

Activité intégrative 2021

Itération 2

Durée : 10 heures

Cette itération a pour but d'implémenter l'édition du livre et un mécanisme de synchronisation entre les superviseurs.

Planification de l'itération

1 : découverte et présentation de l'énoncé (30 min)

En grand groupe, le responsable présente les fonctionnalités attendues.

2 : ajout de la classe `EditSupervisorTests` (30 min)

Nous souhaitons mettre en place un mécanisme de synchronisation entre les superviseurs. Son principe est le suivant : le superviseur d'édition signale toute modification valide du livre à des objets intéressés. Dans notre cas, les objets intéressés seront les autres superviseurs. Ceux-ci implémentent à cette fin l'interface ci-dessous et sont transmis au superviseur d'édition au moment de la construction du système.

Définissez l'interface `gamebook.domains.BookEditedEventHandler`. Cette interface déclare une seule méthode `void onBookEdited()`.

Remplacez le package `gamebook.fakes` par celui disponible sous l'énoncé de l'itération 2 sur HELMo-Learn. Le package contient une classe `gamebook.fakes.FakeBookEditedEventHandler` qui est une implémentation factice de votre interface. En cas d'erreurs, vérifiez le niveau d'accès de votre interface, son nom, son package, etc.

Ajoutez le fichier `gamebook.supervisors.EditSupervisorTests`, disponible sur HELMo-Learn.

Modifiez la classe `ReadSupervisor` afin qu'elle implémente cette interface. L'implémentation consiste à rafraichir sa vue.

3 : conception de l'application (2 h 30 min)

Partez des modèles fournis ci-dessous et appliquez le [RDD](#) pour concevoir votre solution. Procédez en trois étapes :

1. Individuellement, élaborez une liste **de responsabilités candidates** (15 min).
2. En petits groupes (3 à 5 étudiants, si compatible avec les règles COVID-19 en vigueur), **complétez le** modèle de cartes CRC et les diagrammes de collaboration. **Créez un diagramme de classe qui met en évidence les méthodes publiques** (~1 h).
3. En grand groupe, vous discutez de solutions possibles (~1h 15 min).

Cette étape est également un moment idéal pour réfléchir à nouveau aux choix de collections opérés lors de l'itération 1 (interface et implémentation).

4 : implémenter et tester de l'application (6 heures)

Nous vous recommandons d'implémenter progressivement l'application, une user story à la fois.

AI-2.1 : éditer le titre

En tant qu'auteur, je souhaite éditer le titre, afin d'en trouver de meilleurs.

Détails

- Le titre d'un livre est défini et non-blanc (il ne peut pas être composé que de caractères d'espacements).
- La mise à jour du titre du livre se fait quand le champ de saisie perd le focus.
- Tout changement de titre provoque la remise à zéro de la lecture en cours.
- Il est certain que l'écran de vérification devra également être notifié de tout changement de titre

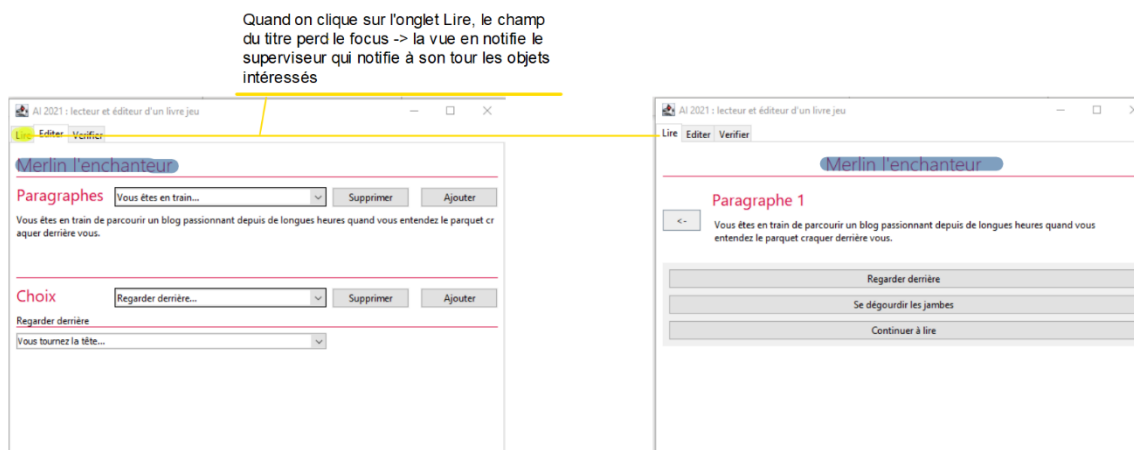


Figure 1 Comportement attendu lors d'une édition valide du titre

Tests d'acceptation

Remarque : Comme leur nom l'indique, les tests d'acceptation déterminent si votre programme fonctionne comme attendu. La classe `gamebook.superviseurs.EditSupervisorTests` implémente ces tests. **Nous l'utiliserons pour valider la partie fonctionnelle de l'itération.**

