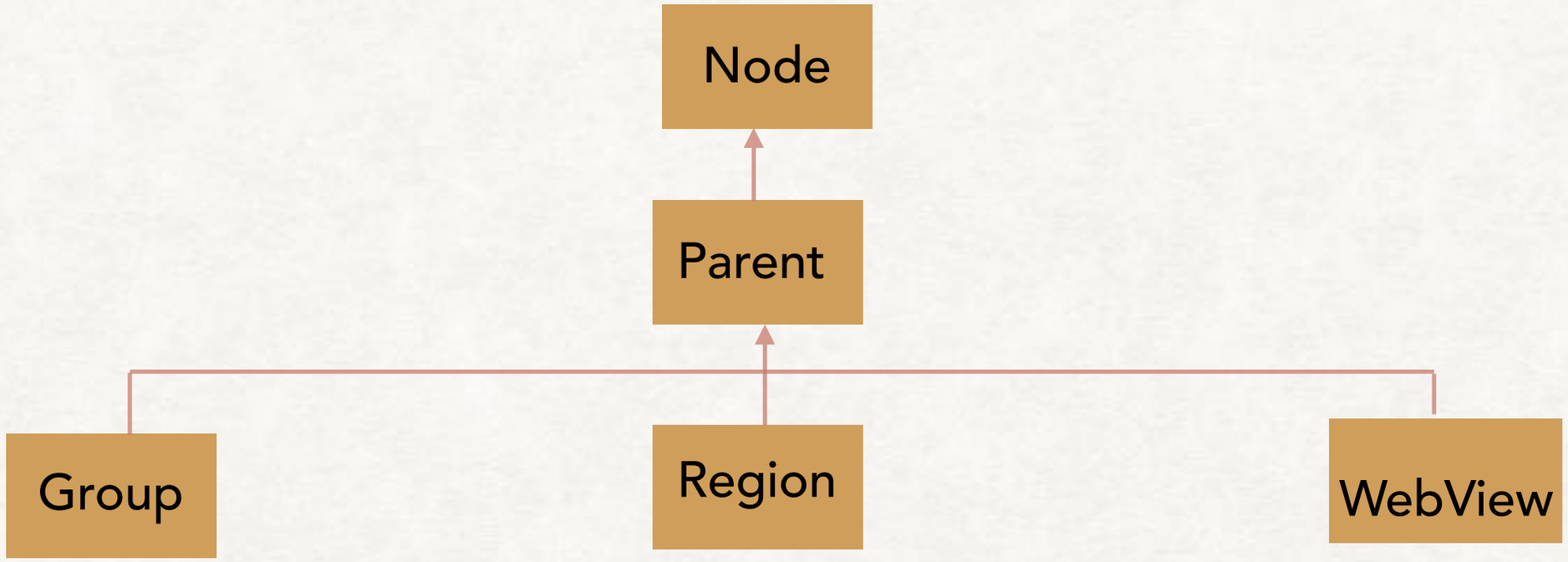
LAYOUT PANE

LAYOUT PANE

A quoi sert LayoutPane?

- c'est un gestionnaire de position
- permet d'organiser les Nodes de l'interface graphique
- implémente la gestion de position spécifique à chaque type de layout pane
- automatiquement repositionne et redimensionne les Nodes
- plus besoin de positionner les Nodes manuellement

