

Laboratoire 1

JavaFX, les premiers pas

Objectifs d'apprentissage

- Comprendre la notion de graphes de scene
- Comprendre et utiliser les classes Group et Region
- Utiliser les panneaux de mise en page (Layout panes)
- Construire des interfaces utilisateur graphiques simples
- Créer des zones de texte, des boutons, des étiquettes
- Créer des boutons radio, des cases à cocher

Évaluation du laboratoire

Ce laboratoire est **formatif**, mais il vous permettra d'obtenir des points lorsque le professeur jugera que vous avez atteint les objectifs d'apprentissage. Répondre le mieux possible aux questions posées, puis validez vos solutions avec le professeur. Votre code doit respecter les normes de programmation du cours.

Important :

Pour tous les contrôles, importer les classes qui se trouvent dans l'ensemble de classes **javafx.scene.....**

Créez un projet JavaFX *laboratoire 1*. Pour chaque exercice, créez une sous-classe de la classe *Application*. N'oubliez pas d'y inclure la méthode *main()* qui contient l'instruction *launch()*

Exercice 1 : Quelques classes de base

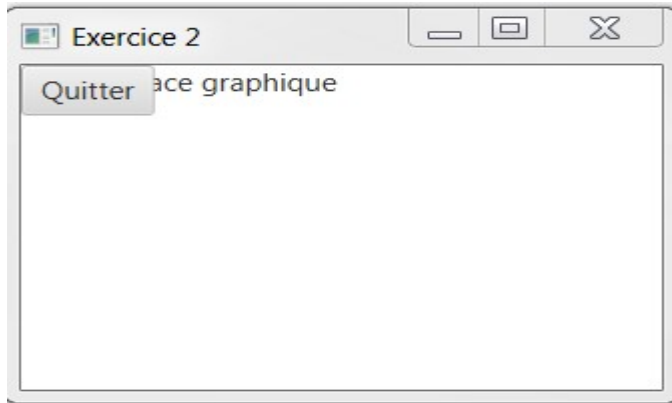
En utilisant l'aide en ligne de Java, étudiez les classes suivantes :

- **Node** : c'est la classe (abstraite) de base de tous les nœuds d'un graphe de scène.
- **Parent** : c'est la classe (abstraite) de base de tous les nœuds (composants) qui sont des branches.
- **Scene** : C'est la classe conteneur pour tous les nœuds d'un graphe de scène.

Exercice2 : Création d'une première interface graphique JavaFX, utilisation de la classe Group

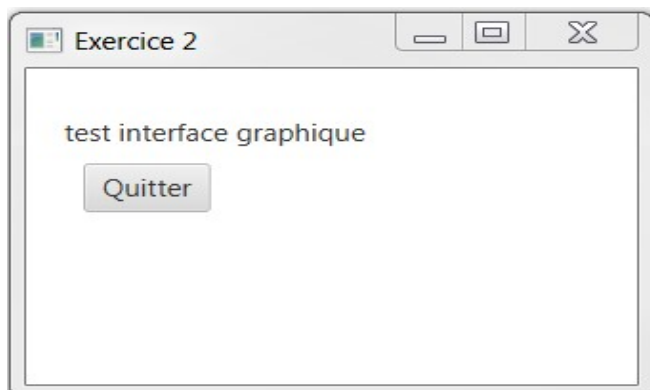
- Créez un nouveau projet JavaFX (laboratoire 1)
- Renommez la classe Main en *Exercice2*
- Remplacez l'instruction *BorderPane root = new BorderPane()* (générée par JavaFX) par *Group root = new Group();*

- La taille de la fenêtre doit être de 320 x 200
- Le titre de la fenêtre est «*Exercice 2*»
- Ajoutez une étiquette (Label) «*Test Interface Graphique* » et un bouton (Button) avec le texte « *Quitter* » au groupe *root*.
- L'exécution de votre application donne l'affichage suivant :



