

[résolu] taille dessin javacurve

Posté par damien - 25-11-2009 à 03:14

Bonjour,

J'ai encore un problème !

Je ne comprends pas comment est gérée la taille des dessins avec javacurve. La syntaxe a l'air identique à celle de coord, mais en passant du second au premier, le dessin change de taille.

Pour expliquer plus clairement, j'envoie deux exemples (mon exercice, en simplifié) :

```
\title{parallélogramme III}
\language{fr}
\author{Damien Delwarde}
\format{html}
```

```
#####
## choix des points ##
#####
```

```
## point de départ : A ##
```

```
\integer{xa=random(-5..5)}
\integer{ya=random(-5..5)}
```

```
## vecteurs pour traduire A ##
```

```
\integer{xu=0}
\integer{yu=0}
\while{\xu=0 and \yu=0}{
\integer{xu=random(-3..3)}
\integer{yu=random(-3..3)}
}
```

```
\integer{xv=0}
\integer{yv=0}
\while{\xv=0 and \yv=0}{
\integer{xv=random(-3..3)}
\integer{yv=random(-3..3)}
}
```

```
## points B, C et D ##
```

```
\integer{xb=\xa+\xu}
\integer{yb=\ya+\yu}
```

```
\integer{xc=\xb+\xv}
\integer{yc=\yb+\yv}
```

```
\integer{xd=\xa+\xv}
\integer{yd=\ya+\yv}
```

```
#####
## dessin ##
#####
```

```
## calcul des dimensions ##
```

```
\text{x=\xa,\xb,\xc,\xd}
\integer{xmin=0}
\for{i=1 to 4}{
\integer{xmin=maxima(min(\xmin,\x))}
}
\integer{xMin=\xmin-1}
```

```

\integer{xmax=0}
\for{i=1 to 4}{
\integer{xmax=maxima(max(\xmax,x))}
}
\integer{xMax=\xmax+1}

```

```

\text{y=\ya,\yb,\yc,\yd}
\integer{ymin=0}
\for{i=1 to 4}{
\integer{ymin=maxima(min(\ymin,y))}
}
\integer{yMin=\ymin-1}

```

```

\integer{ymax=0}
\for{i=1 to 4}{
\integer{ymax=maxima(max(\ymax,y))}
}
\integer{yMax=\ymax+1}

```

```

\integer{dx=\xMax-\xMin}
\integer{zx=\dx-\xMax}
\integer{dy=\yMax-\yMin}
\integer{zy=\dy-\yMax}

```

les différents dessins

```

\text{O=slib(graphpaper/millimetre \dx,\dy,\zx,\zy,,,,,)}

```

```

\text{O=wims(replace line number 2 by xrange 0,10*\dx in \O)}
\text{O=wims(replace line number 3 by yrange 0,10*\dy in \O)}

```

```

\text{OOO=draw(100,100
\O)}

```

```

\real{Xa=60*(\zx+\xa)}
\real{Ya=60*(\dy-\zy-\ya)}

```

```

\real{Xb=60*(\zx+\xb)}
\real{Yb=60*(\dy-\zy-\yb)}

```

```

\real{Xc=60*(\zx+\xc)}
\real{Yc=60*(\dy-\zy-\yc)}

```

```

\real{Xd=60*(\zx+\xd)}
\real{Yd=60*(\dy-\zy-\yd)}

```

```

#####
## énoncé ##
#####

```

\statement{On se place dans le plan muni d'un repère (O, I, J) .

Soient $(A:(x_a, y_a))$, $(B:(x_b, y_b))$ et $(C:(x_c, y_c))$.

Cliquez pour représenter le point (A)

```
\embed{ reply1 }
}
```

```
#####
## solutions ##
#####
```

```
\answer{{\OOO;circle,\Xa,\Ya,12}{type=coord}{option=noanalyzeprint}}
```

