

A decorative graphic on the left side of the slide, consisting of a network of white lines and circles on a blue gradient background, resembling a circuit board or a neural network.

PRÉSENTATION DESIGN PATTERN

PAR LUKAS BRASSELEUR, ALESSANDRO ALTERNO ET YANI FOUGHALI

The background is a blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines and small circles representing nodes.

COMMENT ON A TRAVAILLÉ ?

The background is a blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines and small circles representing nodes.

C'EST QUOI UN DESIGN PATTERN ?

DESIGN PATTERN : OBJECT POOL

- Qu'est ce ?

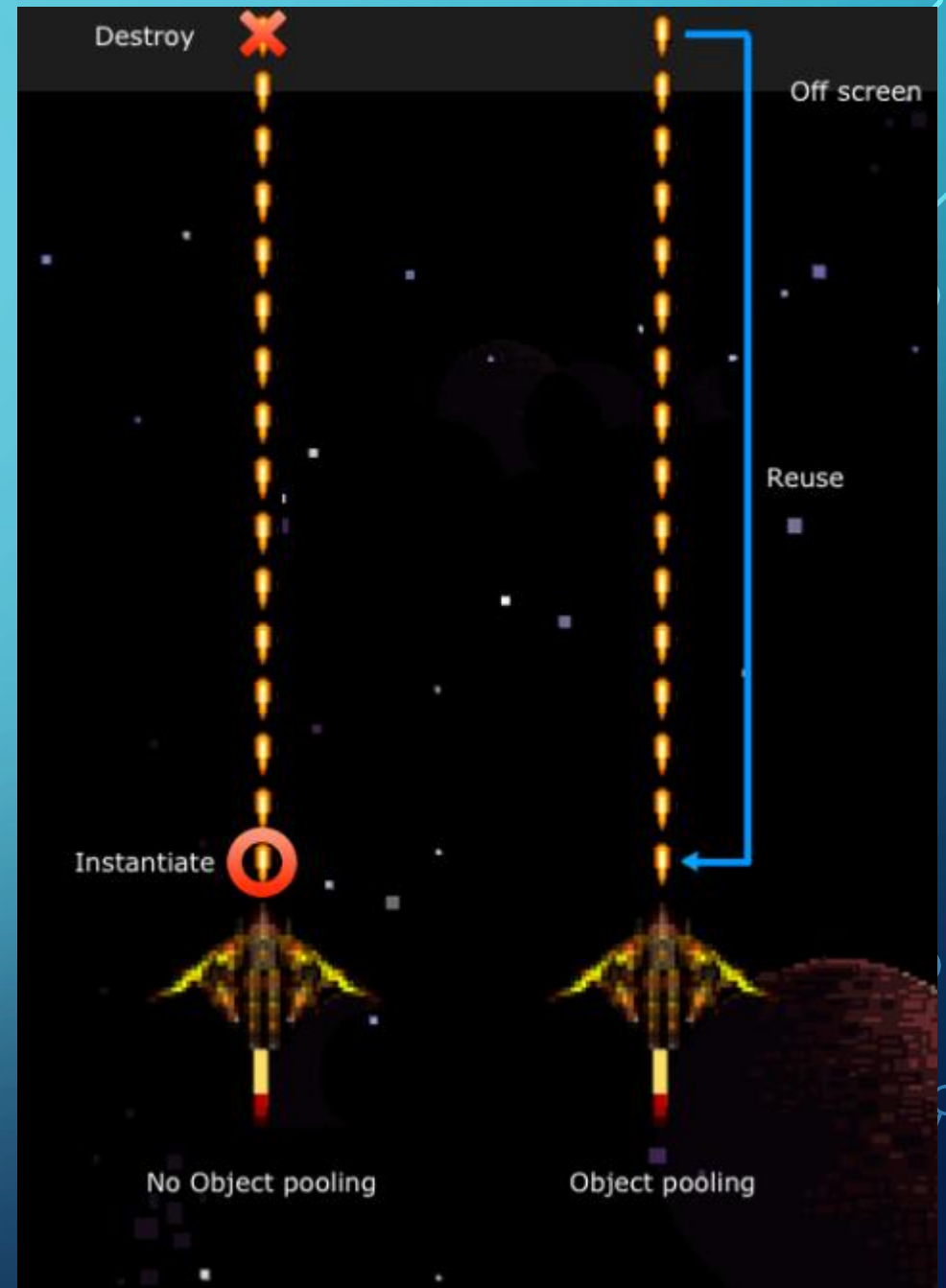
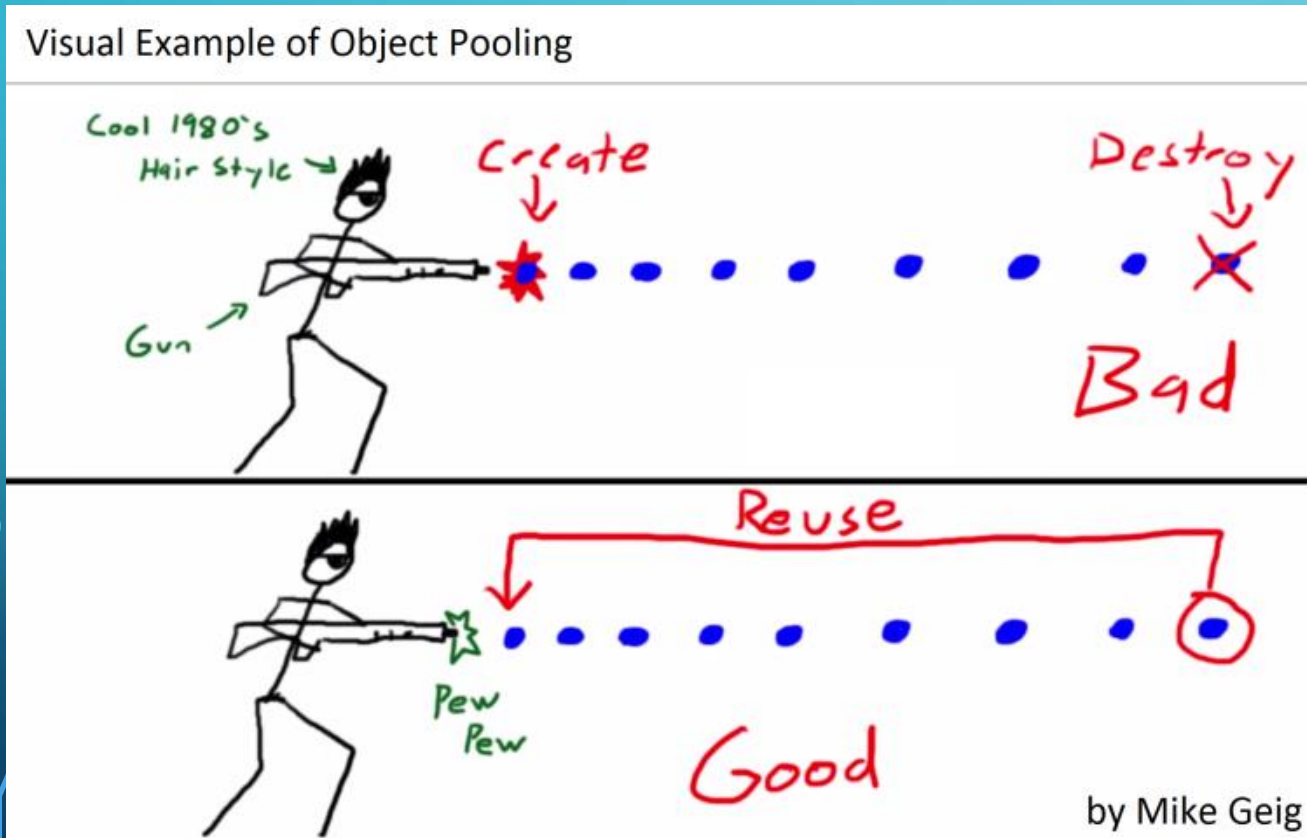
Le pattern **Object Pool** (ou Réservoir d'Objets) permet de stocker une quantité **finie** d'instances d'une classe afin de les distribuer à qui en a besoin et qui sont rendus en fin d'utilisation.

