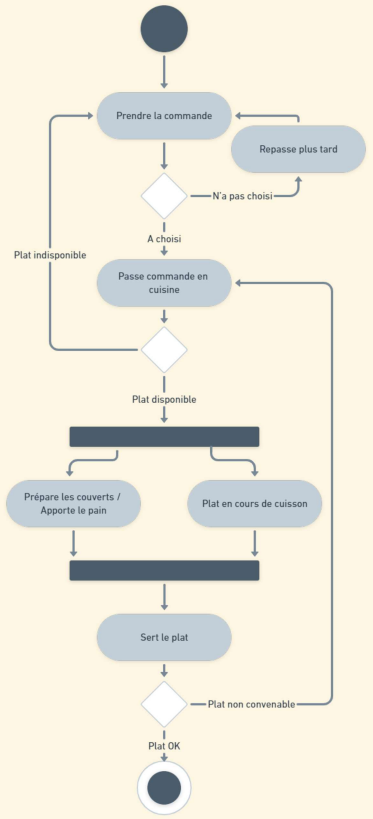
DIAGRAMME D'ACTIVITÉ



DÉFINITION

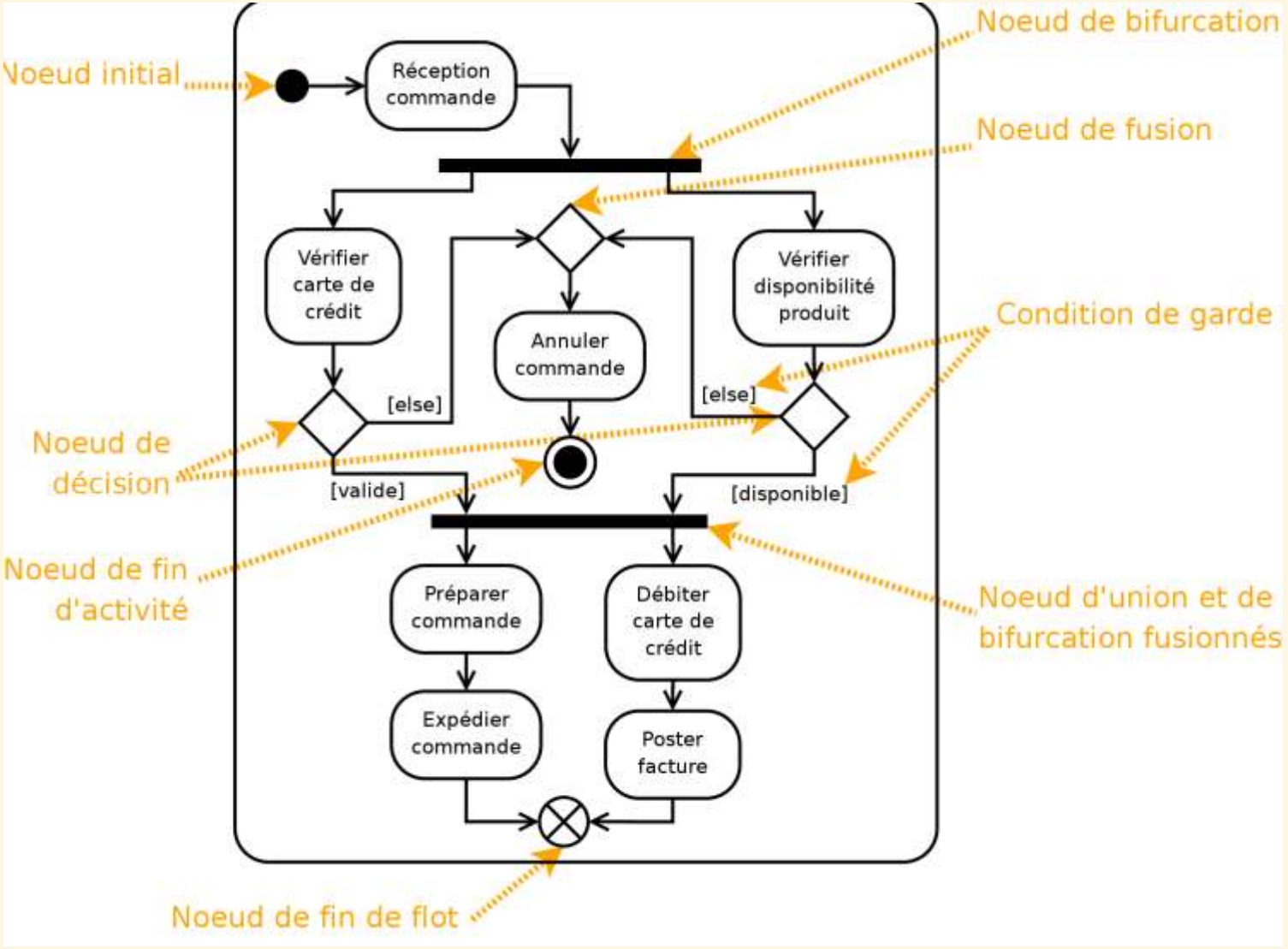
- Variant du diagramme d'état-transitions,
- Permet de visualiser le comportement interne d'une méthode, un cas d'utilisation ...
- Comme un workflow, représente l'enchainement des actions et décisions

