

ING 1 - POO Java

Autoformation à JavaFX

Gestionnaire de positionnement

Positionnement de composants

Comme vous pouvez le constater, les composants graphiques se rajoutent les uns à la suite des autres et même si c'est suffisant pour certaines applications basiques, c'est insuffisant et complexe pour construire de vraies applications.

Il est parfois utile de placer les composants de manière absolue sur l'écran. Mais cela présente de nombreux inconvénients (taille de l'écran différente suivant les machines, choix de l'utilisateur...)

Le gestionnaire de positionnement permet entre autre de redimensionner automatiquement certains objets. L'idée principale est de définir des **règles de disposition** (des contraintes) que le gestionnaire se chargera de faire respecter en fonction du contexte spécifique de la machine cible.

Les différents composants que l'on peut placer dans un gestionnaire de positionnement possèdent des propriétés qui peuvent être prises en compte lors du calcul de leur position :

- `minWidth` : Largeur minimale souhaitée pour le composant
- `prefWidth` : Largeur préférée (idéale) du composant
- `maxWidth` : Largeur maximale souhaitée pour le composant
- `minHeight` : Hauteur minimale souhaitée pour le composant
- `prefHeight` : Hauteur préférée (idéale) du composant
- `maxHeight` : Hauteur maximale souhaitée pour le composant

Ces valeurs sont des souhaits et sont respectés dans la mesure du possible mais parfois c'est impossible.

Comme pour son prédécesseur Swing, JavaFX propose différents gestionnaires de positionnement, voici ci-dessous quelques-uns des principaux gestionnaires de positionnement disponibles :

- [HBox](#)
- [VBox](#)
- [FlowPane](#)
- [BorderPane](#)
- [GridPane](#)

HBox

Le principe du gestionnaire [HBox](#) est de positionner les nœuds JavaFX sur une ligne horizontale. Chaque nœud ajouté est rajouté à la droite du précédent. Par défaut, le gestionnaire [HBox](#) (et tous les autres gestionnaires) essayent de respecter la taille des nœuds enfants.

L'alignement des composants enfants est déterminé par la propriété `alignment`. (par défaut type énuméré `TOP_LEFT`).

L'exemple suivant vous donne un exemple de positionnement avec HBox. Quatre rectangles de couleur aléatoire sont créés et ajoutés avec un objet de type text. Le constructeur qui est utilisé pour HBox et qui prend un entier en paramètre indique l'espacement entre les objets.

Le code suivant générera la fenêtre ci-dessous :

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Rectangle;
import javafx.scene.layout.HBox;
import javafx.scene.layout.FlowPane;
import javafx.stage.Stage;
import javafx.scene.paint.Color;
import javafx.scene.text.Text;
import javafx.scene.text.Font;
import java.util.Random;

public class TestHBox extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {
```

```

// création du gestionnaire HBox
HBox hbox = new HBox(20);

Random rand = new Random();

//Création de la Scene en lui passant le noeud racine
//ainsi que la largeur et la hauteur
Scene scene = new Scene(hbox ,800, 600);

Rectangle r1 = new Rectangle(50, 50);
Rectangle r2 = new Rectangle(20, 20);
Rectangle r3 = new Rectangle(100, 100);
Rectangle r4 = new Rectangle(80, 80);

Color c = Color.rgb(rand.nextInt(255),
                    rand.nextInt(255),
                    rand.nextInt(255));

r1.setFill(c);
c = Color.rgb(rand.nextInt(255),
              rand.nextInt(255),
              rand.nextInt(255));

r2.setFill(c);
c = Color.rgb(rand.nextInt(255),
              rand.nextInt(255),
              rand.nextInt(255));

r3.setFill(c);
c = Color.rgb(rand.nextInt(255),
              rand.nextInt(255),
              rand.nextInt(255));

r4.setFill(c);

Text t = new Text("Test HBox");
t.setFont(new Font(24));

hbox.getChildren().addAll(r1,r2,r3,r4,t);

//On remplit la Scene d'une couleur
//scene.setFill(lg);

//Ajout d'un titre à la fenêtre --> Stage
primaryStage.setTitle("Gestionnaire HBox");

//On ajoute la Scene au stage --> Stage
primaryStage.setScene(scene);