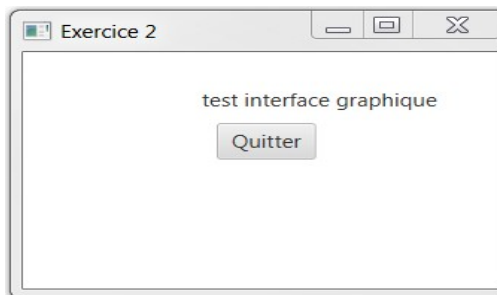
Noter que les deux composants sont à la même position (0,0)

- Utilisez les méthodes *setLayoutX* et *setLayoutY* de la classe *Node* pour placer l'étiquette à la position (20,30) et le bouton à la position (30,60). Vous pouvez également utiliser la méthode *relocate* de la classe *Node*.
- L'exécution de votre application donne l'affichage suivant :



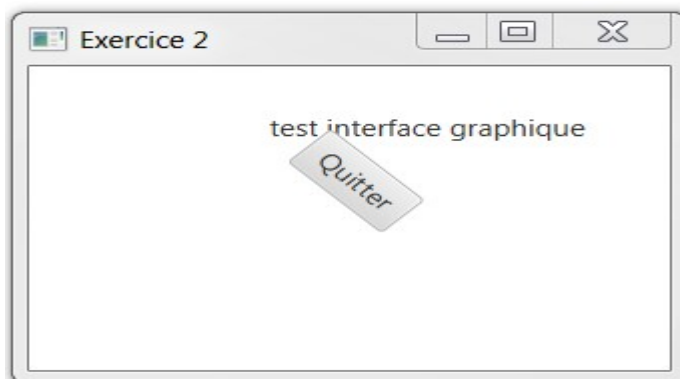
Appliquez une transformation simple (translation en X) au groupe. Utilisez l'instruction *root.setTranslateX(100);*

- L'exécution de votre application donne l'affichage suivant :



Notez que **tous les composants du groupe** ont subi la transformation.

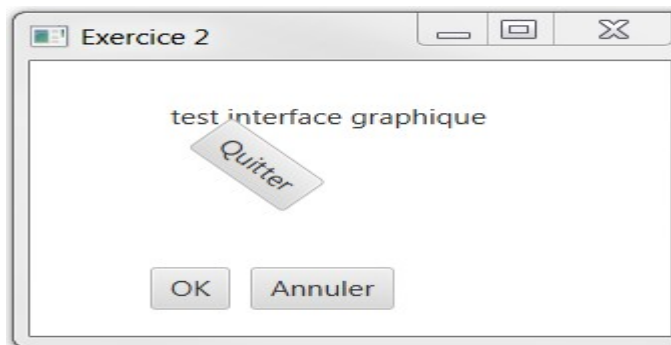
- Appliquez une transformation de votre choix à un des composants, puis testez.
- Voici un exemple avec une rotation de 45° du bouton



- **Positionnement d'un composant par rapport à un autre**
 - Créez un bouton *btnOK* à la position (10,150).
 - Créez un bouton *btnAnnuler*. Ce bouton sera positionné relativement au bouton *btnOK* comme suit :

```
NumberBinding layX = btnOK.layoutXProperty().add(btnOK.widthProperty().add(10));
btnAnnuler.layoutXProperty().bind(layX);
btnAnnuler.layoutYProperty().bind(btnOK.layoutYProperty());
root.getChildren().addAll(btnOK,btnAnnuler);
```

- Bien comprendre ces instructions. L'exécution de l'application affiche ceci :



Exercice 3 : La classe Region

C'est la classe de base pour tous les panneaux de mise en page et les contrôles. Utilisez l'aide en ligne pour comprendre ses caractéristiques.

