

LOG210 ANALYSE ET CONCEPTION DE LOGICIELS: SÉANCE 02

ADMINISTRIVIA

1. Accès G Suite (**ETSMTL.NET**)

- Google Classrooms (Travaux pratiques)
- Faire attention au quota de 15GB

2. Invitation Lab 0 (GitHub Classroom)

3. Équipes (à décider plus tard)

4. Invitation Discord pour lab



AUXILIAIRES D'ENSEIGNEMENT

- Philippe Côté (philippe.cote.5@etsmtl.ca)
- Mario Munoz (mario-francisco.munoz.1@ens.etsmtl.ca)

Laboratoire est le jeudi après-midi.

La première séance est le 22 septembre 2022

SURVOL SÉANCE 02

- Rétroaction feuille d'une minute
- Rétroaction Mini-test
- Rappel méthodologie (MDD, DSS, Contrats, RDCU)
- MDD: Catégories (exercice)
- DSS: Introduction
- Architecture en couches

RÉTROACTION

FEUILLE D'UNE MINUTE

1. Une seule pause de 15 minutes ...

RÉTROACTION

MINI-TEST



QUESTIONS DIFFICILES

Selon les statistiques de hier matin ...



Socrative → ETSLOG210

Pourquoi fait-on un modèle du domaine (MDD)?

- A. Un modèle du domaine comprend tous les concepts importants dans le contexte du problème que le logiciel va résoudre. Il est primordial d'avoir une compréhension presque parfaite du problème au début d'un projet.
- B. Dans le développement de logiciel orienté-objet, si on commence avec un modèle du problème en termes de classes et d'objets, cela favorise une solution claire en termes de classes et d'objets logiciels.
- C. Un modèle du domaine est la base de toutes les classes logicielles qui vont exister dans la solution.
- D. Un modèle du domaine est fait pour respecter le processus uniifié. no, pour bien comprendre le problème.

COMPLEXITÉ: VRAI OU FAUX?

L'existence de plusieurs versions d'Android avec les politiques différentes pour les autorisations d'exécution est un exemple de complexité inhérente.

- A. Vrai
- B. Faux

COMPLEXITÉ: VRAI OU FAUX?

Un logiciel contient un moteur de règles pour supporter toutes les politiques d'une entreprise en ce qui concerne les congés et les vacances selon la convention collective avec les employés. Ceci est un exemple de complexité circonstancielle.

- A. Vrai
- B. Faux

MDD ÉLÉMENTS

Qu'est-ce qui ne fait pas partie du modèle du domaine?

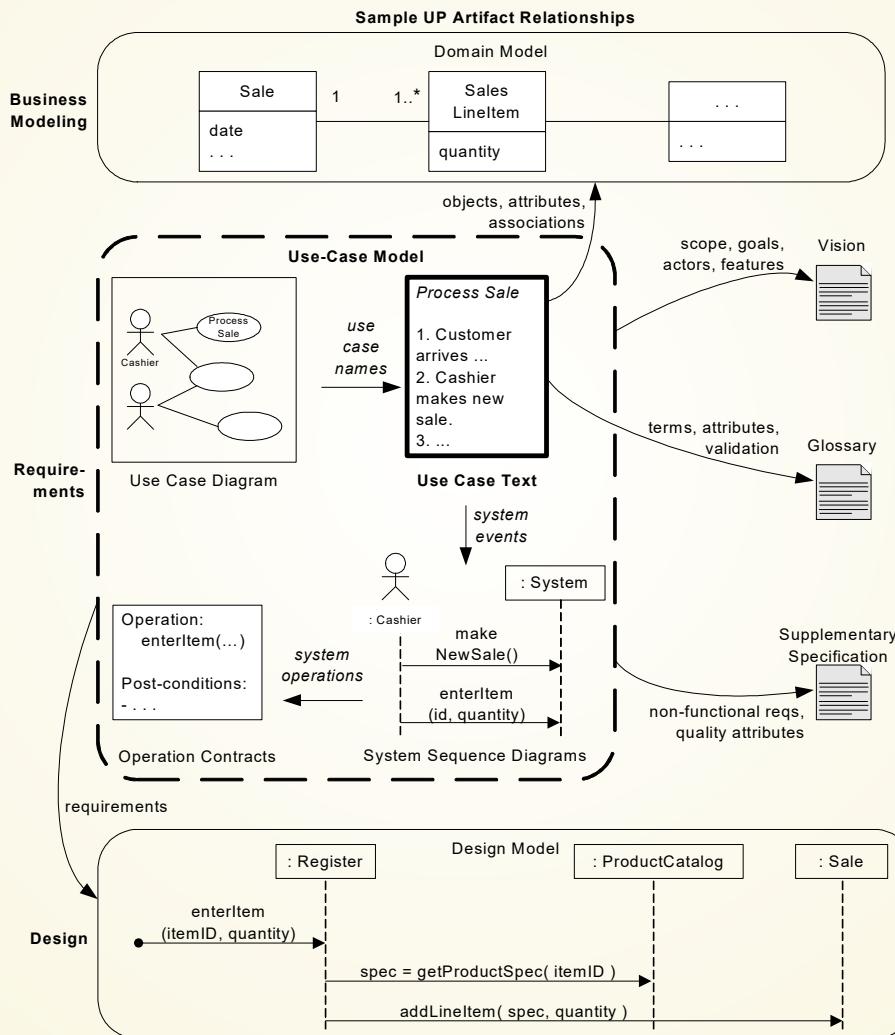
- A. Les associations entre les classes conceptuelles
- B. Les classes conceptuelles
- C. Les attributs dans les classes conceptuelles
- D. Les méthodes dans les classes conceptuelles.

MÉTHODOLOGIE RAPPEL

- Décrire (comprendre) le problème (Analyse)
- Décrire (proposer) une solution (Conception)

Itérer pour valider les hypothèses, approfondir la compréhension, etc.

Survol Visuel



Est-ce que l'artefact permet de démontrer l'analyse de cas d'utilisation?

- Diagramme de séquence **Non**
- Diagramme de séquence système (DSS) **Oui**
- Réalisation d'un cas d'utilisation (RDCU) **Non**
- Modèle du domaine **Oui**
- Diagramme de classes logicielles **Non**
- Contrat **Oui**

MODÈLE DU DOMAINE (MDD)

CLASSES CONCEPTUELLES

Analyse de logiciel pour les ressources humaines d'une université. Dans ce logiciel les cours qu'enseignent les professeurs ne sont pas importants. Indiquer les classes à inclure dans le modèle du domaine (MDD) pour ce logiciel.

