

Partie 1 : MopsConstruction, le langage

MopsConstruction est un langage de script (inspiré d'instrumentPoche) destiné à créer des animations pour illustrer des constructions géométriques aux instruments (règle, équerre, compas, rapporteur)

Remarque : à l'origine à usage personnel, ce n'est pas destiné à être utilisé par les élèves

Un script MOPSConstruction est un fichier xml de la forme :

```
<MopsConstruction>
  <script>
    ...
    <instruction paramêtres />
    ...
  </script>
</MopsConstruction>
```

Les instructions disponibles sont :

<new /> : créations de nouveaux objets.

<tool /> : gestion des instruments.

<image /> : manipulation d'images.

<wait /> : temporisation.


<step/> : crée un point d'arrêt possible lors de la lecture de l'animation

<style/> : définit le style par défaut

Un script peut-être créé avec un éditeur de texte ou en utilisant « [MopsConstructionEditor](#) »

Le script peut-être interprété soit en utilisant « [MopsConstructionEditor](#) » en ligne sur le site [My Own Private Square](#) ou en utilisant le programme « [MopsConstructionPlayer](#) ».

Remarques :

- Les paramètres sont des chaînes de caractères, exemples : id = "11" vis = "on"
- La dimension de la « zone de travail » est de 1280x720, l'origine du repère est en bas à gauche.
- Avec la règle par défaut, 40 pixels représentent 1 cm
- Pour les unités de temps, 10 unités représentent environ 1 seconde.
- Dans la suite, **p** représente le point de coordonnées (x;y) , **p0** représente le point de coordonnées (x0;y0) , **p1** représente le point de coordonnées (x1;y1) , **p2** représente le point de coordonnées (x2;y2)
- Pour les angles notamment, le sens positif est le sens trigonométrique : 
- Dans la "timeline", un identifiant ne peut être appelé qu'après avoir été créé.
- Les couleurs sont définies en écriture hexadécimale exemple : "#A233BF", la transparence peut être prise en compte en rajouter 2 'chiffres' : "#A233BFC0"

<new />

Instruction permettant la création de nouveaux objets. Le paramètre **type** est obligatoire.

Les types disponibles sont :

- [point](#)
- [segment](#)
- [vecteur](#)
- [droite](#)
- [demidroite](#)
- [milieu](#)
- [compte](#)
- [cercle](#)
- [arc](#)
- [polygone](#)
- [texte](#)
- [code](#)
- [inter](#)
- [interDteCer](#)
- [interCer](#)
- [pointsur](#)
- [sym](#)
- [rot](#)
- [trans](#)
- [ptmed](#)
- [ptbiss](#)
- [perp](#)
- [para](#)
- [media](#)
- [biss](#)
- [pointRap](#)
- [markRap](#)
- [surl_rect](#)

➤ type = "point"

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x, y : coordonnées du point (par défaut x = "640" et y = "360")

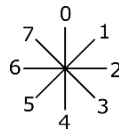
col : couleur (par défaut col = "#000000")

name : nom du point

size : taille de la police de caractère (par défaut size = "24")

ep : épaisseur du trait (par défaut ep = "2")

pos : position du nom du point (par défaut pos = "0")



lg : taille de la croix (par défaut lg = "4")

exemple : `<new type = "point" x="900" y="250" id="1" name="A" col="#ff0000" />`

➤ type = "segment" : segment [p1p2]

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées d'une extrémité ou **pt1id** : identifiant d'une extrémité.

x2 y2 : coordonnées de l'autre extrémité ou **pt2id** : identifiant de l'autre extrémité.

⚠ les identifiants pt1id et/ou pt2id doivent avoir été créés au préalable.

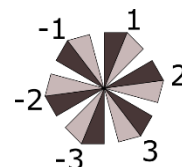
col : couleur (par défaut col = "#000000")

ep : épaisseur du trait (par défaut ep = "2")

dsh : pointillés (par défaut dsh = "0", trait continu)

trace : temps pour le tracé du segment (par défaut trace = "0")

pen : position du stylo pendant le tracé. (par défaut pen = "0" invisible)



exemple : `<new type = "segment" x1="900" y1="250" pt2id="1" id="2" col="#00ff00" />`

➤ **type = "vecteur"** : vecteur $\overrightarrow{p_1p_2}$

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées d'une extrémité ou **pt1id** : identifiant d'une extrémité.

x2 y2 : coordonnées de l'autre extrémité ou **pt2id** : identifiant de l'autre extrémité.

