

GeoCatching

Rapport rédigé par:

- Laura Daufeld
- Vincent De Robert
- Loïc Mennella
- Marine Landraudie
- Ludivine Fafournoux
- Jérémy Paroux

LP SIL DAM

**Université de Nice - Sophia Antipolis
Département Informatique**

Table des matières

Organisation de l'équipe	3
Explication des spécifications fonctionnelles	3
Architecture de notre solution	4
Présentation incluant les diagrammes UML	4
Cas d'utilisation	4
Diagramme d'activité	4
Diagramme de classe	5
Avantages	6
Inconvénients	6
Justification de choix conceptuels	7
Rétrospective sprint #2 / #1	7
Perspectives	9
Améliorations	9

Organisation de l'équipe

Pour le premier sprint, nous avons sommairement défini les tâches dans le Kanban de Jira, puis nous avons tous ensemble réparti les tâches à réaliser à chacune des personnes sur l'ensemble du sprint #1.

Pour le second Sprint, il a été décidé de scinder notre équipe en deux parties. Trois personnes se sont occupées de finir le sprint #1 avec la finalisation du jeu (Loïc, Vincent, Jeremy). Trois autres personnes ont alors développé la fonctionnalité unique qui a été choisie.

Nous avons essayé cette méthode de travail afin de travailler de manière plus efficace et d'éviter trop de conflits au niveau du code.

Explication des spécifications fonctionnelles

L'application permet à un joueur connecté d'accéder à une carte Google Maps et de jouer à capturer des zones.

Un joueur non connecté ne peut accéder qu'à l'interface de connexion, il faut donc demander la création de compte à un administrateur du jeu. Une fois les identifiants obtenus, le joueur peut passer l'étape de la connexion, qui passe par un service externe d'authentification.

Le joueur a la possibilité de choisir parmi plusieurs choix de jeu. Il peut créer une partie, pour cela il devra renseigner plusieurs informations obligatoires comme le nom de la partie, sa date de fin, le nombre d'équipes pouvant y participer.

Il pourra ensuite créer un terrain de jeu en plaçant au moins trois marqueurs sur la carte. S'il se trompe, il a la possibilité, en maintenant le clic sur le marqueur, de le déplacer ou de le supprimer. Dès qu'il a le terrain qu'il souhaite, il doit le valider en cliquant dessus afin de pouvoir passer à la création de zones. Il les crée de la même manière que le terrain. Le joueur ne peut pas mettre un marqueur en dehors du terrain.

Une fois les zones de jeu validées en cliquant simplement dessus, il rejoint directement sa partie.

S'il le souhaite, le joueur peut rejoindre une partie déjà créée par un autre joueur. Il aura donc accès au terrain et aux zones créées. Il ne pourra cependant pas en créer.

Peu importe le mode de jeu qu'il a choisi parmi les deux cités précédemment, le joueur peut capturer une zone s'il est dans celle-ci en réussissant un mini-jeu. Il consiste à reproduire une séquence aléatoire montrée par l'ordinateur. Une fois la zone capturée, elle devient verte.

Le joueur peut choisir le dernier mode de jeu, qui est le mode libre, celui-ci lui permet de voir tous les joueurs connectés, chacun représenté par un marqueur.

Architecture de notre solution

Présentation incluant les diagrammes UML

Voici les schémas de l'application qui ont permis de développer plus rapidement mais aussi permet à chacun d'avoir une vision globale du fonctionnement. Nous retrouverons trois type de schéma : Cas d'utilisation, diagramme d'activité et diagramme de classe.