La méthode setBackground()

- Créez une classe *Exercice3* (sous classe de *Application*)
- Créez un panneau racine de votre scène, *Pane root = new Pane();*
- La taille de la fenêtre doit être de 400 x 400
- Le titre de la fenêtre est «*Exercice 3*»
- Créez les fonds (remplissages) comme suit :

```

BackgroundFill bgFillVert = new BackgroundFill(Color.LIGHTGREEN, new CornerRadii(10), new Insets(20));
BackgroundFill bgFillRouge = new BackgroundFill(Color.RED, new CornerRadii(25), new Insets(30));
Background bg = new Background(bgFillVert, bgFillRouge);
root.setBackground(bg);

```

- L'exécution de l'application donne l'affichage suivant :



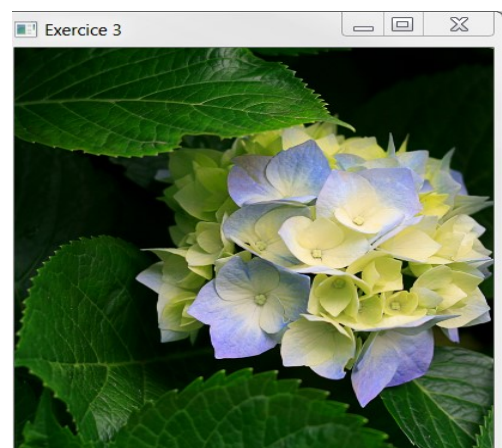
- Étudiez les constructeurs des classes **BackgroundFill** et **Background**. Changez les valeurs des paramètres et ré exécutez.
- Il est aussi possible d'utiliser une image lors de l'appel de la méthode **setBackground()**. Voici un exemple

```

Image image = new Image("hydrangeas.PNG");
BackgroundSize bgTaille = new BackgroundSize(100,100,true,true,false,true);
BackgroundImage bgImage = new BackgroundImage(image, BackgroundRepeat.SPACE, BackgroundRepeat.SPACE, BackgroundPosition.DEFAULT, bgTaille);
Background bg1 = new Background(bgImage);
root.setBackground(bg1);

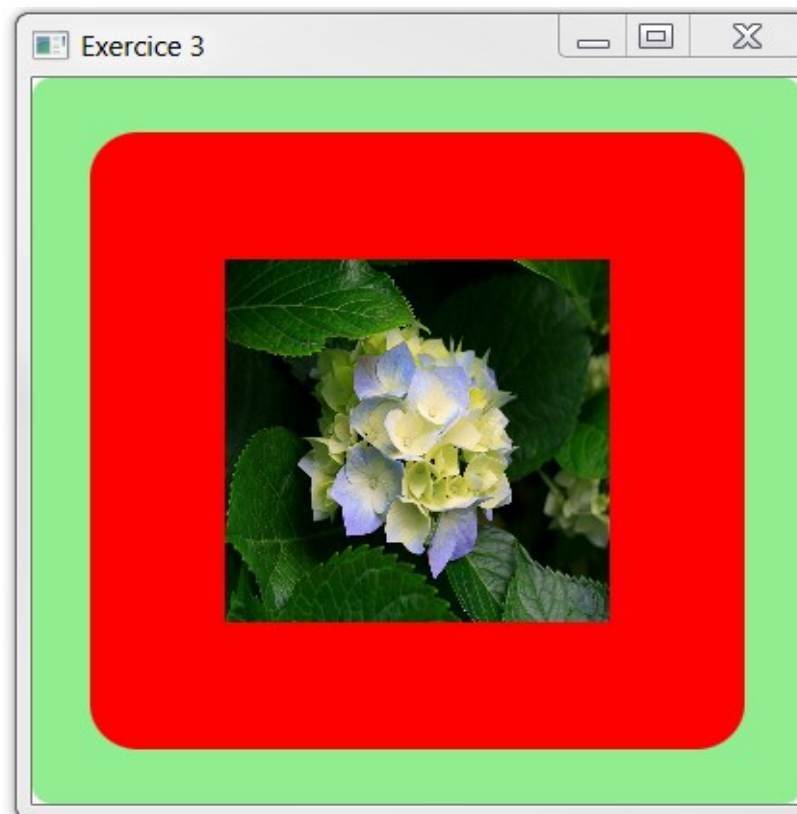
```

- Notez que l'image doit se trouver dans un dossier ressource que vous devez créer (New/Source Folder).
- Étudiez les constructeurs des classes **BackgroundSize** et **BackgroundImage**
- La fenêtre ressemble à ceci :



