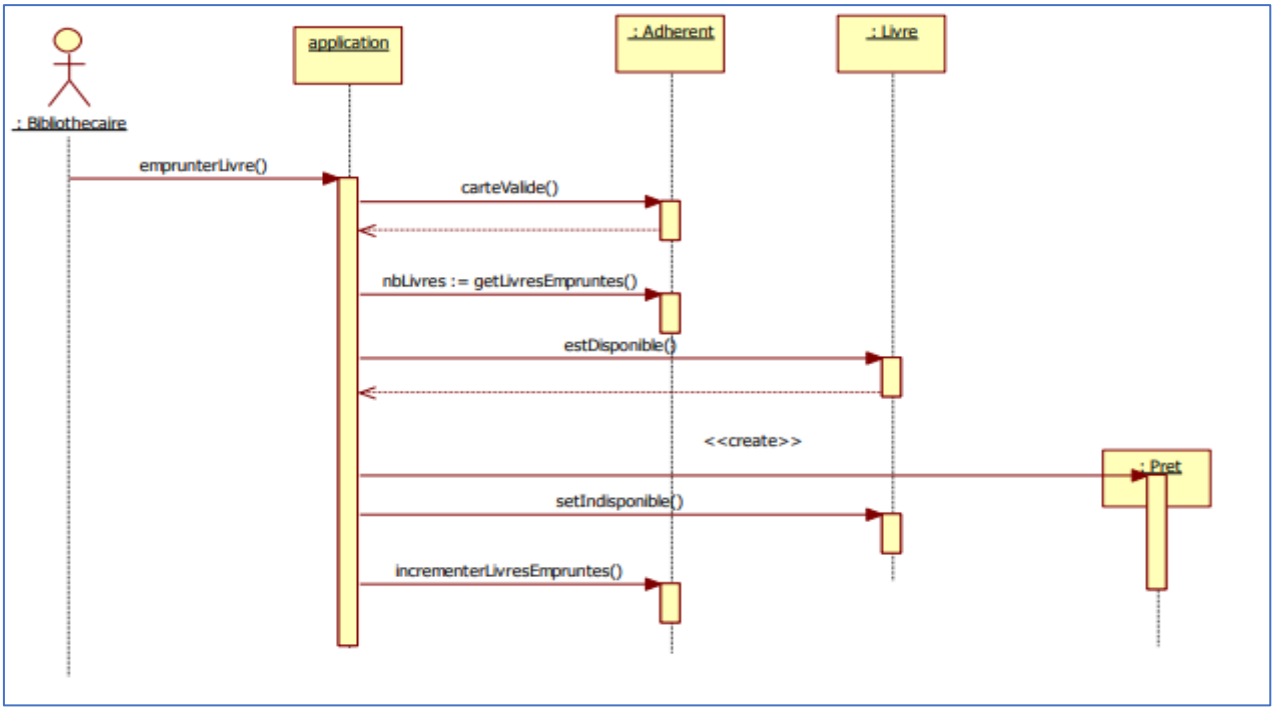
Université Larbi ben M'Hidi
Faculté des sciences exactes & sciences de la nature et de la vie
Département MI **2022/ 2023**
3ème Année Licence SI **Dr. Kouah Sofia**

TD 4 : Diagrammes d'interactions : Diagramme de Séquences et Diagramme de communication (Collaboration)

Exercice 1 : On s'intéresse à la **modélisation dynamique de la gestion d'une bibliothèque**. Pour emprunter un livre, on a le **scénario** suivant :

1) L'adhérent se présente au comptoir et la **bibliothécaire** saisit la fonctionnalité pour emprunter un **livre** de **l'application**.
2) D'abord, il faut vérifier si l'adhérent a le droit d'emprunter des livres (carte valide, nombre de livres déjà empruntés ne dépasse pas un seuil fixé, …).
3) En suite, il faut vérifier si le livre est disponible.
4) Si tout va bien, on crée un nouveau **prêt** avec la date de prêt et la date de retour, associé avec l'adhérent et le livre choisit.
5) On rend le livre indisponible.
6) On incrémente le nombre de livres empruntés par l'adhérent.
Question : Etablir le diagramme de séquences du scenario décrit ci-dessus en considérant seulement le cas nominal (on ne traite pas les cas d'erreurs, tels que : carte non valide, nbre de livres empruntés dépasse un seuil fixe, …)

