Interfaces graphiques - JavaFX

responsable : Wiesław Zielonka
zielonka@liafa.univ-paris-diderot.fr
https://www.irif.univ-paris-diderot.fr/~zielonka

April 5, 2016

Habiller l'application avec CSS

Créer style sheet:

```
Scene scene = new Scene(root,650,800);
scene.getStyleSheets().add(``/path/stylesheet.css'');
```

path – le chemin vers le fichier css.

Exemple : le fichier css se trouve dans le répertoire `src/convertisseur/stylesheets/` dans le projet NetBeans alors

scene.getStyleSheets().add("/convertisseur/stylesheets/stylesheet.css");

Notez

- / au début du chemin et
- le fait que le nom de package de mon application se trouve dans le chemin.

Description de CSS pour javaFX

http://docs.oracle.com/javase/8/javafx/api/javafx/scene/doc-files/cssref.html

Habiller la racine:

```
.root{   -fx-font-size: 16pt;
         -fx-font-family: "Courier New";
         -fx-text-fill: #00FFAA;
}
```

Sélecteurs

```
.label{ -fx-font-size: 18pt;
        -fx-text-fill: #FF00FF;
        -fx-effect: dropshadow(gaussian,black,8, 0.0, 2, 0);
        }

.titled-pane{ -fx-text-fill: #FFFF11; }
```

- `label` et `titled-pane` – les noms de classes (sélecteurs),
- `-fx-font-size` – propriété
- `18pt` – la valeur de la propriété,
- `-fx-text-fill: #FF00FF;` – une règle.

Si le nom de la classe contient plusieurs mots, ces mots sont séparés par `-`.
Même chose pour le noms de propriétés.

Sélecteurs pour les pseudo-classes

```
.check-box{
    -fx-color: #00AA00;
    -fx-border-width: 3px;
    -fx-border-color: #000000;
    -fx-alignment: center;
}
.check-box:focused{
    -fx-color: #AA0000;
}
.check-box:selected{
    -fx-color: #FFFF00;
}
```

CheckBox change la couleur si elle est dans l'état "focused" ou dans l'état "selected".

`.check-box:selected` est une pseudo-classe.

Définir un nouveau style

```
.mylabel{
    -fx-text-fill: rgba(0,255,255,0.7);
    -fx-font-size: 18pt;
    -fx-font-family: "Cambria";
    -fx-background-color: rgb(200,200,200);
}
```

```
Label label = new Label(title);
label.getStyleClass().clear();
label.getStyleClass().add("mylabel");
```

Le style `mylabel` est utilisé uniquement avec ce label et non pas avec les autres.
Le style défini de cette façon s'ajoute au style de la classe Label. Si nous ne voulons pas que cela arrive il faut supprimer le style précédent avec `clear()`.

Styles pour Pane

Les différents Pane n'ont pas de style par défaut.
Par exemple GridPane ,'a pas de style par défaut. Dans ce cas il faut utiliser la méthode suivante :
Dans le fichier css :

```
.grid-pane{
    -fx-hgap: 5mm;
}
```

et dans le programme :

```
GridPane lowGrid = new GridPane();
GridPane highGrid = new GridPane();
lowGrid.getStyleClass().add("grid-pane");
highGrid.getStyleClass().add("grid-pane");
```

Création de style par ID

Dans le fichier css :

```
#font-button{
  -fx-font: bold italic 20pt "Arial";
}
```

Le selector qui commence par # est in ID selector.

```
Button button = new Button("Font");
button.setID("font-button")
```

Seulement les selectors avec ID `font-button` utilise le style.

Spécifier le style directement dans le code (sans fichier css)

```
Button button = new Button("Font");
button.setStyle("-fx-background-color: slateBlue; "+
                "-fx-text-fill: white");
```

JavaFX 3D

La classe `Shape3D` avec 4 classes dérivées :

- `Sphere`
- `Cylinder`
- `Box`
- `MeshView`

Les propriétés de Shape3D

- `Material` – pour contrôler l'interaction avec la lumière, réalisation concrète `PhongMaterial`,
- `DrawMode` – deux valeurs possibles: `DrawMode.LINE` et `DrawMode.FILL`
- `CullFace` – les valeurs possibles : `CullFace.NONE`, `CullFace.BACK`, `CullFace.FRONT`

Camera

Deux types de caméra : `ParallelCamera` et `PerspectiveCamera`.

```
PerspectiveCamera camera = new PerspectiveCamera(true);
//eloigner le camera pour voir l'objet
camera.setTranslateZ(-1000);
//objets plus proches que nearClip deviennent invisibles
camera.setNearClip(0.1);
//objets plus loins que farClipp sont invisibles
camera.setFarClip(5000.0);
//l'angle de projection en degrees (30 par defaut)
camera.setFieldOfView(50);
//mettre le camera sur la scene
scene.setCamera(camera);
```

Éloigner/rapprocher le caméra

Handler pour éloigner ou rapprocher le caméra avec la molette de souris :

```
root.addEventHandler(ScrollEvent.ANY,
    new EventHandler<ScrollEvent>() {
    @Override
    public void handle(ScrollEvent event) {
        camera.setTranslateZ(camera.getTranslateZ()
        + event.getDeltaY());
    }
});
```