⚠ les identifiants pt1id et/ou pt2id doivent avoir été créés au préalable.

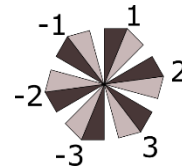
col : couleur (par défaut col = "#000000")

ep : épaisseur du trait (par défaut ep = "2")

dsh : pointillés (par défaut dsh = "0", trait continu)

trace : temps pour le tracé du segment (par défaut trace = "0")

pen : position du stylo pendant le tracé. (par défaut pen = "0" invisible)



op : option lors du tracé du vecteur,

op = "0" trace puis place la point de flèche, **op = "1"** pointe de flèche pendant le tracé.

exemple : `<new type = "vecteur" x1="240" y1="100" x2="300" y2="160" id="3"/>`

➤ **type = "droite"** : droite (p1p2)

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

col : couleur (par défaut col = "#000000")

dist1 : distance de « dépassement » de la représentation de la droite au-delà de p1 par rapport au segment [p1p2].

ratio1 : proportion de « dépassement » de la représentation de la droite au-delà de p1 par rapport au segment [p1p2]. Si ratio1 est défini, dist1 est ignoré.

Si ratio1 = 0 (par défaut) « pas de limite » coté p1

Si ratio1 = 1 la représentation s'arrête à p1 .

dist2 : distance de « dépassement » de la représentation de la droite au-delà de p2 par rapport au segment [p1p2].

ratio2 : proportion de « dépassement » de la représentation de la droite au-delà de p2 par rapport au segment [p1p2]. Si ratio2 est défini, dist2 est ignoré.

Si ratio2 = 0 (par défaut) « pas de limite » coté p2

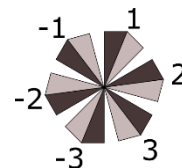
Si ratio2 = 1 la représentation s'arrête à p2 .

ep : épaisseur du trait (par défaut ep = "2")

dsh : pointillés (par défaut dsh = "0", trait continu)

trace : temps pour le tracé du segment (par défaut trace = "0")

pen : position du stylo pendant le tracé. (par défaut pen = "0" invisible)



➤ **type = "demidroite"** : demi – droite [p1p2]

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées de l'extrémité ou **pt1id** : identifiant de l'extrémité.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

col : couleur (par défaut col = "#000000")

dist2 : distance de « dépassement » de la représentation de la demi-droite au-delà de p2 par rapport au segment [p1p2].

ratio2 : proportion de « dépassement » de la représentation de la demi-droite au-delà de p2 par rapport au segment [p1p2]. Si ratio2 est défini, dist2 est ignoré.

Si ratio2 = 0 (par défaut) « pas de limite » coté p2

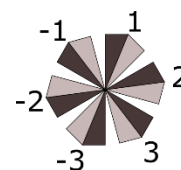
Si ratio2 = 1 la représentation s'arrête à p2 .

ep : épaisseur du trait (par défaut ep = "2")

dsh : pointillés (par défaut dsh = "0", trait continu)

trace : temps pour le tracé du segment (par défaut trace = "0")

pen : position du stylo pendant le tracé. (par défaut pen = "0" invisible)



➤ **type = "milieu"** : milieu de [p1p2] ou milieu de l' "objet"

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

ou obid : identifiant d'un objet.

⚠ **obid** : peut-être l'identifiant d'un segment mais aussi d'une droite, demi-droite ou d'un vecteur ... dans ces cas le point créé est le milieu des 2 points qui définissent l'objet.

cf. "point"

exemple : `<new type = "milieu" x1="240" y1="100" x2="300" y2="160" id="4"/>`
ou : `<new type = "milieu" obid="2" id="5"/>`

➤ **type = "compte"** : illustration d'un "comptage"

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

col : couleur (par défaut col = "#000000")

ep : épaisseur du trait (par défaut ep = "2")

trace : temps pour le tracé du segment (par défaut trace = "0")

pc : hauteur du point, valeur entre 0 et 1

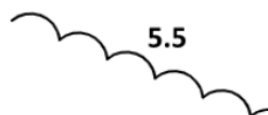
deca : décalage au départ, valeur entre 0 et 1

step : pas du comptage en pixel.

txt : affiche la valeur (txt = "on" ou txt = "off")

dir : +1 ou -1, orientation des ponts et du texte

dt : valeur de -5 à 5 déterminant la distance du texte. (par défaut dt = "1")



➤ **type = "cercle"** : cercle de centre p1 passant par p2 ou cercle de centre p1 et de rayon ray.

