Programmation orientée objet
TP9 : Gestion de boutons

Romain Marie & Hervé Gaudin

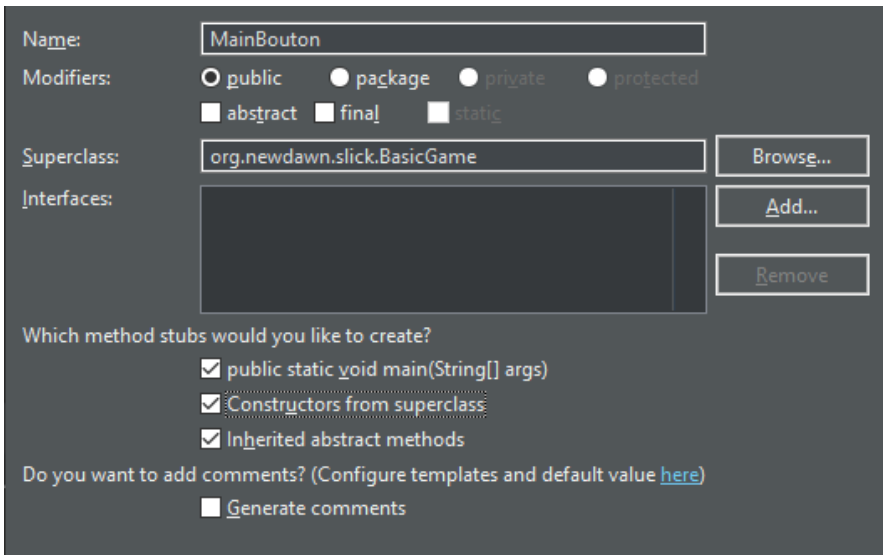
Novembre 2019

3iLINGENIEURS
</les sciences informatiques>

Dans ce TP, vous allez découvrir une fonctionnalité relativement avancée de Slick 2D : la création et l'utilisation de boutons, c'est à dire d'éléments graphiques prévus pour réagir au passage ou au clic de la souris.

1 Mise en place du projet

1. Ouvrez Eclipse en prenant bien soin de choisir votre workspace configuré pour utiliser Slick2D.
2. Créez un nouveau projet que vous nommerez "ProjetBoutons".
3. Configurez ce projet pour qu'il utilise Slick2D.
4. Ajoutez à votre projet une classe **MainBouton** qui hérite de **BasicGame**. Cette classe inclura une méthode *main()*, le constructeur faisant appel à celui de la classe mère, et les méthodes abstraites. Pour vous aider, la configuration de la classe vous est présentée ci-dessous :



5. Ecrivez le contenu de la méthode *main()* pour qu'elle initialise et démarre l'application Slick. Il faut pour cela écrire le code source qui vous est rappelé ci-dessous :

```
public static void main(String[] args) throws SlickException {
    AppGameContainer app = new AppGameContainer(new MainBouton("MainBouton"));
    app.setShowFPS(false);
    app.start();
}
```

6. Exécutez votre projet, et vérifiez qu'une fenêtre vide sur fond noir s'ouvre. Si ce n'est pas le cas, appelez votre enseignant.

2 Création d'un bouton

1. Commencez par changer la couleur de fond de la fenêtre, en la choissant blanche. Vous utiliserez pour cela *g.setBackground(Color.white)* dans la méthode *render()*

2. Ajoutez une propriété privée **bouton** de type **MouseOverArea**.

3. Dans la méthode *init()*, créez une variable **img** de type Image, que vous instancierez grâce au fichier "btn1.png" fourni avec l'énoncé.