//On affiche le stage
primaryStage.show();
}
public static void main(String args[]){
    launch(args);
}
}

```

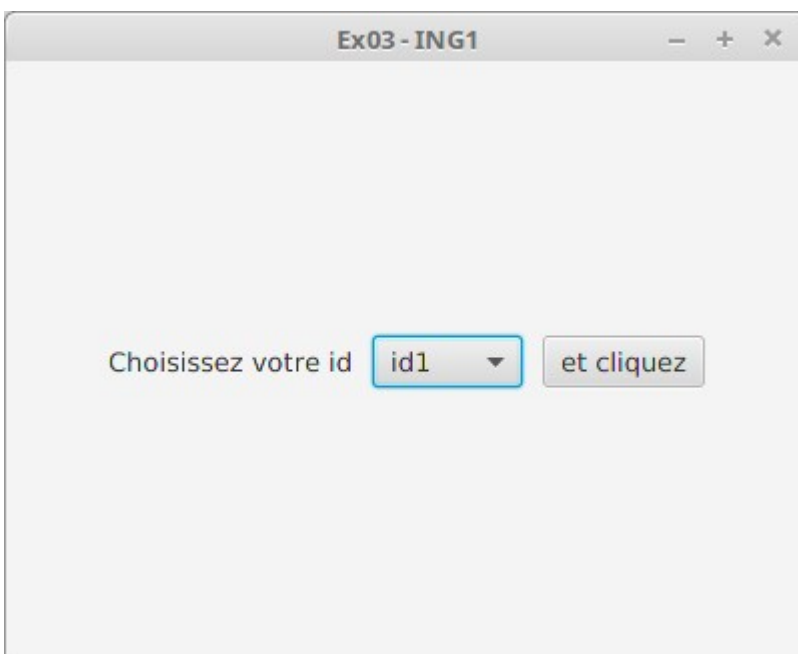


Le constructeur de Hbox accepte comme paramètre une valeur d'espace entre tous les nœuds, cette valeur peut être écrasée par la méthode `setSpacing(xx)`. C'est par défaut.

Si possible, le gestionnaire respecte la taille « préférée » des composants. Si la fenêtre devient trop petite, il les affiche à leur taille minimale (`minWidth`)

Exercice

Réaliser l'exemple suivant :



La « correspond » correspond bien évidemment à une HBox, dans lequel ont été successivement rajoutés un Label, une ComboBox et un bouton. Vous trouverez les informations ici les informations concernant la [ComboBox](#).

Vous vous souvenez du cours ? (sinon reportez lui à lui !). Il existe une propriété padding qui permet de définir des marges entre les bords du gestionnaire et son contenu. La méthode `setPadding()` qui est en fait une méthode héritée de la classe [Region](#) permettant de définir des marges. Cette méthode réclame qu'on lui passe en paramètre un objet de la classe [Inset](#) :

