

Le widget canvas est un widget qui implémente un mode dessin à structure d'affichage. Cette structure est une liste d'*items* 2D1/2. Les items peuvent être manipulés par programme, et on peut définir des liaisons d'événements au niveau des items. On peut manipuler des groupes d'items grâce à la notion de *tag*. La commande de création est "canvas".

**Options principales** - Ces options de configurations sont utilisées lors de la création d'un canvas dans les méthodes `configure` et `cget` d'un widget.

`-width nn / -height nn` largeur / hauteur du canvas  
`-xscrollcommand script` callback pour le défilement horizontal  
`-yscrollcommand script` callback pour le défilement vertical

**Création d'items** - Un item est créé par la méthode `create` du canvas. Chaque item a un numéro unique dans le canvas, appelé son identificateur.

`set c [canvas .c]`  
`set id [$c create <type> <coordonnées> <options>]`

Les *types* disponibles avec leurs principales options sont les suivants :

`arc` arc de cercle ou d'ellipse  
`bitmap` image monochrome  
`image` image quelconque  
`line` polyligne ou courbe spline  
`oval` rectangle arrondi  
`polygon` polygone ou courbe spline fermée  
`rectangle` rectangle  
`text` texte multiligne  
`window` widget

Les *coordonnées* sont spécifiées sous forme d'une liste `x1 y1 x2 y2 ... xn yn`. Les items de type `line` et `polygon` prennent une liste de longueur variable. Les items de type `bitmap`, `image`, `text` et `window` prennent un point `x y` ; les autres items prennent deux points `x1 y1 x2 y2` spécifiant le rectangle englobant la forme.

Les *options* sont spécifiées sous la forme de couples `-option valeur` comme pour les widgets. La plupart des types prennent en compte les options suivantes :

`-fill color` couleur de remplissage pour les formes fermées  
`-outline color` couleur du bord pour les formes à contour  
`-width nn` épaisseur du contour pour les formes à contour  
`-tags list` liste de tags (voir ci-dessous)  
`-smooth on/off`  
`-text texte` texte pour un item `text`  
`-anchor nlslelwlne...` point d'ancrage pour les items ayant une seule coordonnée `x y`

**Tags** - Chaque item peut avoir un ou plusieurs tags ou étiquettes. Un tag est une chaîne de caractères quelconque qui ne doit pas commencer par un chiffre. Certains noms sont réservés : `all` (qui est associé à tous les items) et `current`, qui est associé à l'item sous la souris s'il y en a un. La plupart des méthodes de manipulation des items prennent en argument soit l'identificateur de l'item (retourné par exemple par la commande de création), soit un tag. Dans ce dernier cas, la commande s'applique, selon les cas, à tous les items qui ont ce tag ou au premier d'entre eux. Les tags servent également pour définir des liaisons d'événements.

**Recherche d'items** - La méthode `find` permet de récupérer une liste d'identificateurs d'items selon plusieurs critères spécifiés dans une expression de recherche "searchSpec" :

`$w find withtag tagorid` items ayant le tag ou l'identificateur `tagorid`  
`$w find closest x y halo` item le plus proche de la coordonnée `xy`  
`halo` spécifie une tolérance  
`$w find enclosed x1 y1 x2 y2` items contenus dans le rectangle `x1 y1 x2 y2`  
`$w find overlapping x1 y1 x2 y2` items intersectant le rectangle `x1 y1 x2 y2`  
`$w find above/below tagorid` item juste au-dessous/au-dessus des items `tagorid`  
`$w find all` tous les items

## Manipulation des items -

\$w <u>itemcget</u> tagorid -option	retourne la valeur d'une option d'un item
\$w <u>itemconfigure</u> tagorid -option valeur ...	change la valeur d'une ou plusieurs options d'un item
\$w <u>coords</u> tagorid	retourne les coordonnées d'un item
\$w <u>delete</u> tagorid	détruire tous les items spécifiés par tagorid
\$w <u>addtag</u> tag searchSpec arg ...	ajoute 'tag' à tous les items qui satisfont l'expression searchSpec (voir la méthode addtag)
\$w <u>dtag</u> tag	enlève le tag 'tag' de tous les items
\$w <u>dtag</u> tagorid tag	enlève le tag 'tag' de tous les items spécifiés par tagorid
\$w <u>gettags</u> tagorid	retourne la liste des tags d'un item
\$w <u>lower</u> tagorid	fait passer les items spécifiés par tagorid à l'arrière-plan
\$w <u>raise</u> tagorid	fait passer les items spécifiés par tagorid au premier plan
\$w <u>move</u> tagorid dx dy	déplace les items spécifiés par tagorid de dx dy
\$w <u>scale</u> tagorid xo yo sx sy	applique une homothétie de centre xo yo, de rapport horizontal sx et de rapport vertical sy aux items tagorid.
\$w <u>bbox</u> tagorid ...	retourne le rectangle englobant tous les items tagorid.

**Liaison d'événements** - On peut définir des liaisons d'événements sur le widget canvas lui-même comme pour tout widget. De plus, on peut définir des liaisons d'événements sur les items du canvas grâce à la méthode bind, de syntaxe similaire à la commande bind :

\$c bind tagorid <evenement> script

Les seuls types d'événements qui peuvent être utilisés sont ceux relatifs aux dispositifs d'ent (ButtonPress, ButtonRelease, Motion, Key, KeyPress, KeyRelease) et les événements Enter et Leave correspondent à l'entrée et la sortie du curseur de la souris sur un item.

Si l'on fait la liaison sur un tag, tous les items (présents et à venir) qui auront ce tag bénéficieront de liaison. Lorsqu'un événement se produit, plusieurs liaisons peuvent éventuellement être déclenchées l'item a plusieurs tags et qu'il existe des liaisons pour cet événement associées aux différents tags. Dans ce cas, toutes les liaisons sont exécutées. De plus les liaisons d'événements du widget canvas lui-même si elles existent, sont toujours déclenchées.

**Items de type texte** - Un certain nombre de méthodes supplémentaires sont dédiées aux items de type texte, afin de pouvoir éditer leur contenu. Le contenu d'un item texte est accédé à l'aide d'indices. L'indice peut être :

nombre i	la position avant le i-ème caractère (le premier caractère a l'indice 0)
<u>end</u>	la position après le dernier caractère
<u>insert</u>	la position du curseur d'insertion de texte
<u>sel.first</u>	le début de la sélection
<u>sel.last</u>	la fin de la sélection
<u>@x,y</u>	la position la plus proche des coordonnées xy

Les principales méthodes spécifiques des items textes sont :

\$c <u>insert</u> tagorid indice chaine	insère la chaine à l'indice donné de l'item texte
\$c <u>dchars</u> tagorid first [last]	détruit les caractères d'indice first à last
\$c <u>focus</u> tagorid	spécifie que l'item reçoit les événements du clavier
\$c <u>icursor</u> tagorid indice	positionne le curseur à l'indice donné
\$c <u>index</u> tagorid indice	retourne l'indice numérique correspondant à indice
\$c <u>select from</u> tagorid index	spécifie l'indice du début de la sélection
\$c <u>select to</u> tagorid index	spécifie l'indice de la fin de la sélection

**Items de type window** - Les items de type window sont des widgets de Tk qui apparaissent dans le canvas, comme n'importe quel autre widget. Lorsque l'on crée un item window, il faut spécifier le chemin d'accès par l'option -window.

**Sortie postscript** - La méthode postscript permet de produire une représentation postscript du contenu du canvas, par exemple pour l'imprimer.