

Placement : les Panes avec gestionnaire de placement

19 janvier 2021

Les Panes avec gestionnaire de placement

Layout-Pane = Pane avec un *Layout Manager* (gestionnaire de placement)

- héritent de Pane,
- placent automatiquement les noeuds fils à l'intérieur du panneau
- chacun place d'une certaine manière : en ligne, en grille, etc...
- utilisent les propriétés des noeuds fils : `minHeight`, `prefHeight`, `margin`, etc... pour calculer le placement.
- on peut toujours paramétrer le placement ; espacement entre les noeuds, alignement, centrage, etc...
- peuvent changer les tailles des noeuds fils pour respecter les contraintes.

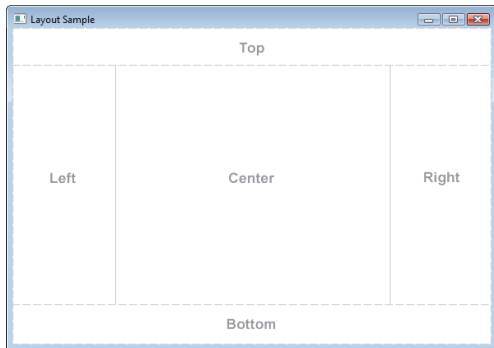
Les Panes avec gestionnaire de placement: Les différents Layout-Pane

- **BorderPane** : découpe le Pane en 5 zones (*Top*, *Left*, *Right*, *Bottom* et *Center*). Ne permet de placer qu'un noeud par zone. Une zone non occupée attribuée est répartie sur les autres zones
- **HBox** et **VBox** : place les composants en une ligne (HBox), ou en une colonne (VBox)
- **FlowPane** horizontal/vertical : place les composants en une ligne/colonne et passe à la ligne/colonne suivante s'il n'y a plus de place dans le conteneur.
- **GridPane** : place les composants dans une grille, les largeurs des colonnes et hauteur des lignes s'adaptent au plus grand noeud qui s'y trouve, les cellules peuvent donc avoir des tailles différentes. Un composant peut s'étaler sur plusieurs lignes et colonnes.
- **TilePane** : place les composants à la suite dans une grille, toutes les cellules (les tuiles) ont la même taille.
- **StackPane** : empile les composants les uns au dessus des autres.
- **AnchorPane** : place les composants en fonction d'une contrainte de distance par rapport aux bords de l'AnchorPane.

BorderPane

BorderPane Place les composants fils en 5 zones avec `setTop`, `setLeft`, `setRight`, `setBottom`, `setCenter`

- Un seul composant par zone.



- Découpage de la zone.
 - Top et Bottom prennent la hauteur préférée des composants qui s'y trouvent. Ces composants peuvent s'élargir jusqu'à prendre la largeur du panneau.
 - Left et Right prennent la largeur préférée des composants qui s'y trouvent. Ces composants peuvent s'agrandir jusqu'à prendre toute la hauteur du Panneau (moins la hauteur des zones Top et Bottom)
 - Center occupe l'espace restant, le composant qu'y trouve peut s'agrandir pour occuper toute la zone.
- Alignement dans une zone, les méthodes statiques :
 - `BorderPane.setAlignment(Node n, Pos)` permet d'aligner un composant dans sa zone (`Pos.CENTER`, `Pos.BOTTOM_LEFT`,...)
 - `BorderPane.setMargin(Node n, Insets)` permet de laisser une marge autour d'un composant.

Remarque : les `Insets` sont des écarts dans les 4 directions, ils servent à définir des marges, des bordures, etc... Par exemple `Insets i = new Insets(10,20,30,40)` crée un écart de 10 pixels en haut, 20 à droite, 30 en bas, 40 à gauche.

GridPane place ses composants fils dans une grille

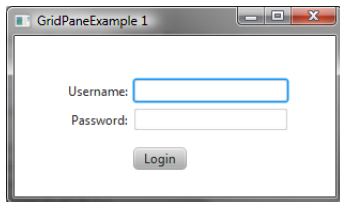
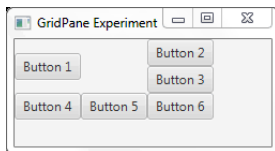
```
add(Node n, int x, int y)
```

```
add(Node n, int x, int y, int spanX, int spanY)
```

Remarque : les composants de la grille peuvent être de classe différente.

- certains fils peuvent être s'étaler sur plusieurs lignes ou colonnes.
- les largeurs de colonnes et hauteur de lignes sont variables, et s'adaptent en fonction des dimensions préférées du plus grand composant qui s'y trouvent.
- le nombre de lignes et colonnes s'adaptent automatiquement
- on peut spécifier :
 - les écarts entre les lignes `setVgap` et colonnes `setHgap`
 - Des contraintes sur les lignes et colonnes : hauteurs min et max, alignement et agrandissement par défaut des composants d'une case.
 - des contraintes sur un composant précis :
 - son alignement avec `GridPane.setHalignment(Node, Pos)`,
 - son agrandissement avec `GridPane.setHgrow(Node, Priority)`
 - ...

GridPane



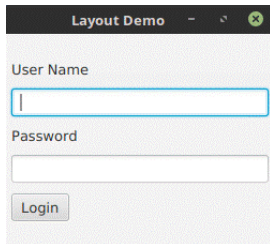
HBox et VBox:

VBox permet de mettre ses fils en ligne horizontale, les composants sont ajoutés de gauche à droite

- on ajoute les éléments par `getChildren().add()`
- HBox respecte si possible la largeur préférée des composants, sinon les réduit à `minWidth`
- alignement par défaut `Pos.TOP_LEFT` (la ligne est calée en haut à gauche de la HBox)
- `spacing` (écart entre les fils) par défaut à 0.
- Possibilité sur un noeud fils de :
 - lui mettre une marge particulière avec `VBox.setMargin(Node, Insets)`, cela permet de décaler le noeud sur la ligne.
 - lui dire d'occuper toute la place possible avec `VBox.setHGrow(Node, Priority)`

Comportement équivalent pour VBox, mais en colonne.

HBox et VBox:



FlowPane:

FlowPane place les composants les uns à la suite des autres en ligne/colonne, et passe à la ligne/colonne suivante quand il atteint le bord du panneau

- ➊ ajout des composants avec `getChildren().add();`
- ➋ `orientation` permet de choisir un `FlowPane` vertical ou horizontal.
- ➌ `Hgap` et `Vgap` pour les espacements dans la ligne et entre la colonne,
- ➍ alignement global dans le conteneur, par défaut en `Pos.TOP_LEFT`
- ➎ alignement dans les lignes,

TilePane:

TilePane place les composants dans une grille.

- orientation
 - Tile panel horizontal = `prefColumns` fixé.
 - Tile panel vertical = `prefRows` fixé.
- ajout des éléments par `getChildren().add()`
- toutes les tuiles ont la même taille (contrairement au `GridPane`) adaptée au plus grand des composants.
- `prefTileWidth`, `prefTileHeight` : dimension préférée des tuiles.
- Propriétés habituelles `hgap` `vgap` alignement,...