- ✓ Quand le programme a démarré, je peux voir le titre du livre, le contenu et les choix du paragraphe 1 dans la fenêtre d'édition.
- ✓ Quand je remplace le titre par « Merlin l'enchanteur », alors les objets intéressés sont notifiés de l'édition.
- ✓ Quand j'encode un titre blanc, alors la demande de modification est ignorée, le titre dans la vue d'édition est défini à la valeur précédente et les objets intéressés ne reçoivent aucune notification d'édition.

AI-2.2 : éditer les paragraphes

En tant qu'auteur, je souhaite éditer les paragraphes, afin d'améliorer le contenu du livre.

Détails

- On peut modifier le contenu d'un paragraphe (directement dans la zone de texte), ajouter de nouveaux paragraphes à la fin du livre (en cliquant sur le bouton d'ajout) et supprimer le paragraphe sélectionné (bouton « supprimer »).
- Le contenu d'un paragraphe ne peut pas être indéfini ou blanc. Un nouveau paragraphe a pour libellé de départ « Nouveau paragraphe » et ne comporte aucun choix.
- Toute demande de suppression est ignorée s'il ne reste plus qu'un seul paragraphe

- La suppression d'un paragraphe entraîne la suppression des choix où ce paragraphe apparaît
- Tout changement dans le texte d'un paragraphe provoque la mise à jour des 2 combo box de choix d'un paragraphe. Tout changement dans le texte d'un choix provoque la mise à jour de la combo box des choix.
- L'ordre des choix d'un paragraphe est quelconque.
- Tout changement apporté aux paragraphes provoque la remise à zéro de la lecture.

Tests d'acceptation

- ✓ Quand je choisis un paragraphe non-terminal, les choix du nouveau paragraphe sélectionné sont affichés dans la combo box, un premier choix est sélectionné (peu importe lequel) et le paragraphe de destination de ce choix est affiché sous le choix.
- ✓ Quand je choisis un paragraphe terminal, les choix du nouveau paragraphe sélectionné sont affichés (il n'y en a aucun) et il n'y a plus de choix sélectionné ni de paragraphe affiché
- ✓ Quand j'ajoute un nouveau paragraphe, ce dernier est ajouté à la fin du livre avec un contenu par défaut, il est directement sélectionné pour l'édition.
- ✓ Quand je mets à jour le contenu du paragraphe sélectionné, les paragraphes affichés sont mis à jour avec le nouveau contenu et **les objets intéressés sont notifiés.**
- ✓ Quand je supprime un paragraphe, la liste des paragraphes est mise à jour, le paragraphe sélectionné correspond à celui à la position du paragraphe supprimé, **et les objets intéressés sont notifiés.**
- ✓ Quand j'essaie de supprimer le seul paragraphe d'un livre, alors la demande est ignorée.

AI-2.3 : éditer les choix

En tant qu'auteur, je souhaite éditer les choix, afin d'améliorer les interactions du livre.

Détails

- On peut modifier le libellé d'un choix (directement dans la zone de texte correspondante), ajouter de nouveaux choix à un paragraphe (bouton « ajouter ») et supprimer le choix sélectionné (bouton « supprimer »).
- Le libellé d'un choix ne peut pas être indéfini ou blanc.
- Un paragraphe peut être sans choix (il est alors un paragraphe de fin).
- Un nouveau choix a pour libellé de départ « Nouveau choix » et cible le premier paragraphe du livre.
- Tout changement apporté aux choix provoque la remise à zéro de la lecture.

Tests d'acceptation

- ✓ Quand je choisis un choix, le libellé et le paragraphe ciblé affichés sont mis à jour.
- ✓ Quand j'ajoute un nouveau choix, la liste des choix est mise à jour, les données du nouveau choix sont présentées et les objets intéressés sont notifiés.
- ✓ Quand je supprime un choix, la liste des choix est mise à jour et les objets intéressés sont notifiés.
- ✓ Quand j'édite le libellé d'un choix, la liste des choix est mise à jour et les objets intéressés sont notifiés.
- ✓ Quand j'édite le libellé d'un choix avec un texte blanc, la demande est ignorée.
- ✓ Quand je change le paragraphe ciblé par un choix, les objets intéressés sont notifiés.

