

Programmation orientée objet

TD12 : Un peu d'interfaces

Romain Marie & Hervé Gaudin

Novembre 2019

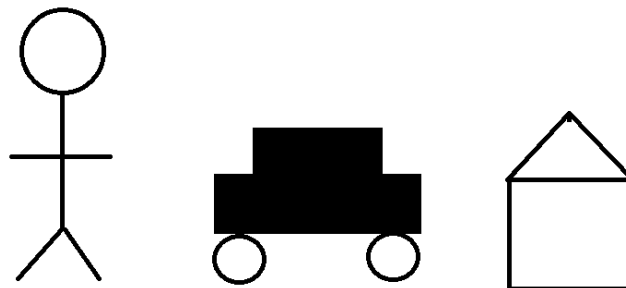


Dans ce TD, vous allez vous familiariser avec les interfaces. Toujours sur Slick2D, un code de base vous est cette fois fourni (voir verso). Ce code prévoit l'affichage et la mise à jour des éléments d'une ArrayList de **Slickable**, une interface que j'ai créé pour l'occasion.

Le but du TD va donc être de définir ces éléments, c'est à dire de créer des classes qui implémentent l'interface **Slickable**. Le résultat sera un programme qui dessinera et mettra à jour vos objets, alors que le code qui vous est fourni a été réalisé sans savoir quels objets vous alliez réaliser.

1. A partir du code qui vous est fourni, écrivez l'interface **Slickable**. Vous devrez en particulier déduire quelles méthodes abstraites elle contient.
2. proposez un premier type d'objet (une classe) implémentant l'interface **Slickable**. Vous avez libre choix sur son contenu, à condition de respecter les contraintes suivantes :
 - La position et la taille de l'objet doivent être configurable à sa construction,
 - L'objet devra changer de couleur toutes les secondes (nouvelle couleur générée aléatoirement à chaque fois),
 - L'objet sera composé d'au moins 3 éléments géométriques (cercles, rectangles, lignes).

Pour ceux qui n'auraient pas d'inspiration, voici quelques exemples sur lesquels vous appuyer (je préfère que vous laissiez parler votre imagination !)



3. Complétez la méthode init du code fourni pour que 10 de vos objets soient ajoutés à l'ArrayList. Leurs tailles et leurs coordonnées seront aléatoires, dans les limites du raisonnable (que vous fixerez).

4. Proposez un nouveau type d'objets à afficher dans votre application. Les contraintes sont cette fois les suivantes :
 - La position, la couleur et la taille de l'objet doivent être configurables à sa construction,
 - L'objet devra se téléporter toutes les 2 secondes
 - L'objet sera composé d'au moins 3 éléments géométriques (cercles, rectangles, lignes).
5. Finalisez la méthode init pour que 10 de ces objets soient pris en compte par votre programme.

1 Pour aller plus loin

On souhaiterait maintenant faire en sorte qu'une seconde interface **Deplacable** permette de prévoir le déplacement des objets qui l'implémentent dans la fenêtre Slick, lorsque les flèches gauche, droite, haut ou bas sont pressées.

1. Proposez le contenu de cette interface
2. Faites en sorte que la première de vos deux classes l'implémente.
3. Complétez la méthode init et la méthode update pour que vos objets de la première classe se déplacent lorsque vous appuyez sur les flèches du clavier !

```

public class MainTD12 extends BasicGame {

    ArrayList<Slickable> objets = new ArrayList<Slickable>();

    public MainTD12(String title) {
        super(title);
    }

    @Override
    public void init(GameContainer gc) throws SlickException {

    }

    @Override
    public void render(GameContainer gc, Graphics g) throws SlickException {
        for (int i = 0; i < objets.size(); i++) {
            objets.get(i).dessiner(gc, g);
        }
    }

    @Override
    public void update(GameContainer gc, int delta) throws SlickException {
        for (int i = 0; i < objets.size(); i++) {
            objets.get(i).mettreAJour(gc, delta);
        }
    }

    public static void main(String[] args) throws SlickException {
        AppGameContainer app = new AppGameContainer(new MainTD12("TD 12"));
        app.setShowFPS(false);
        app.start();
    }
}

```