

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

<https://docs.oracle.com/javase/8/javase-books.htm>

et les transparents et programmes étudiés en cours :

<https://moodlesupd.script.univ-paris-diderot.fr>

**L'utilisation de toute autre ressource web (en particulier des forums de développeurs), de moteur de recherche (google et autre), de mail ou d'autre moyens de communication sera traitée comme une fraude. Les documents papier autorisés : vos notes de cours ou TP, les photocopies de transparents de cours, notes personnelles. Livres et photocopies de livres interdits. Vos mobiles doivent être impérativement éteints. L'écran de votre voisin n'est pas une ressource autorisée.**

**Le sujet comporte 5 pages.**

---

## Consignes

Votre classe principale doit porter votre nom, par exemple pour un dénommé Mathias Dupont la classe principale s'appellera `DupontMathias`.

Chaque exercice est un programme à développer découpé en plusieurs questions. Parfois les questions sont indépendantes, c'est-à-dire on peut faire la question  $i + 1$  même si on a pas fait la question  $i$ .

Dans votre programme mettez en commentaire l'information quelles questions vous avez implémentées.

Il est possible de déposer plusieurs versions numérotées (`DupontMathiasExo1a`, `DupontMathiasExo1b` etc.) du même exercice. Cependant c'est la dernière version qui sera examinée pour noter votre travail. Si pour une raison quelconque vous voulez que j'examine aussi d'autres versions faites un fichier README avec un message approprié.

Si vous développez sur NetBeans le plus simple c'est exporter le programme vers zip (File → Export Projet → to Zip) et déposer le fichier zip avec le projet complet. Sinon vous pouvez aussi faire zip ou tar contenant juste les fichiers sources.

Le fichier archive doit être déposé sur moodle, cours Interfaces graphiques, dans dépôt d'examen groupe 1.

N'attendez pas la fin de l'examen pour déposer vos programme, le dépôt sur moodle ferme à 11h30 pile.

**Vos programmes seront examinés par un logiciel de détection de plagiat efficace.** Le dépôt sur moodle 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.

**Dans tous les exercices et question, 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.**

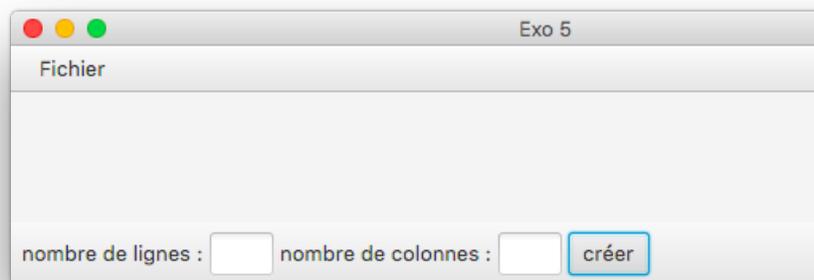
---

## Exercice 1

**Question 1:** Écrire une application qui comporte initialement un menu en haut avec juste un seule item qui permet de quitter l'application, un Pane vide au centre un **ToolBar** en bas. Dans le **ToolBar** il y a les éléments comme sur cet image :

L'utilisateur écrit dans les deux **TextFields** deux nombres entiers positifs (nombre de lignes et nombres de colonnes) et clique sur le bouton **créer**. Si les deux nombres sont positifs alors votre programme les mémorise (dans des variables) et désactive (disable) les deux **TextFields** ainsi que le bouton pour que l'utilisateur ne puisse plus les utiliser.

Sinon le programme vide le deux **TextField** et on recommence.



**Question 2:** Ajouter le comportement suivant. Quand l'utilisateur clique sur le bouton **créer** et les nombres de lignes `nbLignes` et colonnes `nbColonnes` sont positifs on construit le nombre `nbLignes * nbColonnes` de rectangles et on les positionner de façon suivante :



On note que les rectangles sont des carrés : (`largeur = hauteur = L`,  $L$  doit être déclaré comme une constante dans votre programme, par exemple  $L = 25$  pixels).

Les rectangles sont positionnés en `nbLignes` et `nbColonnes`.

Il est commode d'utiliser une classe dérivée de `Rectangle` que vous trouverez sur moodle et qui permet de mémoriser dans chaque rectangle son numéro de la ligne et le numéro de la colonne. Les numéros commencent par 0, donc le premier rectangle (en haut à gauche) est le rectangle  $(0, 0)$ , le suivant dans la même ligne est  $(0, 1)$  etc.