- Il ne fait pas état de la collaboration ni du comportement des objets
- Utile pour représenter les processus métiers et les cas d'utilisation
- On l'utilise en complément des use case pour apporter du détail

IL COMPREND

- Les activités
  - étapes qui représente un mécanisme du déroulement de l'action
  - Le passage d'une activité à un autre
- Les transitions
  - Représentés par des flèches,
  - Se déclenchent automatiquement à la fin d'une activité

LÉGENDE

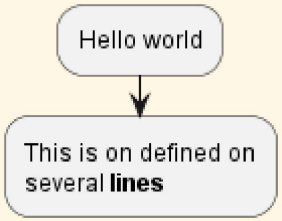

Noeud initial
Réception commande
Noeud de bifurcation
Noeud de fusion
Vérifier carte de crédit
Annuler commande
Vérifier disponibilité produit
Condition de garde
[else]
[else]
Noeud de décision
[valide]
[disponible]
Noeud de fin d'activité
Préparer commande
Débiter carte de crédit
Noeud d'union et de bifurcation fusionnés
Expédier commande
Poster facture
Noeud de fin de flot

LES ACTIVITÉS

- Définit un comportement décrit par un séquencement organisé d'éléments,
- Le flot est modélisé par des neouds et relié par des acrs (transitions)

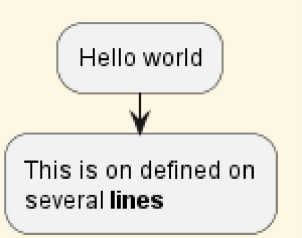
REPRÉSENTATION

Les activité sont représentées comme suit :



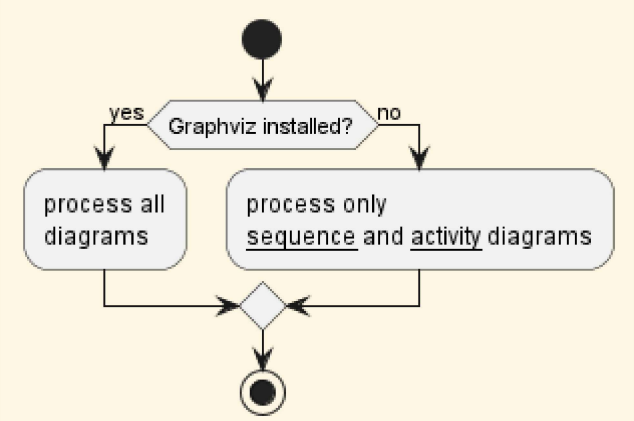
- Ici, l'activité 'Hello World' sous entend que l'on enverra un message

TRANSITIONS



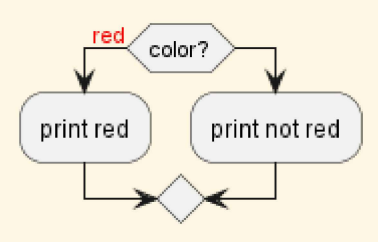
- L'enchainement des actions et activités est représenté par une flèche
- Une transition se déclenche automatiquement lorsque l'action est terminée

DÉCISIONS / TRANSITIONS COMPOSITES



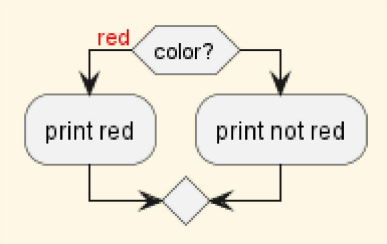
- On peut représenter les décisions par un losange.
- On peut y glisser ou non du détail
- Une décision implique souvent une action

EXEMPLE :



- Les décisions representent des embranchement à partir desquelles les actions suivantes divergent

EXEMPLE



- ici, la couleur est rouge : on affiche rouge
- Dans les autres cas, on affiche 'no color'

EN CODE

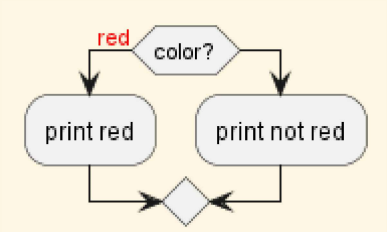
En javascript, on le traduirait comme ceci :

```
if (color == 'red') {console.log('red')}
else {console.log('no color')}
```