Consignes et contraintes

Algorithmique

Répondez aux questions ci-dessous en commentaires dans votre code source. Pour chaque question, nous indiquons l'endroit où la réponse doit être écrite.

Documentation de la méthode de suppression d'un paragraphe

Dans la Javadoc en tête de la méthode de Book permettant de supprimer un paragraphe.

Rédigez le vade-mecum de cette méthode.

OK

- La javadoc doit mentionner la description du **but** de la méthode, les **post-conditions** (= les conditions qui seront vérifiées par la classe Book après l'exécution de la méthode – Soyez complets !) et une explication des **étapes de l'algorithme** de cette méthode.
- Derrière les annotations @param, on doit retrouver une **description de chaque paramètre** mentionnant s'il est modifié par la méthode ou pas (**IN** = accès en lecture uniquement, **IN/OUT** = accès en lecture et modification) ainsi que les éventuelles **préconditions** qui le concernent
- Derrière l'annotation @return, s'il y a un résultat renvoyé, on doit trouver une description de ce résultat et des conditions qu'il respecte

CTT de la méthode de suppression d'un paragraphe

Pour répondre à cette question, identifiez les boucles, les imbrications, mais aussi les collections utilisées et leurs opérations. Quand vous répondez, n'oubliez pas d'indiquer à quoi correspondent vos libellés N,M, etc.

OK

Répondez à cette question dans la Javadoc de la méthode de Book permettant de supprimer un paragraphe.

Choix de collection pour les paragraphes

Réévaluez vos choix de collections (interfaces et classes concrètes) : sont-ils toujours pertinents pour réaliser efficacement l'itération 2 ?

Un livre jeu possède des paragraphes.

OK

Quelle interface de collection (List, Queue, Deque, (Sorted)Set, (Sorted)Map, etc.) allez-vous choisir pour mémoriser les paragraphes d'un livre jeu ? Justifiez votre choix en identifiant les principales opérations dont vous aurez besoins au cours des 2 premières itérations. Avez-vous notamment besoin d'accéder à un élément précis ? Si oui, sur base de quelle sorte de clé ?

Quelle implémentation de collection (ArrayList, LinkedList, ArrayDeque, HashSet, TreeMap, etc.) allez-vous choisir pour mémoriser les paragraphes d'un livre-jeu ? Justifiez votre choix en déterminant les CTT des principales opérations que vous utiliserez pour l'itération 1 **et la 2**.

Répondez à cette question dans la Javadoc en-tête de la classe GameBook. Si vous ne modifiez rien aux choix et justifications de l'itération 1, indiquez «Choix non modifiés » au début de la javadoc.

Choix de collection pour mémoriser les choix possibles à partir d'un paragraphe ?

Un paragraphe mémorise des choix possibles. Un choix est caractérisé par un texte et un paragraphe de destination (sa référence ou son indice ou ...).

OK

Quelle interface de collection (List, Queue, Deque, (Sorted)Set, (Sorted)Map, etc.) allez-vous choisir pour mémoriser ces choix ? Justifiez votre choix en identifiant les principales

opérations dont vous avez besoin au cours des 2 premières itérations. Avez-vous notamment besoin d'accéder à un élément précis ? Si oui, sur base de quelle sorte de clé ?

OK

Quelle implémentation de collection (`ArrayList`, `LinkedList`, `ArrayDeque`, `HashSet`, `TreeMap`, etc.) allez-vous choisir pour mémoriser ces choix possibles? Justifiez votre choix en déterminant les CTT des principales opérations que vous utiliserez pour l'itération 1 **et la 2**.

Répondez à cette question dans la Javadoc en-tête de la classe `Paragraphe`.

Programmation Orientée Objet

Déclarez et implémentez l'interface `BookEditedEventHandler` comme expliqué dans la section 2 : AJOUT DE LA CLASSE `EDITSUPERVISERTESTS` (30 MIN). Assurez-vous que le superviseur d'édition collabore avec les autres superviseurs à l'aide de références sur cette interface.

Notifiez les objets intéressés par tout changement apportés au livre (voir les tests d'acceptation).

