R2.02_Prog_TP6
Prototypage haute-fidélité et conception avec Scene Builder (2/2)

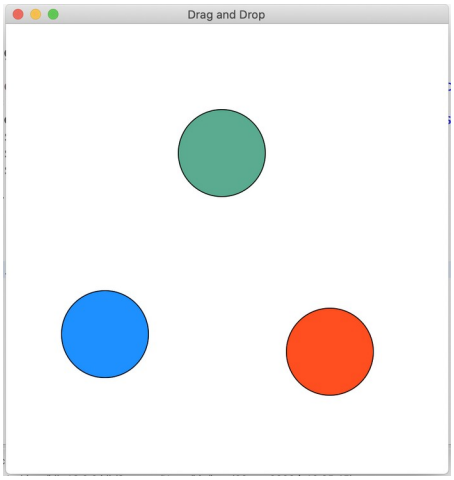
Objectifs :

- Utilisation de Scene Builder pour la programmation déclarative (suite)
- Manipuler des formes géométriques (création, positionnement,...)
- **Se familiariser avec la technique du « Drag and Drop »**

Exercice 1 : Drag and Drop avec Scene Builder

Dans cet exercice votre but sera de déplacer des formes géométriques à l'aide de votre souris via la technique du « Drag and Drop » (voir vidéo R2.02_Prog_TP6_Exo1.mov).

La figure suivante vous fait voir 3 formes qui devront être déplacées suivant les mouvements de votre souris. (Vous choisissez en fait les formes que vous voulez ! C'est juste un exemple).
Ces formes seront créées par Scene Builder dans un Anchor Pane (de préférence).



Indices :

- Pour chaque forme, vous aurez deux évènements à gérer : « MousePressed » et « MouseDragged » **Essayer de faire en sorte qu'il n'y est que DEUX fonctions génériques pour toutes les formes.**
- « MousePressed » détecte si une forme a été « cliquée » par l'utilisateur :
  - Détection/enregistrement des coordonnées X et Y du point « cliqué » (le point de départ)
  - Définit les coordonnées X et Y de la translation ajoutée à la transformation de la forme cliquée (en gros on ajoute une translation qui suivra les mouvements de la souris).
  - Exemple de code à intégrer :

```
PointCliqueX = evenement.getSceneX();
PointCliqueY = evenement.getSceneY();
orgTranslateX = ((Circle)( evenement.getSource())).getTranslateX();
orgTranslateY = ((Circle)( evenement.getSource())).getTranslateY();
```

- « MouseDragged » détecte la fin de la translation :
  - Détection/enregistrement des coordonnées X et Y du point « relâché » final
  - Calcul des nouvelles coordonnées à ajouter à la translation finale
  - Exemple de code :

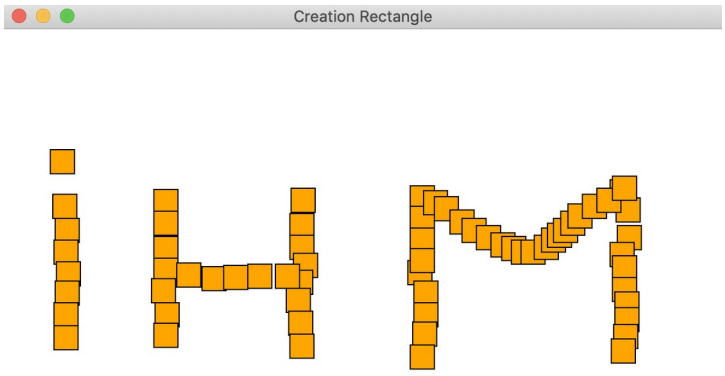
```
double offsetX = evenement.getSceneX() - PointCliqueX;
double offsetY = evenement.getSceneY() - PointCliqueY;
double newTranslateX = orgTranslateX + offsetX;
double newTranslateY = orgTranslateY + offsetY;

((Circle)(evenement.getSource())).setTranslateX(newTranslateX);
((Circle)(evenement.getSource())).setTranslateY(newTranslateY);
```