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées du centre ou **pt1id** : identifiant du centre.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

ray : rayon du cercle (le point p2 ne sert alors que pour définir le départ du tracé du cercle)

obid : identifiant d'un segment, d'une droite, d'une demi-droite, d'un vecteur pour définir le rayon par la distance de leur 2 points de référence. (si obid est défini, ray est ignoré. Le point p2 ne sert alors que pour définir le départ du tracé du cercle)

col : couleur (par défaut col = "#000000")

fill : couleur de remplissage (par défaut fill = "no", pas de remplissage)

bckg : remplissage placé en arrière-plan (bckg = "on" ou bckg = "off", par défaut bckg = "off")

ep : épaisseur du trait (par défaut ep = "2")

trace : temps pour le tracé du segment (par défaut trace = "0")

⚠ si le compas est visible, il "trace" le cercle.

➤ **type = "arc"** : arc de cercle de centre p0 .

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x0 y0 : coordonnées du centre ou **pt0id** : identifiant du centre.

x1 y1 : coordonnées du centre ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

ang : angle définissant l'arc. ⚠ si ang est défini alors p2 est ignoré.

op : option **op = "0"** arc centré sur p1, **op = "1"** arc "d'origine" p1 et p2 est alors l'autre extrémité.

col : couleur (par défaut col = "#000000")

fill : couleur de remplissage (par défaut fill = "no", pas de remplissage)

bckg : remplissage placé en arrière-plan (bckg = "on" ou bckg = "off", par défaut bckg = "off")

ep : épaisseur du trait (par défaut ep = "2")

trace : temps pour le tracé du segment (par défaut trace = "0")

⚠ si le compas est visible, il "trace" l'arc.

➤ **type = "polygone"** :

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

vertex : liste des identifiants des sommets. Attention le séparateur est le point virgule ;
exemple : vertex = "1;3;4". Les points doivent avoir été créés au préalable.

bckg : remplissage placé en arrière-plan (bckg = "on" ou bckg = "off", par défaut bckg = "off")

col : couleur du contour (par défaut col = "#000000")

fill : couleur de remplissage (par défaut fill = "no", pas de remplissage)

ep : épaisseur du trait (par défaut ep = "2")

➤ **type = "texte"** : affiche du texte.

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x y : coordonnées pour placer le texte ou **ptid** : identifiant du point pour placer le texte

say : texte

col : couleur (par défaut col = "#000000")

size : taille de la police de caractère (par défaut size = "24")

Nota Bene : il existe une différence de rendu du texte entre l'editor (javascript) et le player (processing). Eviter les majuscules, en début de ligne notamment, semble limiter le problème.

➤ **type = "code"** : codage des segments, des angles ou des angles droits.

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x0 y0 : coordonnées d'un point ou **pt0id** : identifiant d'un point.

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

de : type de codage :

de = "segment" code le segment $[p_1p_2]$

de = "droit" code l'angle droit $p_1\widehat{p_0p_2}$

de = "angle" code l'angle $p_1\widehat{p_0p_2}$

col : couleur du contour (par défaut col = "#000000") uniquement pour le codage d'un angle

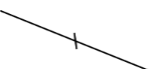





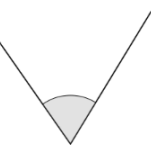

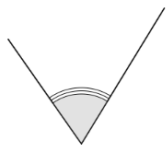

fill : couleur de remplissage (par défaut fill = "no", pas de remplissage) uniquement pour le codage d'un angle

ep : épaisseur du trait (par défaut ep = "2") uniquement pour le codage d'un angle

bckg : remplissage placé en arrière-plan (bckg = "on" ou bckg = "off", par défaut bckg = "off") uniquement pour le codage d'un angle

op : option **op = "1"** code l'angle rentrant.

n : différent codage.

n	1	2	3	4	5	6
segment						
angle						

sc : échelle

➤ **type = "interDteCer"** : intersection droite / cercle

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

ob1id : identifiant de la droite.

ob2id : identifiant du cercle.

op : option **op = "1"** ou **op = "-1"** choix d'une des 2 intersections lorsqu'elles existent.

