

TD 2 – JAVA FX : ECOUTEURS ET EVENEMENTS

Durée : 2h00.

Matériel nécessaire : aucun.

Objectif : Comprendre la gestion événementielle avec Java FX.

Pré-requis : Eclipse, Java 8 et JavaFX disponibles sur votre poste (optionnel la javadoc).

EXERCICE UN – EVENEMENTS : A partir de l'interface mise en œuvre la fois précédente vous allez réaliser la partie événementielle en appliquant les règles ci-dessous :

Lorsque l'utilisateur clique sur le bouton valider, il faudra **mettre à jour le libellé en bas de page** à partir des saisies de l'utilisateur :

1. il faudra **recupérer le login saisi**. S'il est vide, le libellé de bas de page contiendra la phrase suivante « aucun login saisi ».
2. de plus, il faudra **ajouter l'application sélectionnée dans la ComboBox**. Vous noterez que par défaut, c'est l'application comptabilité qui sera sélectionnée, mais elle ne sera affichée dans le libellé uniquement si on clique sur le bouton valider.
3. Bonus : on pourra spécifier la date et l'heure de la validation de la saisie.

Application login

Veuillez saisir vos informations de login

utilisateur: username

mot de passe: ***

application: Comptabilité

Valider

login de <xxxx> pour application <xxxx> le xx/xx/xxxx à hh:mm:ss

Pour réaliser cela, vous devrez donc ajouter **des écouteurs d'évènement sur le bouton valider, sous forme de classe anonyme** :

- a. rattachez au bouton un EventFilter pour écouter les événements de type MouseEvent
- b. rattachez au bouton un autre écouteur, cette fois-ci de type EventHandler
- c. rattachez au bouton un 3ème écouteur via la "convenience method" (méthode utilitaire) : *setOnMouseClicked*.

Pour chacun de ces écouteurs, loggez dans la console l'évènement et observez l'ordre de leur déclenchement. Pour la suite, on conservera la dernière manière (la "convenience method").

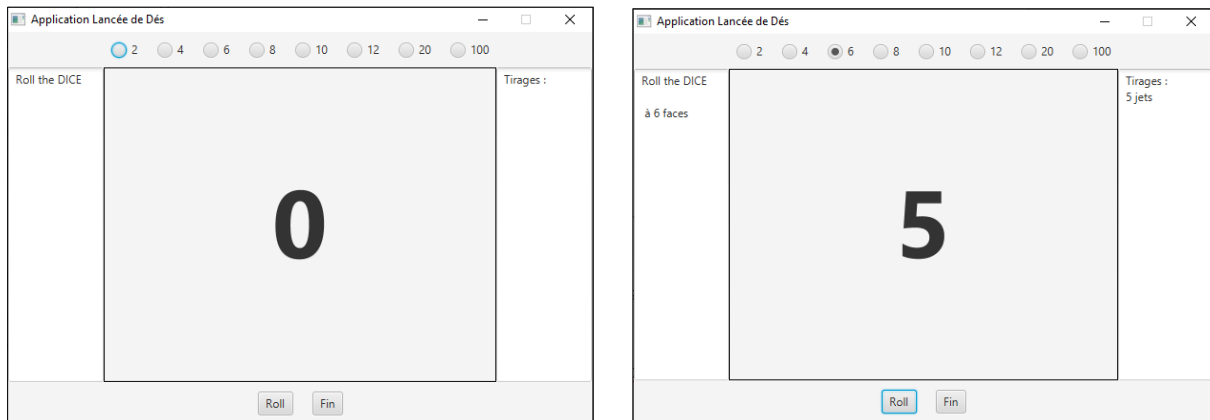
Il nous reste à présent à **gérer les erreurs** : au cas où le champ utilisateur ou mot de passe serait vide, vous devrez rajouter un libellé dans la partie gauche de la fenêtre pour faire office de message d'erreur. Pour rendre l'interface uniforme, ce libellé devra occuper une place minimum de 250 pixels même s'il n'y a pas d'erreur affichée. Pour cela, il faudra donc augmenter la largeur totale de votre interface et le positionner à 600 pixels. Le message d'erreur en rouge et en gras contiendra le texte suivant : « la saisie d'un xxxxxx est obligatoire ».

DEUXIEME PARTIE : Un programme pour rôlistes

Vous allez réaliser un programme permettant de gérer des lancers de dés.

L'utilisateur choisira le type de dés et décidera de lancer le dé. Le résultat sera ensuite affiché.

Sur la base de la maquette ci-dessous, réfléchissez à la présentation puis à la gestion des événements nécessaires à la réalisation de ce programme.



La liste des radios boutons est gérée dans un ToggleGroup.

Les zones d'affichage situés à gauche et à droite sont des TextArea.

La zone centrale est un Label ayant des bordures

A chaque fois que l'on sélectionnera un bouton radio : on effacera le libellé central et on mettra à jour la zone de texte de gauche à partir de cette sélection.

Le lancer de dés est simulé par l'appel à une méthode privée basée sur `Math.random()` qui permet de générer un nombre aléatoire. Celle-ci devra prendre en compte le type de dé sélectionné.

On pourra effectuer jusqu'à 10 lancers de dés : au bout de 10 clicks, le bouton "Roll" devra être désactivé.

Dans cet exercice, les écouteurs d'évènements devront être des classes internes (ou classes imbriquées).

Quelles sont les autres manières d'attacher des écouteurs ?

Bonus - gérer des statistiques : on pourra rajouter un troisième bouton en bas de l'interface, libellé "Stats". Celui-ci permettrait d'afficher les statistiques (la moyenne par type de dés par exemple).