Exercice 2 : Création de figures par pointage avec la souris avec Scene Builder

Dans cet exercice, vous devrez créer des carrés (ou d'autres formes) par un clic de souris (bouton gauche) tel que présenté dans la vidéo R2.02_Prog_TP6_Exo2.mov.
Vous utiliserez le conteneur « Anchor Pane » comme à l'exercice précédent.
Chaque clic de souris permettra de créer un rectangle « orange », cerclé de « noir » de cette manière : AJOUTER CLIC DROIT POUR CHANGER LA COULEUR DE CREATION et en CSS

```
Rect(event.getSceneX(), event.getSceneY(), 20, 20)
```

ATTENTION: la couleur « orange » et le cerclage « noir » du rectangle seront gérées dans une feuille de style (donc la partie « vue »)



Vous devrez certainement utiliser une fonction spécifique liée au « AnchorPane » qui ressemble à celle-là à chaque clic de souris sur AnchorPane pour la création de rectangle : (à vous de choisir le type d'évènement)

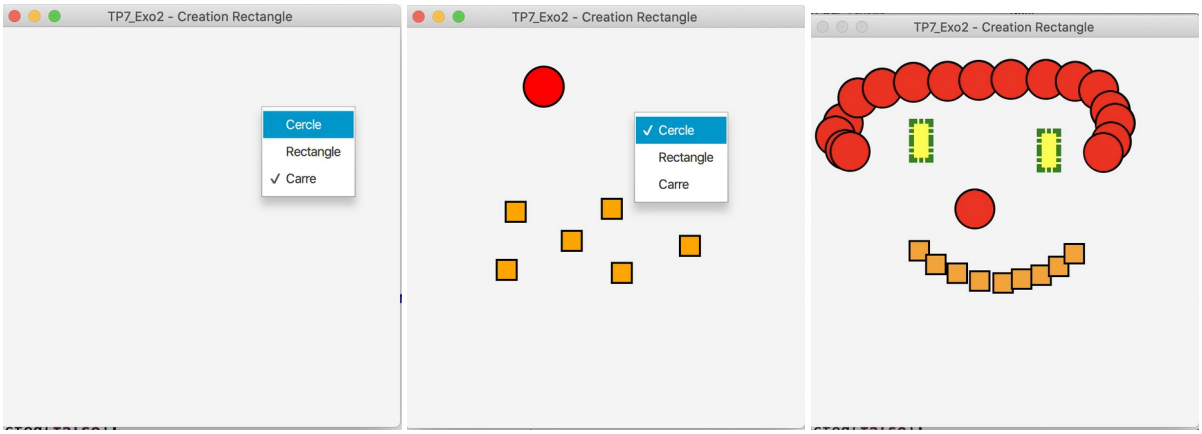
```
@FXML
void addShape(……) {
        // Créer le rectangle ici
        // Ajouter au AnchorPane        }
```

Exercice 3 : Plus loin avec plus de figures (plus chaud aussi !)

Continuité de l'exercice 2 mais avec une possibilité de choisir la figure que vous voulez créer par un clic de souris (limitez-vous au cercle, rectangle et carré).

Vous ferez intervenir un menu contextuel, notion qui a déjà été abordée dans l'exercice 3 du TP2, avec des Checkbox pour choisir la figure souhaitée.

Vous devez vous rapprocher le plus possible de la vidéo R2.02_Prog_TP6_Exo3.mov et des aperçus suivants :



ATTENTION:

- les différentes couleurs et cerclage doivent être dans un fichier CSS (partie « vue »)
- Pour vous faciliter la vie vous pouvez ne pas tenir compte de la partie « model » et déposer un max de code dans la partie « controleur »
- Pour vous compliquer la vie, tenez compte de la partie « model » et faites en sorte que les figures n'apparaissent pas dans la partie « controleur » mais dans la partie « model »