Solution :



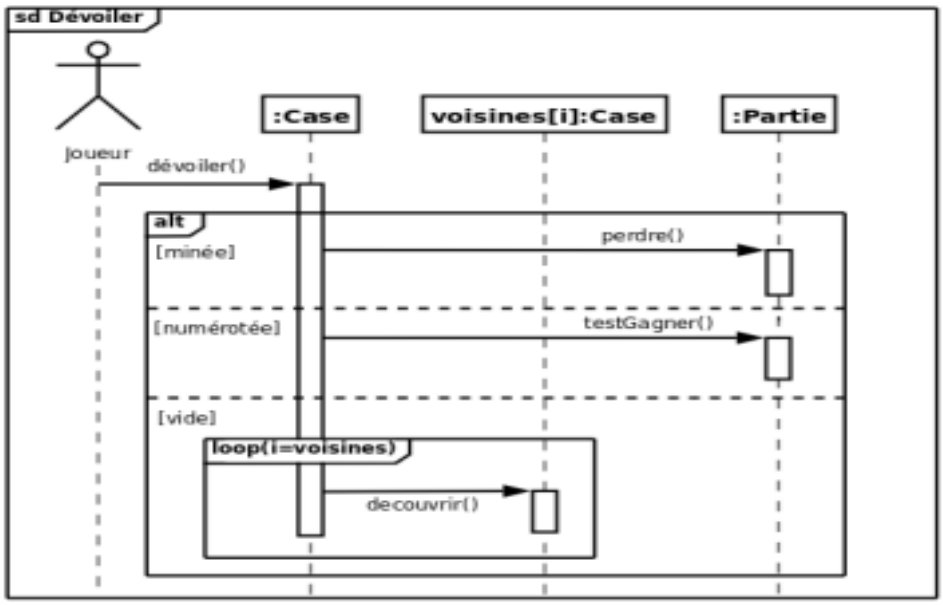
Explication :

- On distingue les rôles suivants : **Bibliothécaire**, **Application**, **Adhérent**, **Livre** et **Prêt**. On constate que les quatre premiers rôles existent dès le lancement du système (des instances qui existe déjà), par contre le rôle **Prêt** est créé lors de l'interaction et ce après la vérification de la disponibilité du livre. Donc, sa ligne de vie ne commence qu'après la demande de sa création.
- Les messages échangés correspondent aux invocations des méthodes correspondantes, donc ce sont des messages de type synchrone qui nécessite la réception d'une réponse afin de poursuivre l'interaction. Un message de réponse (message de retour) est représenté par une flèche en pointillées. Il est à noter qu'on peut omettre les messages de retour.

Exercice 2 : **Jeu de démineur**

Elaborer le **diagramme de séquence** quand un joueur **dévoile une case** :
- Si la **case est minée**, la **partie est perdue**
- Si la **case est numérotée**, il faut **tester si la partie est gagnée**
- Si la **case est vide**, on **découvre toutes les cases voisines**.

Solution :