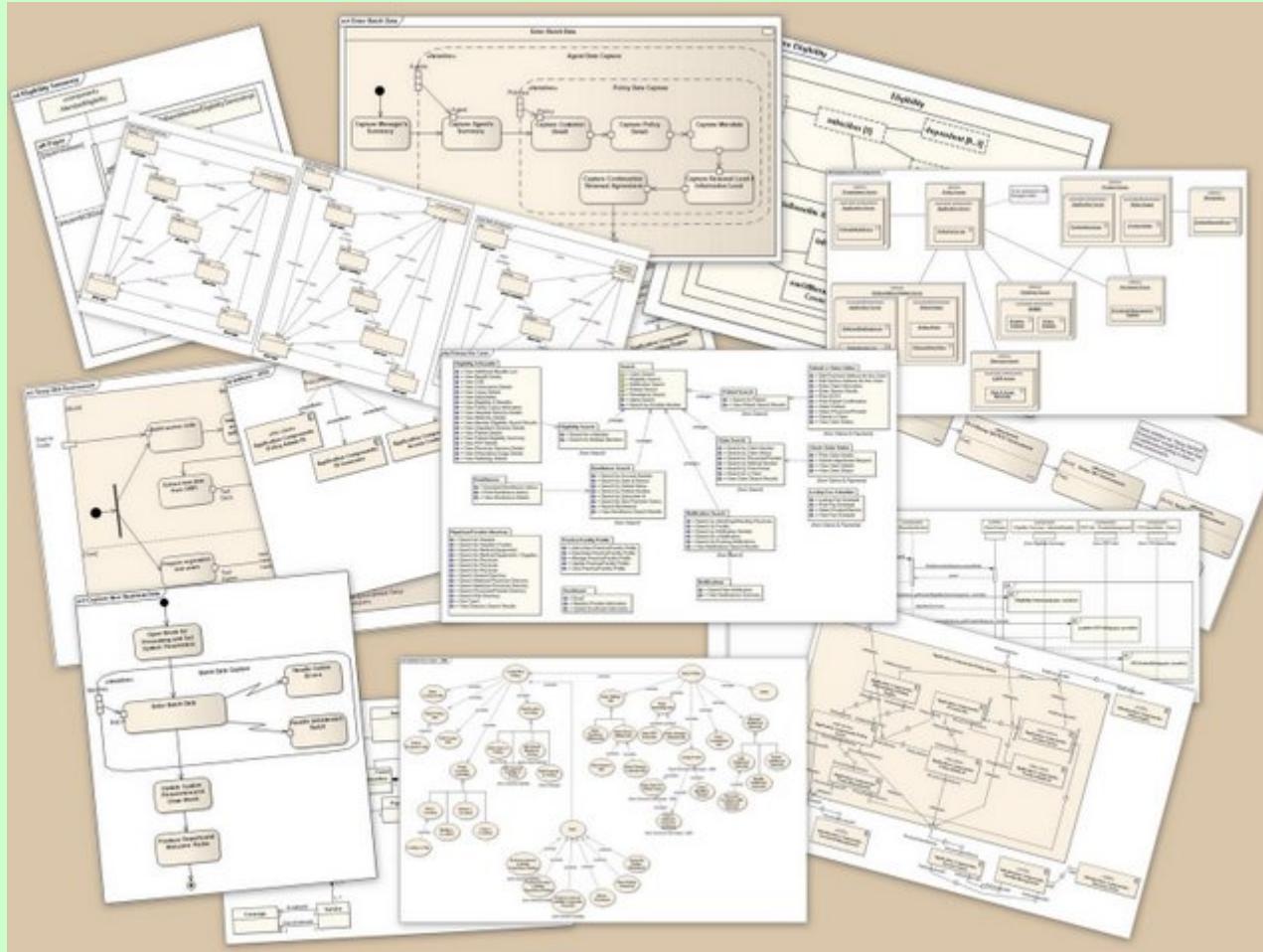
- Étudiant
- Université
- Professeur
- Devoir
- Cours



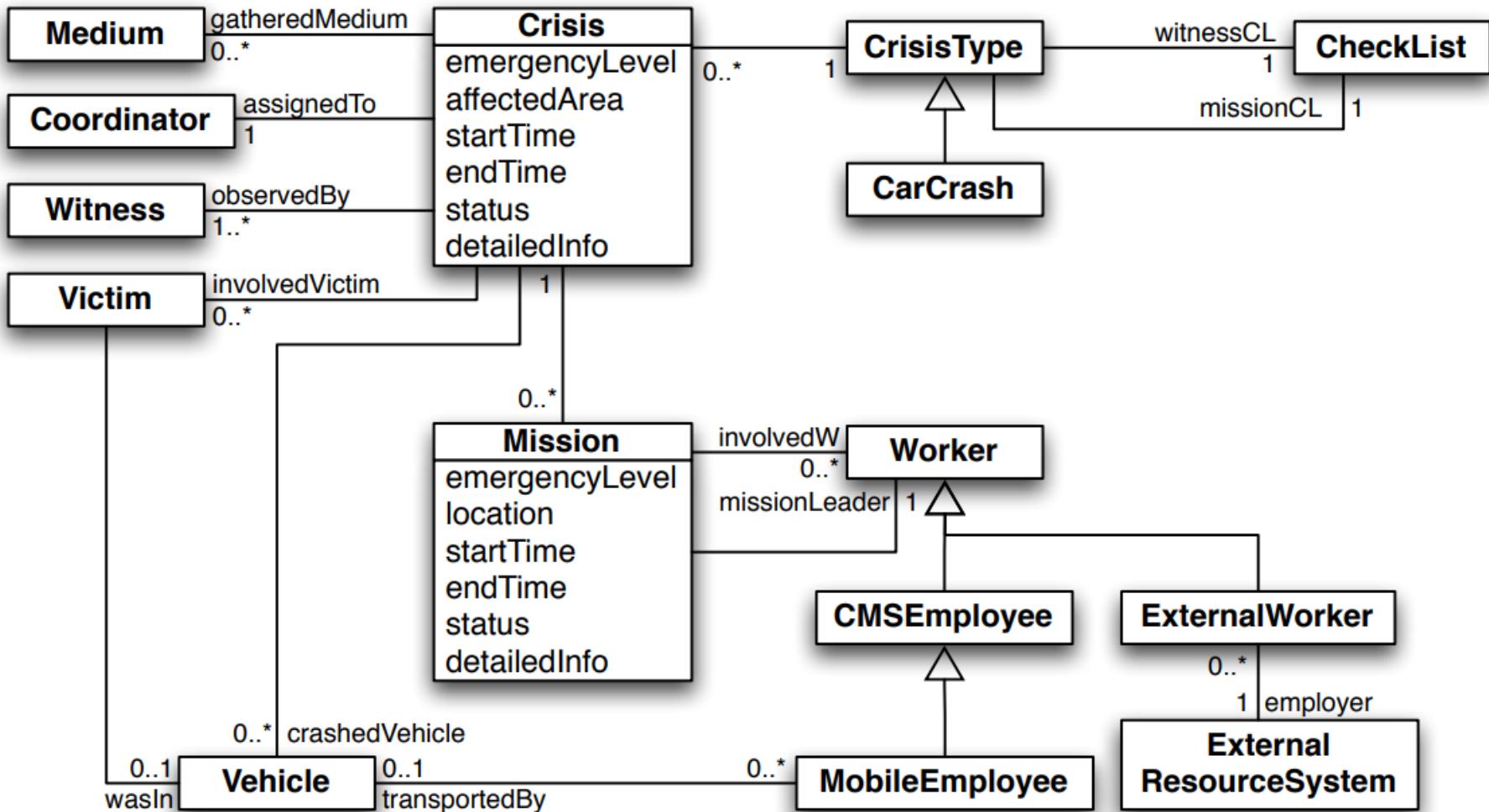
TROUVER DES CLASSES CONCEPTUELLES

1. Réutiliser modèle(s) existant(s)
2. Catégories de classes
3. Groupes nominaux

MÉTHODE 1: MODÈLE EXISTANT



MÉTHODE 1: MODÈLE EXISTANT



MÉTHODE 2: CHERCHER PAR CATÉGORIE



Section 9.5 du livre.

Produit ou Service
lié à une
transaction

Rôle des
personnes
(acteurs)

Catalogues

Transaction
d'affaires

Lignes d'une
transaction

Description
d'entités

Documents
financiers,
contrats, etc.

Où la transaction
est-elle
enregistrée?

Organisations
liées à la
transaction

Conteneurs

Événements
notables

Instruments
financiers

Lieu de la
transaction ou
service

Objets physiques

Contenu

Systèmes
externes

Plannings,
manuels, etc.
(consultés pour
faire un travail)

CATÉGORIE DE CLASSE



Cas d'utilisation « S'inscrire à un groupe-cours »

1. L'Étudiant commence une inscription.
2. L'Étudiant entre le sigle du cours.
3. Le Système affiche la liste des groupes-cours ainsi que l'horaire de chaque groupe-cours.
4. L'Étudiant sélectionne le groupe-cours auquel il veut s'inscrire.
- 5....

Catégories dans les Notes de cours



MÉTHODE 3: IDENTIFIER LES GROUPES NOMINAUX

- Analyse linguistique du texte des spécifications (Cas d'utilisation, Récit utilisateur, etc.)
- Noms → Classes conceptuelles ou attributs possibles
- Approche *simple*, mais *imprécise*

IDENTIFIER LES GROUPES NOMINAUX(...)



Cas d'utilisation « S'inscrire à un groupe-cours »

1. L'Étudiant commence une inscription.
2. L'Étudiant entre le sigle du cours.
3. Le Système affiche la liste des groupes-cours ainsi que l'horaire de chaque groupe-cours.
4. L'Étudiant sélectionne le groupe-cours auquel il veut s'inscrire.
- 5....

IDENTIFIER LES GROUPES NOMINAUX(...)



Cas d'utilisation « S'inscrire à un groupe-cours »

1. L'Étudiant commence une inscription.
2. L'Étudiant entre le sigle du cours.
3. Le Système affiche la liste des groupes-cours ainsi que l'horaire de chaque groupe-cours.
4. L'Étudiant sélectionne le groupe-cours auquel il veut s'inscrire.
- 5....

