( d , x ) { rouge };
r_A =rotation( A , 90 ) { noir };
D =image( r_A , B ) { rouge };
sBC =segment( B , C ) { rouge , 2 };
sCD =segment( C , D ) { rouge , 2 };
sDA =segment( D , A ) { rouge , 2 };
```

`{ rouge , 2 }` indique que chacun des segments est rouge et épais (à cause du 2, mais vous pouvez essayer aussi 3, 4 ou 6, 7, 8 et 9).

Pour que le pointeur de la souris reprenne son aspect normal, il suffit de cliquer ailleurs sur la figure.

Vous pouvez continuer à travailler même si la palette est affichée.

On peut également modifier l'aspect des points et coder les segments en modifiant directement le script.

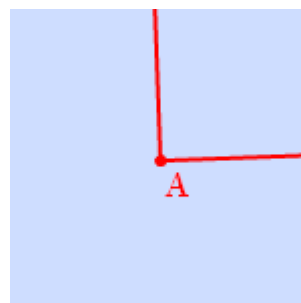
Les points peuvent avoir l'aspect d'une croix, d'un rond ou être masqués.

Modifiez le script comme suit : remplacez { rouge } par { rouge , rond2 } à la fin de la déclaration du point A puis actualisez la figure :

Le script est donc devenu :

Et le point A a changé d'aspect :

```
@figure;
A = point(-2.73 , -4.77) { rouge , rond2 };
B = point(4.6 , -4.5) { rouge };
```

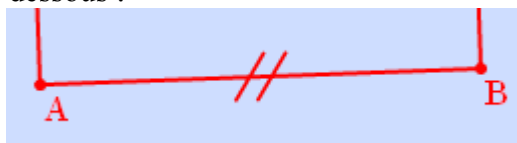


A la place de **rond2** vous pouvez essayer **rond1** ou **croix0** (ce dernier attribut masque la marque du point).

Vous pouvez coder les segments avec différents symboles : /, //, ///, \, \\, \\\, x, o.

Pour cela, il faut rajouter le symbole voulu dans les accolades avec les autres options (séparées d'une ,).

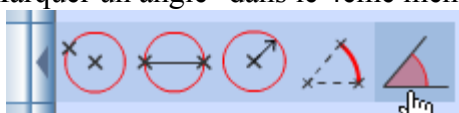
Par exemple, en remplaçant les attributs actuels du segment [AB] par : { rouge , 2 , // }, vous obtiendrez le résultat ci dessous :



Coder les angles.

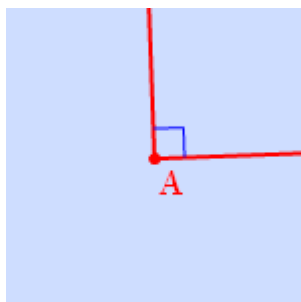
Rien n'est plus simple, chacun des angles étant droit il suffit de les construire pour qu'ils soient codés en tant que des angles droits.

Pour cela, utilisez le bouton "Marquer un angle" dans le 4ème menu :



Pour marquer l'angle de sommet A, c'est à dire \widehat{BAD} il faut cliquer sur B, A et puis D dans cet ordre.

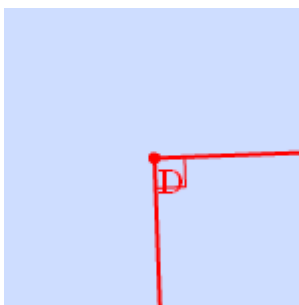
Vous devriez obtenir ce résultat :



Vous pouvez marquer l'angle en rouge en ajoutant { rouge } dans le script.

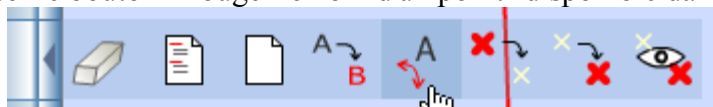
Une fois tous les angles marqués en rouge, vous allez vous rendre compte qu'il reste une chose à améliorer : le point D.

En effet le nom du point se superpose avec le marquage de l'angle, ce qui, il faut avouer, ne fait très "beau".

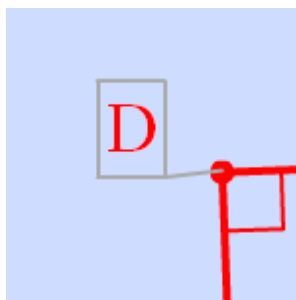


TracenPoche vous donne la possibilité de bouger l'étiquette du nom de point afin de le disposer où on veut par rapport au point lui même.

Pour cela il faut utiliser le bouton "Bouger le nom d'un point" disponible dans le 7ème menu :



En cliquant sur le nom d'un point et en maintenant le bouton de la souris appuyé vous pouvez bouger le nom du point :



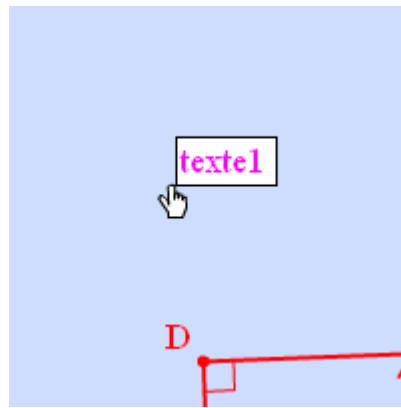
Vous pouvez également bouger les noms des autres points, le bouton restant actif tant que vous n'appuyez pas dessus une 2ème fois.

Le script a été modifié. Dans les attributs de chacun des points, dont vous avez bougé le nom, apparaissent des coordonnées qui correspondent au déplacement du nom que vous avez appliqué :