Vous devez ajouter de nouvelles méthodes publiques à vos classes métiers. Essayez cependant de ne pas exposer trop de détails sur l'implémentation de vos classes (si une classe possède une `Map` parmi ses attributs, faut-il exposer un getter sur cette `Map` ?). Au minimum, assurez-vous que vos attributs d'objets restent privés.

Si vous leur trouvez une valeur ajoutée pertinente, n'hésitez pas à ajouter de nouvelles classes ou interfaces métiers.

Documents de départ pour la conception

Les documents suivants doivent vous aider à concevoir votre solution pour l'itération 2.

Conseil : concentrez-vous sur les récits utilisateurs présentés dans ce document et à ne cherchez pas à anticiper des fonctionnalités qui n'ont pas été encore présentées.

Modèle de cartes CRC

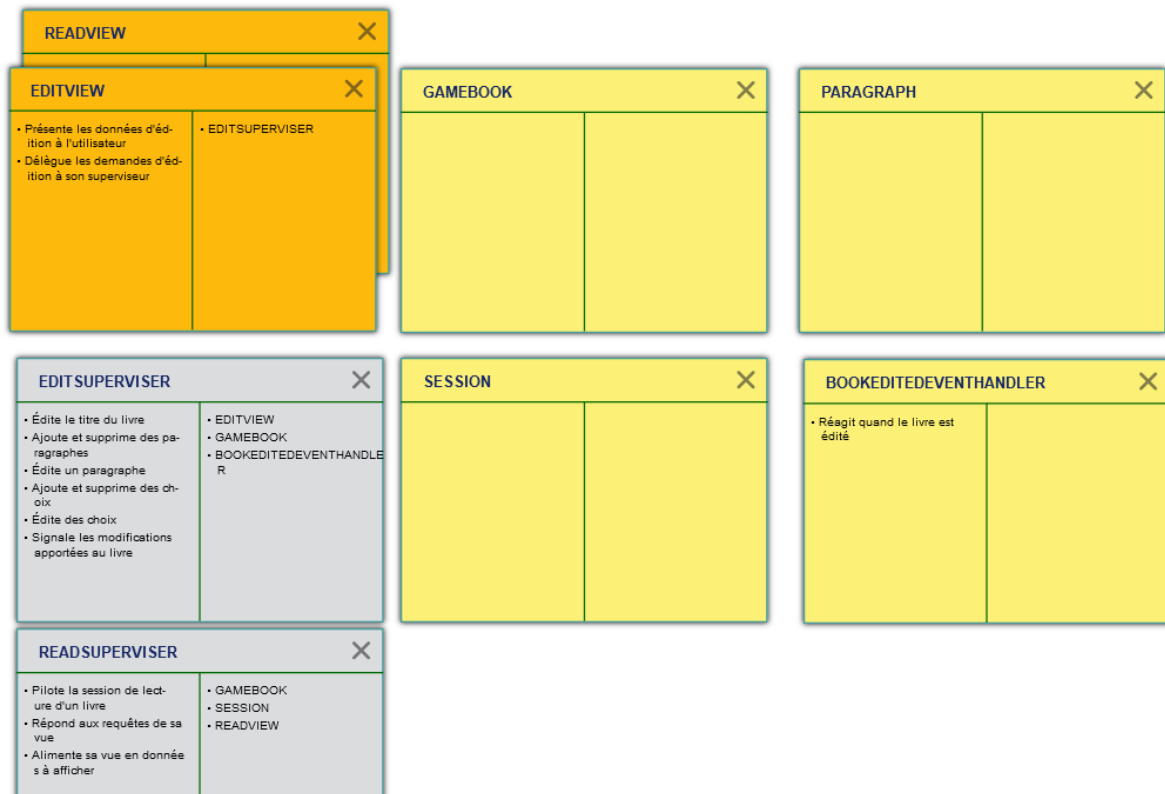


Figure 2 Squelette de modèle de cartes CRC à compléter

Diagrammes de collaboration

Complétez les diagrammes de collaboration ci-dessous pour en déduire des méthodes à déclarer dans vos classes. **Attention**, il se peut que certains objets soient inutiles.