IDENTIFIER LES GROUPES NOMINAUX(...)



Résultat: *Classes candidates*

- Beaucoup de directives (Chapitre 9)
- Classe vs Attribut?
- Attribut vs Association?

C'est l'analyse (pas encore la conception)

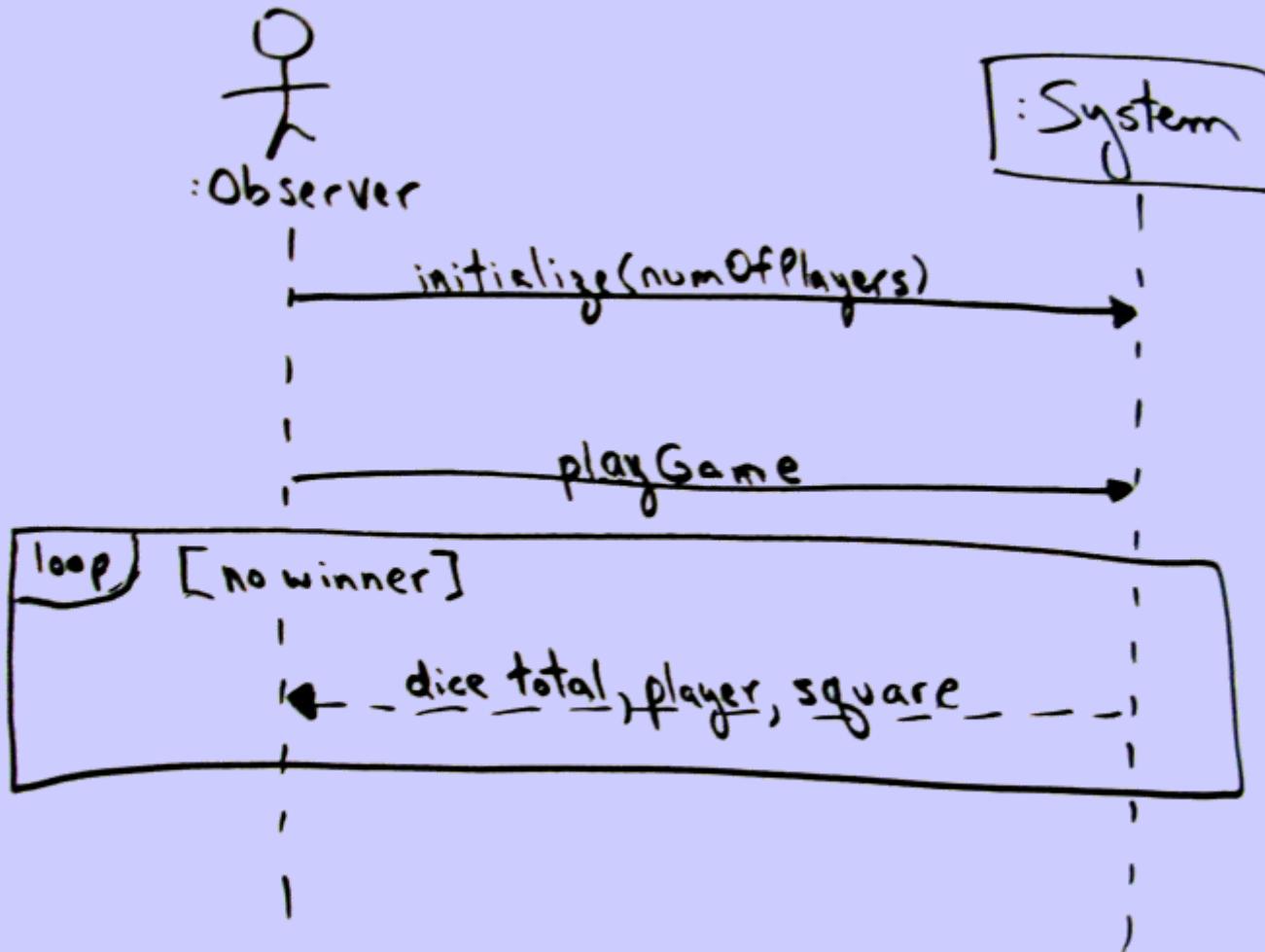
L'étude du problème



EXERCICE: MDD CLASSES CONCEPTUELLES

Google Classroom – Rétroaction

DIAGRAMME SÉQUENCE SYSTÈME (DSS)



DSS

- Crée à partir d'un cas d'utilisation
- Modélise l'interaction
 - Opérations système
 - Messages de retour (au besoin)
- Notation UML
- **But : définir l'API (haut niveau)**
- Système est objet sans détails

TRACER UN DSS

- Identifier l'acteur principal
- Modéliser le système comme objet sans détails
- Identifier les évènements système
 - *Évènements externes en entrée*
 - Décrits dans cas d'utilisation
- Proposer une **opération système** pour chaque **événement système**
 - Types primitifs pour arguments
 - Messages de retour si nécessaire

DSS: CU01 - AJOUTER UN LIVRE À ÉCHANGER

1. Le Client démarre un nouvel ajout de livre.
 2. Le Client entre le code ISBN du livre, ainsi que le code de sa condition.
 3. Le Système enregistre le livre et présente sa description.
- Le Client répète les étapes 2 à 3 jusqu'à ce qu'il ait saisi tous les livres à échanger.*
4. Le Système présente la liste de livres que possède le Client.

DSS: « ATTAQUER UN PAYS »

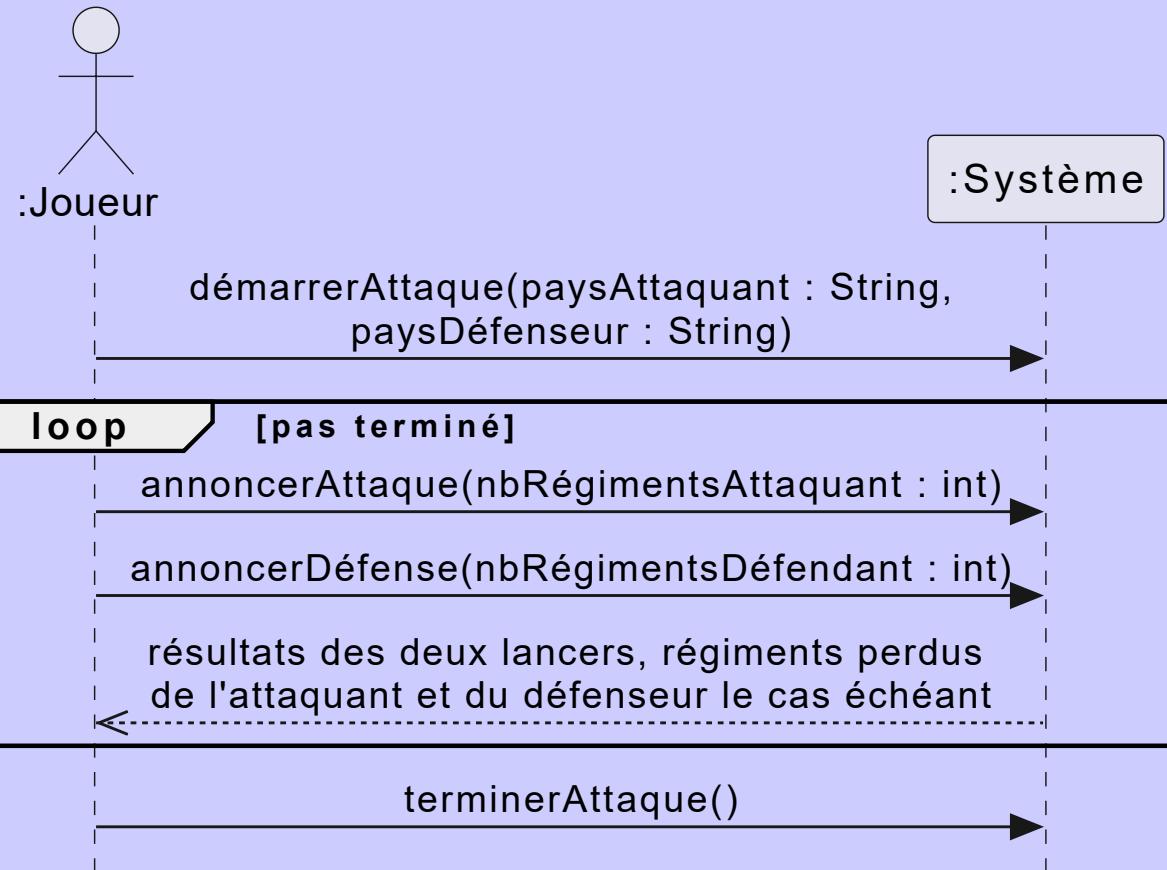
1. Le Joueur attaquant choisit d'attaquer un pays voisin du Joueur défenseur.
2. Le Joueur attaquant annonce combien de régiments il va utiliser pour son attaque.
3. Le Joueur défenseur annonce combien de régiments il va utiliser pour sa défense.
4. Les deux Joueurs jettent le nombre de dés selon leur stratégie choisie aux étapes précédentes.
5. Le Système compare les dés et élimine les régiments de l'attaquant ou du défenseur selon les règles et affiche le résultat.