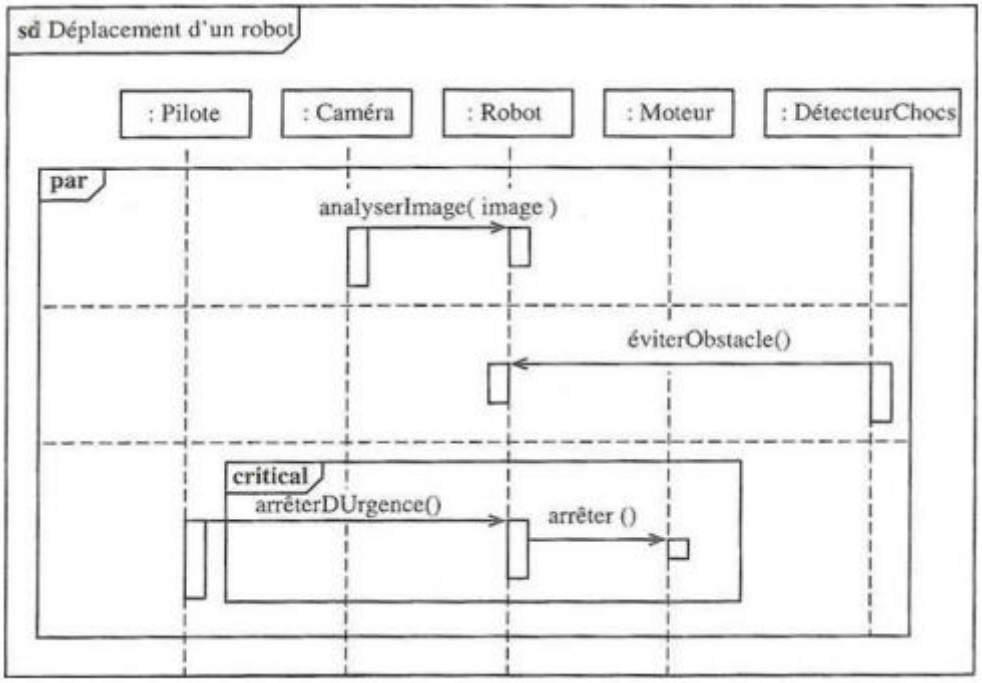
Explication :

- D'après l'énonce, il est clair que les différentes interactions décrivant le Jeu **s'exécutent de manière alternative**, **selon l'état de la case** qui peut être **: minée, numérotée ou vide**. Donc, on utilise l'opérateur d'interaction « **alt** » et on modélise chaque interaction par un **fragment ayant la condition correspondante**. Ces conditions portent sur l'état de la case ([minée], [numérotée] et [vide]). Par exemple si l'état de la case = minée (c a d [minée]), alors on exécute le premier fragment. Par contre, si l'état de la case = **numérotée** on exécute le deuxième fragment.
- Aussi, on distingue les rôles suivants : **Joueur**, **Case** et **Partie**. Pour les cases, on constate que le traitement pour une case vide, à un instant donné, concerne la case elle-même et ses voisines. Afin de différencier entre une case quelconque et ses voisines**, on nomme les cases voisine par voisine(i)**, tel que **i** représente le numéro de la case voisine parmi **n cases voisines**.
- Le traitement, dans le cas de la case vide, consiste à découvrir toutes les cases voisine, donc, c'est un **traitement itératif** dont le nombre d'itération est égale au nombre de case voisines. Pour modéliser **l'itération**, on utilise l'opérateur d'interaction « **loop** ». Le nombre d'itérations est modélisé par [**i=voisines**], tel que **voisines** est le nombre total des voisins.
- **« sd Dévoiler »** désigne que le diagramme s'agit bien du diagramme de séquences modélisant le **cas dévoiler du jeu**. Il peut être éventuellement réutilisé dans un autre diagramme de séquences en utilisant l'opérateur « **ref »** tout en mentionnant son nom dans le fragment lui-même sans redessiner l'interaction.

Exercice 3 (Optionnel): Modélisation du comportement d'un Robot

On veut modéliser le comportement d'un **robot** mobile, doté d'une **caméra** et d'un **détecteur de chocs**. Pendant son fonctionnement normal, le robot doit analyser l'image qui provient de la caméra et le détecteur de chocs doit lui permettre d'éviter les obstacles. En **cas d'urgence**, à tout moment un **pilote** humain peut faire arrêter le robot, ce qui entraine l'arrêt immédiat de son **moteur** : cette opération doit être accomplie de **façon atomique**. Élaborer le diagramme de séquences correspondant.

