Échanges de message entre objets (Diagrammes de Séquence)

Sébastien Mosser - INF5153
Chapitre 2 - Capsule 1
Automne 2020

Dans les épisodes* précédents

- On s'est intéressé à la **dimension "statique"** des objets
  - Avec les **diagrammes d'objets** (objets collaborant à l'exécution)
  - Avec les **diagrammes de classes** (structure de l'application)
- On a **négligé le comportement dynamique** des objets
  - En fait on l'a délégué au code : "*Les objets s'envoient des messages*"
- **Quels messages ? Dans quel ordre ? Qui parle à qui ?**

Incluant les pré-requis du cours

L'exemple du domicile intelligent


```
public void netflixAndChill(SmartHome s) {

  HomeCinema hc = s.findTV("TV Salon");
  SmartLight light = s.findLight("Philips Light Hue 1");

  hc.startTv();
  hc.setSoundMode(Modes.DOLBY_5_1);
  hc.setChannel(Services.NETFLIX)

  light.setLumPercentage(0.40);

}
```

Comment modéliser les échanges entre objets ?

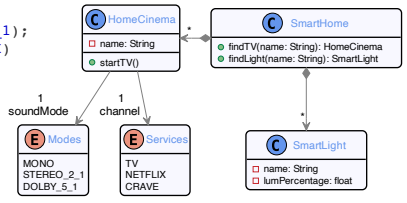
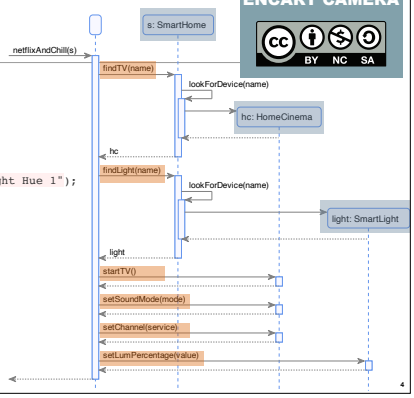


Diagramme de séquence

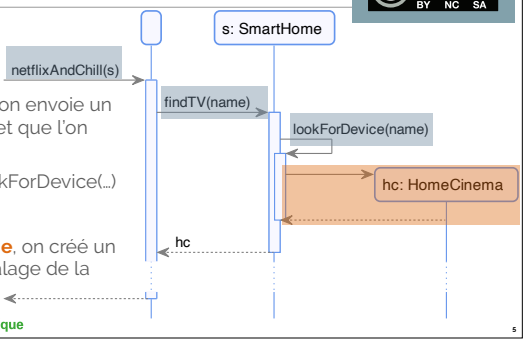
ENCART CAMÉRA

```
public void netflixAndChill(SmartHome s) {

  HomeCinema hc = s.findTV("TV Salon");
  SmartLight light = s.findLight("Philips Light Hue 1");

  hc.startTv();
  hc.setSoundMode(Modes.DOLBY_5_1);
  hc.setChannel(Services.NETFLIX)

  light.setLumPercentage(0.40);

}
```



Différencier l'utilisation d'un objet de sa création

ENCART CAMÉRA



- **Dans le cas bleu**, on envoie un message à un objet que l'on connait déjà :
  - s.findTV(…), s.lookForDevice(…)
- **Dans le cas orange**, on créé un nouvel objet (décalage de la ligne de vie)

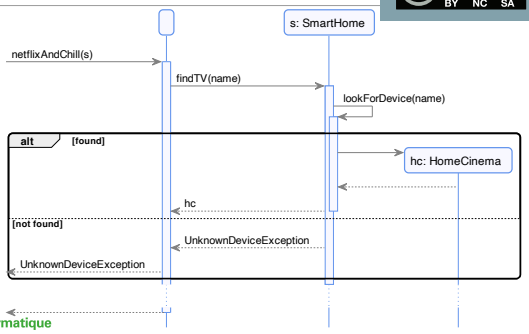
Attention au niveau de granularité

ENCART CAMÉRA

- **On n'utilise pas les diagrammes de séquence comme un langage de programmation**
  - *(en vrai on peut le faire, mais **ce n'est pas l'objet de ce cours**)*
- Privilégiez des **diagrammes de séquences ciblés** sur les parties critiques
  - *Création des objets*
  - *Coordination critique*
  - *Point difficile à voir dans le diagramme de classes*
- **Un diagramme doit servir une intention de conception**

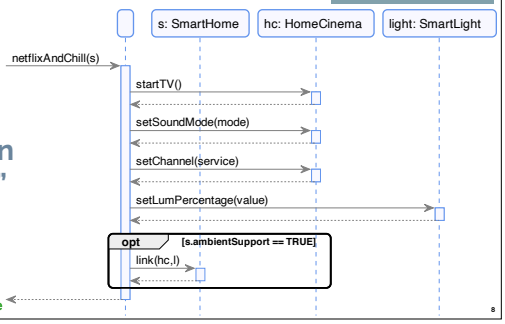
Gestion des flots de contrôles alternatifs

Ifs
Switchs
Exceptions



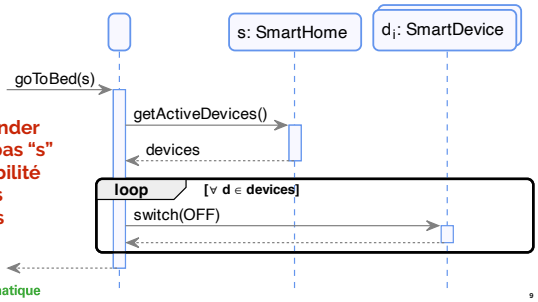
Gestion des cas optionnels

Équivalent d'un
"if" sans "else"



Gestion des boucles

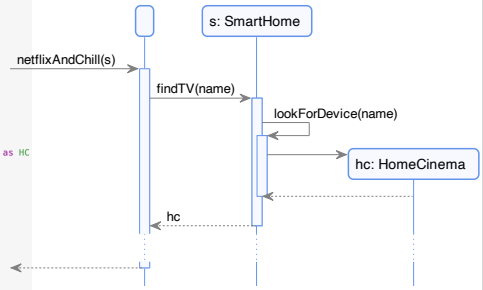
On peut se demander pourquoi ce n'est pas "s" qui a la responsabilité d'éteindre les périphériques



Utilisation de PlantUML

ENCART CAMÉRA

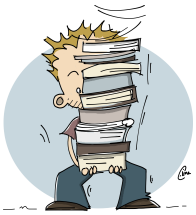
```
@startuml
!include ./../../../_commons.style

participant " " as M
participant "s: SmartHome" as S

[-> M: netflixAndChill(s)
activate M
  M -> S: findTV(name)
  activate S
    S -> S: lookForDevice(name)
    activate S
      create participant "hc: HomeCinema" as HC
      S -> HC
      HC --> S
    deactivate S
    S --> M: hc
  deactivate S
  ...
  [<--M:
deactivate M

@enduml
```



UQÀM | Département d'informatique

FACULTÉ DES SCIENCES
Université du Québec à Montréal

ENCART CAMÉRA



https://mosser.github.io/

ace

https://ace-design.github.io/

***Abonne toi** à la chaine,*
et met un pouce bleu !