DSS: « ATTAQUER UN PAYS »

Les Joueurs répètent les étapes 2 à 5 jusqu'à ce que l'attaquant ne puisse plus attaquer ou ne veuille plus attaquer.

DSS: « ATTAQUER UN PAYS »

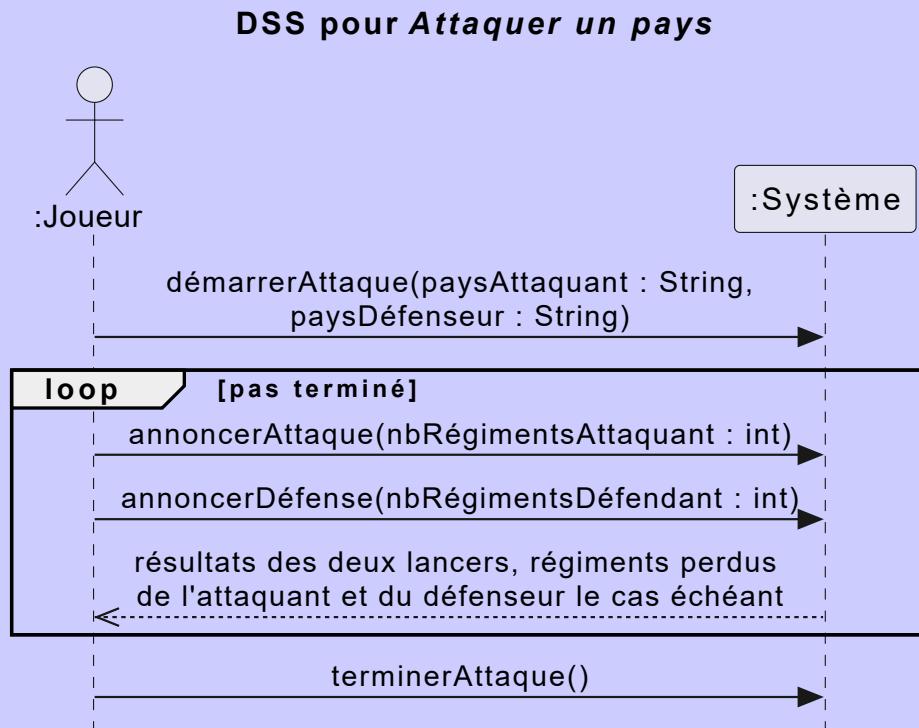
DSS pour *Attaquer un pays*

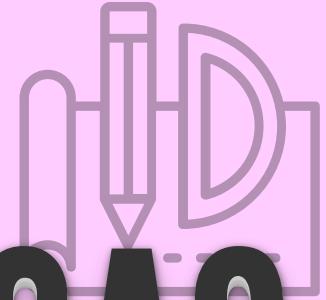




Quiz → ETSLOG210

Vrai/Faux: Il y a 5 opérations système dans ce DSS





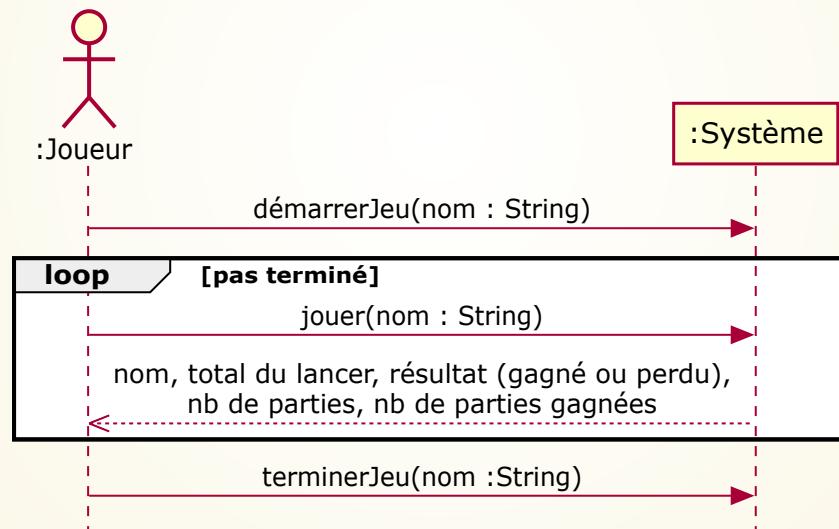
RÉALISATIONS DE CAS D'UTILISATION (RDCU) CONTRÔLEUR GRASP

QU'EST-CE QU'UNE RDCU

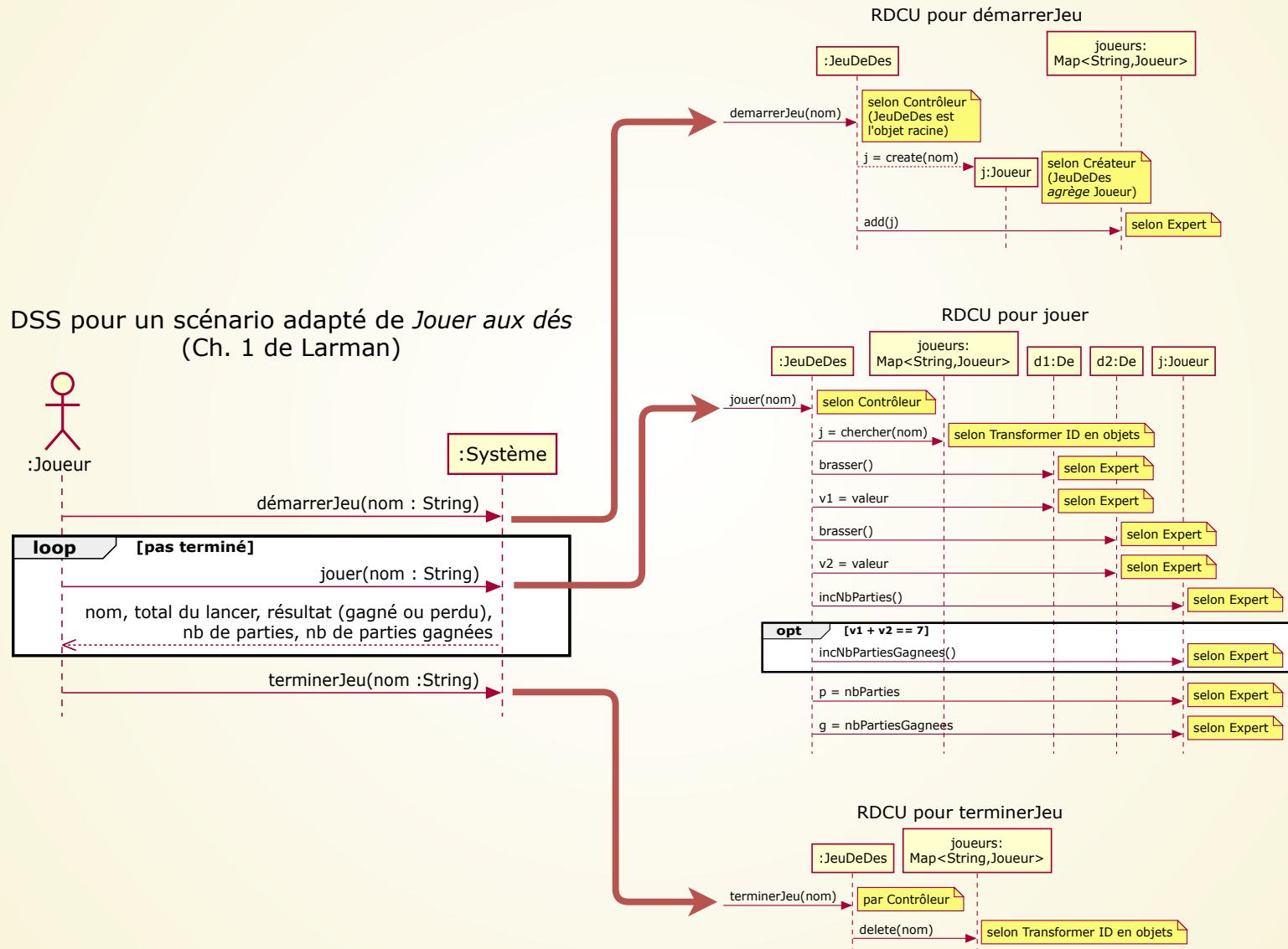
- C'est la conception des opérations système.
- Diagrammes d'interaction (de séquence)