Group

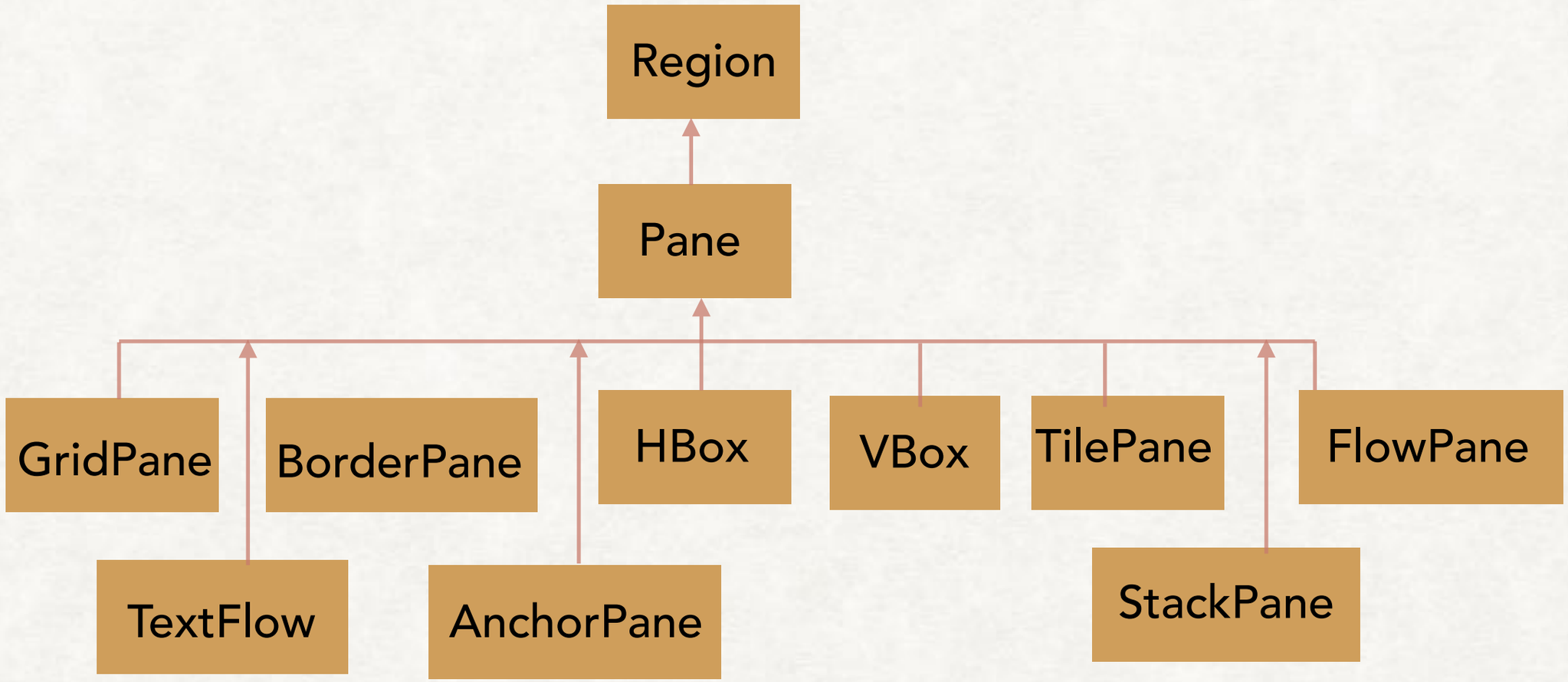


Group - pour regrouper les éléments pour applique la même transformations à tous les enfants du groupe

WebView - affichage pages web

Region

Region


Region
Pane
GridPane
BorderPane
HBox
VBox
TilePane
FlowPane
TextFlow
AnchorPane
StackPane

Region

Region peut changer les dimension et style (CSS)

Propriétés intéressantes :

- width, height
- minWidth, minHeight
- maxWidth, maxHeight
- prefWidth, prefHeight
- padding — top, bottom, left, right, espace autour de contenu du région

Placement layout

Pos Enums :

- TOP_LEFT TOP_RIGHT TOP_CENTER
- CENTER_LEFT CENTER_RIGHT CENTER
- BOTTOM_LEFT BOTTOM_RIGHT BOTTOM_CENTER
- BASELINE_LEFT BASELINE_RIGHT BASELINE_CENTER

HPos Enums:

- CENTER LEFT RIGHT

VPos Enums

- TOP CENTER BOTTOM BASELINE

Priorités Layout

Les priorités spécifiques pour chaque enfant de layout.

- déterminent si on grossit/rétrécit l'enfant quand layout change les dimensions

Priority Enums:

- ALWAYS — toujours changer la taille avec d'autres éléments qui ont la priorité ALWAYS
- NEVER — jamais changer la taille
- SOMETIMES —- si l'espace disponible partager avec d'autres noeuds qui ont la priorité SOMETIMES

Pane

Classe mère de tous les layout.