```
hbox.setPadding(new Insets(20, 10, 10, 20));
```

Un point très important sur lequel nous reviendrons ultérieurement est l'application possible des feuilles de style, qui permettent de passer un style CSS en caractères.

Voici par exemple, le style pour le bouton passé sous forme de CSS : `btnC.setStyle("-fx-text-fill: #006464;-fx-background-color: #DFB951;-fx-background-radius: 20;-fx-padding: 5;");`

Testez-le

Le gestionnaire [VBox](#) se comporte comme le gestionnaire HBox à la différence que les objets sont positionnés verticalement.

Il existe pour tous les gestionnaires de nombreuses options comme l'espacement entre les objets, la bordure, etc.

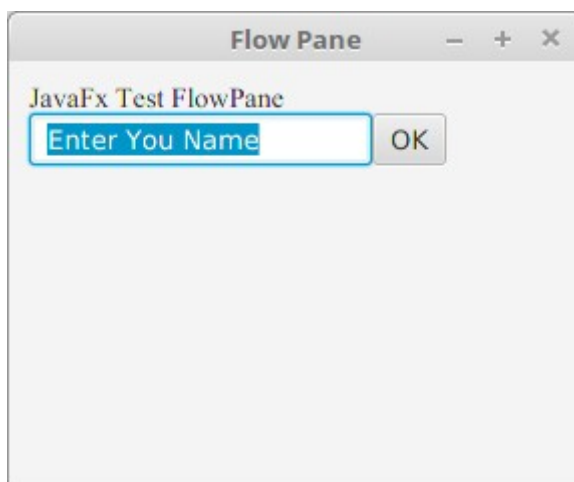
FlowPane

Le gestionnaire de positionnement FlowPane est un gestionnaire assez aisé à manipuler. S'il ne reste plus de place horizontalement, les objets positionnés sur une ligne virtuelle au-dessous. Par défaut, le gestionnaire FlowPane positionne les objets horizontalement, mais il est possible de positionner les objets verticalement.

Exercice :

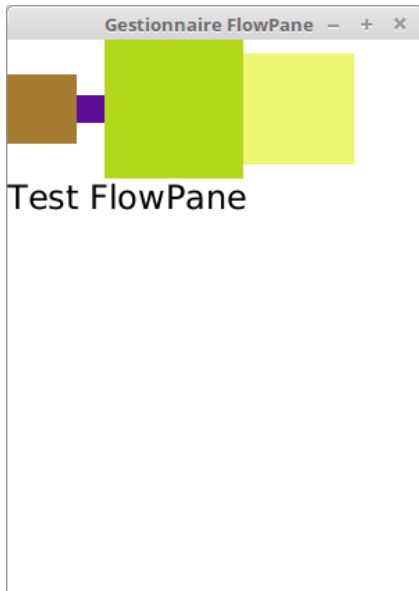
Réaliser l'exemple suivant :

Le « pane » est bien évidemment de type FlowPane. La fonte utilisée est du Times New Roman 14 et j'utilise un objet de type [TextField](#) et un objet de type [Button](#)



Les bordures

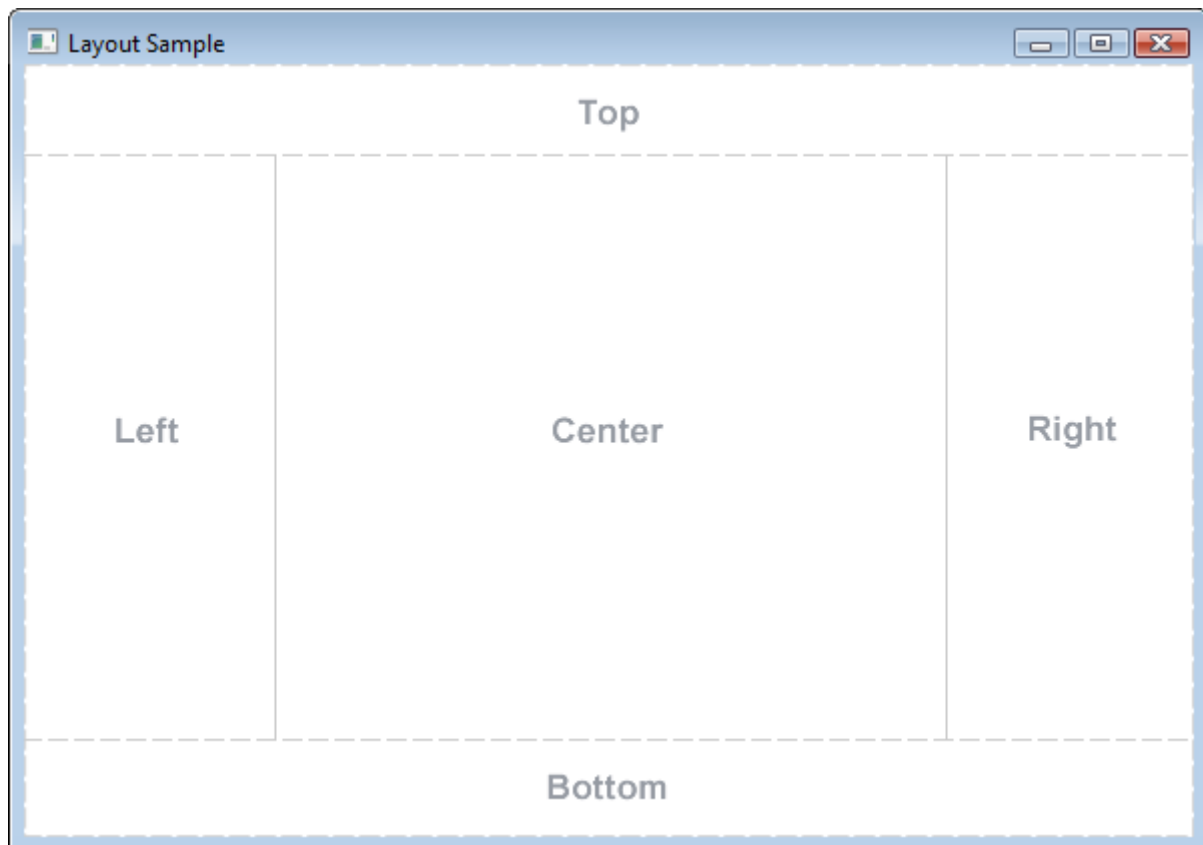
Toutes les sous-classes de la classe [Region](#) peuvent avoir une bordure



BorderPane

C'est un gestionnaire de positionnement un peu plus complexe mais très utile.

Le gestionnaire BorderPane définit 5 zones et place 1 seul composant (conteneur ou contrôle graphique) par zone :



Ce gestionnaire de positionnement assez simple permet de spécifier la position de placement d'un objet en indiquant sa position :

- `setTop(Node)`
- `setBottom(Node...)`
- `setCenter(Node...)`
- `setLeft(Node...)`
- `setRight(Node...)`

De méthodes **statiques** (`setAlignment()`, `setMargin()`) de `BorderPane` peuvent être invoquées pour appliquer des contraintes de positionnement à un composant passé en paramètre.