et le deuxième, le même avec javacure :

```
\title{parallélogramme II}
\language{fr}
\author{Damien Delwarde}
\format{html}
#####
```

```
#####
## choix des points ##
#####
```

```
## point de départ : A ##
```

```
\integer{xa=random(-5..5)}
\integer{ya=random(-5..5)}
```

```
## vecteurs pour traduire A ##
```

```
\integer{xu=0}
\integer{yu=0}
\while{\xu=0 and \yu=0}{
\integer{xu=random(-3..3)}
\integer{yu=random(-3..3)}
}
```

```
\integer{xv=0}
\integer{yv=0}
\while{\xv=0 and \yv=0}{
\integer{xv=random(-3..3)}
\integer{yv=random(-3..3)}
}
```

```
## points B, C et D ##
```

```
\integer{xb=\xa+\xu}
\integer{yb=\ya+\yu}
```

```
\integer{xc=\xb+\xv}
\integer{yc=\yb+\yv}
```

```
\integer{xd=\xa+\xv}
\integer{yd=\ya+\yv}
```

```
#####
## dessin ##
#####
```

```
## calcul des dimensions ##
```

```
\text{x=\xa,\xb,\xc,\xd}
\integer{xmin=0}
\for{i=1 to 4}{
\integer{xmin=maxima(min(\xmin,\x))}
```

```

}
\integer{xMin=\xmin-1}

\integer{xmax=0}
\for{i=1 to 4}{
\integer{xmax=maxima(max(\xmax,x))}
}
\integer{xMax=\xmax+1}

\text{y=\ya,\yb,\yc,\yd}
\integer{ymin=0}
\for{i=1 to 4}{
\integer{ymin=maxima(min(\ymin,y))}
}
\integer{yMin=\ymin-1}

\integer{ymax=0}
\for{i=1 to 4}{
\integer{ymax=maxima(max(\ymax,y))}
}
\integer{yMax=\ymax+1}

\integer{dx=\xMax-\xMin}
\integer{zx=\dx-\xMax}
\integer{dy=\yMax-\yMin}
\integer{zy=\dy-\yMax}

## les différents dessins ##

\text{O=slib(graphpaper/millimetre \dx,\dy,\zx,\zy,,,,,)}
\text{O=wims(replace line number 2 by xrange 0,10*\dx in \O)}
\text{O=wims(replace line number 3 by yrange 0,10*\dy in \O)}

\text{OOO=draw(500,500
\O)}

\real{Xa=60*(\zx+\xa)}
\real{Ya=60*(\dy-\zy-\ya)}

\real{Xb=60*(\zx+\xb)}
\real{Yb=60*(\dy-\zy-\yb)}

\real{Xc=60*(\zx+\xc)}
\real{Yc=60*(\dy-\zy-\yc)}

\real{Xd=60*(\zx+\xd)}
\real{Yd=60*(\dy-\zy-\yd)}

#####
## énoncé ##
#####

\statement{On se place dans le plan muni d'un repère  $(O, I, J)$ .

```

Soient $A:(x_a, y_a)$, $B:(x_b, y_b)$ et $C:(x_c, y_c)$.

Cliquez pour représenter les points A , B et C

```
\embed{r1}
}
```

```
#####
## solutions ##
#####
```

```
\answer{}{\OOO;points,\Xa,\Ya,\Xb,\Yb,\Xc,\Yc}{type=javacurve}{option=noanalyzeprint}
```

Je ne comprends pas pourquoi les deux dessins sont affichés différemment.

Damien.

=====

Re:taille dessin javacurve

Posté par damien - 25-11-2009 à 03:18

je précise que le changement de taille dans la ligne

```
\text{\OOO=draw(100,100 \O)}
```

n'a aucun effet sur l'un et l'autre exercice.

=====

Re:taille dessin javacurve

Posté par bernadette - 25-11-2009 à 07:20

Le problème vient du slib utilisé graphpaper qui impose une taille et ne tient pas compte de la taille indiquée. Il y a eu une discussion à ce sujet récemment il me semble.

Bernadette

=====

Re:taille dessin javacurve

Posté par damien - 25-11-2009 à 17:02

Oui, j'ai lu cette discussion et c'est ce qui explique pourquoi les paramètres dans draw n'ont aucun effet, graphpaper est "prioritaire" pour définir la taille.

En revanche, je n'arrive pas à expliquer la taille de l'image avec javacurve. Je n'arrive même pas à la modifier ! Si je me souviens bien, donné comme ça, le dessin est affiché seulement en partie et je n'ai pas trouvé le moyen de l'agrandir. La seule chose que j'arrive à faire c'est, en modifiant la première ligne retournée par graphpaper, changer l'échelle du dessin. Ce qui fait que, pour l'avoir en entier, je suis obligé de l'afficher en petit. Le problème est que, outre la lecture plus difficile, il faut être d'autant plus précis pour placer les points.

Il doit bien y avoir un moyen de changer la taille de l'affichage du dessin...

=====

Re:taille dessin javacurve

Posté par bernadette - 25-11-2009 à 18:51

Oui, il faut mettre la taille voulue dans \embed{} sinon c'est une taille par défaut. Donc quelque chose comme

`\text{si=wims(line 1 of \O)}`
`\text{si=wims(replace new by in \si)}`
`\integer{sizex=\si}`
`\integer{sizey=\si}`
(il faut que `\sizex` soit calculé)

avant le statement et dans le statement

`\embed{r1,\sizex x \sizey}`

Je vais le rajouter dans la documentation en ligne.
(voir page 42 de la documentation en pdf de Sophie et moi-même).

=====

Re:taille dessin javacurve

Posté par damien - 25-11-2009 à 20:31

Ça marche bien, je n'avais pas pensé à mettre la taille dans embed.

Merci beaucoup.

=====