⚠ ob1id peut-être l'identifiant d'un segment ou d'une demi-droite, dans ces cas la droite support est prise en compte.

cf. "point"

➤ **type = "inter"** : intersection de 2 droites

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

ob1id : identifiant du 1^{er} objet.

ob2id : identifiant du 2nd objet.

⚠ si les objets sont des segments, des demi-droites ou des vecteurs, calcule l'intersection des droites supports.

cf. "point"

➤ **type = "interCer"** : intersection de 2 cercles

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 r1 : coordonnées du centre et rayon du 1^{er} cercle ou **ob1id** : identifiant du 2nd cercle.

x2 y2 r2 : coordonnées du centre et rayon du 2nd cercle ou **ob2id** : identifiant du 2nd cercle.

op : option **op** = "1" ou **op** = "-1" choix d'une des 2 intersections lorsqu'elles existent.

cf. "point"

➤ **type = "pointsur"** : point sur la droite (p1p2) situé à une distance donnée de p1.

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

ou obid : identifiant d'un objet.

dist : distance au point p1. Si $dist < 0$ alors le point n'appartient pas à la demi-droite [p1p2)

ratio : coefficient multiplicateur de p1p2. Si $ratio < 0$ alors le point n'appartient pas à la demi-droite [p1p2). Si "**ratio**" est défini le paramètre "**dist**" est ignoré.

ecart : utilise l'écartement du compas (visible ou non)

Si "**ecart**" est défini le paramètre "**dist**" et "**ratio**" sont ignorés

⚠ **obid** : peut-être l'identifiant d'un segment, d'une droite, demi-droite ou d'un vecteur ... dans ces cas le point créé utilise les points p1 et p2 définissant l'objet.

cf. "point"

➤ **type = "sym"** : symétrique du point p par rapport à la droite (p1p2).

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x y : coordonnées d'un point ou **ptid** : identifiant d'un point

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

ou obid : identifiant d'un objet.

⚠ **obid** : peut-être l'identifiant d'un segment, d'une droite, demi-droite ou d'un vecteur ... dans ces cas le point créé utilise les points p1 et p2 définissant l'objet.

cf. "point"

➤ **type = "rot"** : rotation du point p2 de centre p1.

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

ou obid : identifiant d'un objet.

ang : angle de rotation.

dist : si **dist = 0** (par défaut) rotation du point p2 de centre p1. Si **dist > 0** "rotation" du point p2 de centre p1 mais donne le point à une distance **dist** de p1.

⚠ **obid** : peut-être l'identifiant d'un segment, d'une droite, demi-droite ou d'un vecteur ... dans ces cas le point créé utilise les points p1 et p2 définissant l'objet.

cf. "point"

➤ **type = "trans"** : translation du point p de vecteur $\overrightarrow{p_1p_2}$.

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x y : coordonnées d'un point ou **ptid** : identifiant d'un point

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

ou obid : identifiant d'un objet.

⚠ **obid** : peut-être l'identifiant d'un segment, d'une droite, demi-droite ou d'un vecteur ... dans ces cas le point créé utilise les points p1 et p2 définissant l'objet.

cf. "point"

➤ **type = "ptmed"** : point sur la médiatrice du segment [p1p2].

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

ou obid : identifiant d'un objet.

dist : distance du point par rapport à p1.

op : option **op = "1"** ou **op = "-1"** choix d'une des 2 possibilités.

⚠ **obid** : peut-être l'identifiant d'un segment, d'une droite, demi-droite ou d'un vecteur ... dans ces cas le point créé utilise les points p1 et p2 définissant l'objet.

cf. "point"

➤ **type = "ptbiss"** : point sur la bissectrice de $\widehat{p_1 p_0 p_2}$.

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x0 y0 : coordonnées d'un point ou **pt0id** : identifiant d'un point.