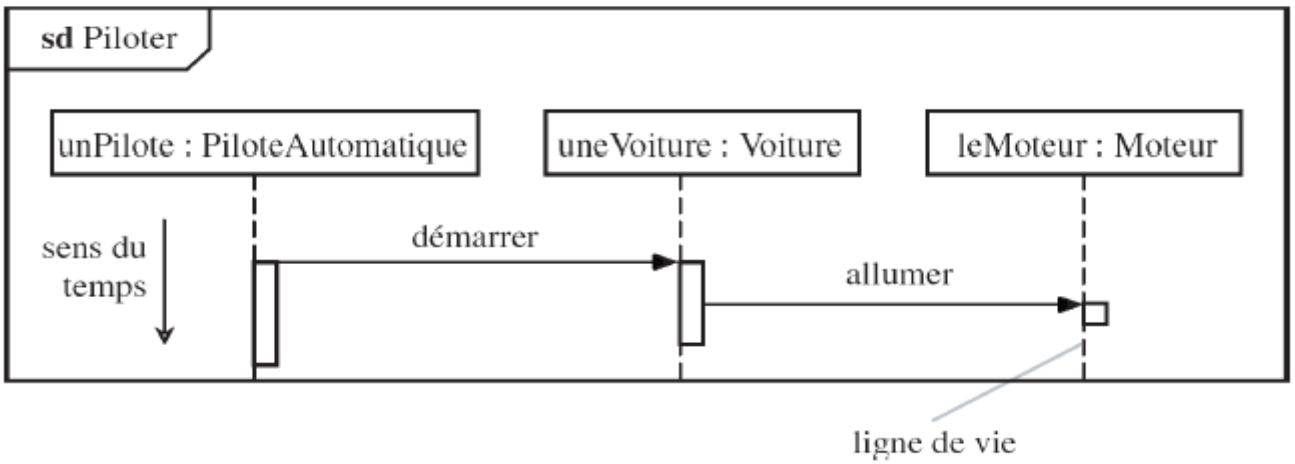
Solution :



Explication :

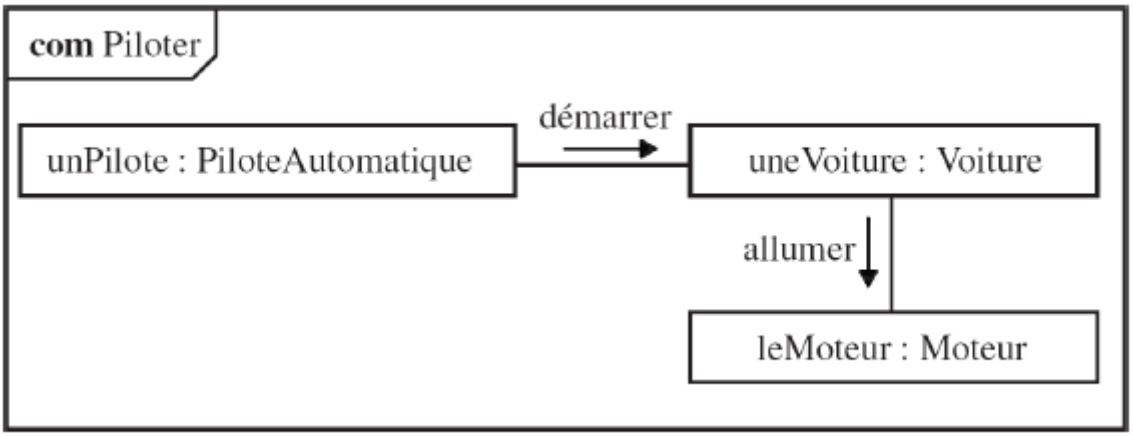
- **Désignation des Objets participants à l'interaction :** On distingue les rôles suivants : **Pilote**, **Camera**, **Robot**, **Moteur** et **DetecteurChocs**. Ces objets existent au départ, ils ne sont pas créés lors de l'interaction.
- **« sd Déplacement d'un robot »** désigne que ce fragment représente un **diagramme de séquences** nommé **Déplacement d'un robot.** Donc, il peut être éventuellement réutilisé dans d'autres interactions en utilisant l'opérateur d'interaction **ref** (référence) et en indiquant son nom a l'intérieur du fragment.
- D'après l'énoncé, on constate que **l'analyse de l'image** et la **détection de chocs** sont **réalisée en parallèle**, Le **parallélisme** est modélisé par l'opérateur d'interaction « **Par »**.
- En plus, en cas d'urgence**, une opération atomique** doit être déclenchée : le pilote doit arrêter le robot tout en arrêtant son moteur. Donc, **la demande d'arrêt doit être impérativement suivie de l'arrêt du moteur.** Pour représenter ça, on utilise l'opérateur « **critical »**, qui permet de définir **une région critique,** c'est-à-dire une région d'une interaction où les **traitements sont atomiques (indivisibles**)

Exercice 4 : Soit le diagramme de séquences suivant :



Question : Transformer ce diagramme de séquences en un diagramme de communication.