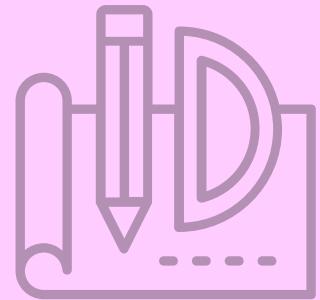
EXEMPLE DSS

DSS pour un scénario adapté de *Jouer aux dés*
(Ch. 1 de Larman)



EXEMPLE DSS + RDCU

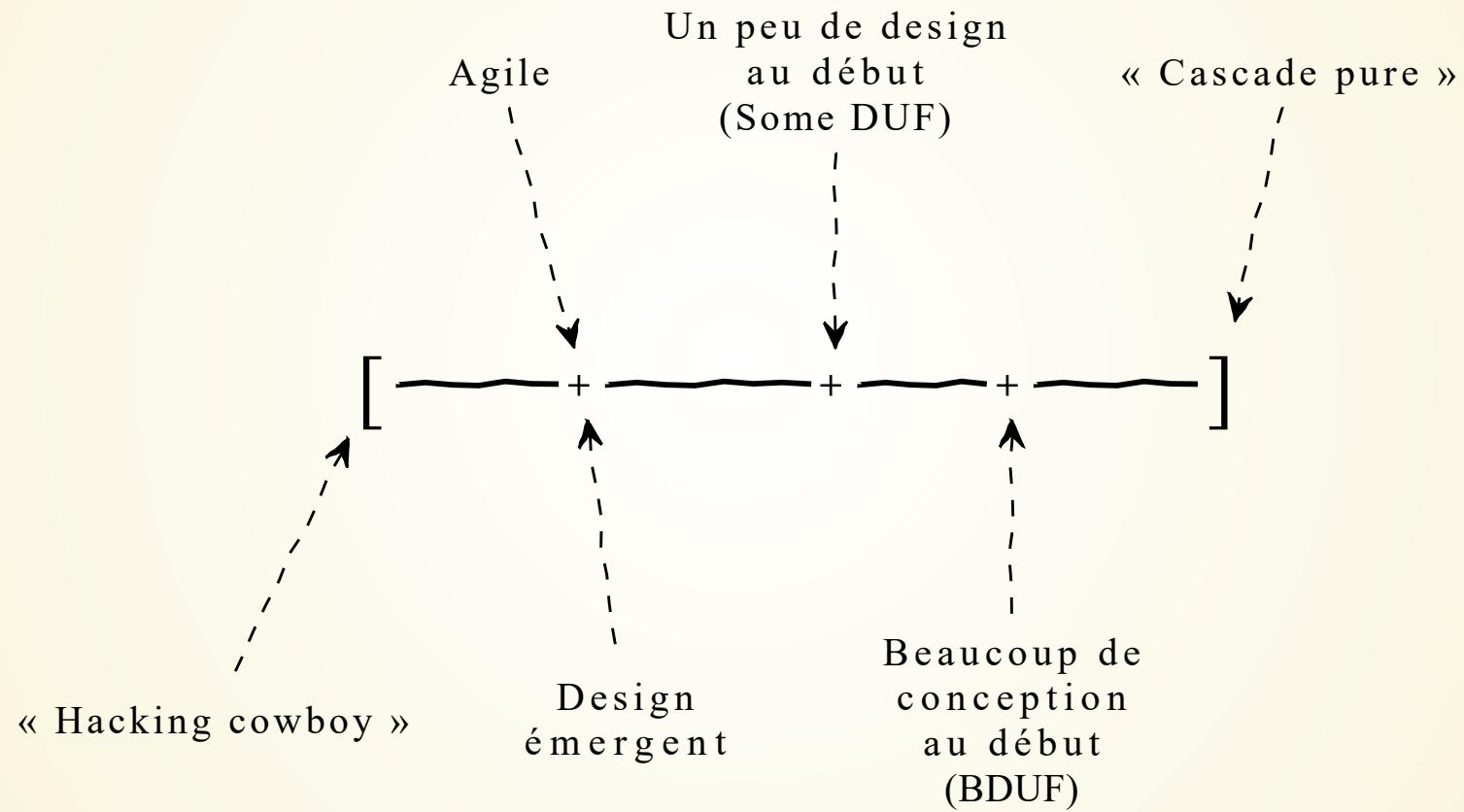


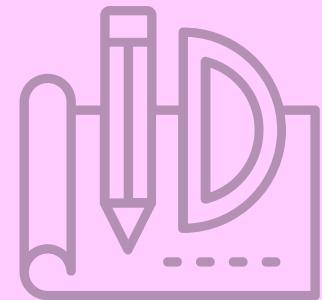


:POURQUOI FAIRE UNE RDCU?

Pour apprendre à faire une solution
modulaire et intuitive

Spectre de la conception

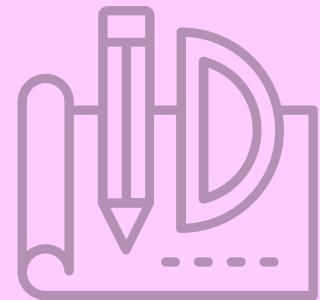




RDCU: ASPECTS

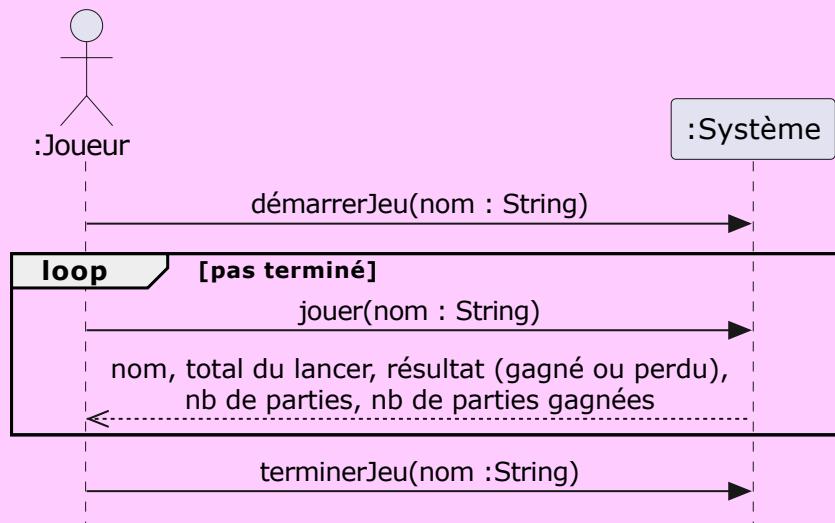
- Proposer des **classes logicielles** correspondant aux **classes conceptuelles**
 - Inspirées du MDD
 - Pour réduire le *décalage des représentations*
- Utiliser les principes GRASP
 - **Faible Couplage et Forte cohésion (LOG121!)**
 - **Contrôleur** (architecture en couches)