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

dist : distance du point par rapport à p0. Si dist < 0 le point est sur la bissectrice "extérieure"

⚠ **obid** : peut-être l'identifiant d'un segment, d'une droite, demi-droite ou d'un vecteur ... dans ces cas le point créé utilise les points p1 et p2 définissant l'objet.

cf. "point"

➤ **type = "perp"** : perpendiculaire à (p1p2) passant par p

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

ou obid : identifiant d'un objet.

x y : coordonnées d'un point ou **ptid** : identifiant d'un point

⚠ **obid** : peut-être l'identifiant d'un segment, d'une droite, demi-droite ou d'un vecteur ... dans ces cas le point créé utilise les points p1 et p2 définissant l'objet.

cf. "droite"

➤ **type = "para"** : parallèle à (p1p2) passant par p

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

ou obid : identifiant d'un objet.

x y : coordonnées d'un point ou **ptid** : identifiant d'un point

⚠ **obid** : peut-être l'identifiant d'un segment, d'une droite, demi-droite ou d'un vecteur ... dans ces cas le point créé utilise les points p1 et p2 définissant l'objet.

cf. "droite"

➤ **type = "media"** : médiatrice du segment [p1p2]

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

ou obid : identifiant d'un objet.

⚠ **obid** : peut-être l'identifiant d'un segment, d'une droite, demi-droite ou d'un vecteur ... dans ces cas le point créé utilise les points p1 et p2 définissant l'objet. **cf. "droite"**

➤ **type = "biss"** : bissectrice de l'angle $p_1\widehat{p_0p_2}$

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x0 y0 : coordonnées d'un point ou **pt0id** : identifiant d'un point.

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un point ou **pt2id** : identifiant d'un point.

cf. "demidroite"

➤ **type = "pointRap"** : point sur le bord du rapporteur.

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

ang : angle.

cf. "point"

➤ **type = "markRap"** : marque sur le bord du rapporteur.

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

ang : angle.

cf. "segment"

➤ **type = "surl_rect"** : rectangle pouvant servir au surlignage de texte. p1 et p2 sont deux « coins » du rectangle .

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées d'un « coin » ou **pt1id** : identifiant d'un point.

x2 y2 : coordonnées d'un autre « coin » ou **pt2id** : identifiant d'un point.

col : couleur (par défaut col = "#000000")

fill : couleur de remplissage (par défaut fill = "no", pas de remplissage)

bckg : remplissage placé en arrière-plan (bckg = "on" ou bckg = "off", par défaut bckg = "off")

ep : épaisseur du trait (par défaut ep = "2")

rd : arrondi

<tool />

Instruction permettant gestion des instruments : règle, équerre, compas, rapporteur.

Le paramètre **type** est obligatoire.

Les types disponibles sont :

- regle
- equerre
- compas
- rapporteur
- grille

➤ type = "regle"

Une seule règle peut être affichée, elle est toujours « présente ».

vis : visibilité (vis = "on" ou vis = "off", initialement vis = "off")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point ➔ **zéro de la règle**

x2 y2 : coordonnées d'un autre point ou **pt2id** : identifiant d'un autre point.

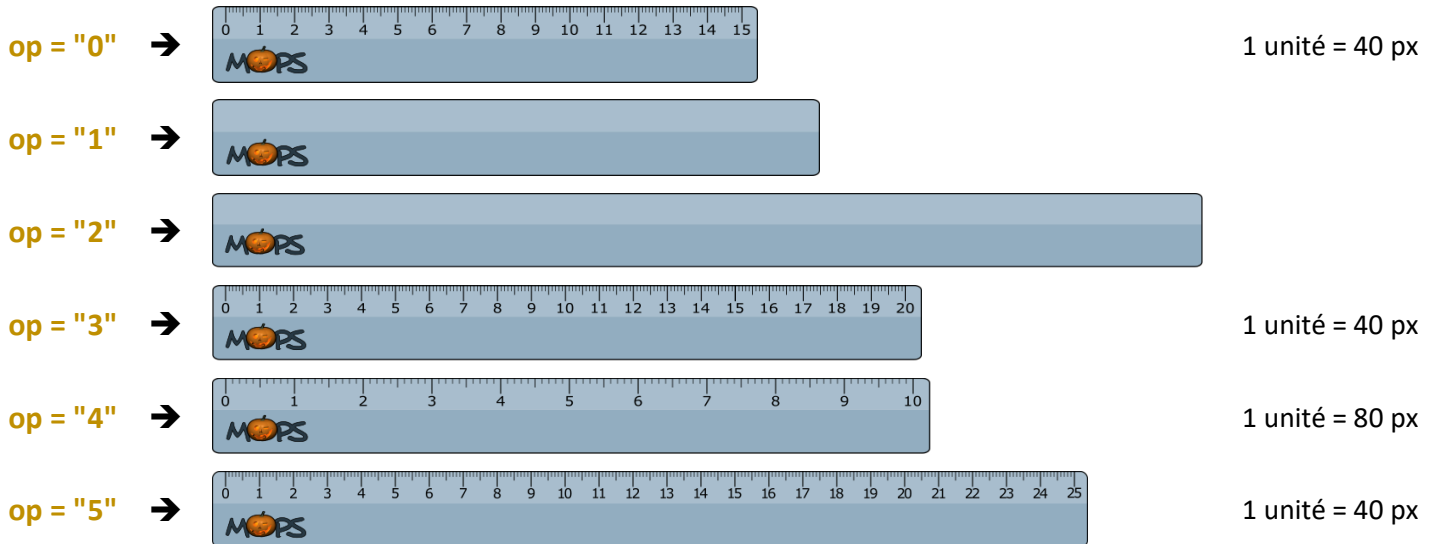
dist : distance du zéro de la règle au point p1.