Permet de spécifier le positionnement absolu de tous les enfants.

Les enfants gardent la position absolue même si on change les dimensions de la fenêtre.

HBox

- Les noeuds enfants arrangés sur une ligne
- pas de passage à la ligne suivante (donc certains enfants peuvent être invisibles si la largeur de la fenêtre insuffisant)

Propriétés:

- alignment — alignement des enfants dans HBox (Pos Enums, par défaut VERTICAL_TOP, HORIZONTAL_LEFT)
- fillHeight — si vrai alors les enfants redimensionnés pour remplir la box en hauteur (vrai par défaut)
- spacing — espace entre deux enfants (0 par défaut)

On peut ajouter les contraintes sur les enfants de HBox:

- hgrow — la priorité horizontale (pour le grossissement de l'enfant)
- margin — espace autour (à l'extérieur) de l'enfant

HBox

```
Button plus = new Button("+");

Button moins = new Button("-");

Label label = new Label(compteur+"");

Text text = new Text("nombre de click");

HBox.setMargin(text, new Insets(20,10,0,0));

HBox box = new HBox();

box.getChildren().addAll(text,plus,label,moins);

box.setAlignment(Pos.CENTER);

box.setSpacing(30);
```

VBox

- Les noeuds enfants arrangés sur une colonne
- pas de passage à la colonne suivante (donc certains enfants peuvent être invisibles si la hauteur de la fenêtre insuffisant)

Propriétés:

- alignment — alignement des enfants dans HBox (Pos Enums, par défaut VERTICAL_TOP, HORIZONTAL_LEFT)
- fillWidth — si vrai alors les enfants redimensionnés pour remplir la box en largeur (vrai par défaut)
- spacing — espace entre deux enfants (0 par défaut)

On peut ajouter les contraintes sur les enfants de HBox:

- vgrow — la priorité verticale (pour le grossissement de l'enfant)
- margin — espace autour (à l'extérieur) de l'enfant

FlowPane

- Les noeuds enfants rangés soit horizontalement soit verticalement
- A la fin de la ligne (ou colonne) les enfants qui "débordent" passent à la ligne (colonne) suivante.

Propriétés:

- hgap — distance entre les enfants sur la ligne (horizontal flow) ou entre les colonnes (vertical flow)

- vgap — distance entre les enfants dans la colonnes (vertical flow) ou distance entre les lignes (horizontal flow), par défaut 0. Les enfants peuvent chevauchés si vgap négatif.

- alignment — alignement de contenu de FlowPane (vertical TOP et horizontal LEFT par défaut)
- columnHalignment — alignement horizontale de noeuds dans FlowPane verticale
- rowValignment — alignement verticale de noeuds dans FlowPane horizontale
- prefWrapLength — largeur ou hauteur préférée
- orientation — Orientation.HORIZONTAL Orientation.VERTICAL

FlowPane

Contrainte à imposer sur un noeud enfant:

- margin — espace autour de noeud enfant (type Insets)

Exemple:

```
FlowPane fp = new FlowPane(Orientation.HORIZONTAL);
FlowPane.setMargin(label, new Insets(5,5,5,5));
fp.setVgap(12.0);
fp.getChildren().add(label);
```

TilePane

Espace divisé en grille de tuiles de même taille. La taille d'une tuile déterminée par la taille du plus grand enfant.
Les enfants peuvent être rangés soit horizontalement soit verticalement (les tuiles remplies ligne après ligne ou colonne après colonne).
Propriétés:

- hgap, vgap — espace entre les enfants sur la ligne ou sur la colonne
- alignement
- prefRows, prefColumns — les préférences sur le nombre de lignes et colonnes
- prefTileWidth, prefTileHeight — préférences sur la taille d'un tuile
- tileWidth, tileHeight — la taille réelle d'une tuile (read only)
- orientation — Orientation.HORIZONTAL, Orientation.VERTICAL

TilePane

Les propriétés que nous pouvons spécifier pour chaque enfant (avec des méthodes statiques de TilePane):

- margin — espace autour d'un enfant
- alignment — position de l'enfant à l'intérieur d'une tuile

Exemple:

```
TilePane tilePane = new TilePane();
TilePine.setAlignment(text, Pos.TOP_CENTER);
TilePane.setMargin(bouton, new Insets(5,10,10,20));
tilePane.getChildren().addAll(text,bouton);
```

StackPane

StackPane empile les enfants un sur l'autre. Utiliser par exemple pour mettre un text sur un autre objet (rectangle, cercle, peu importe).

