

Projet Transpromotion 2022

# Baroudeurs



1ère année : Marie Apparailly, Vanessa Pellefigue, Matéo Moreau, Juliette Meunier

2ème année : François Domecq, Baptiste Laran, Corentin Léger, Milo Toumine

# Sommaire

<b>Introduction</b>	<b>4</b>
<b>Gestion de projet</b>	<b>5</b>
I.1. Organisation générale	5
I.2. Ressources	6
Collaboration	6
Développement	6
Création	7
I.3. Plannings	8
Planning prévisionnel	8
Planning effectif	9
<b>Pôle UX</b>	<b>11</b>
Introduction sur la partie UX	11
II.1. État de l'art	11
Applications de découverte	11
Jeux vidéos	12
Gamification	14
II.2. Entretiens	20
Objectifs et méthode	20
Conclusion	22
II.3. Elaborations des fonctionnalités de l'application	22
Brainstorming	23
Spécification des fonctionnalités	24
II.4. Questionnaire	26
Objectifs et méthode - le concept des marqueurs	26
Résultats du questionnaire	27
Design final	30
II.5. Maquettage	30
Conclusion et acquis pédagogiques	31
<b>III. Pôle Développement</b>	<b>32</b>
Introduction sur la partie Dev - Objectifs et Méthodes	32
III.1. Phase d'apprentissage	32
III.2. Modélisation	33
III.3. Programmation du prototype	38
Outils utilisés	38
Exigences fonctionnelles	38
Résultats	40

Conclusion et Acquis pédagogiques	44
<b>Conclusion</b>	<b>45</b>
<b>Bibliographie</b>	<b>46</b>
Développement	46
État de l'art	46
UX	47
<b>Annexes</b>	<b>48</b>
Cahier des charges.pdf	48
Entretiens _ Grille.pdf	48
Dépot Github - Code source du prototype	48
Questionnaire	49
Maquettes	49

# Introduction

Ce projet a pour but de fournir aux baroudeurs curieux et avides de culture un médium leur permettant de parcourir une ville et d'en apprendre plus sur son histoire de façon ludique.

L'idée est d'utiliser les codes du jeu vidéo afin de développer une application, créant ainsi une expérience innovante reposant sur la découverte, la curiosité et l'émerveillement de découvrir quelque chose de nouveau.

C'est un projet à la fois **technique** et **créatif**. Il présente donc de nombreux enjeux pour les étudiants et leur permet de développer de nombreuses connaissances et compétences, tant dans le domaine de l'expérience utilisateur (UX) que dans le développement informatique.

Les principaux objectifs sont :

- Faire un état de l'art sur les jeux vidéos, les applications touristiques et les principes de *gamification*<sup>1</sup> notamment.
- Concevoir des maquettes selon une démarche UX.
- Développer un prototype représentant un quartier d'une taille restreinte (quelques rues).

Ce rapport vise à faire un état des lieux sur le déroulement et l'avancement du projet tout au long de ce semestre. Il s'articule autour de trois axes majeurs : la gestion de projet, les démarches UX et le développement de l'application.

L'ensemble des fonctionnalités et leurs spécifications sont décrites dans le [Cahier des charges.pdf](#).

---

<sup>1</sup> La gamification consiste à transposer des techniques de motivation du jeu dans un autre domaine afin de le rendre plus ludique.

# I. Gestion de projet

## I.1. Organisation générale

Le projet Baroudeurs s'est articulé autour de deux axes majeurs, les démarches UX d'une part et le développement d'autre part. L'objectif du pôle développement était de livrer le squelette fonctionnel de l'application. Les considérations liées à la conception centrée utilisateur n'arrivant qu'après la phase d'apprentissage et tard dans le développement du prototype, les deux axes pouvaient être travaillés en parallèle.

La division en deux pôles a donc été décidée, et la répartition des membres au sein de ceux-ci en fonction des préférences de chacun.e, selon le schéma ci-dessous.

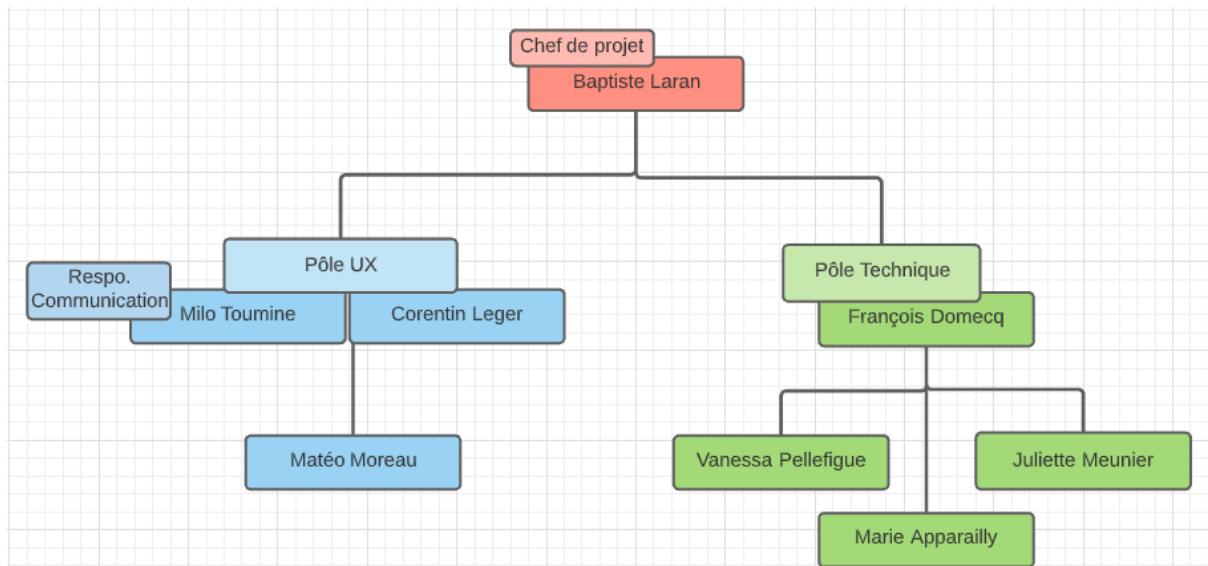


figure : schéma de répartition des membres entre les pôles

Une réunion bi-hebdomadaire avec l'ensemble du groupe permettait de rendre compte des avancées de chacun et d'orienter la suite du projet. Les réunions entre membres d'un même pôle étaient plus fréquentes. Formelles ou informelles, en ligne ou non, des échanges réguliers ont

permis de consolider les liens intra et inter - pôles et ainsi de garder une vision d'ensemble du projet.

## I.2. Ressources

### Collaboration

#### [Google Drive](#)

Pour tout ce qui concerne le stockage et le partage de fichiers textes, des présentations et des livrables, Google Drive est l'outil le plus pratique car il est simple d'utilisation et accessible par tous.

#### [Discord](#)

Pour la communication à distance, Discord s'avère être plus pratique que Messenger pour garder une trace des messages et fichiers envoyés.

### Développement

#### [React Native](#)

Nous avons utilisé le Framework de développement d'application mobile React Native basé sur le langage Javascript.

#### [Git et Github](#)

Git a été utilisé pour la gestion de versions de code et GitHub pour la mise en commun de ces versions.

#### [Visual Studio Code - Atom](#)

Nous avons également utilisé un éditeur de code pour programmer le prototype. Visual Studio Code ou Atom ont été utilisés par les membres du groupe.