Cas d'utilisation

Il représente sommairement les actions que peuvent faire les types d'utilisateur

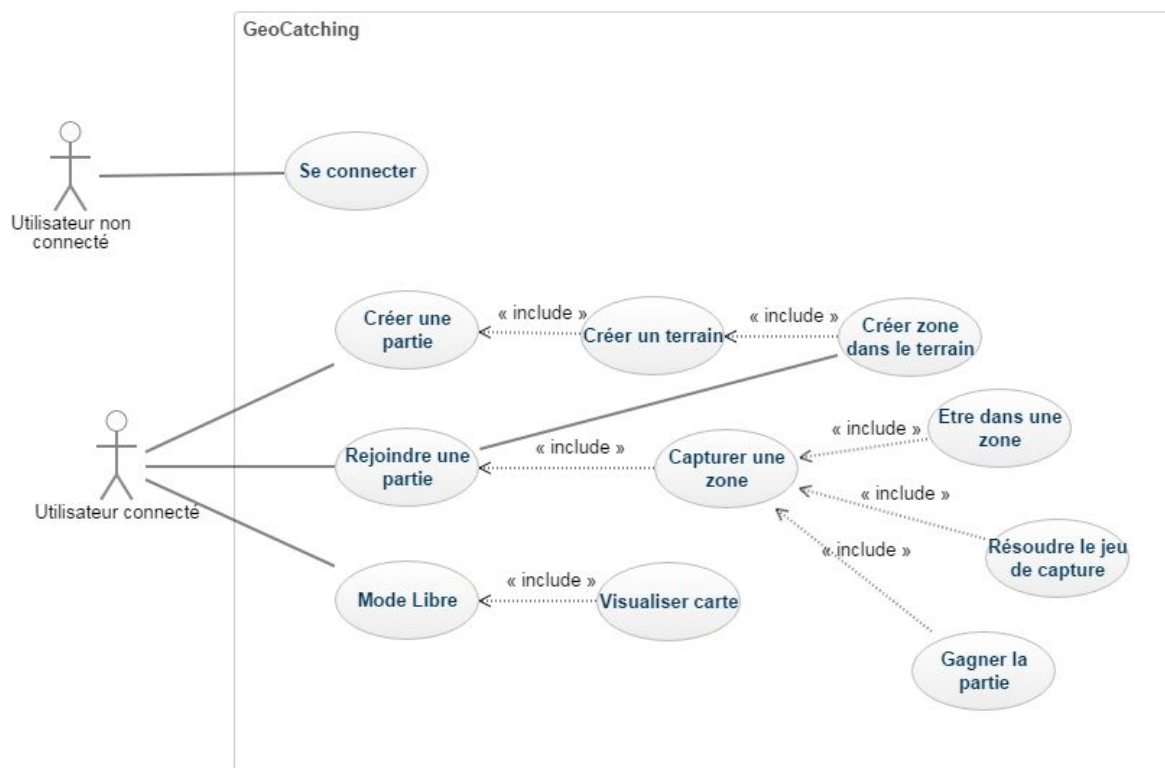


Illustration 1: Usecase de l'application

Diagramme d'activité

Ce diagramme permet d'avoir une arborescence des fonctionnalités de l'application.

