

# Rappels, la notion d'échappement

Fabrice.Kordon@lip6.fr



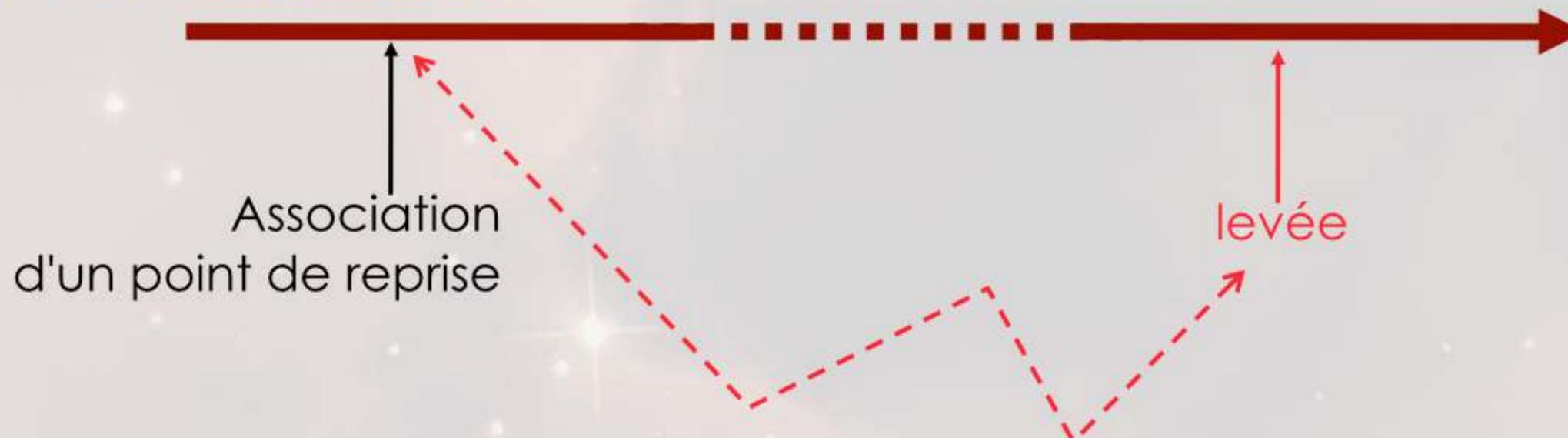
# Vous avez dit «échappement»?

## Mécanisme de déroutement

-  Associé à une «condition de levée» (raise condition)
-  Association à un «traitement» (handler)

## Déroutement «propre»

-  Un échappement préserve le contexte du programme au niveau du traitement associé



## Vision «de haut niveau» des interruptions

-  Variation 1, empilement ou pas des échappements
-  Variation 2, gestion des points de reprise

# Utilité de cette notion

## «Blinder» un programme sans ce mécanisme...

- Alourdit considérablement l'écriture
- Altère la lisibilité.. et nuit à l'évolutivité
- Incommode de prévoir un «code d'erreur»
  - ▶ Cela alourdit considérablement le code
  - ▶ Comment faire dans certains cas  
Exemple: l'addition entre 2 entiers peut générer une erreur (dépassement de capacité) quelle valeur associer à ce cas de figure?

## Traitement des situations exceptionnelles

- Impossible à intégrer dans les sous-programmes
  - ▶ Comment savoir ce que veut faire «l'appelant»?
- D'où la règle
  - ▶ Toujours différer les traitements dépendants de l'appelant

# En guise de conclusion...

4

 Les échappements permettent de factoriser le traitement de cas exceptionnels (e.g. erreurs)

 C'est un mécanisme qui préserve la lisibilité des programmes

 Sont dotés d'un mécanisme d'exception

 Modula, Ada, Java, C++, etc.