A quoi ça sert ?

L'usage premier est le **contrôle** du nombre d'instances d'un objet vivant dans la mémoire, mais il sert principalement à **économiser des ressources**, en évitant d'instancier un objet + de fois que nécessaire.

EXAMPLE 1 : JEUX VIDEOS

Visual Example of Object Pooling



EXEMPLE 2 : LA BIBLIOTHEQUE

Livre : Pooled Object

Bibliothèque : Pool

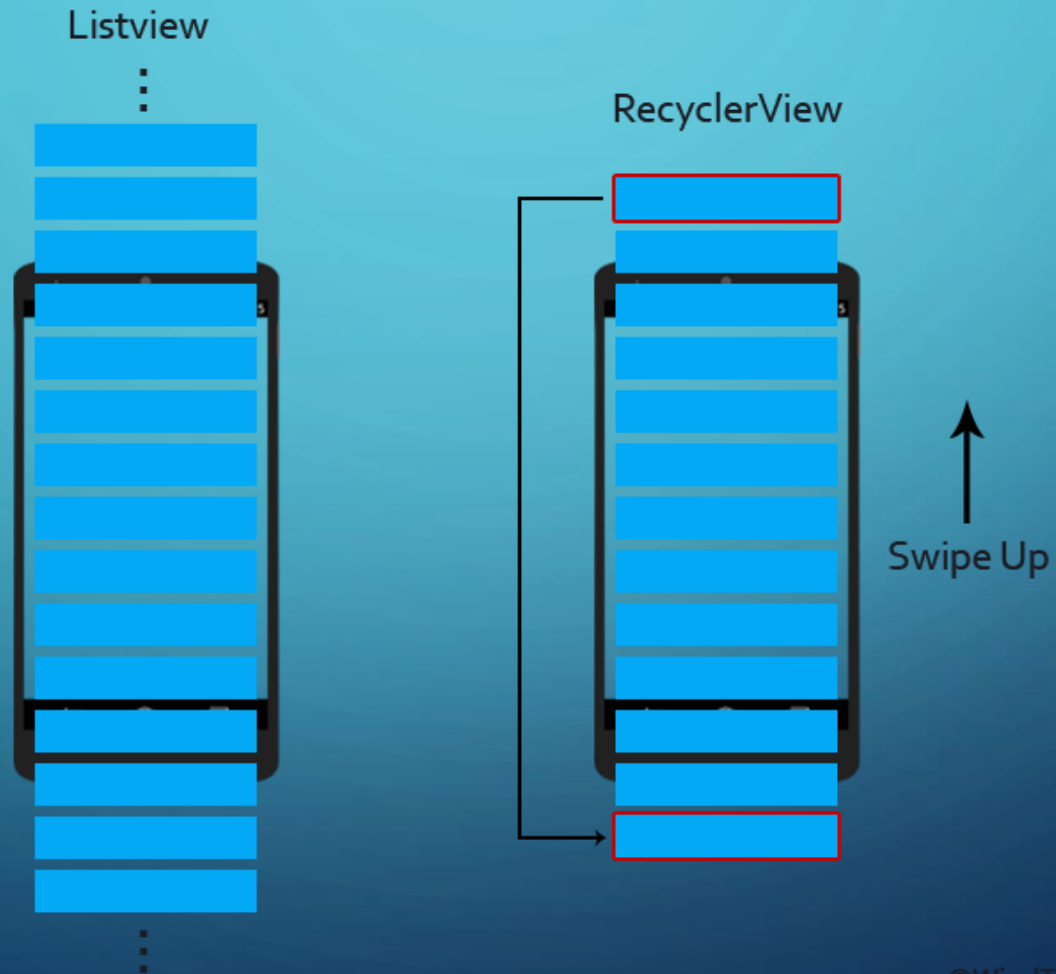
La classe Bibliothèque(le réservoir dans notre cas), gère l'emprunts des livres qu'elle contient. Ces livres sont automatiquement ajoutés à leur création dans la bibliothèque.

2 fonctions importantes :

`getBook(String title)` : elle permet de récupérer un livre (pooled object) dans la bibliothèque (pool).

`returnToPool(Book book)` : elle permet de remettre un livre emprunté à l'état "disponible". De cette manière on reçoit le livre dans le même état que pendant son initialisation : pas de création/destruction d'objet.

EXAMPLE 3 : RECYCLER VIEW



QUELLES CONSÉQUENCES ?

Avantages

- Très utile dans certaines applications où il y a une perte de performances à cause de la création/destruction d'objets identiques.
- Permet de contrôler et de limiter le nombre d'instances « vivantes »

Inconvénients

- S'assurer de bien synchroniser l'accès aux objets, pour éviter tout problème.
- De moins en moins utilisé car la création/destruction d'objets requiert moins de ressources que par le passé. Peut se révéler inutile.

LIENS UTILISÉS

- <https://www.youtube.com/watch?v=CxGM9VFTgqU>
- <https://www.bart-konieczny.com/fr/blog/design-patterns/object-pool>
- https://sourcemaking.com/design_patterns/object_pool