RDCU - JEU DE DÉS

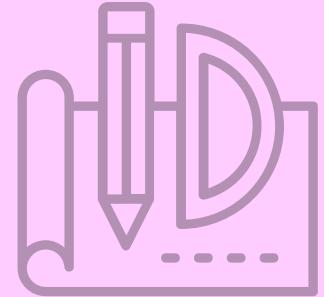


Opérations système du DSS

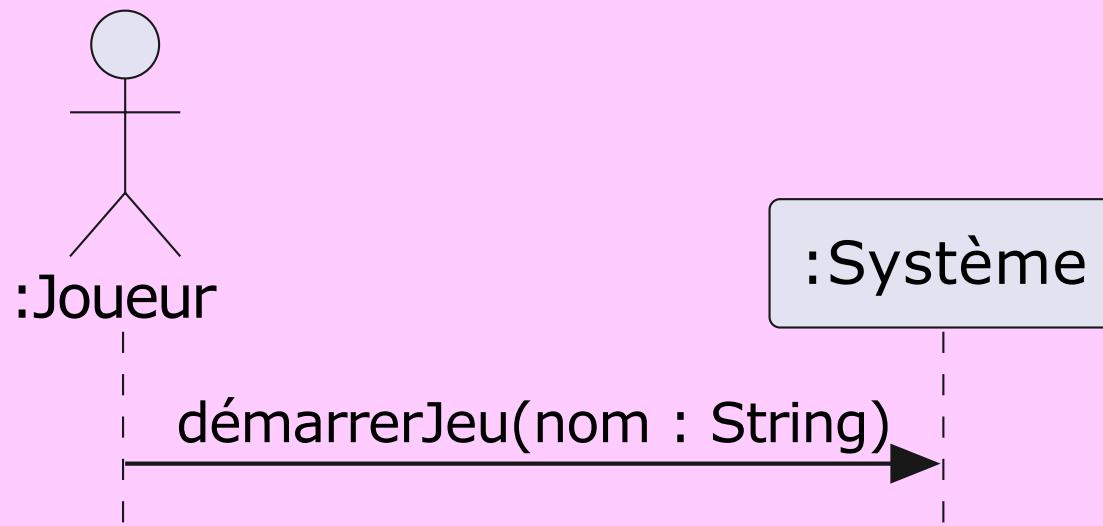
DSS pour un scénario adapté de *Jouer aux dés*
(Ch. 1 de Larman)



RDCU - JEU DE DÉS

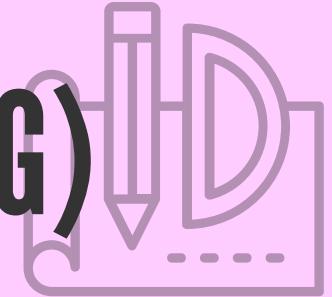


Première opération système:

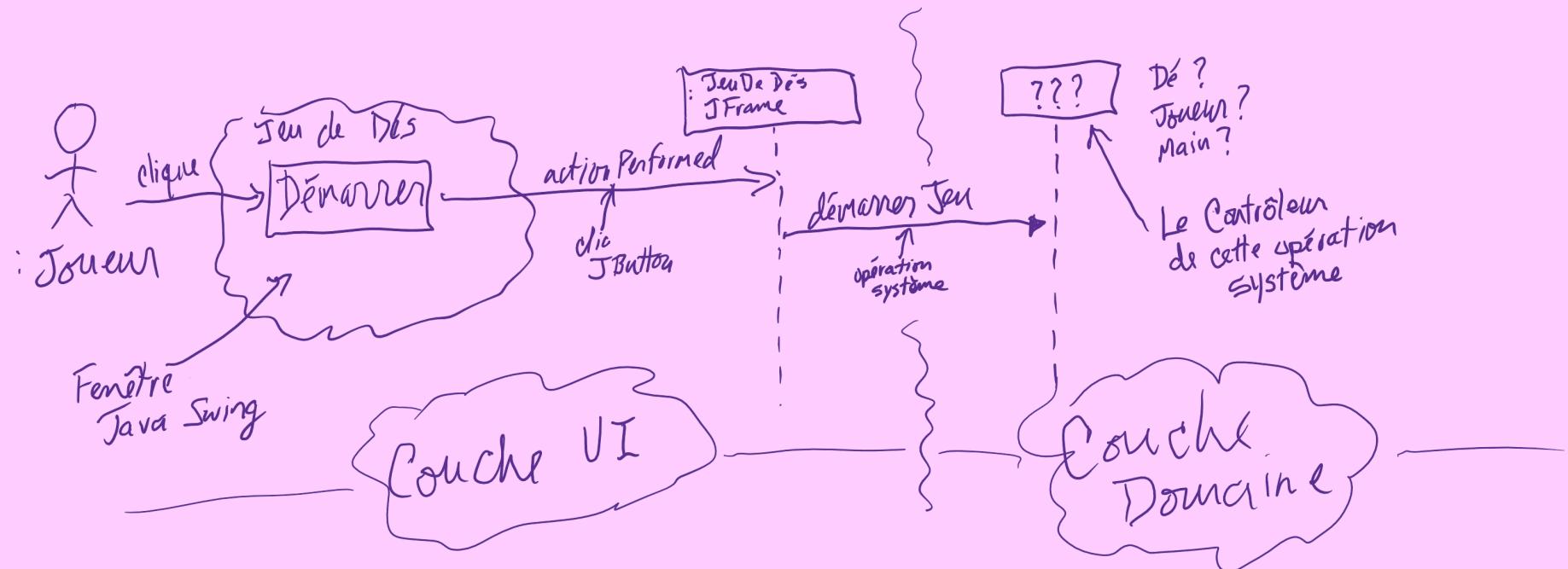


Qui envoie l'opération? Qui la reçoit?

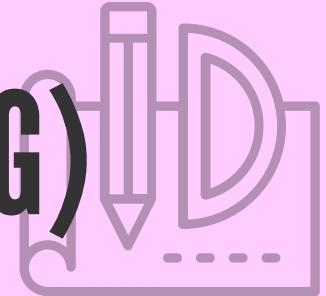
RDCU (SOLUTION JAVA SWING)



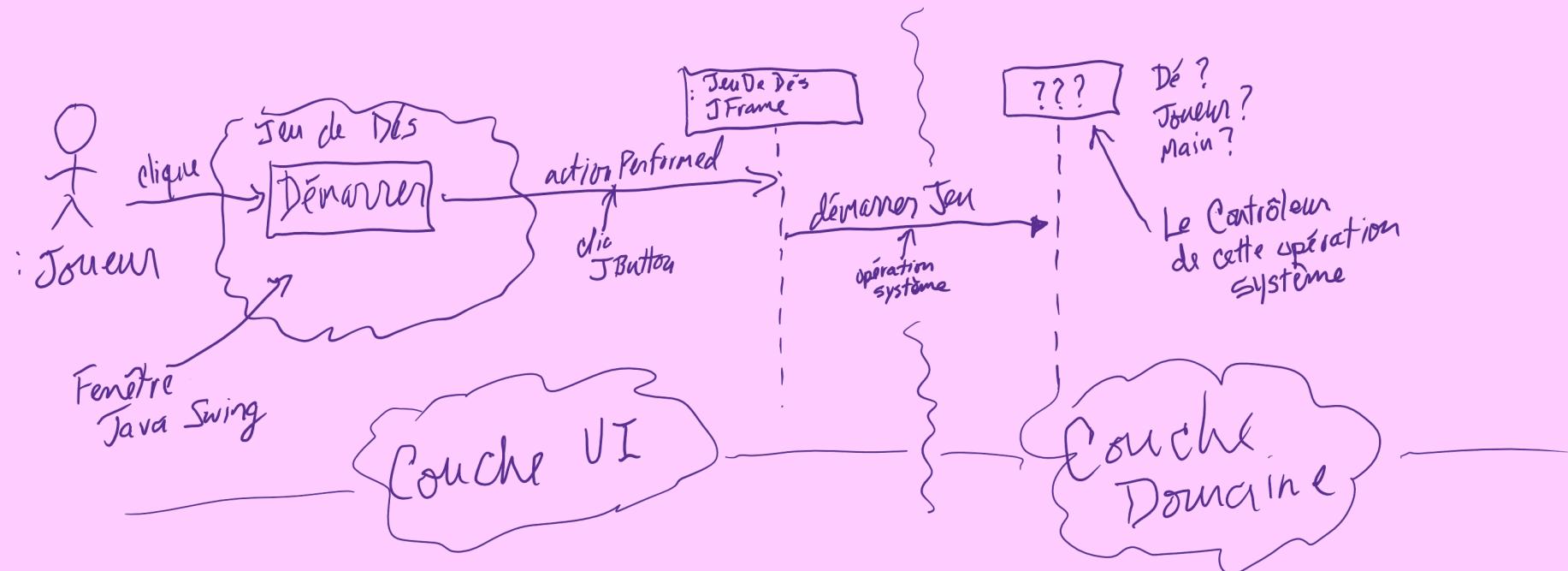
Opération démarrerJeu - qui envoie cette opération?