```
r_A =rotation( A , 90 ) { noir };
D =image( r_A , B ) { rouge , rond2 , (-0.69,-0.82) };
sBC =segment( B , C ) { rouge , 2 , // };
```

Apportons une touche finale à notre figure en lui ajoutant une zone texte dans laquelle nous afficherons la longueur des côtés du carré.

Appuyez sur le bouton "Afficher un texte"  puis cliquez dans la **zone figure** à l'endroit ou vous voulez l'insérer :



Effacez `textel` et tapez le texte suivant :

ABCD est un carré de côté cm

Nous ne pouvons pas indiquer la longueur des côtés du carré car cette longueur est variable.

Nous allons donc afficher la valeur de la variable `x` dans la zone texte.

Pour le faire, il faut modifier le script, l'objet texte est déclaré comme ceci :

```
textel =texte( -3.3 , 5.7 , "ABCD est un carré de côté cm" ) { magenta
, dec2 };
```

Le texte qui s'affiche dans la zone figure est écrit entre guillemets.

Pour afficher la valeur d'une variable il faut l'écrire entre 2 symboles `$`. Ici il faut donc modifier l'objet texte comme ceci :

```
textel =texte( -3.3 , 5.7 , "ABCD est un carré de côté $x$ cm" )
{ magenta , dec2 };
```

Et le texte affiché maintenant ressemble à :

ABCD est un carré de côté 7.33 cm

Si vous ne vous êtes pas trompé, et il n'y a aucune raison pour qu'il en soit ainsi, le contenu de la zone script devrait ressembler à ceci :

```
@options;
```

```
  repereortho(310,270,30,1,1){ 0 , moyen , noir , num1 ,i};
```

```
@figure;
```

```
  A = point( -2.73 , -4.77 ) { rouge , rond2 , (-0.53,0.07) };
```

```
  B = point( 4.6 , -4.5 ) { rouge , rond2 , (-0.03,-0.03) };
```

```
  sAB = segment( A , B ) { rouge };
```

```
  d = perpendiculaire( B , sAB ) { i };
```

```
  var x =AB { 7.33497102925431 };
```

```
  C = pointsur( d , x ) { rouge , rond2 , (0,-0.73) };
```

```
  r_A = rotation( A , 90 ) { noir };
```

```
  D = image( r_A , B ) { rouge , rond2 , (-0.53,-0.8) };
```

```
  sBC = segment( B , C ) { rouge };
```

```
  sCD = segment( C , D ) { rouge };
```

```
  sDA = segment( D , A ) { rouge };
```

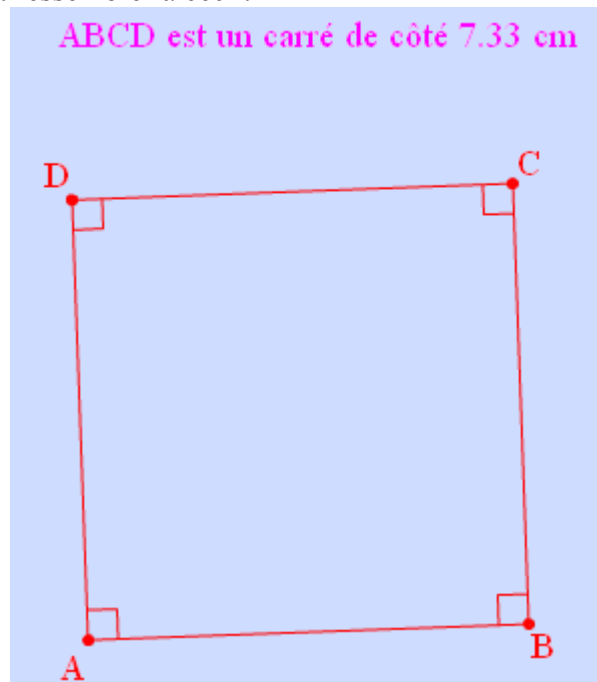
```
  angleBAD = angle( B , A , D ) { rouge };
```

```
  angleABC = angle( A , B , C ) { rouge };
```

```
  angleBCD = angle( B , C , D ) { rouge };
```

```
angleCDA = angle( C , D , A ) { rouge };  
textel = texte( -3.3 , 5.7 , "ABCD est un carré de côté $x$ cm")  
{ magenta , dec2 };
```

Et votre figure devrait ressembler à ceci :



Vous connaissez maintenant quelques unes des fonctionnalités élémentaires de TracenPoche.

Pour découvrir toutes les autres possibilités de TracenPoche, vous pouvez consulter le mode d'emploi et visiter le site www.tracenpoche.net

Bonnes figures et bons scripts !