#### [Expo](#)

Pour réaliser notre prototype, nous avons utilisé Expo qui est une plateforme permettant de créer des applications Cross Platform

## Création

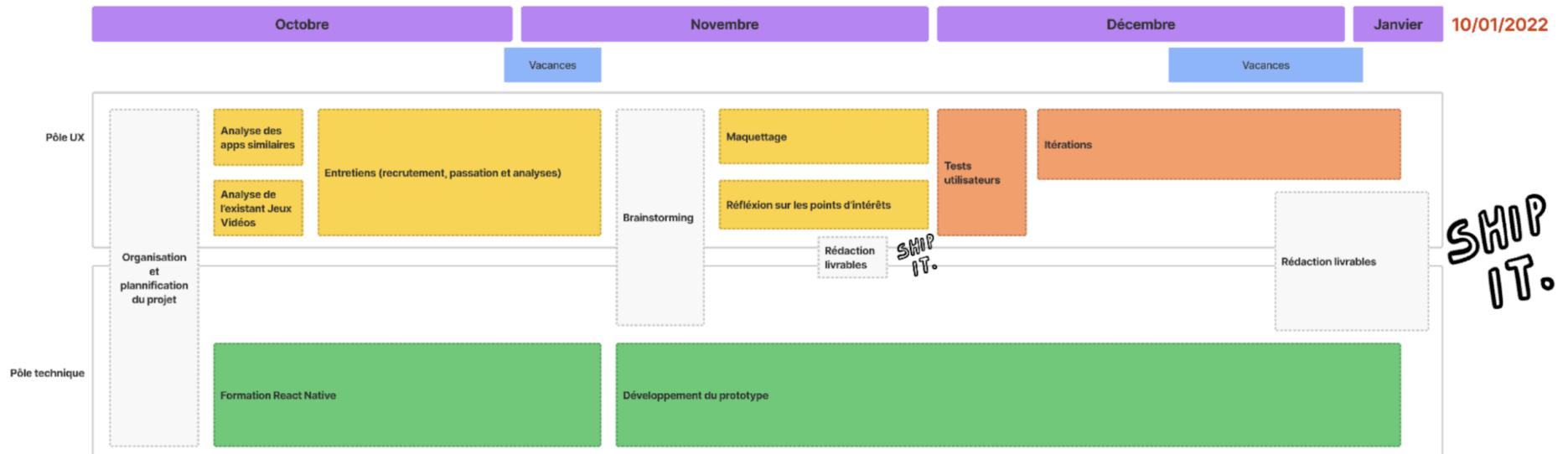
### [Figma](#)

Nous avons utilisé Figma pour regrouper les images et les screenshots issus de notre analyse sur l'existant, les plannings, réaliser des brainstorming et les maquettes/prototypes.

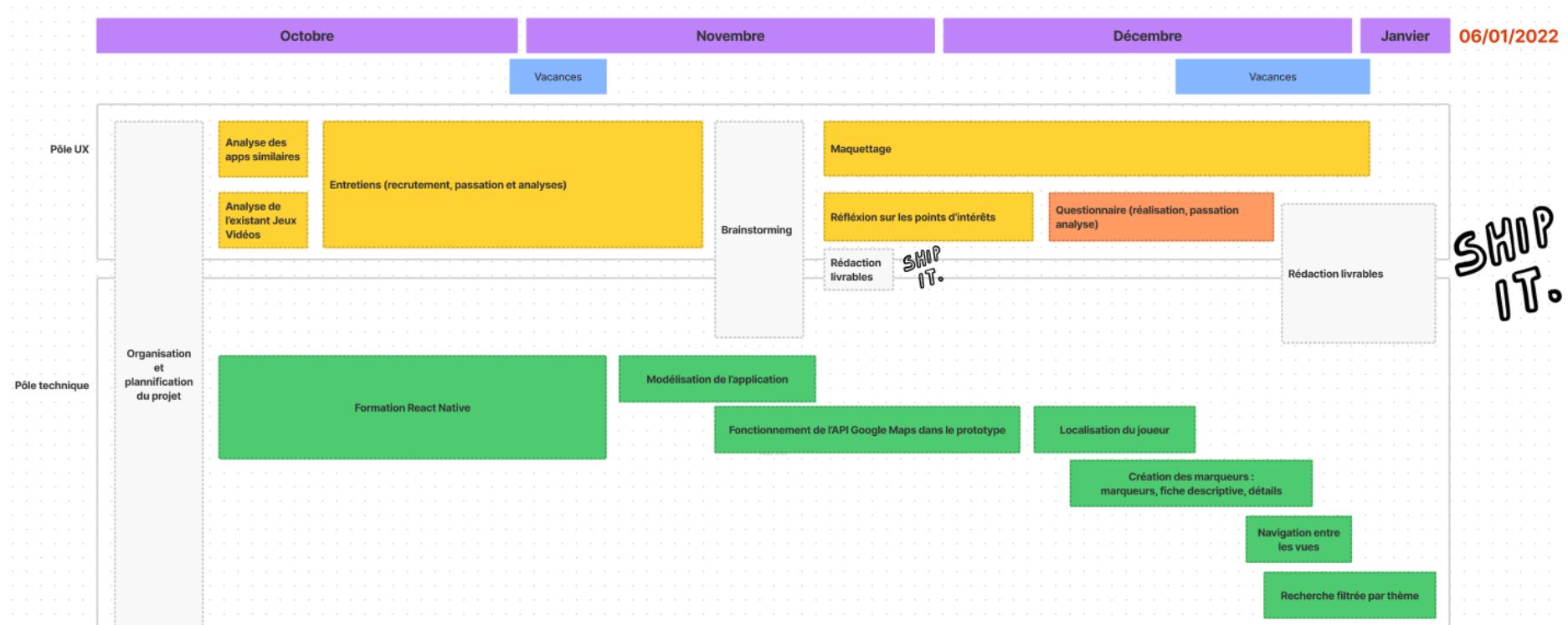
Comme c'est un outil collaboratif de création, il est possible de travailler à plusieurs en direct sur le même fichier. L'utilisation peut se faire sur le site web ou en téléchargeant l'application. De plus, avec notre mail étudiant, nous avons accès au programme Education de l'application.

## I.3. Plannings

### Planning prévisionnel



## Planning effectif



Comme nous pouvons le voir, le planning effectif est totalement différent du planning prévisionnel pour la partie Développement. En effet, au lancement du projet, les objectifs de programmation du prototype étaient assez flous car nous ne connaissions que peu de choses de la technologie React-Native. C'est la raison pour laquelle nous n'avions pas détaillé de jalons précis dans la programmation du prototype. De plus, nous avons fait le choix de rajouter une phase de modélisation de l'application en utilisant les résultats des entretiens du pôle UX. Cette phase a d'abord commencé par une phase d'apprentissage pour les 1A, puis un brainstorming commun des deux pôles nous a permis d'établir un cahier des spécifications que nous avons utilisé pour réaliser un diagramme de classe de l'application. En parallèle de la modélisation, nous avons fait le choix de commencer le prototype de l'application en implémentant l'API Google Maps dans notre application. Cependant, cette étape du projet nous a pris beaucoup plus de temps que prévu. Une fois cette étape réalisée, nous avons pu commencer le développement du prototype.

La partie gérée par le pôle UX a pris du retard lors du maquettage, et les tests et leurs itérations initialement prévues n'ont pas été réalisés. Un questionnaire, non prévu initialement, a été créé et diffusé pour valider et faire certains choix concernant le choix des object map : forme, couleur, sens.. des marqueurs affichés sur la carte.