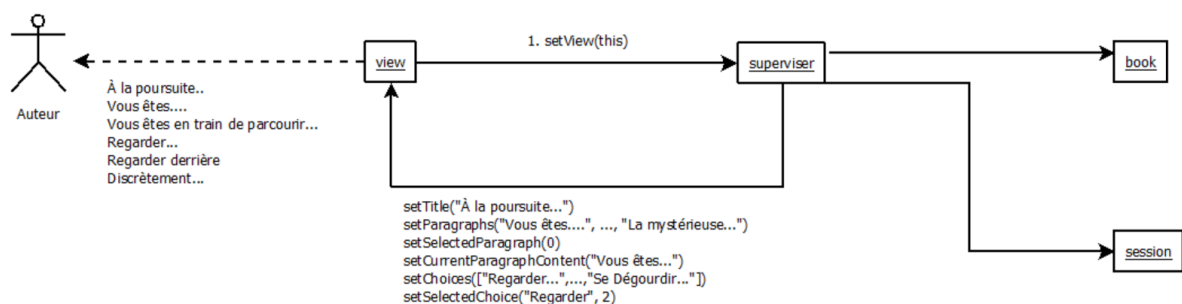


Figure 3 squelette à compléter pour afficher le contenu de l'onglet d'édition

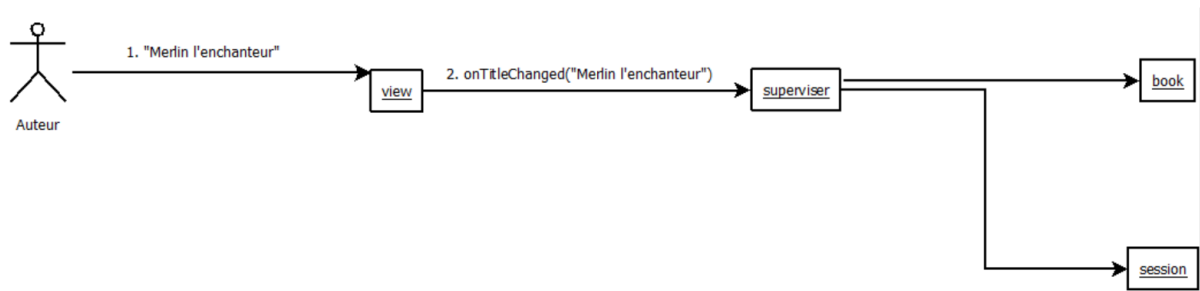


Figure 4 squelette pour éditer le titre avec le texte « Merlin l'enchanteur »

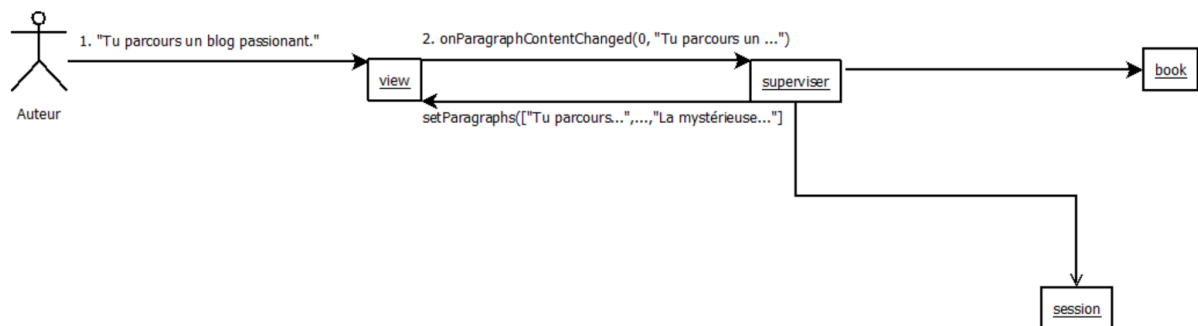


Figure 5 squelette pour éditer le paragraphe 1

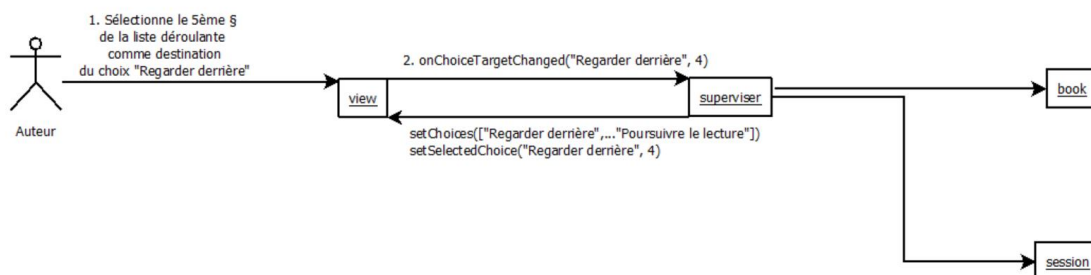


Figure 6 squelette pour changer la cible du choix « Regarder derrière » de paragraphe 1