

3dev3a – Développement 3**Ascii Paint***Grouper, dégrouper. . . faire et défaire***Consignes**

Dans cet exercice nous allons ajouter la possibilité de grouper plusieurs formes afin d'en obtenir une nouvelle ; le groupe de formes.

Table des matières

1	Mini Projet : AsciiPaint - the end	2
2	Groupes	3
2.1	Le modèle	3
2.2	Le contrôleur	3
3	Undo et redo	4
3.1	Le modèle	4
3.2	Le contrôleur	4

1 Mini Projet : AsciiPaint - the end

Le but de cet exercice est la mise en œuvre de deux *design patterns* (patrons de conception) : *composite* et *command*. Pour cela, nous allons ajouter des fonctionnalités de sorte que votre application permette maintenant de :

- ▷ **quitter** ;
- ▷ **grouper ou dégrouper** des formes ;
- ▷ défaire et refaire des commandes (**undo** / **redo**).

Ces manipulations se feront dans la console via des commandes textuelles. Par exemple :

- ▷ **group 1 2 5**
grouper les formes 1, 2 et 5.
Le groupe devient la dernière forme créée, les 3 formes groupées disparaissent de la liste¹.
- ▷ **ungroup 2**
dégrouper le groupe qui se trouve être en position 2 dans la liste des formes, les formes qui le composaient sont remises dans le dessin ;
- ▷ **undo**
annuler la dernière commande ;
- ▷ **redo**
réexécuter la dernière commande annulée ;
- ▷ **exit**
permet de quitter l'application.

Les commandes qui doivent faire partie du mécanisme d'undo/redo sont :

- ▷ ajout d'une forme ;
- ▷ suppression d'une forme ;
- ▷ changement de couleur d'une forme ;
- ▷ déplacer une forme ;
- ▷ grouper et dégrouper des formes (*bonus facultatif*).

Composite pattern

Le groupement et le dégroupement de formes seront implémentés avec le patron de conception composite (*composite pattern*).

Voir la fiche associée.

Command pattern

La gestion de l'*undo* et du *redo* sera implémentée avec le patron de conception commande (*command pattern*).

Voir la fiche associée.

1. Il faudra gérer les couleurs. Le groupe a une couleur et chaque forme aussi. La couleur du groupe peut, par exemple, être la couleur de la première forme du groupe.

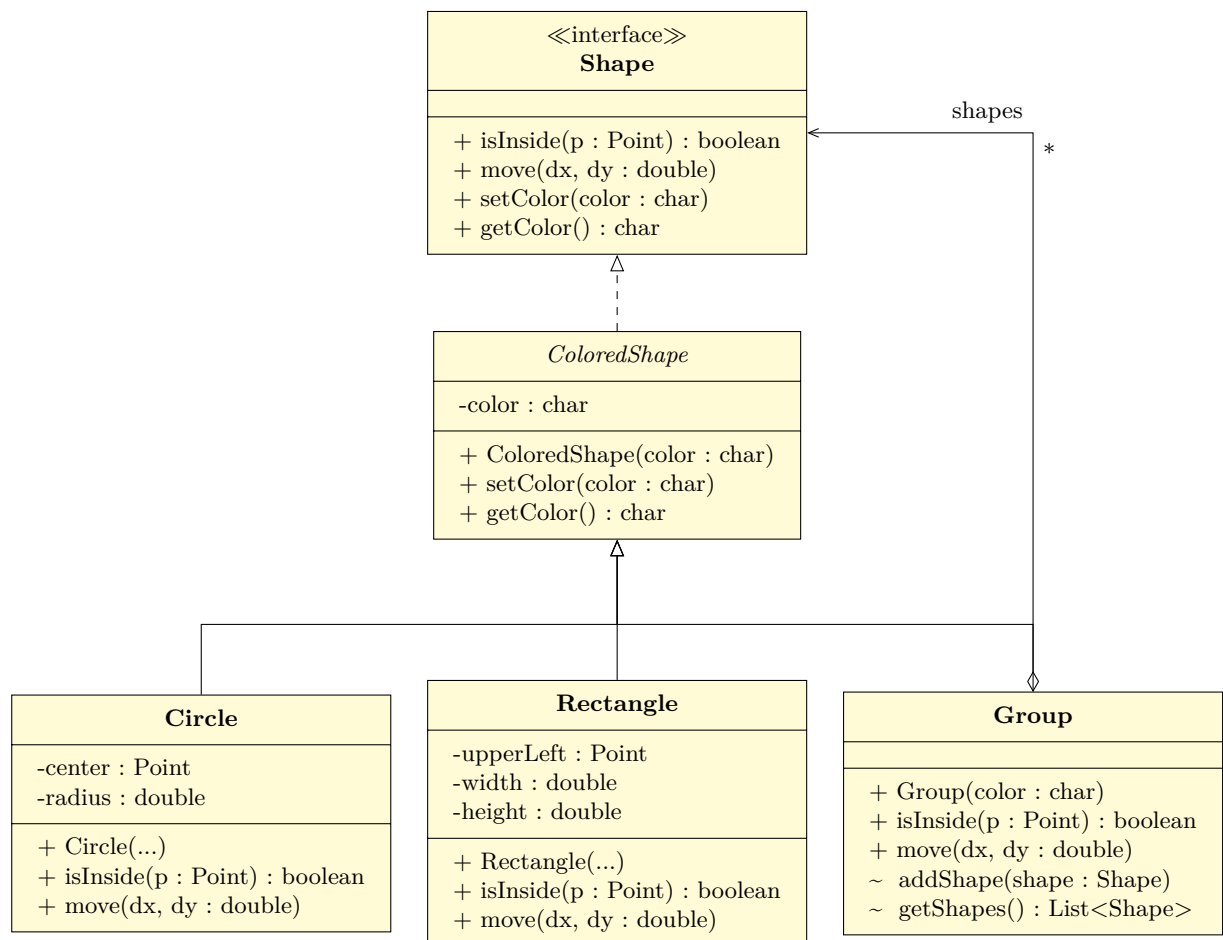
2 Groupes

2.1 Le modèle

Dans le modèle, vous ajoutez une classe **Group** qui implémente l'interface **Shape** (via la classe **ColoredShape**) et contient une liste de **Shapes**, les formes qui le composent. C'est la structure du pattern *Composite*. Consultez la fiche *Composite* sur la page du cours.