## II. Pôle UX

### Introduction sur la partie UX

Les objectifs fixés par le pôle UX étaient :

- Comprendre et réutiliser les principes de ludification qui créent de l'engagement, comprendre le fonctionnement des applications de tourisme. (État de l'art et recherches)
- Établir une gamification de l'application.
- Mettre en place un processus de design UX afin de mieux cerner les utilisateurs cibles, s'appuyer sur les méthodes UX pour sélectionner les fonctionnalités de l'application, créer des maquettes puis les tester.

À partir de ces objectifs, nous avons défini un planning, et les tâches successives étaient réparties au fur et à mesure entre les membres du pôle. L'ouvrage de Carine Lallemand et Guillaume Gronier, Méthodes de design UX a été consulté pour préparer chacune des étapes de notre démarche de conception centrée utilisateur.

#### III.1. État de l'art

La première étape a été l'analyse de ce qui existait déjà, que ce soit des applications tournées vers l'exploration et le voyage, ou des mécaniques de jeux vidéo en monde ouvert.

##### Applications de découverte

Le projet Baroudeurs reposant sur la conception d'une application d'exploration et de tourisme, un état de l'art a été nécessaire concernant les applications de tourisme existantes.

Nous avons retenu une dizaine d'applications aux fonctionnalités intéressantes. Par exemple, le principe d'associer des histoires et anecdotes à des lieux que nous avons décidé d'implémenter existe dans l'application "Monument Tracker". Cette application permet de découvrir les monuments d'une ville sous forme d'un parcours regroupant les monuments à visiter avec les visuels des monuments et les anecdotes et éléments historiques que l'utilisateur peut consulter sur l'application.

La plupart des applications proposent des parcours et itinéraires prédéfinis pour découvrir ludiquement des villes en apportant des informations pertinentes sur ces lieux. Certaines sont spécialisées pour renseigner sur le trafic des transports en commun comme "City Mapper" ou de noter des lieux comme "Trip Advisor". Le principe de navigation dans l'espace et de visuel de la carte est inspiré de "Google Maps".

Le projet Baroudeurs s'inspire donc des anecdotes sur les lieux et sur le principe de visualisation des lieux sur un plan avec des icônes. C'est donc le concept d'exploration libre du projet et la gamification de l'application qui permet de rendre original notre projet d'application.

## Jeux vidéos

Les jeux vidéos étudiés sont des jeux vidéos à monde ouvert. Ce sont des jeux vidéo qui permettent l'exploration de territoires réalistes importants, qui se veulent quasiment entièrement explorables.

Lorsque le joueur aperçoit par exemple une montagne dans son champ de vision, il peut aller l'explorer, ce ne sont pas seulement des décors graphiques mais aussi des éléments du jeu. De plus, le joueur est plutôt

libre de ses déplacements, il n'est pas entièrement guidé par le jeu et peut décider d'arrêter la progression de l'histoire pour explorer.

L'analyse d'une bibliographie concernant notamment la création de cartes dans les jeux vidéos, a permis d'appréhender certains mécanismes intéressants, tels que le brouillard de guerre ou le principe d'icônes guidant l'exploration par exemple. Cela permet de créer une découverte progressive, donner des indices sur les lieux sans les dévoiler entièrement. Certains jeux proposent également des fonctionnalités qu'il serait possible d'intégrer à notre application. Par exemple, Assassin Creed Valhalla et ses 3 modes d'exploration différents :

- Adventurer : les quêtes et icônes sont directement disponibles sur la map
- Explorer : les icônes apparaissent au fur et à mesure que le joueur parcourt un endroit de la carte.
- Pathfinder : les icônes s'affichent dans un rayon réduit sur la mini map. Il n'y a pas de marqueurs sur la grande map.

Une idée serait alors de donner à l'utilisateur la possibilité de personnaliser son expérience d'exploration en fonction de ses envies, comme réduire ou augmenter le rayon de la zone où les marqueurs sont visibles, ou encore la possibilité pour l'utilisateur qu'une fois arrivé dans la zone d'un des marqueurs, il ait quand même à explorer un peu autour de lui pour trouver l'endroit recherché.

L'utilisation d'un compas montrant l'aspect des objectifs et leur direction est aussi répandue dans les jeux en monde ouvert et aurait tout à fait sa place dans Baroudeurs. Le principe de découverte de lieux via des icônes permet d'inciter l'utilisateur à se déplacer et à explorer pour découvrir des lieux et leur histoire. De plus, nous nous sommes inspirés des jeux vidéos pour gamifier l'application. Le but étant de faire un mélange entre une application de tourisme et les principes de jeux vidéo, nous retrouvons un système de récompenses, de progression, un visuel ludique etc.

## Gamification

Le processus de gamification est central dans le développement de ce projet car Baroudeurs a pour but de rendre ludique la découverte d'une ville. De plus, le principe d'exploration prend une part importante dans le développement de cette application. Ainsi nous nous sommes inspirés des fonctionnalités des jeux vidéo d'exploration en monde ouvert pour apporter une touche vidéoludique à notre application. Cette partie décrit quelques concepts clés à prendre en compte dans notre future application.

## Mécaniques de jeu

Défis	Objectifs à atteindre et obstacles à surmonter
Récompenses	Bénéfices obtenus
Chance	Part de hasard et incertitude
Feedback	Indicateur de progression en temps réel
Acquisition de ressources	Possibilité d'acquérir des ressources et outils pour progresser dans le jeu
Victoire	État du jeu lorsque les objectifs sont atteints

Contraintes	Limites imposées par les règles du jeu
Emotions	États affectifs que le jeu veut communiquer
Narration	Structure qui regroupe les éléments et mécaniques en un tout cohérent
Progression	Accomplissement et objectifs à atteindre
Relations	Interactivité et dynamique sociale

Les boucles d'engagement consistent à définir la progression des utilisateurs dans le système. Il faut définir quels feedbacks mettre en place

pour encourager des comportements cibles. Par exemple, l'ajout d'un onglet statistiques dans l'application permet à l'utilisateur d'avoir un retour direct sur ses actions.

Le niveau de difficulté doit augmenter au fur et à mesure que le niveau de compétences des joueurs augmente, et les défis doivent augmenter en quantité et en variété.

Il existe différents types de joueurs (voir typologie des joueurs de Marczewski). Pour Baroudeurs, les catégories concernées sont : Esprit libre, Ambitieux et Joueur.

On distingue également 3 phases majeures dans l'expérience de jeu :

- Phase de découverte : convaincre l'utilisateur d'utiliser le système
- Phase d'apprentissage : accompagner le joueur dans la maîtrise du jeu
- Phase de maîtrise : proposer des défis et une progression continue pour créer de l'engagement et de la satisfaction.

Besoins Universels	Eléments de jeu
Compétence	Feedback positif, défis, points, niveaux, tableaux de classements, progression de la difficulté, contrôles intuitifs, tutoriels intégrés
Autonomie	Profils, avatars, interfaces personnalisables, activités alternatives, contrôle de la vie privée, contrôle des notifications.
Relationnel	Groupes, équipes, messages, blogs, fonctions de chat, connexion aux réseaux sociaux, actions collaboratives, cadeaux et dons, transactions

Type de récompenses :

- Les récompenses externes comprennent la monnaie, les items, les manuels etc. Les ressources consommables, telles que, entre autres, les pièces, l'expérience requise pour des améliorations, des matériaux font également partie de ce type de récompense. Dans baroudeurs ces récompenses seront des skins pour les avatars, des points et des succès.

- Les récompenses dites internes sont plus abstraites, et plus liées à l'expérience subjective du joueur, telle que le sentiment d'accomplissement, la satisfaction sociale, etc. Elles proviennent de sentiments personnels. Elles peuvent provenir de cinématiques appuyant l'obtention d'un succès, elles peuvent soutenir la progression du joueur. Le sentiment d'accomplissement peut aussi provenir de la mise à jour des barres de progression dans l'application. Le feedback en temps réel.