ratio : coefficient multiplicateur de p1p2. Si "**ratio**" est défini le paramètre "**dist**" est ignoré.

dep : temps pour le déplacement de la règle à partir de sa position précédente (par défaut dep = "0")

op : option **0, 1, 2, 3, 4** type de règle.

start : **0** ou **1** (0 par défaut) si start = "0" le point 1 correspond au **zéro de la règle**, si start = "1" le point 1 correspond à l'autre extrémité de la règle.



exemple : `<tool type = "regle" x1="700" y1="250" x2="940" y2="560" vis="on" />`

➤ type = "equerre"

Une seule équerre peut être affichée, elle est toujours « présente ».

Perpendiculaire, passant par le point0, à la droite passant par point1 et point2.

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "off")

t : temporisation (par défaut t = "0")

x0 y0 : coordonnées d'un point ou **pt0id** : identifiant d'un point

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point

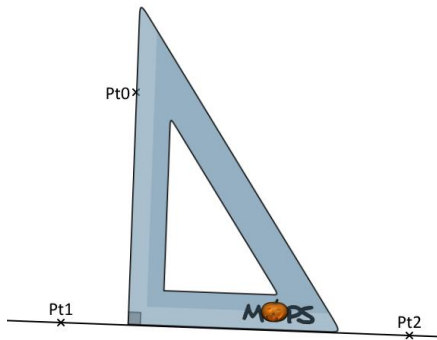
x2 y2 : coordonnées d'un autre point ou **pt2id** : identifiant d'un autre point.

dep : temps pour le déplacement de la règle à partir de sa position précédente (par défaut dep = "0")

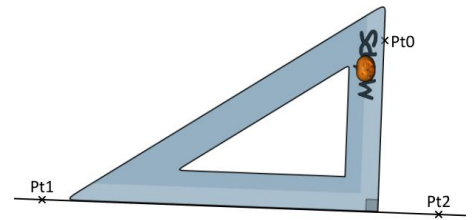
op : option **0, 1, 2, 3** Différents positionnement de l'équerre.

exemple : `<tool type = "regle" x1="700" y1="250" x2="940" y2="560" vis="on" />`

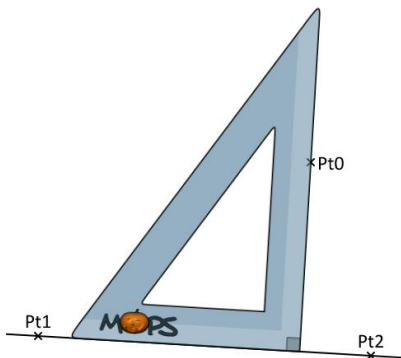
op = "0" →



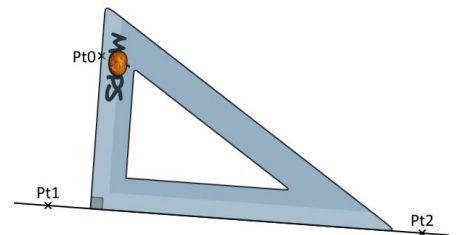
op = "1" →



op = "2" →



op = "3" →



➤ type = "compas"

Un seul compas peut être affiché, il est toujours « présent ».

Compas, la pointe correspond au point1 et le crayon au point2.

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "off")

t : temporisation (par défaut t = "0")

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point : pointe du compas.

x2 y2 : coordonnées d'un autre point ou **pt2id** : identifiant d'un autre point : crayon du compas.