- **Utilisation du remplissage et l'image de fond en même temps.** Utilisez le constructeur :

Background(BackgroundFill[] fills, BackgroundImage[] images) pour appliquer les deux les remplissages *bgFillVert*, *bgFillRouge* et l'image de fond *bgImage*. Vous devez changer les paramètres des constructeurs *BackgroundSize* et *BakgroundImage* pour avoir le résultat suivant :



La méthode `setBorder()`

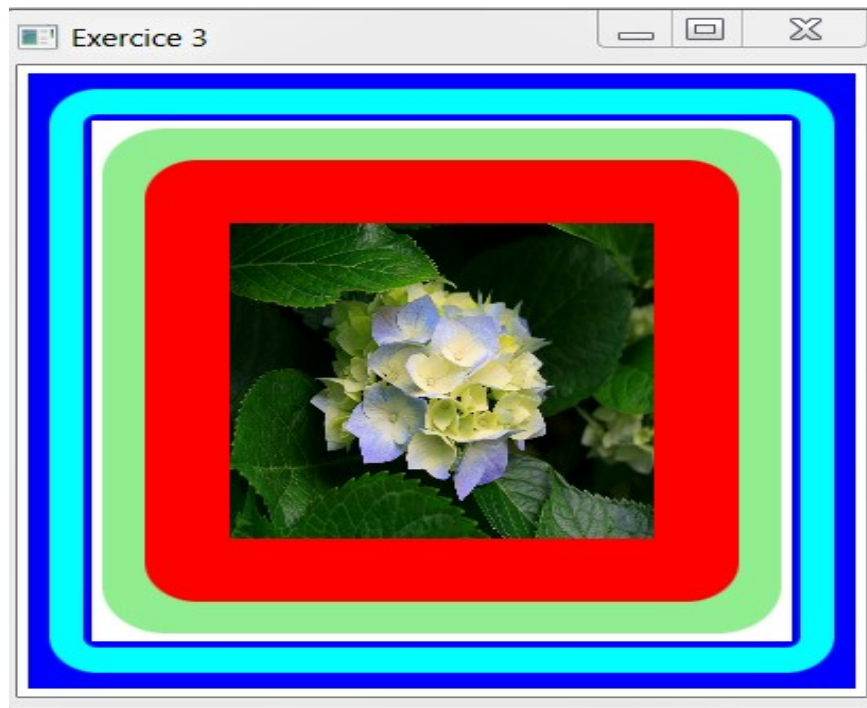
- Pour ajouter des bordures à notre fenêtre, modifiez l'argument du constructeur *Insets* à 40 pour l'objet *bgFillVert* et à 60 pour l'objet *bgFillRouge*, ceci nous permettra de mieux voir les bordures.
- Ajoutez les instructions suivantes à votre classe *Exercice3* :

```
BorderStrokeStyle styleBleu = new BorderStrokeStyle (StrokeType.CENTERED, StrokeLineJoin.MITER, StrokeLineCap.BUTT,10,0,null);
BorderStrokeStyle styleCyan = new BorderStrokeStyle (StrokeType.INSIDE, StrokeLineJoin.MITER, StrokeLineCap.BUTT,10,0,null);

BorderStroke bordureBleu = new BorderStroke(Color.ALICEBLUE.BLUE, styleBleu, CornerRadii.EMPTY, new BorderWidths(30), new Insets(20));
BorderStroke bordureCyan = new BorderStroke(Color.ALICEBLUE.CYAN, styleCyan, new CornerRadii(15), new BorderWidths(16), new Insets(15));

Border bordure = new Border(bordureBleu,bordureCyan);
root.setBorder(bordure);
```

- L'affichage ressemble à ceci :