L'aspect immédiat des récompenses dans les jeux joue un rôle important dans l'engagement du joueur. Même si la récompense est virtuelle, la satisfaction est réelle.

### **Objectifs de la Gamification de Baroudeurs**

1. Engager l'utilisateur dans le processus de découverte et d'exploration.

Le processus de découverte est un aspect important dans l'application. Nous nous inspirons des jeux vidéos pour stimuler l'envie d'explorer des utilisateurs. Par exemple, nous utilisons le brouillard de guerre. Le brouillard de guerre est visuellement une brume sur une carte, qui cache les rues et les routes. Seuls les points d'intérêts sont visibles, ainsi cela donne envie d'aller découvrir quels sont ces points d'intérêts et de pousser à l'exploration car l'itinéraire pour y accéder est momentanément indisponible.



Le principe de brouillard de guerre peut prendre plusieurs formes, à l'heure actuelle du développement de l'application le brouillard est positionné par quartier de Bordeaux et se dissipe lorsqu'on rentre dans la zone géographique.

De plus des points d'intérêts ont été développés, ce sont des icônes figurant sur la carte, qui indiquent le type de lieu, par exemple un restaurant, musée...

Ainsi lorsque l'utilisateur se situe autour de la localisation du point d'intérêt, il découvre celui-ci, c'est-à-dire qu'il reçoit via l'application des informations sur le lieu, peut-être des anecdotes du lieu... Par exemple, si l'utilisateur trouve un restaurant il aura un descriptif du restaurant, les plats proposés, son histoire etc. Une fois un point d'intérêt découvert nous pourrons envisager de pouvoir consulter les informations liées à ce point en le sélectionnant sur la carte, ainsi l'information sera toujours accessible une fois le point découvert.

L'utilisateur ne pourra donc pas connaître le lieu avant de l'avoir découvert dans l'expérience classique de l'application.

## 2. Améliorer l'expérience utilisateur en la rendant plus fun

Les points à chercher ajoutent une dimension ludique à l'application, car c'est l'action de l'utilisateur qui a un impact sur l'application.

Le principe de quête fait partie intégrante de l'application, cela permet de découvrir des lieux reliés entre eux, par un type commun de lieu (musées, restaurants) ou l'appartenance à une même époque historique etc. Ces quêtes permettent de mettre en évidence plusieurs points d'intérêts sur la carte, lors d'une quête nous avons donc plusieurs lieux à découvrir pour réussir celle-ci. L'application proposera un ensemble de quêtes que l'utilisateur pourra choisir, des quêtes pourront être débloquées suite à l'obtention de certaines quêtes ou succès. Nous pourrons choisir une priorité sur les quêtes sélectionnées, mettre en pause celle qu'on ne veut plus faire sur le moment.

De plus, des récompenses ont été mises en place, tel que le système de succès. Les succès sont des objectifs remplis par le joueur, ce sont des distinctions, mentions spéciales attribuées à l'utilisateur. Ces objectifs sont annexes à l'utilisation de base de l'application, il est possible d'utiliser l'application sans les succès, mais ils permettent une gamification importante de l'application. Ces objectifs remplis peuvent être multiples et très variés, par exemple un certain nombre de pas atteint ou de nombre de kilomètres parcourus. Un certain nombre de musées visités etc. Visuellement les succès sont souvent des icônes associées à un titre ou à une brève description du succès. Les descriptions se veulent généralement humoristiques et gratifiantes. Par exemple, lors de la visite d'un certain nombre de monuments, nous pourrions attribuer à l'utilisateur le titre de "Féru d'Histoire!", avec une icône stylisée au graphisme sympathique.

L'aspect visuel de l'application est/sera travaillé pour la gamifier, nous pourrons envisager un visuel de la carte différente, la possibilité de modifier l'aperçu de l'icône qui représente le joueur sur la carte. En effet l'utilisateur pourra choisir l'aperçu de son icône de baroudeur. Cette icône pourrait changer en débloquant un certain nombre de succès, cela fait partie des boucles d'engagement du joueur. La progression du joueur va influer sur certains points de l'application. De plus des cinématiques, petite scènes animées qui peuvent être diffusées lors de la découverte de monuments principaux d'une ville, avec une brève description de ce monument par exemple.

### 3. Véhiculer des histoires particulières

Un des éléments phares de l'application est le fait de délivrer des anecdotes et des informations pertinentes sur les lieux visités. Cela a pour but de renseigner l'utilisateur sur un lieu et de stimuler sa curiosité et son intérêt. Les informations seront historiques et explicatives pour des monuments, elles renseigneront sur l'ensemble des œuvres disponibles dans un musée par exemple, etc.

Le système de quêtes permet de véhiculer l'histoire de lieux, les quêtes permettent de découvrir ludiquement un ensemble de lieux et de comprendre leur liens. Cela est une progression amusante dans l'appropriation de l'histoire des lieux. Par exemple, nous pouvons imaginer une quête qui permet de découvrir les témoins d'une période historique comme le siècle des Lumières dans Bordeaux, en expliquant des bâtiments clés, les personnages historiques principaux etc. Cela apporte des informations supplémentaires et permet de jouer.

### 4. Mettre l'accent sur le sentiment de progression et d'accomplissement lié à la découverte d'une ville

Le sentiment de progression et d'accomplissement est lié fortement au feedback que reçoit l'utilisateur. Le retour direct sur ses actions est important. Cela permet de souligner ses actions, les succès débloqués, les quêtes terminés, le nombre de pas effectués etc. Il est

important de mettre en lumière les actions réussies. Ici cela prend la forme de découvertes de lieux. Ainsi ces retours prennent la forme de récompenses comme les succès par exemple, la possibilité de changer l'icône de l'utilisateur pour le récompenser. Des quêtes à débloquer permettent de stimuler l'envie de jeu de l'utilisateur, qu'il prenne du plaisir à continuer. Un onglet statistiques permet de visualiser toutes les actions accomplies, et permet un retour visible constant. Cela souligne la progression de l'utilisateur. Cela est important de pouvoir permettre à l'utilisateur de consulter ce qui est fait.

## II.2. Entretiens

### Objectifs et méthode

Les études qualitatives ont pour objectif d'expliquer et d'analyser des comportements, elles permettent d'étudier les attitudes, représentations et perceptions des participants sur un ou plusieurs thèmes. Ce type d'étude est donc justifié lorsque l'on cherche à prévoir certains comportements mais également de manière générale afin de faire émerger de nouvelles idées. Il est donc particulièrement adapté à un début de projet, dans un contexte de création et d'innovation.

A ce stade de l'étude, nous souhaitons créer une application qui permet de découvrir les villes de manière très libre, en nous inspirant des codes de certains jeux vidéo afin de laisser une grande part de mystère et d'exploration aux utilisateurs. Nous avons donc besoin de recueillir certaines informations quant aux habitudes de tourisme de ces potentiels utilisateurs, leur intérêt pour ce type de jeu et leurs comportements, ressentis et motivation dans le cadre du tourisme et de la découverte.

Le format de l'entretien semi-directif comme recueil de données a été choisi afin de laisser les sujets libres de s'exprimer sur différents thèmes, mais en restant dans un certain cadre nous permettant d'aborder tous les thèmes qui nous intéressaient. Le choix de l'entretien a également été fait car c'est un format plus simple à mettre en place que le focus group, puisqu'un seul participant doit être présent à la fois et non plusieurs, ce qui simplifie grandement la planification. La possibilité de réaliser ces entretiens à distance (et notamment au regard de la crise sanitaire) est également prévue.