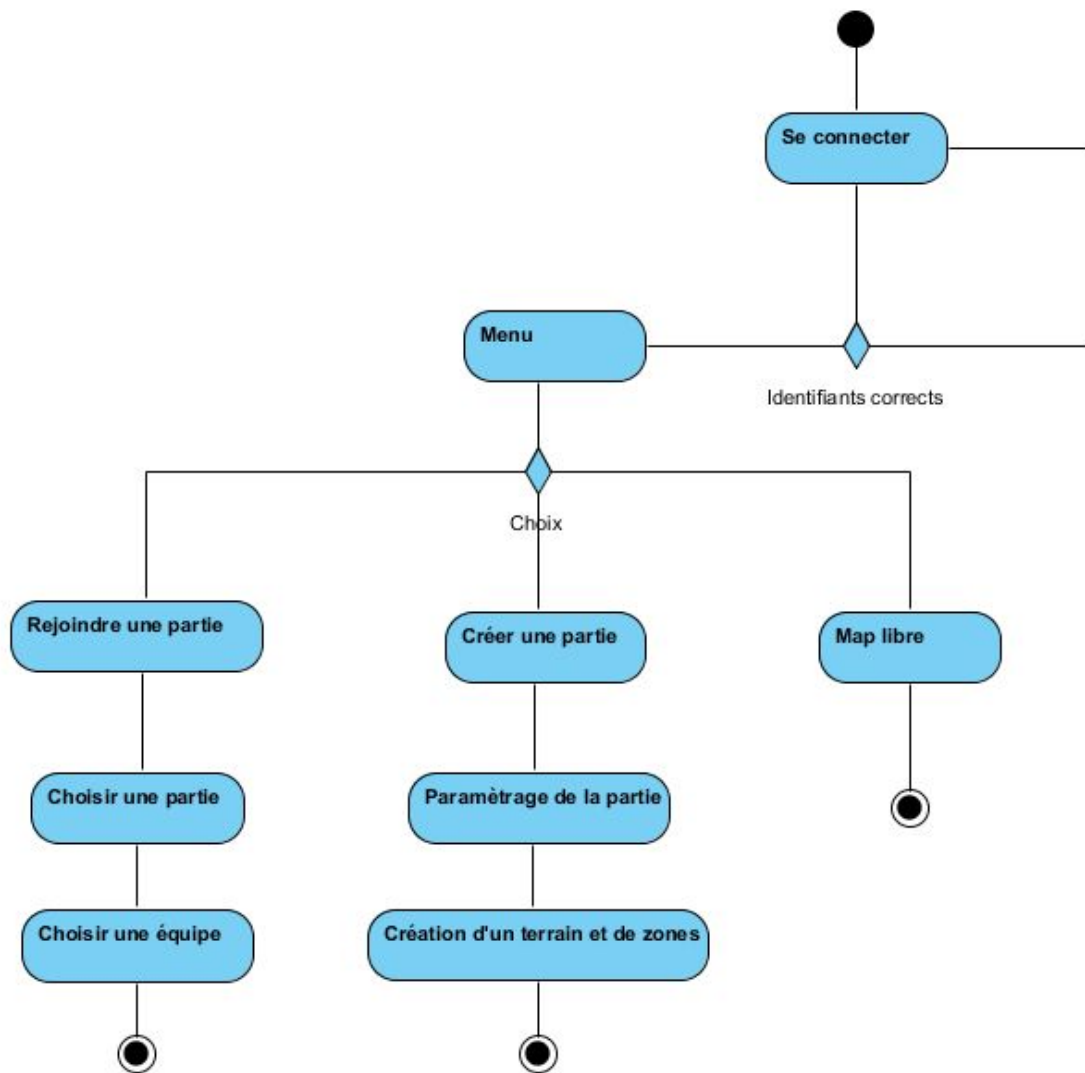


Illustration 2: Diagramme d'activité de l'application

Diagramme de classe

Ce type de schéma sert à comprendre l'architecture du code et servir de modèle pour développer l'application

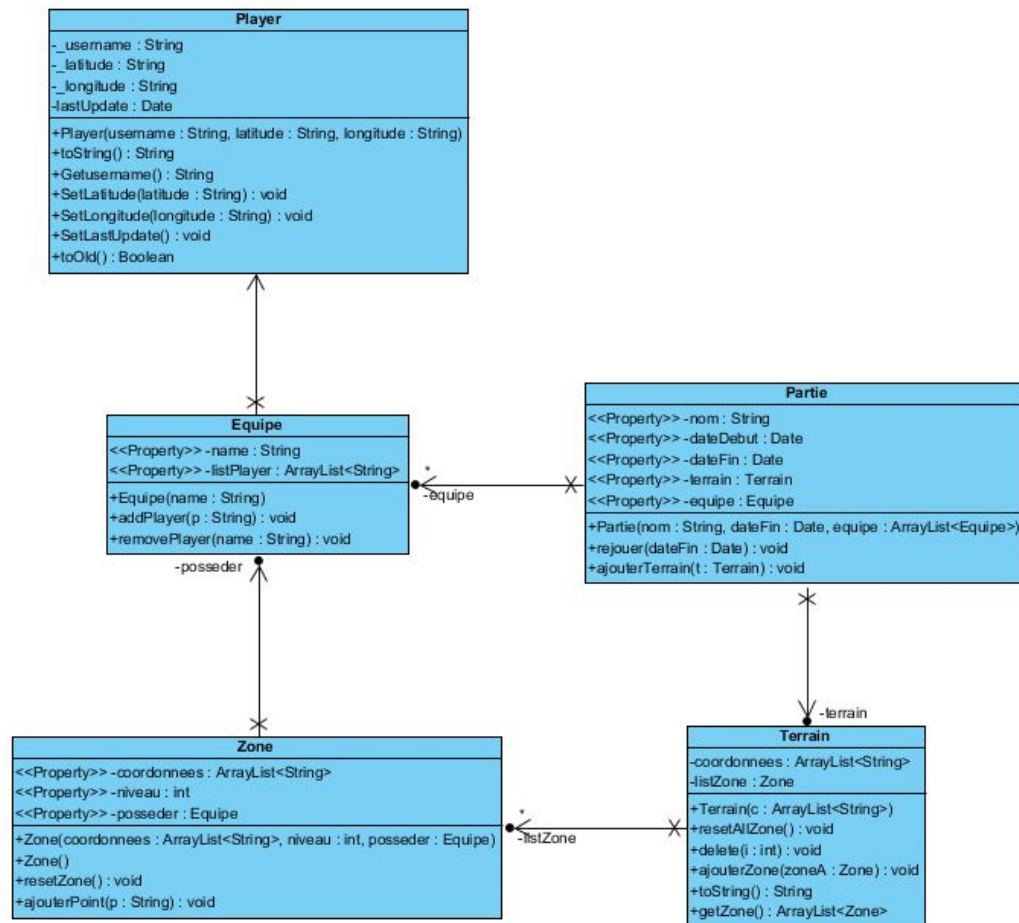


Illustration 3: Diagramme de classe de l'application

Avantages

Notre valeur ajoutée réside dans le jeu qui permet de capturer la zone. En effet, nous avons un jeu qui est totalement fonctionnel, comme nous le souhaitions.