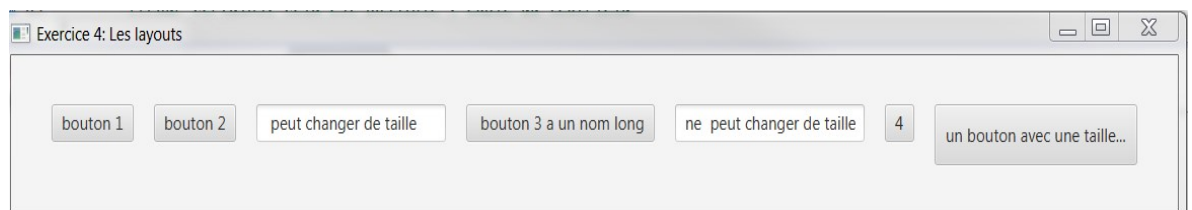


- Étudiez les classes ***BorderStrokeStyle*** et ***BorderStroke*** qui permettent de définir respectivement le style et le trait d'une bordure.
- Changez les valeurs des paramètres des constructeurs des deux classes précédentes et observez l'affichage obtenu.

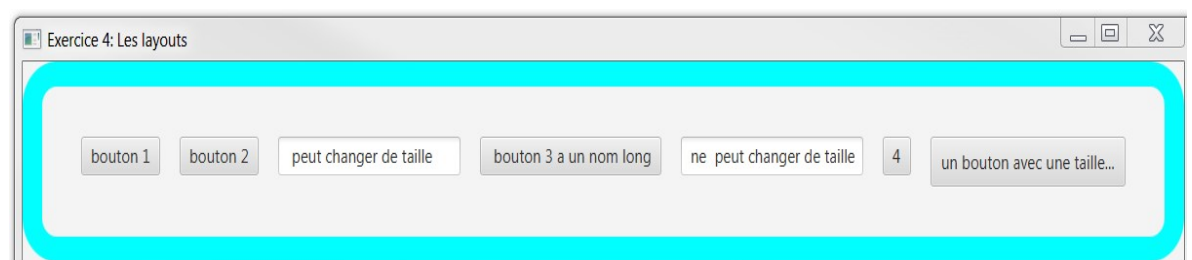
Exercice 4 : Quelques panneaux de mise en page (panneaux conteneurs)

Le HBox

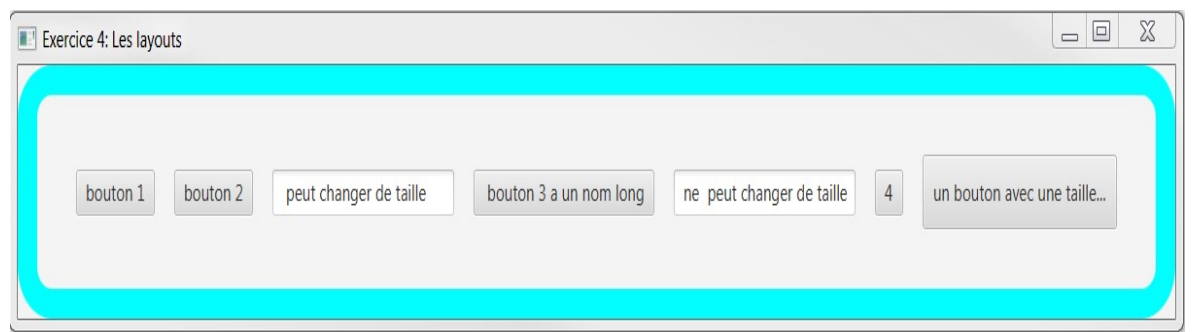
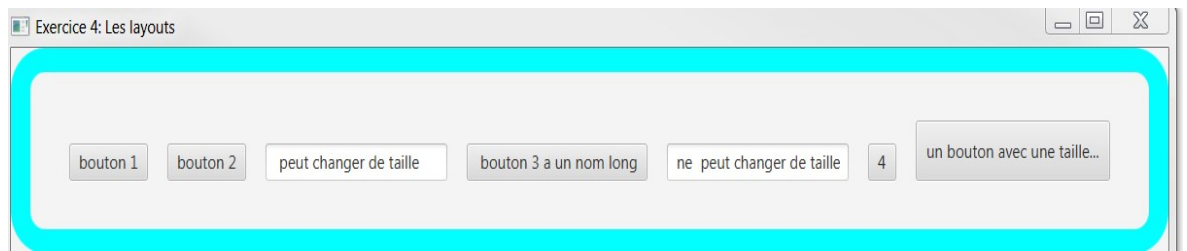
- Utilisez l'exemple du cours pour afficher la fenêtre suivante :