Nous avons établi une grille d'entretien afin de pouvoir mener des entretiens semi-directifs.

Ces entretiens avaient notamment pour buts de :

- Recueillir des données précises sur les habitudes de tourismes d'individus
- Obtenir des renseignements précis sur la façon dont les utilisateurs potentiels appréhendent la découverte d'un lieu, d'une ville ou d'une région
- Cibler les méthodes utilisées par ces individus lors de voyages touristiques, comme l'utilisation d'applications ou de cartes par exemple.

Nous avons mené des entretiens semi-directifs auprès de cinq individus, tous étaient étudiants dans des domaines d'études et sont très variés, avec une moyenne d'âge de 20 ans. Deux femmes et trois hommes ont été interviewés.

## Conclusion

L'état de l'art et les entretiens nous ont permis de dégager de nombreuses idées et un large panel de fonctionnalités envisageables. Les entretiens ont confirmé l'implémentation des descriptifs des lieux ; ces descriptions permettent pour les interviewés de mieux apprécier une visite touristique, en ayant des informations intéressantes et pertinentes. Les interviewés n'ont pas tendance à aller chercher l'information eux-mêmes sur les lieux mais apprécient lorsqu'elle est facilement accessible, à proximité des lieux par exemple. De plus le visuel des descriptions est important pour mieux imaginer un monument dans son contexte historique par exemple.

L'immersion dans la ville est un facteur important dans l'appréciation d'un voyage selon les interviewés.

Suite aux entretiens nous avons également imaginé des fonctionnalités permettant de prendre en compte le budget qu'un utilisateur compte mettre lors d'un voyage. Nous avons aussi songé aux habitants d'une ville concernée et pas seulement aux touristes extérieurs à la ville. Ainsi nous pensions mettre en valeur des événements culturels et activités qui peuvent se dérouler dans la ville pour les découvrir via l'application. De plus, nous avons établi des niveaux de difficulté de découverte de la ville : un résident pourrait par exemple avoir des points d'intérêt de lieux plus difficiles et moins connus à découvrir et ne pas mettre en priorité la découverte des principaux lieux connus.

Il était alors nécessaire de faire un tri et d'entamer l'organisation de ces fonctionnalités.

C'est avec ces objectifs que nous avons organisé un brainstorming avec tous les membres de l'équipe (les deux pôles).

### II.3. Elaborations des fonctionnalités de l'application

## Brainstorming

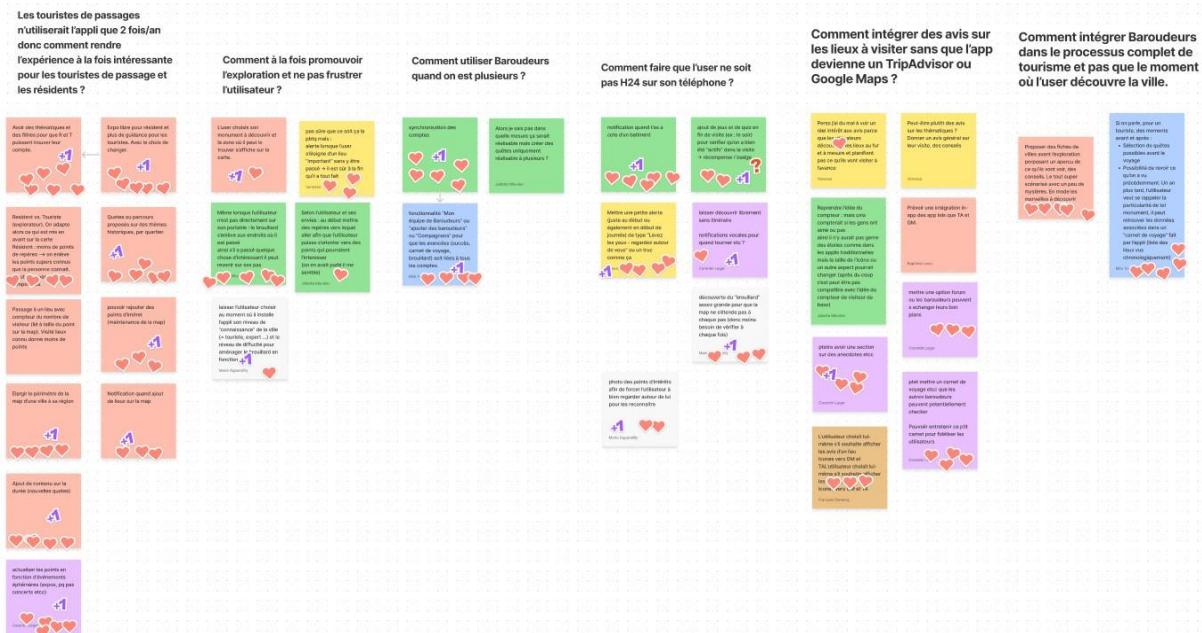
Dans un premier temps, nous avons réalisé un Brainstorming sur Figma avec l'ensemble des étudiants du projet, équipes UX et Développement réunies. L'objectif de ce Brainstorming était de faire émerger les différentes problématiques sur lesquelles nous pencher pour le développement de notre application de découverte de ville.

Les principales problématiques étaient :

- Comment rendre l'expérience à la fois intéressante pour les touristes de passage et les résidents ?
- Comment à la fois promouvoir l'exploration et ne pas frustrer l'utilisateur ?
- Comment utiliser Baroudeurs quand nous sommes plusieurs ?
- Comment faire que l'user ne soit pas continuellement sur son téléphone ?
- Comment intégrer des avis sur les lieux à visiter sans que l'application devienne un TripAdvisor ou Google Maps ?
- Comment intégrer Baroudeurs dans le processus complet de tourisme et pas seulement au moment où l'user découvre la ville.

Pour répondre à ces problématiques, nous avons utilisé Figma qui permettait à chaque étudiant de placer des "post-it" (d'une couleur par étudiant) sous chaque problématique afin de proposer diverses idées. Nous nous sommes ensuite rencontrés lors d'une réunion, et les meilleures idées étaient sélectionnées toujours sur Figma, par un système de vote (utilisation d'icônes cœur et +1 par les membres de l'équipe). Le fait d'être physiquement ensemble nous a permis d'échanger et de commenter au fur et à mesure que nous découvrions des idées, de les confronter et de voter pour les meilleures.

## 2. Identifier les problématiques.



## 3. Idées sélectionnées

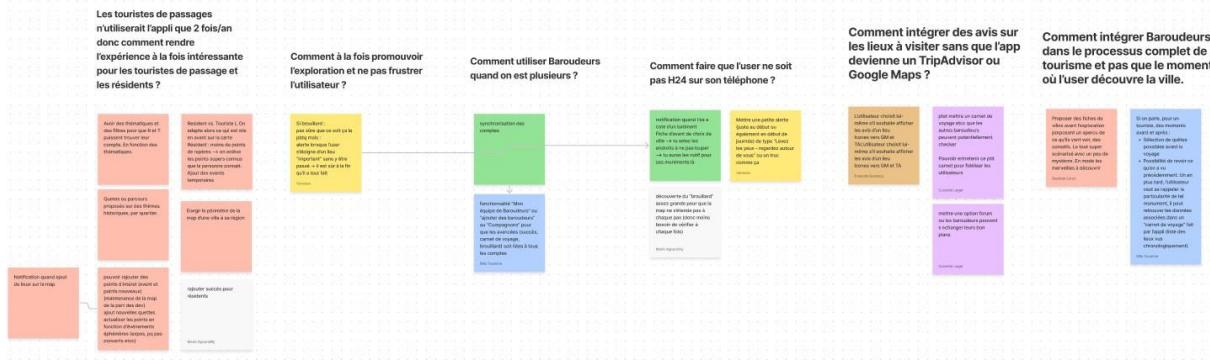


figure : Brainstorming sur Figma 1

## Spécification des fonctionnalités

Une fois que les différentes problématiques ont été identifiées, et les meilleures solutions choisies pour y répondre, nous avons élaboré la liste des fonctionnalités que nous souhaitions développer dans notre projet lors d'une autre réunion. De même que pour le brainstorming, Figma nous a permis de confronter nos idées puis de voter pour les meilleurs, tout en

échangeant à voix haute afin de confronter et de défendre nos idées, mais aussi d'en faire émerger de nouvelles.