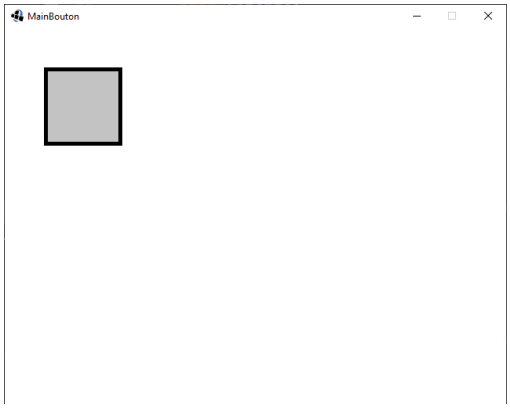
4. Toujours dans la méthode *init()*, instanciez votre bouton grâce au constructeur à 6 paramètres répartis comme suit :

   - gc : le GameContainer fourni en paramètre de la méthode *init()*
   - img : l'Image que vous venez juste de créer, et qui représentera l'aspect visuel de votre bouton
   - x : La coordonnée x du coin supérieur gauche du bouton dans la fenêtre
   - y : même chose mais pour la coordonnée y
   - l : la largeur de votre bouton
   - h : la hauteur de votre bouton.

   En respectant cet ordre de paramètres, instanciez **bouton** à partir de l'image créée ci-dessus, en le placant aux coordonnées (50,50), avec une dimensions de 100x100.

5. Dans la méthode *render()*, faites en sorte que votre **bouton** se dessine. Vous ferez pour cela appel à sa méthode *render(gc,g)*.

6. En exécutant votre programme, vous devriez obtenir une fenêtre blanche avec un bouton gris dans le coin supérieur gauche. Si ce n'est pas le cas, appelez votre enseignant.



3 Réagir au clic

Bien sûr, un bouton n'a de sens que s'il est possible d'effectuer des actions lorsqu'on clique dessus. Pour obtenir ce résultat, plusieurs étapes sont nécessaires. Il faut :

- Faire en sorte que votre classe soit capable de réagir à des boutons.
- Indiquer au bouton que c'est votre classe qui est chargée de réagir au clic
- Ecrire les instructions à exécuter lorsque le bouton est cliqué

1. Modifiez votre classe **MainBouton** pour qu'elle implémente l'interface **ComponentListener**. Votre classe sera ainsi capable de réagir à des boutons, grâce à la méthode *componentActivated()* que vous devez rajouter.
2. Dans la méthode *init()*, Ajoutez **this** comme dernier paramètre au constructeur de votre **bouton**. Cette action indiquera à votre bouton que c'est votre classe qui est chargé de gérer la réaction au clic.
3. Dans la méthode *componentActivated()*, copiez-coller le contenu ci-dessous :

```
@Override
public void componentActivated(AbstractComponent source) {
    if(source==bouton)
        System.out.println("Bouton cliqué !");
}
```

4. En exécutant votre programme, vous devriez constater qu'un message s'affiche dans votre console à chaque fois que votre bouton est cliqué. Si ce n'est pas le cas, appelez votre enseignant !

4 Animer le bouton

En l'état, votre bouton, bien qu'opérationnel, est un peu inerte. Pour animer tout ça, l'idée est d'associer votre bouton à 3 images différentes :

- L'image par défaut, qui s'affiche lorsque le bouton n'est pas sollicité,
- Une image lorsque la souris passe au dessus du bouton
- Une image lorsque le bouton est cliqué.

1. Dans la méthode *init()*, ajoutez deux nouvelles variables de type Image, que vous instancierez grâce aux fichiers "btn2.png" et "btn3.png" fournis avec l'énoncé.
2. Toujours dans la méthode *init()*, utilisez les méthodes *setMouseOverImage()* et *setMouseDownImage()* de votre bouton, pour que l'image 2 soit utilisée lorsque la souris passe au dessus du bouton, tandis que l'image 3 sera utilisée lorsque le bouton est cliqué.

5 Pour aller plus loin

1. Adaptez votre programme, de sorte que, lorsque le bouton est cliqué, le fond de la fenêtre change de couleur (la nouvelle couleur devra être aléatoire).
2. Ajoutez un deuxième bouton à votre programme, qui aura les mêmes propriétés graphiques que le premier. Faites en sorte que, lorsque ce second bouton est cliqué, le premier se téléporte (vous pourrez utiliser la méthode *setLocation()*).
3. Dans la méthode *update()*, faites en sorte que, à partir du moment où il est cliqué, le deuxième bouton suive la souris jusqu'à ce que le clic soit relaché. Vous pourrez vous aider de la méthode *isMouseOver()*.