Inconvénients

Notre jeu fonctionne bien, mais des fonctionnalités additionnelles, qui étaient prévues au départ, n'ont pas été mises en place comme:

- Consulter ses informations de joueur,
- Consulter les informations sur les autres joueurs,
- Pouvoir exclure un joueur de l'équipe,
- Renforcer une zone,

- Savoir à qui appartient la zone.

Lors du développement, nous avons remarqué que la connexion au webservice étant assez lente, ce qui ralentit le rafraîchissement des marqueurs de chaque joueur.

Justification de choix conceptuels

L'application est conçue avec le design pattern MVVM (Model View ViewModel).

Ce choix se justifie par la fréquence à laquelle il est utilisé pour le développement d'applications mobile. La différence avec le modèle MVC (Model View Controller) réside dans le fait qu'il alloue un contrôleur à chaque vue ce qui allège la communication entre la vue et le modèle.

Il permet également de faciliter la maintenance et la lisibilité du code.

Rétrospective sprint #2 / #1

A la fin du sprint 1, nous avons remarqué que nous rencontrons certains problèmes:

- de communication,
- d'organisation,
- d'utilisation des outils mis à notre disposition
- techniques.

Au cours de ce sprint #2, nous devons corriger ces problèmes afin que le projet se déroule mieux et que notre travail soit de meilleure qualité.

Nous avons globalement plus utilisé les outils collaboratifs. Une amélioration significative est visible notamment sur Jira et Git. Comme on peut le voir sur l'illustration 1, les commits sur la branche "develop" que nous avons mis en place à la fin du sprint #1, sont fait des manière régulière contrairement à ce qui a été fait lors du sprint #1 (cf. Illustration 2). Une application dans le projet



Illustration 4: graphique de l'activité sur la branche "develop" lors du sprint #2



Illustration 5: graphique de l'activité sur la branche "master" lors du sprint #1

Comme nous pouvons le voir sur l'illustration 3, nous avons beaucoup plus utilisé Jira dans le but de mieux se répartir les tâches et d'éviter de faire une tâche plusieurs fois comme cela a été le cas lors du sprint #1. Cet outil nous a permis de voir que tout au long du projet, la répartition des tâches n'a pas toujours été équitable.