Solution :



Explication :

- On constate que le nom du diagramme est précédé par **« com »** ce qui indique que c'est un **diagramme de communication.**
- Les objets participants à la communication sont les mêmes du diagramme de séquences (**PiloteAutomatique**, **Voiture** et **Moteur**), sauf que leur représentation ne préserve pas la sémantique du temps, autrement dit : contrairement au digramme de séquences qui représente les interactions sous **un angle temporel en montrant** le séquencement temporel de messages, le diagramme de communication montre **une représentation spatiale des lignes de vie.**
- Le type de message est préservé, les messages « **démarrer »** et « **allumer »** sont des **messages synchrones**. Rappelons que **les messages de retour** peuvent être omis.

Exercice 5 :

Expliquez la syntaxe des messages suivants extraits d'un diagramme de communication.

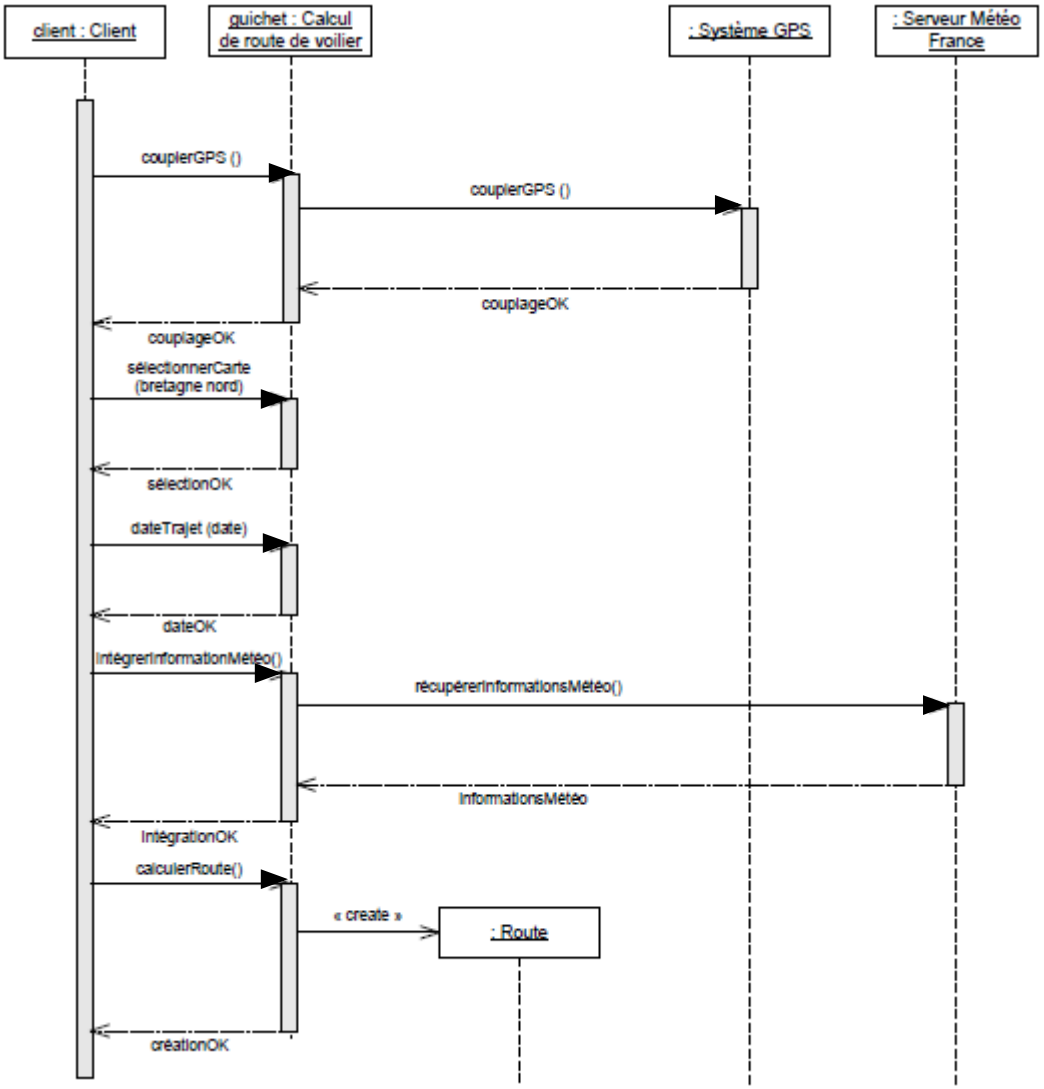
a) 2 : affiche(x,y).
b) 1.3.1 : trouve("Hadock").
c) 4 [x < 0] : inverse(x,couleur).
d) 3.1 *[i:=1..10] : recommencer().

