Université Paris Diderot INTERFACES GRAPHIQUES 18 mai 2017
12h00-14h00

Les seuls supports autorisés :
documentation API java `http://docs.oracle.com/javase/8/docs/api/index.html`,
documentation API JavaFX : `http://docs.oracle.com/javase/8/javafx/api/toc.htm`
les tutoriels de javaFX d'Oracle : `http://docs.oracle.com/javase/8/javase-clienttechnologies.htm`
et les transparents et programmes étudiés en cours :
`https://www.irif.univ-paris-diderot.fr/~zielonka/Enseignement/InterfaceGraphique/2017/HTML/index.html`
L'utilisation de toute autre ressource web (en particulier des forums de développeurs) ou de mail sera traitée comme une fraude. Vos mobiles doivent être impérativement éteints. L'écran de votre voisin n'est pas une ressource autorisée.
Le sujet comporte 3 pages.

Consignes

Votre classe principale doit porter votre nom, par exemple pour un dénommé Mathias Dupont la classe principale s'appellera `DupontMathias`.
Dans votre programme mettez en commentaire l'information quels exercices vous avez implémentés.
Il est possible de déposer plusieurs versions numérotées (DupontMathias1, DupontMathias2 etc.). Cependant c'est la version avec le numéro le plus élevé qui sera examinée pour noter votre travail. Si pour une raison quelconque vous voulez que j'examine aussi une autre version (par exemple une version répond à l'exercice i et une autre répond à l'exercice k) faites un fichier README avec un message appropriés.

Le fichier source du programme doit être déposé sur didel, cours Interfaces graphiques M1 2016-2017, Travaux⟶Examen 18 mai G1.
N'attendez pas la fin de l'examen pour recopier vos solutions, le répertoire et didel ferment à 12h00 pile.

Vos programmes seront examinés par un logiciel de détection de plagiat efficace.
Le dépôt sur didel est protégé donc c'est moi et vous-même qui auront l'accès à votre programme et personne d'autre.
C'est à vous de protéger votre répertoire de travail sur les machines UFR contre les visites non autorisées.

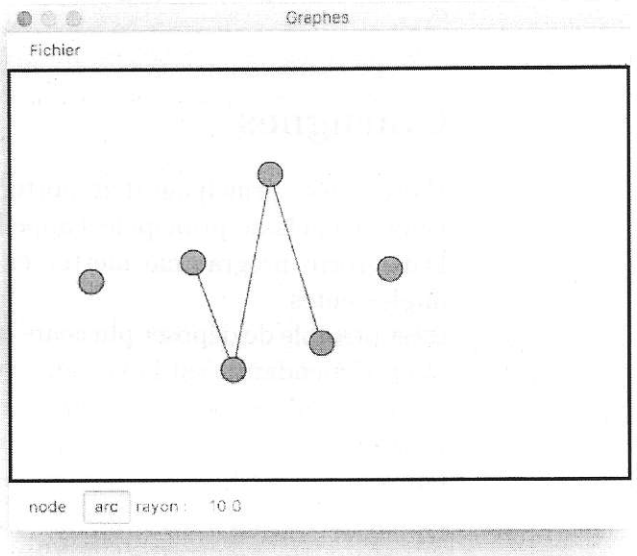
Exercices

Dans tous les exercices, si vous avez à implémenter un interface avec une seule méthode vous devez utiliser plutôt une lambda expression. L'utilisation de classes anonymes (ou non anonymes) là où une lambda expression est suffisante entraînera la diminution de la note.

Exercice 1 Écrire une application dont le titre est votre nom et qui comporte initialement :

- un Pane entouré d'un bord,
- deux ToggleButtons : `node` et `arc`,
- un label avec le texte `rayon :`,
- un TextField.

Tous les éléments sauf Pane se trouvent dans un ToolBar situé au-dessous du Pane.
Les deux ToggleButtons se trouvent dans un ToogleGroup de telle sorte qu'un seul ToggleButton est sélectionné à chaque moment (initialement le ToggleButton node).
Quand le ToggleButon node est sélectionné et l'utilisateur clique sur le Pane un cercle rouge apparaît. Le centre du cercle se trouve sur la position de la souris au moment du clic. Le bord du cercle aura la couleur noire. Le rayon du cercle sera égal à la valeur écrit dans TextField. (TextField doit contenir une valeur supérieure à 0, sinon le cercle ne sera pas construit.)



Indications. Pour faire un bord :

```
pane.setBorder(new Border(new BorderStroke(Color.BLACK, BorderStrokeStyle.SOLID,
        CornerRadii.EMPTY, new BorderWidths(3))));
```

Exercice 2 Quand le ToogleButton `node` est sélectionné le mouvement « drag » appliqué à un cercle provoque le déplacement du cercle suivant le mouvement de la souris (à chaque moment le centre du cercle doit coïncider avec la position de la souris). N'oubliez pas de consommer l'évènement de la souris (le mouvement drag ne doit pas provoquer l'apparition d'un nouveau cercle mais uniquement le déplacement d'un cercle existant).

Exercice 3 Ajouter un menu Fichier avec l'option Quitter qui permet de terminer l'exécution de l'application.

Exercice 4 Quand le ToggleButton `arc` est activé alors l'utilisateur pourra ajouter un segment de ligne entre deux nodes.
Quand l'utilisateur clique sur deux cercles différents et ToggleButton arc est sélectionné alors vous devez construire et ajouter dans le Pane une ligne (`Line`) entre les centres de deux cercles. Nous appelons une telle ligne « arc ».
Les lignes doivent toujours passer au-dessous de cercles (le cercle couvre la partie de la ligne entre son centre et son bord).
Dans la suite les deux cercles utilisés pour construire un arc sont appelés les sommets de l'arc.

Exercice 5 Pour compléter l'exercice précédent on demande que, quand l'utilisateur clique sur le premier cercle et le ToggleButton arc est activé, alors ce cercle changera l'aspect visuel pour indiquer qu'il devient un sommet d'un arc. Ce changement doit être implémenté par un `Effect`, par exemple en ajoutant l'effet DropShadow pour ce cercle.
Ensuite quand l'utilisateur clique sur un deuxième cercle (il choisit le deuxième sommet de l'arc) alors on construit le segment comme indiqué dans l'exercice précédent et on supprime l'effet visuel du premier cercle.

Exercice 6 Ajouter la fonctionnalité suivante : quand l'utilisateur déplace un cercle (le mouvement drag et le ToggleButton node activé) alors pour tous les arcs pour lesquels ce cercle constitue un sommet doivent changer soit l'extrémité **start** soit l'extrémité **end** correspondant pour que cette extrémité reste au centre du cercle.
Donc à tout moment chaque extrémité d'un arc doit être positionné au centre d'un cercle et le déplacement d'un cercle à l'aide de la souris provoquera automatiquement le déplacement d'une extrémité de chaque arc attaché à ce cercle.
Cette fonctionnalité doit être implémentée à l'aide de bind ou de bind bidirectionnel entre les propriétés de l'arc et les propriétés de deux sommets de l'arc. Une implémentation à l'aide de ChangeListener sera nettement moins bien notée.

Exercice 7 Ajouter un ToggleButton `supprimer`. Si ce ToggleButton est sélectionné alors un clic sur un arc provoque la suppression de cet arc.
Un clic sur un sommets provoque la suppression de ce sommet mais uniquement s'il n'y a plus aucun arc dans le Pane.