

Planification du projet

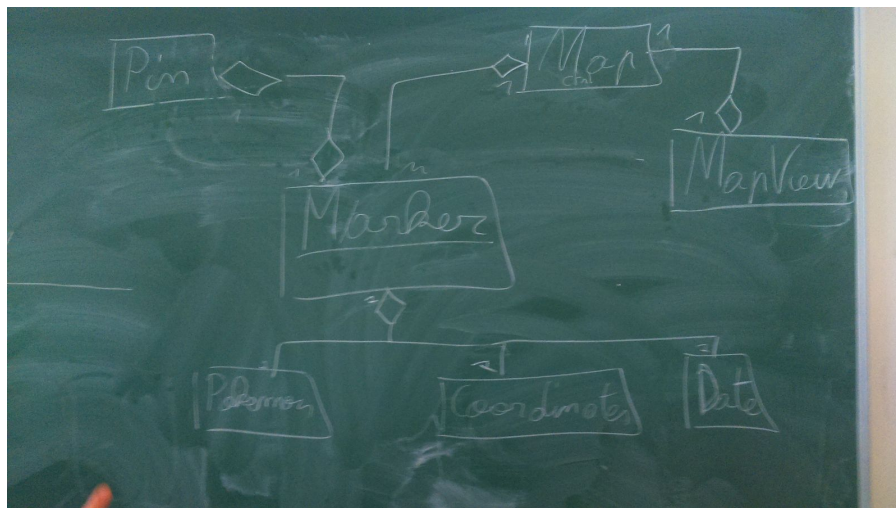
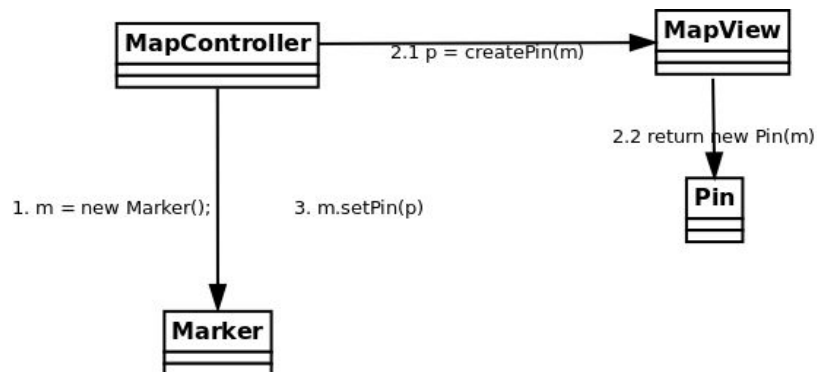
Dans l'histoire une, toutes les sous-histoires ont été implémentées sauf le *grouping* lors du "dézoom" (comme convenu avec le client). De plus, la fonctionnalité de "zoom/dézoom" n'est, à l'heure actuelle, pas fonctionnelle.

Les tâches à accomplir étaient les suivantes :

- Afficher des épingles
- Pouvoir sélectionner des épingles pour voir leurs informations (nom du pokémon et date de parution du pokémon).
- Ajout d'épingle sur la carte (avec complétion des informations : nom et date de parution)
- Système de "zoom/dézoom"

Structure actuelle et choix techniques

Pour le moment, l'architecture générale suit le modèle de classe suivant :



Le modèle ci-dessus ne contient pas tous les widgets ayant été créés.

Le modèle suivant le schéma MVC, nous avons réparti les classes de la manière suivante :

Modèle	Vue	Controller
	MapView	MapController
	Pin	Marker
		Pokemon
		Coordinates
		Date

De manière générale, il a été décidé que les classes faisant parties de la vue ne contiendraient aucune informations et que ces dernières seraient contenues dans les contrôleurs qui leur sont associés.

À titre d'exemple, la *MapView* affiche bien l'image désirée mais ne connais aucunement le chemin menant à l'image. En conséquence, MapView doit faire appel à une méthode de MapController pour obtenir le chemin vers l'image.

De plus, il a été décidé pour des raisons de mise à jour graphiques que ce sont les contrôleurs qui vont indiqué aux vues quand et où dessiner les widgets. Pour cela, les vues possèdent des attributs et des méthodes static.

Difficultés techniques rencontrées

Lors de cette partie, les principales difficultés qui ont été rencontrées étaient liées à l'utilisation correcte de l'API JavaFX. Typiquement, beaucoup de temps a été perdu lors de la conception des fenêtres *pop-up* car l'équipe ignorait qu'elles étaient les widgets les plus adaptées ainsi que les fonctionnalités déjà disponible.

Du point de vue de la conception des classes, des modifications tardives ont dû être effectuées pour permettre d'incorporer les éléments de zoom au MapView. De plus la zone du plan à afficher est difficile à modifier en fonction du facteur de zoom et du point du plan à conserver lors de la translation.