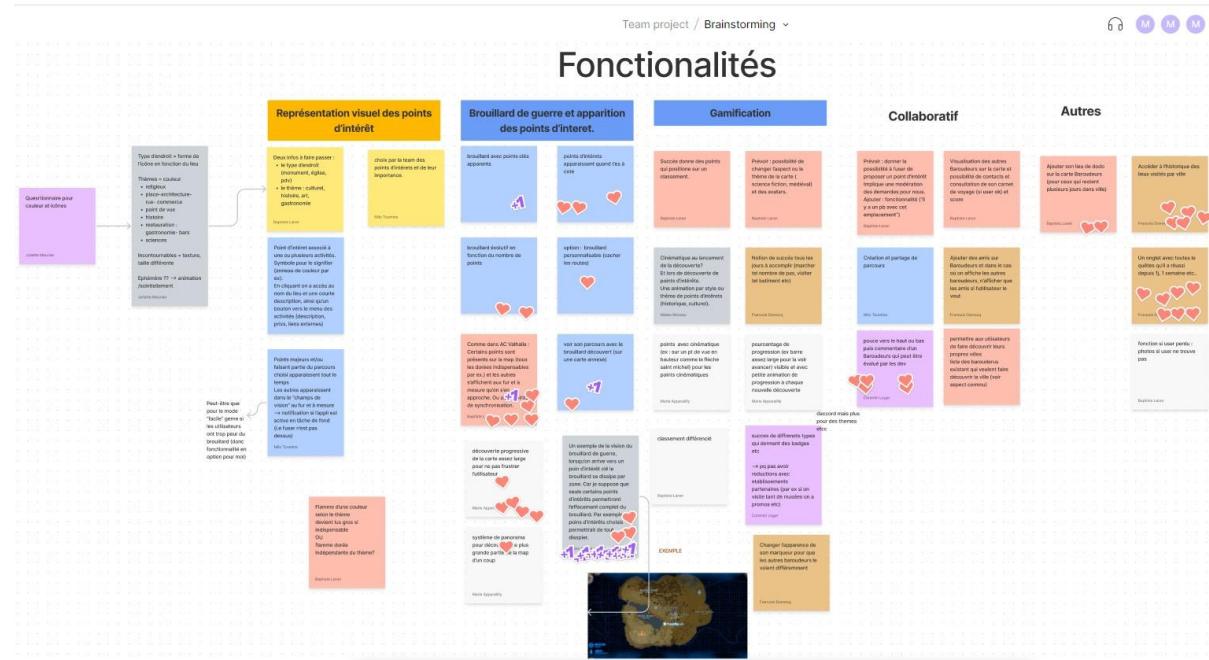


figure : Brainstorming sur Figma 2

Ainsi, plusieurs grandes thématiques des fonctionnalités à développer ont été mises en avant.

Tout d'abord, nous nous sommes intéressés à la représentation visuelle des points d'intérêt. C'est en se questionnant sur ces fonctionnalités que nous avons notamment décidé d'utiliser une couleur pour représenter un thème (histoire, art ...) et une icône pour le type de lieu.

Ensuite, le brouillard de guerre et l'apparition des points d'intérêt restaient à définir. Nous nous sommes par exemple mis d'accord pour que la carte ne laisse apparaître que peu d'informations, hormis les points d'intérêt "phares" de la ville (que nous aurons préalablement choisis et énumérés, en nous appuyant sur des guides touristiques), et les points d'intérêts d'un thème ou d'une quête choisie par l'utilisateur. Des points d'intérêts et informations supplémentaires devraient s'afficher sur l'application lorsque l'utilisateur se déplace dans la ville et s'approche de lieux importants.

Puis, nous souhaitions développer des fonctionnalités permettant une *gamification* de l'application. Nous nous sommes essentiellement retrouvés sur l'importance pour l'utilisateur d'obtenir des succès et des récompenses lors de l'accomplissement de certaines quêtes ou objectifs notamment, ainsi que sur l'affichage visuel de la progression (comme une barre de progression de jeu par exemple)

Enfin, les fonctionnalités collaboratives proposées étaient entre autres de permettre aux utilisateurs de visualiser les autres baroudeurs sur leur carte, de proposer de nouveaux points d'intérêts aux autres baroudeurs, de nouveaux parcours ... La possibilité de réaliser des quêtes à plusieurs a également été proposée.

Enfin, d'autres fonctionnalités plus "annexes" et secondaires ont été proposées, tout en ayant conscience que le temps ne nous permettrait pas de les développer.

## II.4. Questionnaire

### Objectifs et méthode - le concept des marqueurs

Le concept clé de Baroudeur étant l'exploration, l'application s'inspire de certains codes du jeu vidéo, et notamment des codes visuels des cartes des jeux de type RPG<sup>2</sup> open world<sup>3</sup>. L'application est en effet construite sur la base d'une carte, sur laquelle des marqueurs permettront d'identifier des "points d'intérêts" (musées, points de vue, restaurants...) présents dans la ville, qui pourront être filtrés selon un thème ou une "quête" préalablement choisie par l'utilisateur, comme dans ces jeux.

A chaque marqueur est associé une couleur pour le thème auquel il est associé (histoire, panorama ...) ainsi qu'un icône qui s'affiche lorsque

---

<sup>2</sup> Role Player Game

<sup>3</sup> Le mot open world définit un jeu où le joueur peut évoluer librement dans le monde qui l'entoure. Il peut par exemple sortir de la ligne de quête principale pour faire des quêtes secondaires, activités annexes, de l'exploration... Skyrim, Zelda, ou Assassin's Creed en sont des exemples très connus.

l'utilisateur s'approchera afin d'identifier le type de lieu qu'il s'apprête à visiter (espace extérieur, musée ...). Un marqueur plus stylisé permet de pointer les lieux phares de la ville.

Les marqueurs de la carte représentent à la fois cette inspiration des jeux vidéo et l'utilité touristique de pointer les différents lieux d'intérêts d'une ville, c'est pourquoi ils sont au cœur du design visuel de l'application et notamment du logo de Baroudeur.