Tous les rectangles ont le bord blanc, la largeur de bord est fixe (chez moi 2 pixels). Le rectangle  $(0, 0)$  est rouge à l'intérieur (c'est un rectangle empoisonné), tous les autres rectangles ont l'intérieur noir.

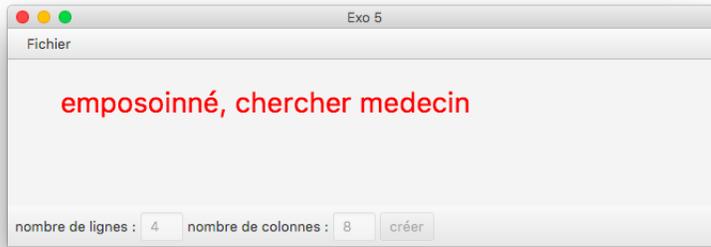
Cet agencement de rectangles représente une tablette de chocolat noir divisée en carré.

Les rectangles sont positionnés de façon absolue dans le `Pane`. Pour cela il suffit de spécifier correctement les valeurs de propriétés `x` et `y` pour chaque rectangle.

Afficher le chocolat.

**Indication.** Il peut être utile de mémoriser les rectangles dans un tableau à deux dimensions `rect` de telle sorte que `rect[i][j]` contient la références vers le rectangles  $(i, j)$ . Mais il est aussi possible de stocker les rectangles n'importe où (une liste, tableau à une dimension) ou même les mettre juste dans le `Pane` et ne pas stocker ailleurs. Ce qui est important c'est que chaque rectangle contient  $(i, j)$  - sa position dans la tablette de chocolat, voir la classe `MyRectangle` sur moodle.

**Question 3:** Quand l'utilisateur clique sur le carré rouge  $(0, 0)$  alors il faut enlever tous les rectangles et afficher le texte (taille au moins 25 de couleur rouge) :



**Question 4:** Quand l'utilisateur clique sur un carré différent de  $(0,0)$  (un carré noir) alors il prend et mange tous les carrés  $(k,l)$  avec  $k \geq i$  et  $l \geq j$ . Les carrés mangés sont supprimés de la tablette.

La tablette après un coup :



et après le deuxième coup :



Chaque coup enlève plusieurs carrées.

**Question 5:** Ajouter dans le menu un item **rejouer** dont l'activation supprime tout le contenu du Pane et débloque les TextFields et le bouton dans le ToolBar. Cela permet de recréer une nouvelle tablette de chocolat.

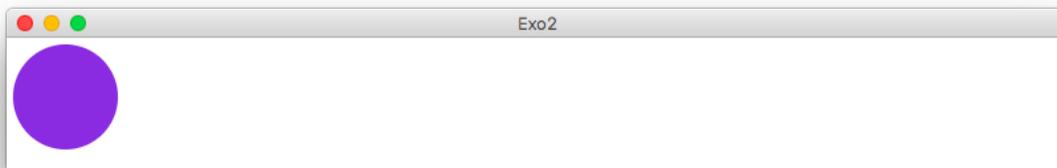
**Exercice 2** Cet exercice sert à tester vos connaissances de bindings. Si vous le faites sans bind alors cela vous apportera **très très peu de points**.

**Question 1:** Placer dans un Pane un cercle. Le rayon du cercle doit être égal à

$$( \text{ la largeur de Pane} - 10 ) \text{ divisé par } 20$$

Le rayon (radius) doit automatiquement s'ajuster à la taille du Pane, si l'utilisateur agrandit la fenêtre le rayon augmente, si l'utilisateur rétrécit la fenêtre le rayon diminue, sa valeur est à chaque moment donnée par la formule ci-dessus.

La position du centre du cercle change aussi avec la taille de la fenêtre. Les propriétés `centerX` et `centerY` du cercle doivent automatiquement s'ajuster de telle sorte que le bord gauche du cercle est à une distance fixe DX du bord de la fenêtre et le bord haut reste à une distance fixe DY du bord nord de la fenêtre (DX et DY constantes de votre choix). Donc le cercle restera coller au nord et à gauche de la fenêtre même si celle-ci change la taille.



**Question 2:** Ajouter d'autres cercles (au total il en faut 10). Les couleurs de cercles de votre choix mais doivent alterner (un rouge, un noir, un rouge etc. par exemple). Le rayon de chaque cercle doit à chaque moment être égal au rayon du premier cercle. Les cercles doivent rester collés comme sur l'image quand on change la largeur de la fenêtre.