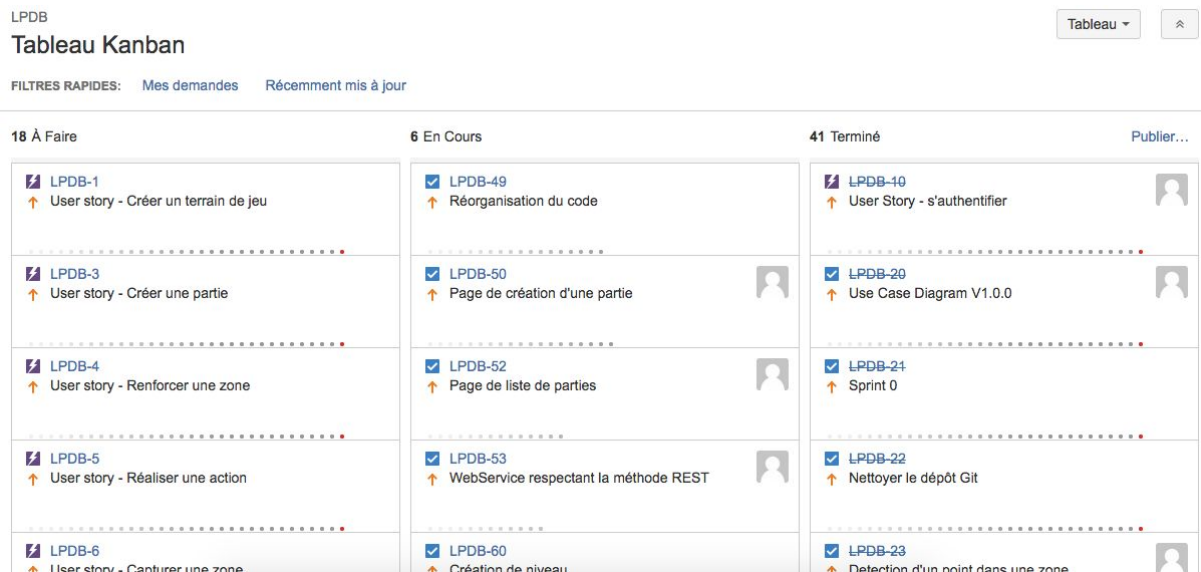


Illustration 6: tableau Kanban de notre projet sur Jira

Nous n'avons, malheureusement, pas résolu certains problèmes que nous avons remarqués lors du précédent sprint. Bien que nous ayons mis en place une conversation de groupe sur Facebook dans le but de faciliter les échanges étant donné que nous sommes en alternance, nous avons du mal à converser.

Le projet a cependant nettement évolué entre les deux sprints, nous avons dû repenser toute notre architecture afin de rendre le jeu fonctionnel.

Voici une comparaison des diagrammes de classes entre les deux sprints :

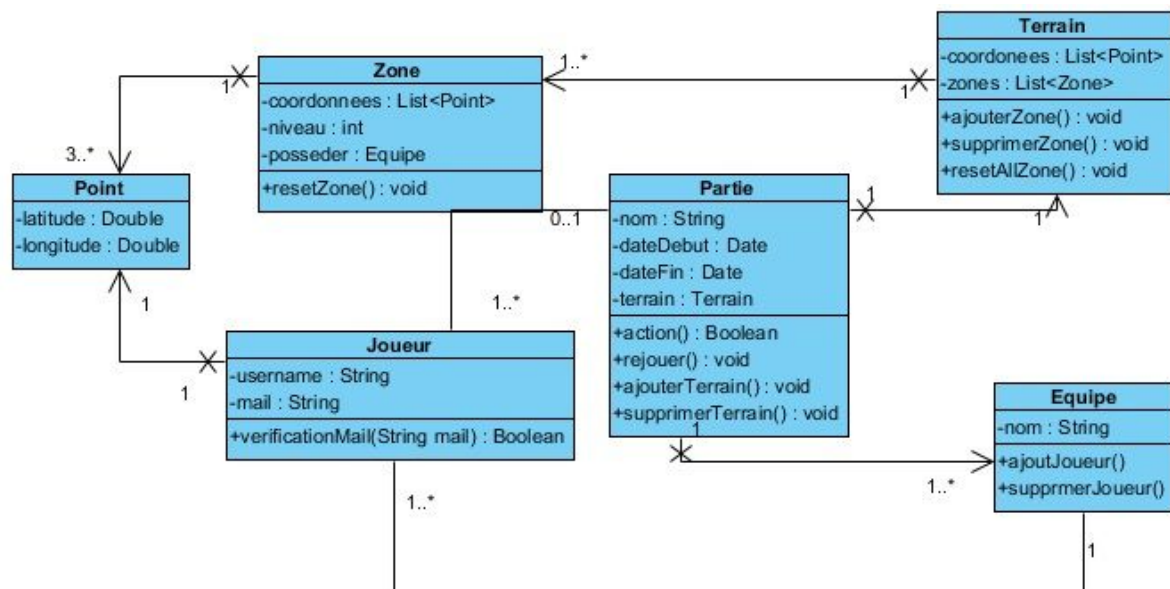


Illustration 7: diagramme de classes lors du sprint 1

Perspectives

Améliorations

Dans un premier temps, les fonctionnalités prévues, au début du projet, devraient y être ajoutées. Elle sont:

- Consulter ses propres informations,
- Consulter les informations des autres joueurs,
- Pouvoir exclure un joueur de l'équipe,
- Renforcer une zone,
- Savoir à qui appartient la zone.

Actuellement, lorsqu'un joueur veut capturer une zone, il le peut que dans le cas où il gagne la partie dans le mini jeu de capture. Il serait intéressant de changer la difficulté de ce jeu en fonction du nombre de fois où une zone a été capturée (plus une zone a été capturée, plus il sera difficile de la capturer à nouveau).

Des améliorations au niveau de l'interface peuvent être considérées étant donné qu'elle est assez basique, mais cela ne gêne en rien l'utilisation. Mettre en place des maker personnalisés sur la carte pourrait renforcer l'identité visuelle de l'application.

On peut citer par exemple le fait qu'un navigation drawer soit intégré et que l'on puisse accéder aux actions plus rapidement.

De nouvelles fonctionnalités peuvent être ajoutées comme la gestion des scores individuels et par équipes avec un classement.