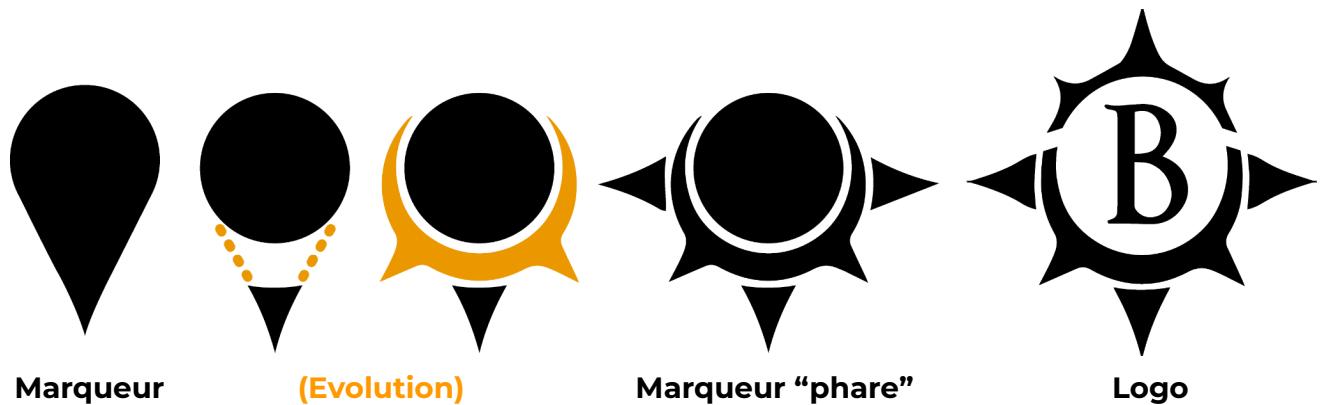


Figure 1 - Crédit : Baroudeur

Un questionnaire a donc été élaboré dans plusieurs buts :

- connaître les connaissances et les attentes des futurs utilisateurs quant aux thèmes et fonctionnalités qu'ils attendent dans l'application
- valider certains choix faits par l'équipe de projet
- leur laisser le choix du design de ces marqueurs : leur forme, leur couleur, leur contexte d'utilisation ... mais également de comprendre le sens qui leur est associé dans une sorte de mémoire collective.

## Résultats du questionnaire

Plus de 80 personnes ont répondu au questionnaire. Les résultats ont permis d'en apprendre plus sur le profil des potentiels utilisateurs. En effet, 70% des répondants avaient déjà joué à des jeux vidéo *open world* et connaissaient donc les codes associés. En revanche, seulement 35% avaient déjà utilisé une application de tourisme ou de découverte de lieux.

Ces réponses ont également permis de définir plus précisément les fonctionnalités d'affichage préférées des utilisateurs ainsi que de définir les marqueurs de points d'intérêts, selon les thèmes, couleurs, types de lieux et d'icônes choisies par les répondants.

## Choix des thèmes

Les thèmes permettent de classifier les points d'intérêts selon un thème de découverte de la ville (selon si l'utilisateur souhaite la visiter d'un point de vue plutôt touristique, ou artistique etc.). Plusieurs thèmes ont donc été proposés aux potentiels utilisateurs, afin de connaître leur avis sur le choix de ces thèmes, ainsi que les couleurs qu'ils souhaitent leur associer.

Ces thèmes étaient : Histoire, Religion, Art, Panorama, Gastronomie et Sciences.

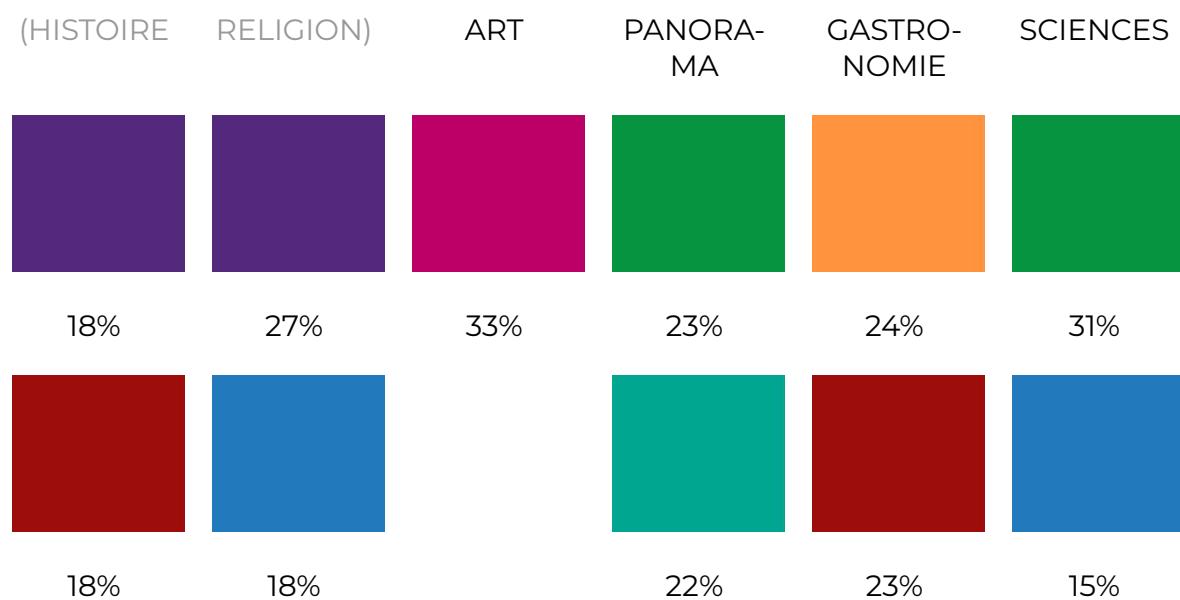


Figure 2 - Réponses sur le choix des couleurs associées aux thèmes

A la demande des répondants, les thèmes Histoire et Religion ont été combinés. Ils permettent en effet tous les deux de découvrir la ville selon un parcours historique, accès sur des bâtiments typiques du développement de la ville au fur et à mesure de l'Histoire.

Enfin, le choix des couleurs a été fait par l'équipe selon les deux premières sélectionnées par les répondants (afin d'éviter des doublons notamment).

## Choix des icônes

Les icônes quant à eux permettent de classifier les points d'intérêts selon le type de lieu (afin que l'utilisateur sache s'il s'attend à visiter une place, un bâtiment historique, un musée etc.). Plusieurs types de lieu ont donc été proposés aux potentiels utilisateurs, afin de connaître leur avis sur le choix de ces types de lieu, ainsi que les icônes qu'ils souhaitent leur associer. Ces types de lieu étaient : Parc, Place, Restaurant/bar, Monument, Musée, Bâtiment historique, Edifice religieux, et Autre bâtiment.