Solution :

a) **2 : affiche(x,y)** : message simple, invocation d'une méthode.
b) **1.3.1 : trouve("Hadock")** : appel emboîté.
c) **4 [x < 0] : inverse(x,couleur)** : conditionnel, Si la condition [x<4] est vérifiée alors l'envoie aura lieu..
d) 3.1 *[i:=1..10] : recommencer() : itération, l'étoile désigne l'itération et la garde (la condition) [i := 1..10] désigne le nombre d'itérations.

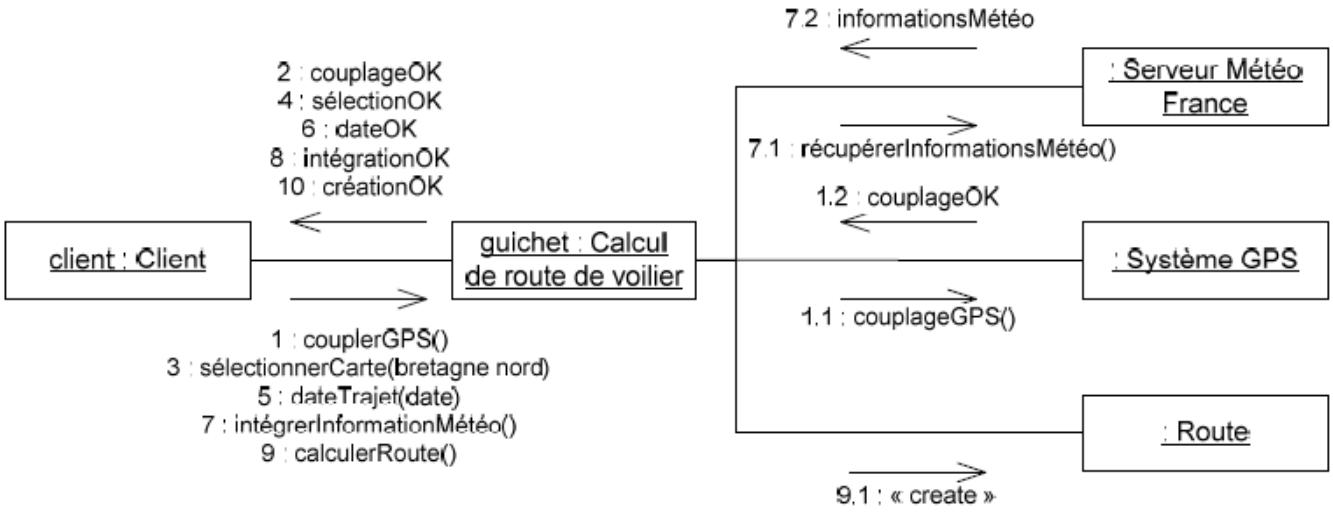
Exercice 6 (Optionnel):

1. Traduire le diagramme ci-après en diagramme de communication.
2. Représenter le diagramme de classe correspondant ainsi que les navigabilités.



Solution :

1. **Diagramme de communication**



- Les numéros de messages désignent leur séquencement.
- Qu'est-ce que vous remarquez ?

2. Diagramme de classes