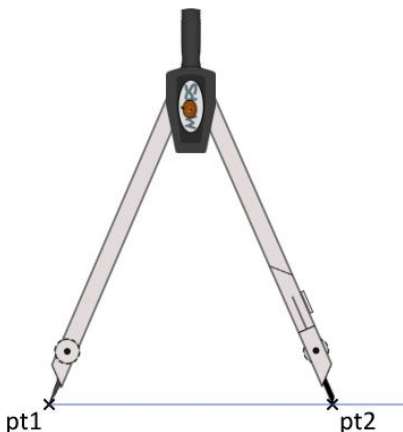
dep : temps pour le déplacement du compas à partir de sa position précédente (par défaut dep = "0")

ecart : définit l'écartement du compas. Si **ecart** est indiqué le point2 définit alors l'orientation du compas.

leve : "on" ou "off" indique si le compas est debout ou non. Par défaut leve = "off"

report : "on" ou "off". Si report = "on" le compas conserve son écartement en se déplaçant. Point2 sert à l'orientation du compas.

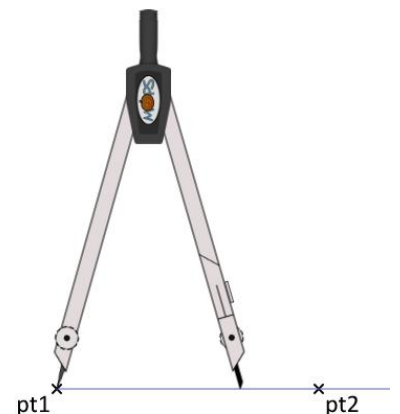
op : option 0, 1 positions du compas.



```
<tool type = "compas" x1="400" y1="250"
x2="650" y2="250" vis="on" />
```



```
<tool type = "compas" x1="400" y1="250"
x2="650" y2="250" leve="on" vis="on" />
```



```
<tool type = "compas" x1="400" y1="250"
x2="650" y2="250" ecart="175" vis="on" />
```

➤ type = "rapporteur"

Un seul rapporteur peut être affiché, il est toujours « présent ».

Rapporteur, le centre correspond au point1 et le zéro au point2.

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "off")

t : temporisation (par défaut t = "0")

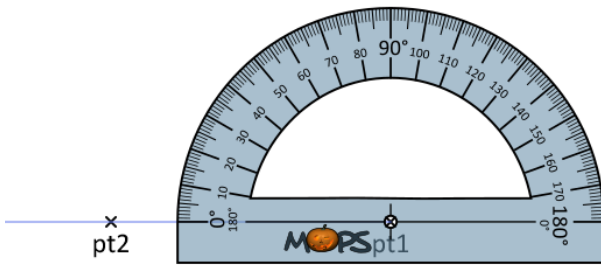
x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point : centre du rapporteur.

x2 y2 : coordonnées d'un autre point ou **pt2id** : identifiant d'un autre point : correspond au 0°.

dep : temps pour le déplacement du rapporteur à partir de sa position précédente (par défaut dep = "0")

sc : échelle du rapporteur.

op : option **0, 1** choix de la graduation.



```
<tool type = "rapporteur" x1="400" y1="250" x2="650" y2="250" vis="on" />
```

➤ type = "grille"

Une seule grille peut être affichée, elle est toujours « présente ».

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "off")

x y : coordonnées d'un point ou **ptid** : identifiant d'un point : coin en bas à gauche de la grille.

unité : 25, 40 ou 50 taille en pixels des carreaux.

repx : répétition horizontale.

repy : répétition verticale.

<image />

Instruction permettant d'afficher une image.

Le paramètre **file** est obligatoire.

id : identifiant

vis : visibilité (vis = "on" ou vis = "off", par défaut vis = "on")

t : temporisation (par défaut t = "0")

file : nom de l'image

x1 y1 : coordonnées d'un point ou **pt1id** : identifiant d'un point : coin inférieur gauche de l'image.

x2 y2 : coordonnées d'un autre point ou **pt2id** : identifiant d'un autre point : coin inférieur gauche d'arrivée de l'image en cas de déplacement.

dep : temps pour le déplacement de l'image.

sc : échelle de l'image.

top : "on" ou "off". Si top = "on" l'image est au-dessus de tous les tracés. Par défaut top = "off"

<wait />

Temporisation.

t : durée de la pause.