- Ajoutez une bordure au panneau avec des propriétés de votre choix. Voici un exemple :

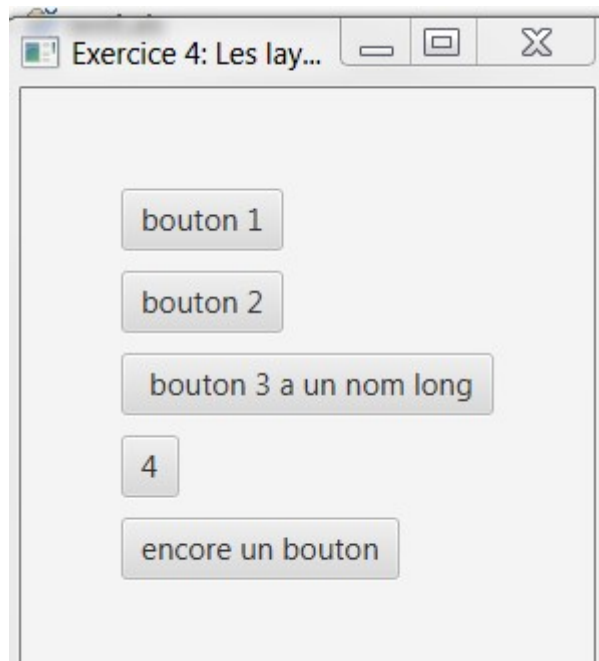


- Utilisez la méthode *setAlignment* de la classe *HBox* pour disposer les composants comme suit : Alignement des bas des composants puis alignement des centres.



Le VBox

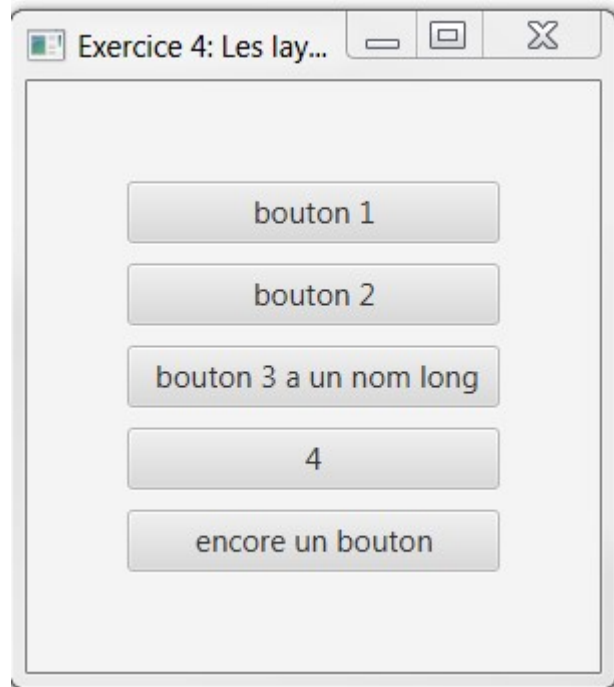
- En utilisant l'exemple du cours, remplacez le panneau racine (root) par un panneau VBox pour afficher ce qui suit :



La largeur du panneau est calculée en fonction du composant le plus large.

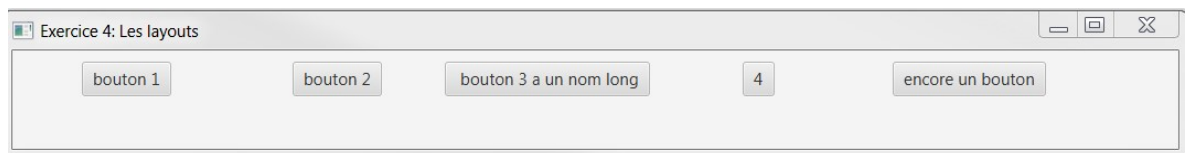
Pour avoir la même largeur pour tous les boutons, il suffit de spécifier une valeur. On pourrait prendre la valeur *Double.MaxValue*.