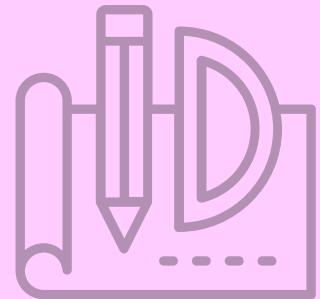


RDCU (SOLUTION JAVA SWING)



Opération démarrerJeu - qui reçoit cette opération?





PRINCIPE CONTRÔLEUR GRASP

Problème: Quel est le premier objet en dehors de la couche présentation qui reçoit et coordonne (« contrôle ») les opérations système ?

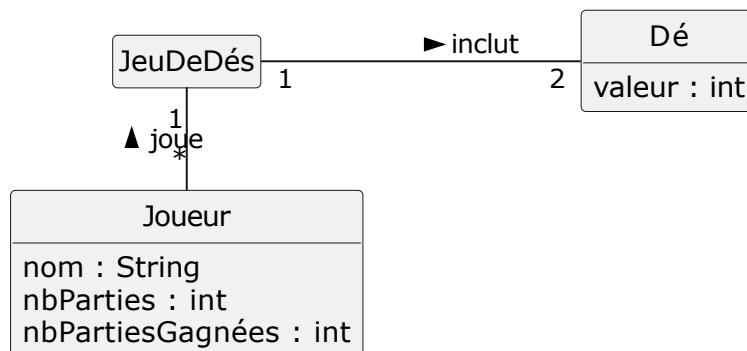
PRINCIPE CONTRÔLEUR GRASP



Solution: Affectez une responsabilité à la classe qui correspond à l'une de ces définitions:

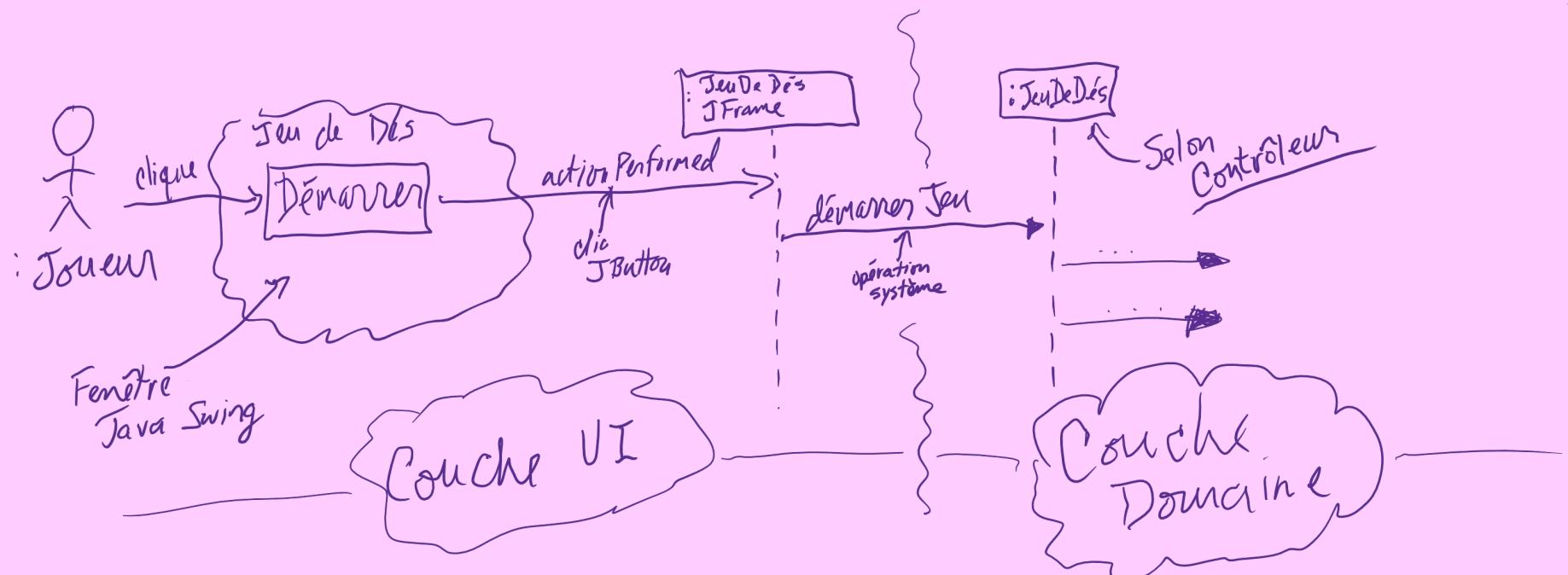
1. Elle représente le **système global**, un « **objet racine** », un **équipement** ou un **sous-système**.
- 2....

Modèle du domaine (adapté du Jeu de dés du Ch. 1 de Larman)



RDCU - CONTRÔLEUR

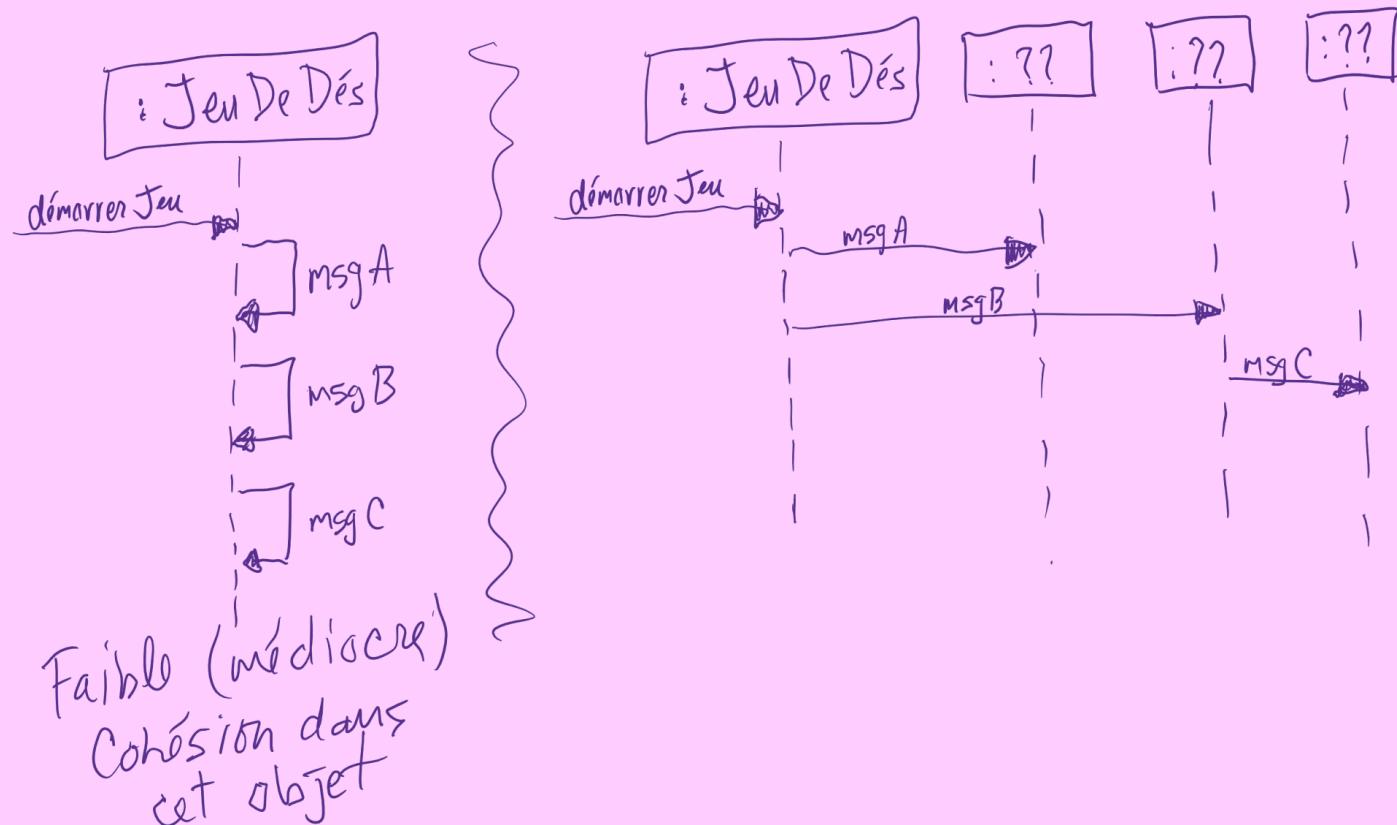
JeuDeDés est le contrôleur GRASP (inspiré du MDD)