<step />

Instruction indiquant un "checkpoint". Pas de paramètre.

<style />

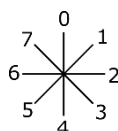
Définit le style par défaut pour la suite du script..

t : temporisation (par défaut t = "0")

size : taille de la police de caractère (par défaut size = "24").

ep : épaisseur du trait (par défaut ep = "2")

pos : position du nom du point (par défaut pos = "0")



lg : taille de la croix (par défaut lg = "4")

dsh : pointillés (par défaut dsh = "0", trait continu)

col : couleur (par défaut col = "#000000")

Partie 2 : MopsConstructionPlayer (version en ligne)

<https://myownprivatesquare.fr/mce.html>

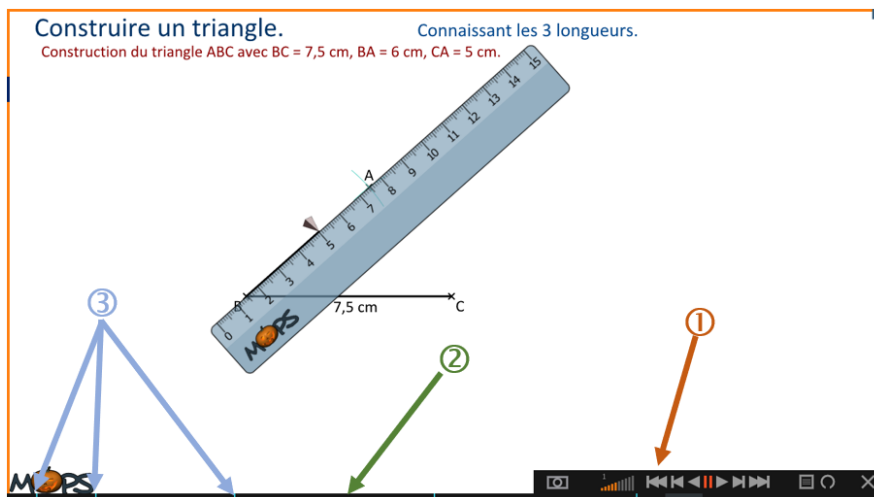
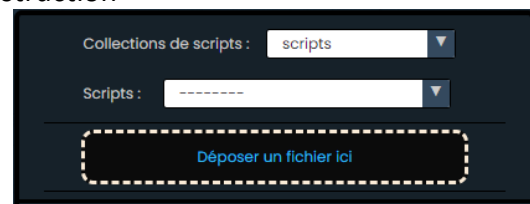
MopsConstructionPlayer permet de visualiser les scripts « MopsConstruction »

Choix d'un script prédéfini.

Certains scripts sont classés par « collections » de scripts

La zone « Drag&Drop » permet déposer et de charger un script personnel ; fichier .xml MopsConstruction.

La touche **[M]** permet d'afficher / masquer le panneau latéral



Remarque : le cadre orange indique que l'écran est sélectionné et donc que les commandes de navigation au clavier sont activées.

①. **Barre de navigation.** (touche **[W]** pour afficher/masquer cette barre)



Enregistre l'image



Vitesse de lecture



lecture en boucle



Commandes de lecture :

début / étape précédente / lecture arrière / pause / lecture / étape suivante / fin

②. **Ligne de temps.**

Vert : animation en pause / **Rouge** : animation en cours de lecture / **Jaune** : animation en lecture arrière
Le rectangle **gris** indique le temps correspondant à la ligne de script.

③. **Indicateurs d'étape dans le script.**

Si les étapes sont activées ces indicateurs apparaîtront en rose (touche **[K]** pour activer/désactiver les étapes). La lecture se mettra alors en pause automatiquement aux étapes.

Autres commandes : (ce panneau peut être affiché/masqué avec la touche **[H]**)



Partie 3 : MopsConstructionEditor (en cours de rédaction)

MopsConstructionEditor permet de faciliter la création de scripts et permet de visualiser l'animation correspondante. MopsConstructionEditor est accessible en ligne sur le site My Own Private Square.

Partie 4 : MopsConstructionPlayer (en cours de rédaction)

MopsConstructionPlayer est un programme permettant de visualiser les animations hors ligne mais aussi de créer des séquences d'images pour la création de vidéos.