Bouger le groupe revient à bouger chaque forme qui le compose. Le groupe se trouve sur un point donné si au moins l'une des formes qui le composent s'y trouve.

Ajoutez dans la façade du modèle, la class **AsciiPaint**, les méthodes **group** et **ungroup** permettant de grouper et dégroupier.



2.2 Le contrôleur

Ajoutez dans le contrôleur les commandes permettant de grouper et dégroupier. Assurez-vous lorsque vous dégroupier que la forme mentionnée soit bien un groupe.

3 Undo et redo

3.1 Le modèle

Le mécanisme d'undo redo utilise le patron de conception *Command*.

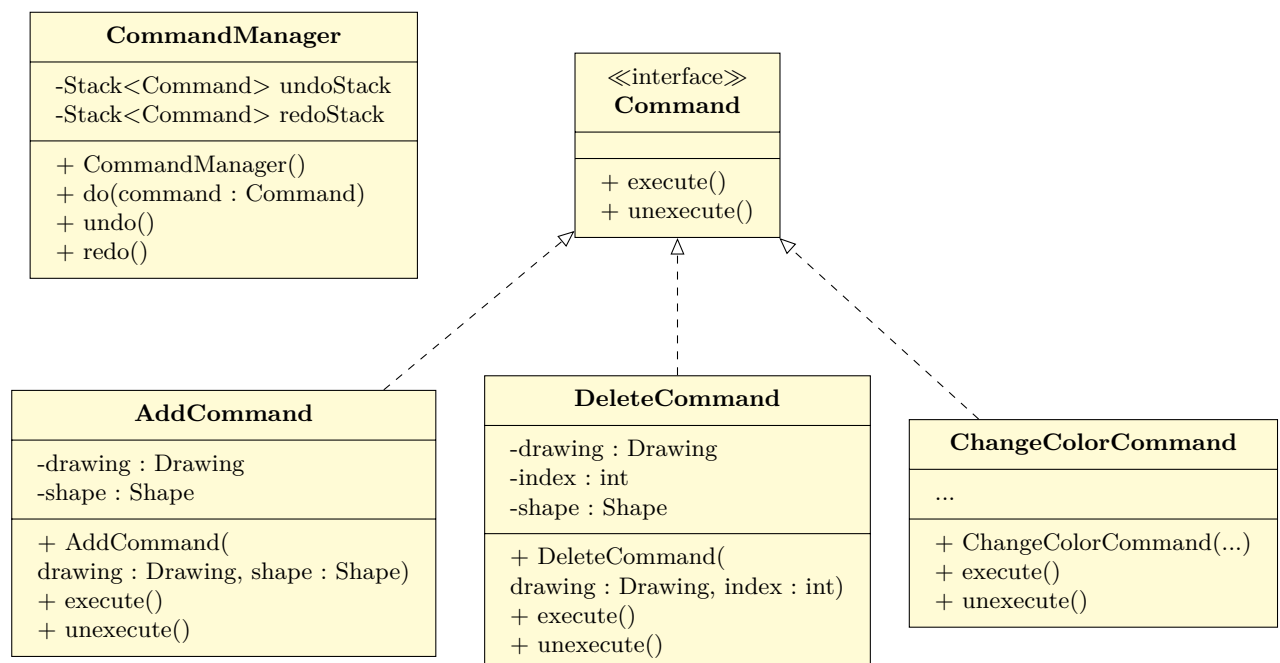
L'interface **Command** représente une commande sur le modèle. Par exemple la commande 'Add' permet d'ajouter une forme au dessin, la commande 'Delete' permet de supprimer une forme du dessin, la commande 'Move' permet d'en déplacer une.

Le constructeur d'une commande prendra en paramètre tout ce qui est nécessaire pour faire et défaire la commande. Par exemple, pour supprimer une forme il faut le dessin (**Drawing**) ainsi que l'indice auquel se trouve la forme. En effet lorsqu'on défait une commande (**unexecute**) il faut remettre le modèle **exactement** dans l'état dans lequel il était avant l'exécution de la commande. Il faut donc remettre la forme dans le dessin à l'indice exact où elle se trouvait avant.

Le mécanisme d'undo/redo s'implémente souvent avec deux piles (*stack*) en utilisant la classe **Stack** par exemple. Une pile, **undoStack** pour les commandes déjà exécutées et une pile pour les commandes que l'on a défaites, c'est la **redoStack**. Le **CommandManager** — aussi appelé **Invoker** — gère ces 2 piles et le mécanisme undo/redo.

Le **CommandManager** fait partie de la façade (attribut de **AsciiPaint**) à laquelle on ajoute les méthodes **undo()** et **redo** qui permettent de faire et défaire les dernières commandes.

Consultez la fiche *Command* sur la page du cours.



3.2 Le contrôleur

Ajoutez dans le contrôleur, les commandes undo et redo.