```
BorderPane.setAlignment(monBouton,  
Pos.CENTER);
```

Pour la méthode `setMargin()`, il faut passer un élément de type `Insets`.

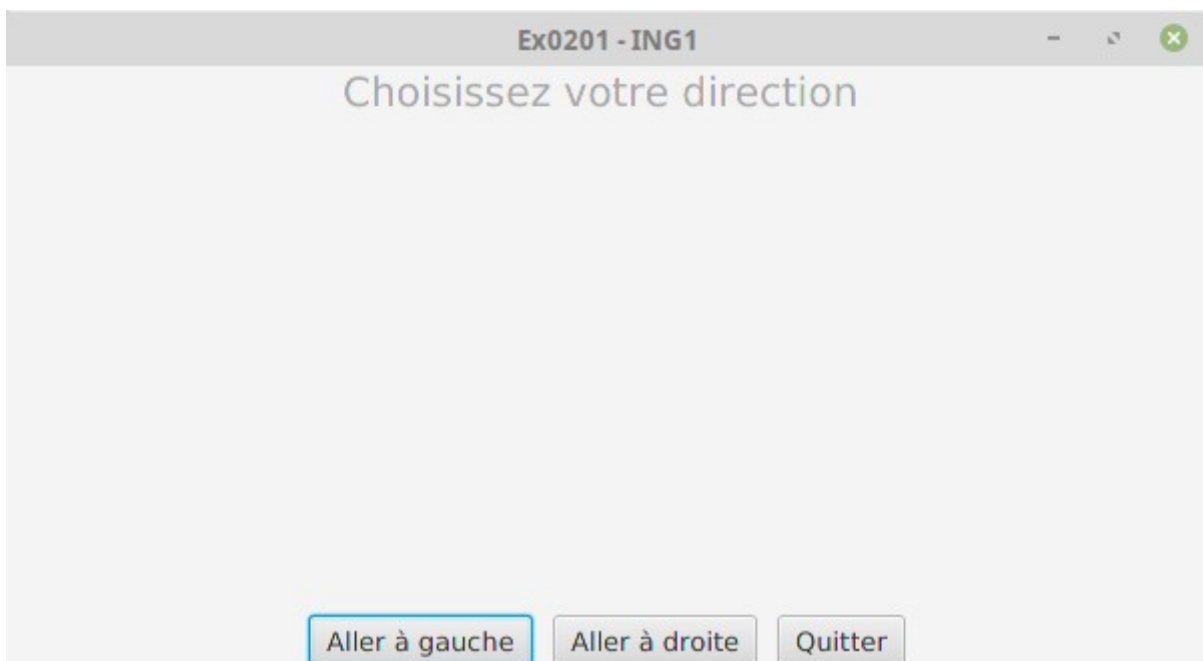
Le gestionnaire `BorderPane` est un des plus utilisé car il représente « assez bien » la manière dont est organisé un écran. On peut bien évidemment rajouter dans la chaque zone un ou plusieurs gestionnaires de positionnement

Il existe d'autres gestionnaires de positionnement comme `GridPane` qui permet de placer les éléments dans un ensemble de lignes et de colonnes qui est également très utilisé. Nous allons le voir et en même temps, nous allons commencer à placer des objets caractéristiques des applications JavaFX

Il est très courant d'utiliser plusieurs gestionnaires de positionnement dans une même application. Très souvent le conteneur `BorderPane` est le conteneur de base car il reprend la structure d'une page web ou d'une application « classique ». Il est ensuite tout à fait possible de rajouter d'autres gestionnaires de positionnement dans les différentes zones de notre `BorderPane`.

Exercice :

Réaliser l'interface suivante qui est un `BorderPane` dans lequel pour placer les différents éléments du bas de l'interface une `HBox` a été utilisée.



J'ai placé trois éléments de type `Button` dans un gestion de positionnement de type `HBox` qui est situé dans la partie basse du `BorderPane`. La fonte choisie est de type `Arial` avec une taille de police de 20 pour la partie haute (couleur `DARKGREY`).

Il existe d'autres gestionnaires de positionnement que nous n'aborderons pas ici