- Utilisez la méthode *setMaxWidth* sur tous les boutons. L'affichage doit ressembler à ceci :



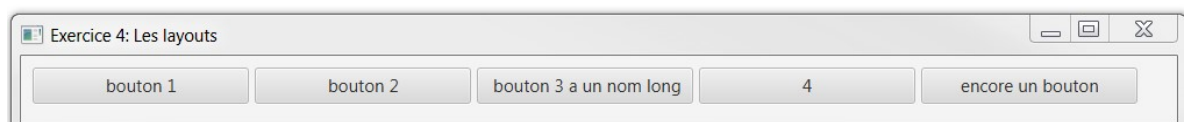
Le TilePane

- Utilisez l'exemple du cours, remplacez *root* par un panneau *TilePane* pour avoir ce qui suit :

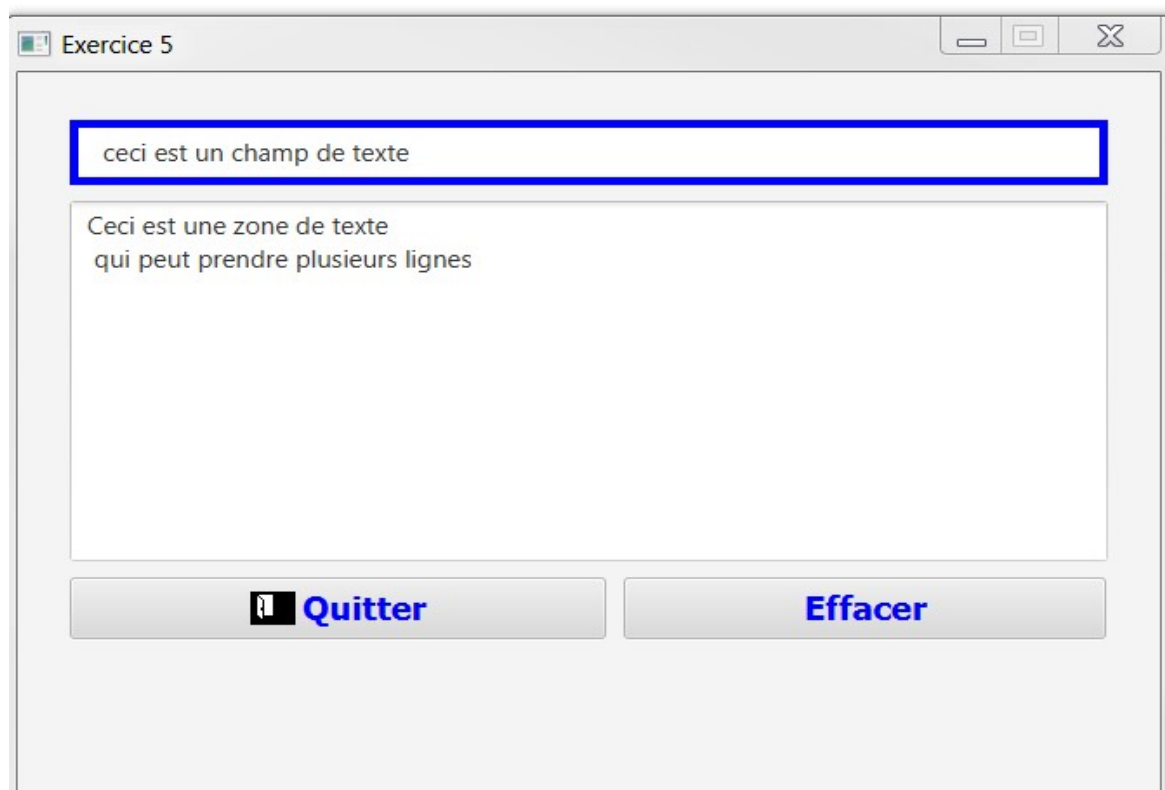


Dans ce cas toutes les cellules ont la même taille. Pour permettre aux boutons d'occuper tout l'espace d'une cellule, il suffit de fixer la taille maximum de tous les boutons en utilisant la méthode *setMaxSize* sur chaque bouton. Exemple : *btn1.setMaxSize(Double.MaxValue, Double.MaxValue);*