RDCU - DÉTAILLÉ

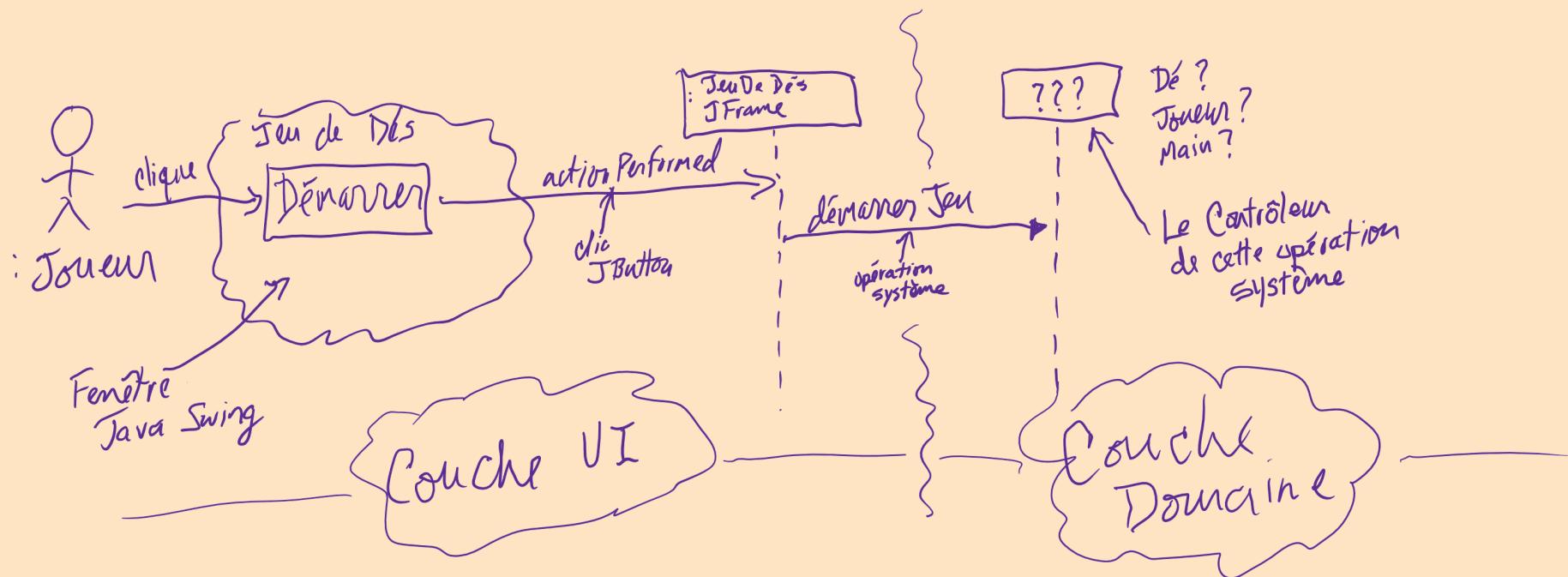


Faire un design modulaire et facile à comprendre.
Affecter les responsabilités aux bonnes classes.



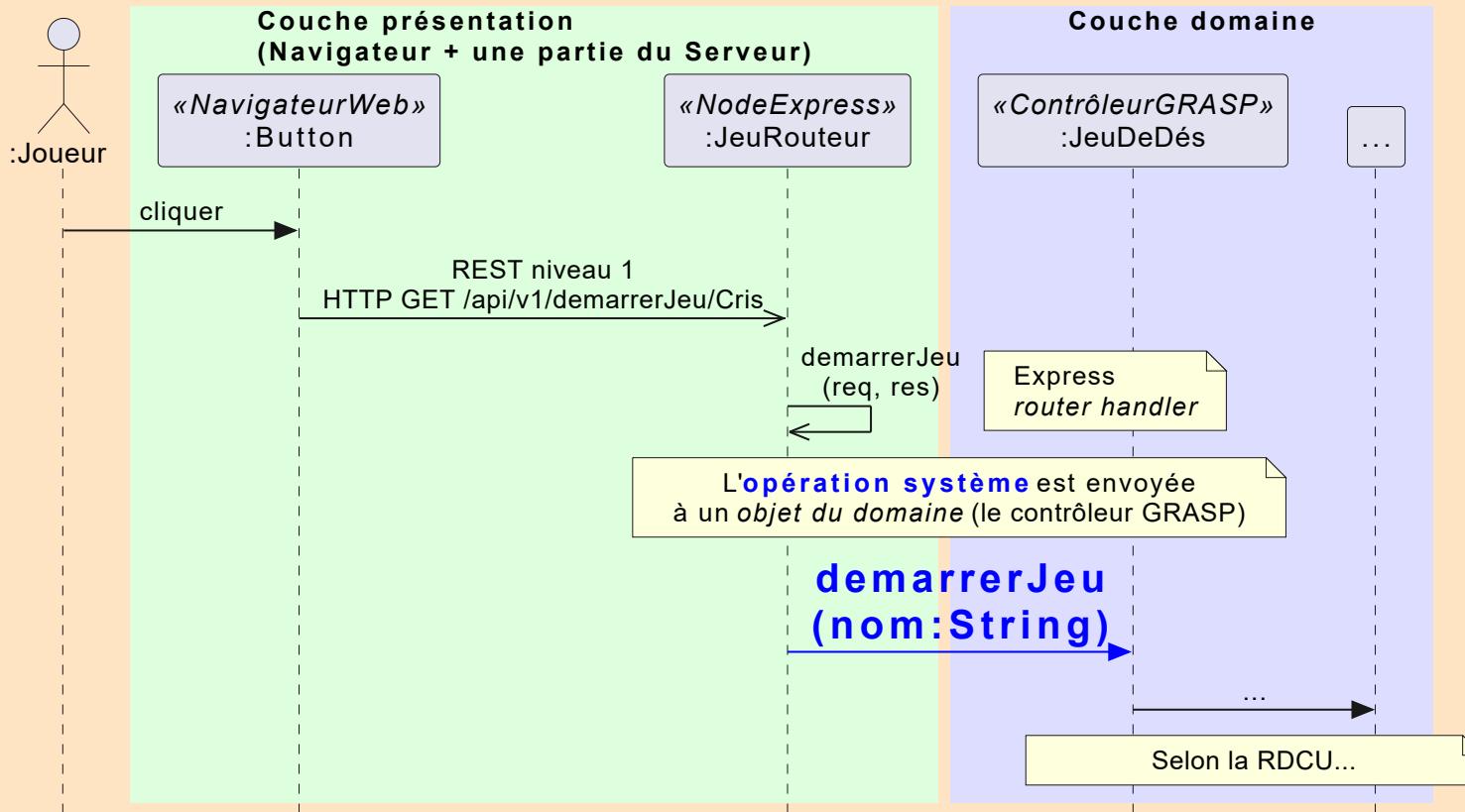
ARCHITECTURE LOGIQUE

Solution avec Java/Swing:



ARCHITECTURE LOGIQUE

Solution avec Navigateur/HTTP/ExpressJS/TypeScript:



COMPRENDRE ET RESPECTER L'ARCHITECTURE EN COUCHES DU LABORATOIRE

[https://log210-cfuhrman.github.io/log210-valider-
architecture-couches/](https://log210-cfuhrman.github.io/log210-valider-architecture-couches/)

RÉSUMÉ

- Rappel méthodologie (MDD, DSS, Contrats, RDCU)
- MDD: Catégories (exercice)
- DSS: Introduction
- Architecture en couches

FEUILLE D'UNE MINUTE

Pour dire ce qu'étaient les points les moins clairs de la séance.