DONC

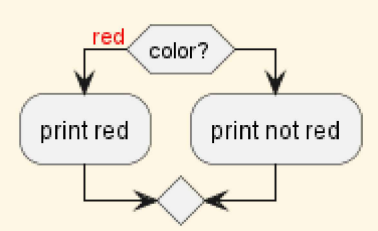
- On comprend donc qu'en réalisant ce schéma, on a le code 'presque prêt' !
- Un développeur saura comment traduire ça en code,
- Un commercial comprendra la vérification réalisée dans le flow

JONCTIONS



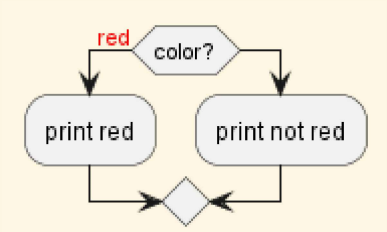
- On peut aussi permettre à différentes branches de se rejoindre

JONCTIONS



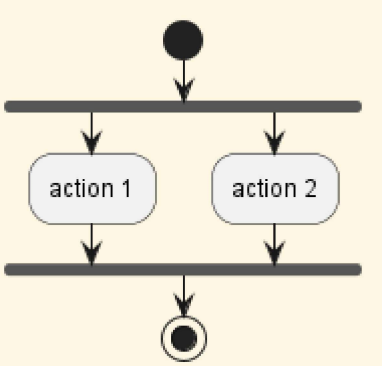
- On utilise le losange pour représenter le point de jonction des actions/activités

JONCTIONS



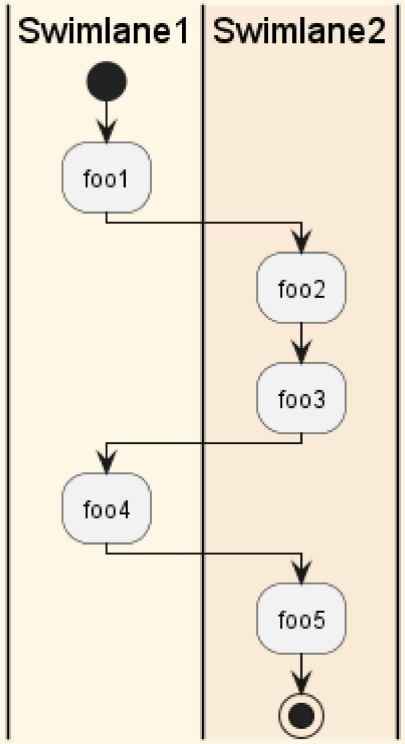
- Ici, les branches séparées lors du choix de la couleur se rejoignent pour terminer l'execution du programme

LES BIFURCATIONS (FORK)



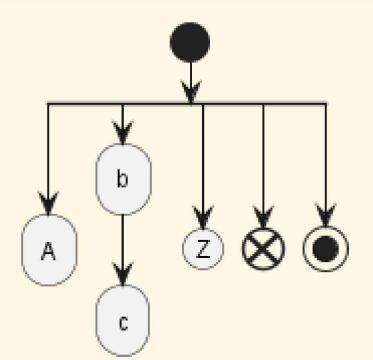
- Servent à représenter deux actions qui se déroulent en simultané
- Elles se rejoignent ensuite

LES SWIMLINES



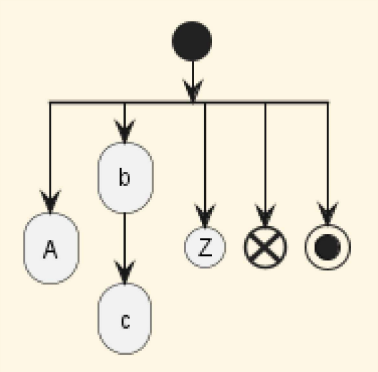
- Quandl le diagramme a besoin d'impliquer plusieurs acteurs, on les représente sous forme de couloir

DÉBUT ET ARRÊT



- On utilise un cercle plein pour indiquer le début de l'application
- On utilise le cercle entourré pour désigner l'arrêt de l'activité

FIN



- Le cercle barré désigne la fin du programme

ARRÊT VS FIN

- Un arrêt peut intervenir avant la fin d'execution de l'application
- Il est souvent lié à une décision qui entraine la fin prématurée de l'application
- Un client qui annule la commande entraine la fin prématurée avant que l'app n'arrive à la fin de son traitement

DÉMONSTRATION

Diagramme d'une commande avec swimlines

EXERCICE

Réalisez l'exercice 4