- L'affichage doit ressembler à ceci :



Exercice 5 : Les zones de texte (TextArea), les champs de texte (Textfield), les boutons (Button).



- Reproduisez l'interface graphique ci-dessus.
- Créez une classe *Exercice5*, sous-classe de *Application*
- Le titre de la fenêtre est *Exercice 5* et contient les composants suivants :
 - Un champ de texte (TextField)
 - Une zone de texte (TextArea)
 - Un bouton *Quitter*
 - Un bouton *Effacer*
- Le redimensionnement de la fenêtre principale n'est pas possible (méthode *setResizable*)
- Ajustez sa dimension au contenu de la scène (méthode *sizeToScene*).
- Le panneau racine est de type *FlowPane* avec une orientation verticale et un espace de 10 pixels entre les lignes.
- Les composants sont à une distance de 10 pixels (Haut, droite, bas et gauche) de la bordure du panneau. Utilisez la méthode *setPadding*.
- Le champ de texte doit contenir le texte : ceci est un champ de texte
- Ce champ n'est pas modifiable. Utilisez la méthode *setEditable*.
- Ajoutez au champ de texte une bordure bleue, rectangulaire, sans arrondis et d'épaisseur 5.
- Le texte de la zone de texte est sur deux lignes. Utilisez `\n` pour obtenir l'affichage demandé et comprenez la méthode *setWrapText*.

- Ajoutez une bulle d'aide contextuelle (texte de votre choix) sur la zone de texte. Utilisez la méthode *setTooltip*.
- Les deux boutons *Quitter* et *Effacer* sont disposés sur un *HBox*. Ils sont espacés de 10 pixels. Notez qu'ils occupent tout l'espace (Utilisez les méthodes *setHgrow* de *HBox* et les méthodes *setMaxWidth* sur les boutons)
- Créez un objet *Font* en utilisant la méthode *static font* de la classe *Font*. La police est de type « Tahoma », le style est gras et la taille est 20.
- Associez cette police à chacun des boutons.
- La couleur du texte des boutons est bleue.
- Ajoutez une icône pour le bouton *Quitter*. Utilisez la méthode *setGraphic*. N'oubliez pas d'ajouter l'image à votre dossier ressource.

Exercice 6 : Le texte (Text), Les boutons radio (RadioButton), les cases à cocher (CheckBox).

- Utilisez des panneaux de mise en page de votre choix pour reproduire l'interface graphique suivante :



- Le titre de la fenêtre est : Exercice 6
- Le redimensionnement de la fenêtre n'est pas possible. Utilisez la méthode *setResizable()*
- Ajustez la taille de la fenêtre à la scène. Utilisez la méthode *sizeToScene()*
- Le titre du panneau principal (formulaire) est : Interface Graphique Java, le texte est en gras, de taille 30, de type Serif et est centré. Utilisez la classe *Text*.
- Au centre du panneau, on trouve à gauche un groupe de trois boutons radio : rouge, vert et bleu. Un seul bouton peut être sélectionné à la fois. Utilisez la classe *ToggleGroup* pour le regroupement logique des trois boutons.
- Utilisez la méthode *setTogglegroup* sur chacun des boutons. Cette méthode est héritée de la classe *ToggleButton*. Initialement, le bouton radio rouge est

sélectionné. À droite, on a deux cases à cocher : gras et italique (gras est cochée initialement)

- Le texte des boutons radio et des cases à cocher est en gras, de taille 20 et de type Serif