Propriété :

- alignment — alignement des enfants à l'intérieur de StackPane, par défaut CENTER

Contraintes à imposer sur chaque enfant :

- alignment — modifie alignement globale donnée par la propriété du même nom
- margin

Appliquer les effets visuels sur StackPane donne de résultats différents que appliquer les effets sur chaque enfant séparément.

GridPane

- Les noeuds enfants placés dans une grille.
- Contrairement à TilePane le placement en lignes/colonnes est figé (les noeuds enfants ne passent jamais sur la ligne ou colonne suivante).

Propriétés:

- hgap — espace entre les colonnes
- vgap — espace entre les lignes
- alignement — alignement à l'intérieur d'une case de la grille
- gridLinesVisible — si true alors les lignes de la grille deviennent visible (utile pour déboggage)

GridPane

Les contraintes:

- rowIndex — la ligne où commence le placement d'un enfant
- columnIndex — la colonne où commence le placement d'un enfant
- rowSpan — le nombre de lignes occupées par l'enfant
- columnSpan — le nombre de colonnes occupées par l'enfant
- halignment — alignement horizontal d'un enfant à l'intérieur d'une case (HPos)
- valignment — alignement verticale d'un enfant à l'intérieur d'une case (VPos)
- margin — autour de l'enfant

Les contraintes sont spécifiées pour chaque noeud enfant avec des méthodes statiques de GridPane.

BorderPane

- Fournit cinq régions pour placer les composants
- Certaines régions peuvent être vide et dans ce cas ne prennent pas de place
- Par défaut s'il y a trop de place alors la place supplémentaire est attribuée à la région au milieu
- Si pas assez de place alors les régions peuvent chevaucher.

Propriétés utiles:

- top — place le composant en haut
- bottom — place le composant en bas
- left — place le composant sur le côté gauche de BorderPane
- center — place le composant au centre de BorderPane
- right — place le composant sur le côté droite de BorderPane

BorderPane

Propriétés à spécifier pour les composants enfants de BorderPane:

- alignment - alignement de l'enfant dans BorderPane
- margin - l'espace autour de l'enfant dans BorderPane

Exemple:

```
BorderPane borderPane = new BorderPane();
BorderPane.setAlignment( rect, Pos.TOP_CENTER);
BorderPane.setMargin(rect, new Insets(5,5,5,5));
borderPane.setCenter(borderPane);
```

AnchorPane

- AnchorPane permet d'accrocher les noeuds enfants en haut, en bas, à gauche et à droite.
- Méthodes statiques pour spécifier les contraintes — la distance entre le composant et le bord de AnchorPane.

Contraintes de AnchorPane (à appliquer sur les composants enfants):

- topAnchor — la distance entre le bord "top" de l'enfant et l'insets top de AnchorPane
- leftAnchor — la distance entre le bord gauche de l'enfant et l'insets gauche de AnchorPane
- rightAnchor — la distance entre le bord droit de l'enfant et l'insets droit de AnchorPane
- bottomAnchor — la distance entre le bord "bottom" de l'enfant et l'insets bottom de AnchorPane

AnchorPane

AnchorPane est utilisé pour que les composants enfants restent à une distance fixe des bords de AnchorPane quand la fenêtre est redimensionnée.

TextFlow

- TextFlow est utilisé surtout pour l'affichage de texte.
- TextFlow gère correctement le passage à la nouvelle ligne quand la largeur est trop petite.
- Il est possible de placer d'autres noeuds que Text dans un TextFlow.

Propriétés:

- lineSpacing — espace verticale entre deux lignes
- textAlignment — alignment horizontale