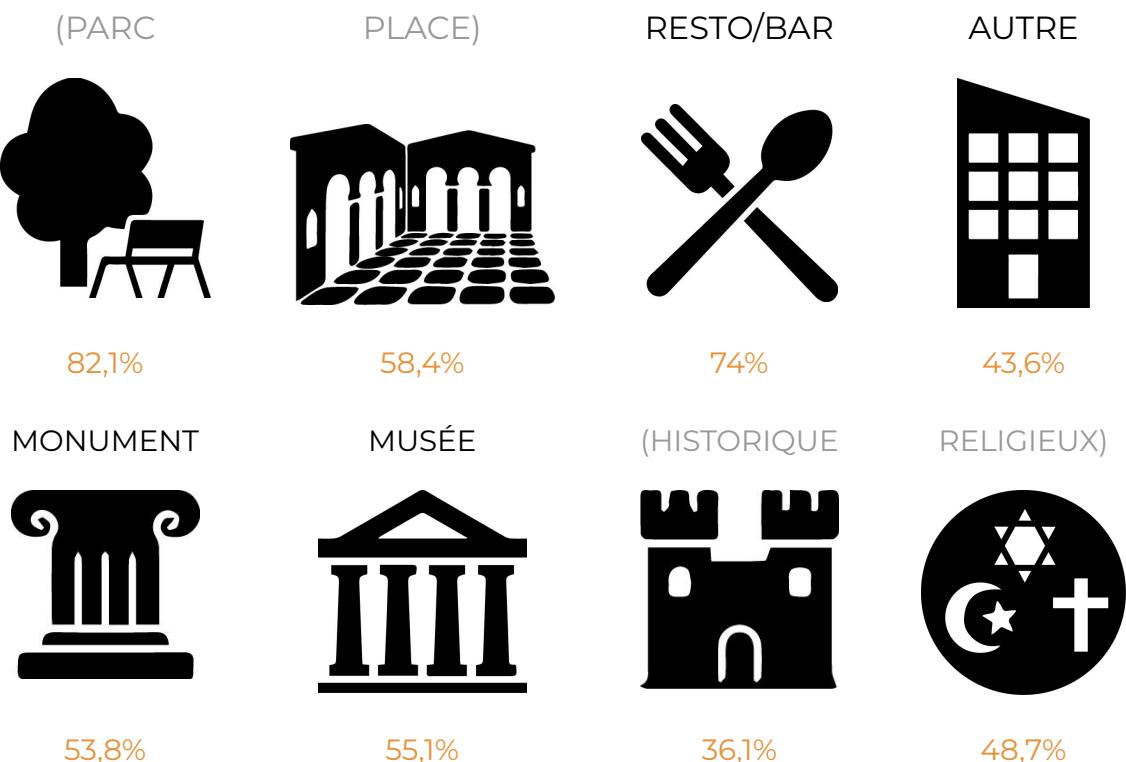


Figure 3 - Réponses sur le choix des icônes associées aux types de lieu

A la demande des répondants, les types de lieux Place et Parc ont été combinés afin de ne créer qu'une seule catégorie espaces extérieurs. De la même manière que pour les thèmes, les édifices religieux ont également été placés dans le type de lieux bâtiment historiques, afin que les utilisateurs s'attendent à un intérêt architectural et historique du bâtiment qu'ils s'apprêtent à découvrir.

## Design final

De manière générale, de nombreuses informations ont été obtenues grâce au questionnaire, et notamment sur les fonctionnalités des jeux vidéo que les utilisateurs souhaitaient retrouver dans notre application.

Ainsi, plusieurs choix ont pu être faits :

- Les thèmes sauvegardés sont : Histoire, Art, Panorama, Gastronomie et Sciences, et chacun se voit une couleur spécifique associée
- Les types de lieux choisis sont : Espaces Extérieurs, Restaurants/Bars Monuments, Musées, Bâtiments historiques et Autres bâtiments, chacun correspondant à un icône bien précis

Mais d'autres fonctionnalités ont également été choisies grâce aux réponses

- Les marqueurs des points d'intérêts déjà explorés par l'utilisateur doivent devenir gris
- Les marqueurs des lieux phares doivent être plus stylisés
- La position du joueur doit être visible sur l'application
- Des informations quant aux horaires d'ouvertures etc. doivent être visibles sur les descriptifs des lieux.

Les marqueurs finaux, qui sont implantés dans le développement informatique de l'application permettent donc d'identifier les lieux. Par exemple, le musée d'Aquitaine sera classé en Histoire (violet) et en Musée (icône musée), la Cité du Vin en Gastronomie (orange) et Musée etc.



Figure 4 - Exemple de marqueurs de points d'intérêts normaux et "phares"

## II.5. Maquettage

La dernière phase prévue dans le planning du pôle UX était la réalisation de maquettes haute fidélité.

Une première étape a été de faire des croquis sur papier afin de se mettre d'accord sur les choix visuels et d'interactions entre les menus. Cette étape a été faite en groupe en présentiel afin de favoriser les échanges d'idées et faciliter la communication. Les maquettes ont ensuite été réalisées sur Figma (cf lien en annexes).

Ces maquettes ne sont finalement pas si détaillées que nous le souhaitions et manquent notamment de style, mais les fonctionnalités prévues sont présentes et la navigation est possible (cf démonstration vidéo en annexe) et permet de se faire une idée de l'appli.

La police (comic sans ms) utilisée dans les maquettes a été choisie pour son esthétique douteuse, afin de concentrer le regard sur la composition et l'agencement des éléments plutôt que sur l'esthétique.

Le retard pris est en partie dû à la phase d'apprentissage de Figma qui n'a pas été prise en compte dans le planning.

La vidéo livrée présente certaines maquettes et les interactions entre elles. Les interactions ne sont pas toutes définies mais les maquettes sont présentées dans le lien figma en annexe.

## Conclusion et acquis pédagogiques

Le travail réalisé par le pôle UX a permis de spécifier les fonctionnalités nécessaires au développement de l'application. Si les visuels livrés ne répondent pas aux objectifs fixés, le reste des objectifs est atteint et les acquis pédagogiques sont nombreux. Les méthodes de CCU éprouvées ont permis de renforcer ou développer des compétences pour tous les membres. La prise en main de Figma est aussi un apprentissage précieux.

## **III. Pôle Développement**

### **Introduction sur la partie Dev - Objectifs et Méthodes**

Le pôle développement avait pour objectif principal de programmer un prototype de l'application Baroudeurs qui en donnerait un aperçu visuel (non-définitif) et qui présenterait quelques-unes de ses principales fonctionnalités. Le temps disponible ne permettant effectivement pas de coder davantage, il s'agit donc d'un squelette fonctionnel et non pas de la version finale de l'application.

Les résultats du pôle UX étaient donc essentiels pour commencer le démarrage du développement pour déterminer les principales fonctionnalités à implémenter dans le prototype. Nous avons donc, en parallèle de l'enquête utilisateur, commencé par une phase d'apprentissage. Cette phase se révélait nécessaire puisque le langage choisi, react native, était inconnu des 1ères années et de 2ème année. Ensuite, une fois les données du pôle UX récoltées, nous avons fait le choix de réaliser une modélisation d'un diagramme de classe selon la version finale de l'application qui serait utile dans le cas où ce projet serait poursuivi. Puis, nous avons commencé la réalisation du prototype de l'application.

#### **III.1. Phase d'apprentissage**

Les élèves de première année ont commencé le projet par une formation sur React Native, langage qu'ils ne connaissaient pas.

Cette phase a commencé par le visionnage de cours délivrés par notre professeur B. Pesquet disponibles sur Youtube. Ces vidéos ont permis de

comprendre les intérêts du langage et d'obtenir les pré-requis pour son utilisation.

L'apprentissage d'un langage nécessitant majoritairement de la mise en pratique, nous avons également suivi un cours sur OpenClassrooms intitulé Développez une application mobile React Native. Nous avons ainsi pu créer une première application qui nous a permis de découvrir les avantages et le fonctionnement de React Native ainsi que d'appréhender certains de ses concepts. Nous nous sommes donc familiarisés avec la gestion des vues et la navigation entre elles, les components, les styles, les props et le state.

## III.2. Modélisation

Afin de modéliser l'application, nous avons utilisé le Langage de Modélisation Unifié (UML). Ce langage permet de réaliser une modélisation visuelle de l'architecture d'une application ou d'un programme dans les domaines du développement logiciel et de la conception orientée objet. Les premières années n'ayant pas encore eu de cours sur l'UML, la première étape fut de reprendre le cours que les deuxièmes années avaient pu suivre, afin que tout le monde soit capable de créer un diagramme de classe.

Une fois formés, nous avons chacun créé un diagramme de classe. Pour ce faire, suite aux entretiens menés par le pôle UX, un cahier des charges avec les différentes fonctionnalités a été créé à la suite d'un brainstorming des deux pôles.

Nous nous sommes ainsi appuyés sur ce cahier afin de déterminer les différentes classes, attributs et méthodes utiles à la réalisation d'un diagramme de classe en UML. Enfin, nous avons mis en commun nos différents diagrammes afin de garder les meilleures idées de chacun.

Pour réaliser ce diagramme de classe, nous avons donc utilisé les résultats obtenus par le pôle UX afin de créer une modélisation de l'application telle qu'elle serait en version finale.

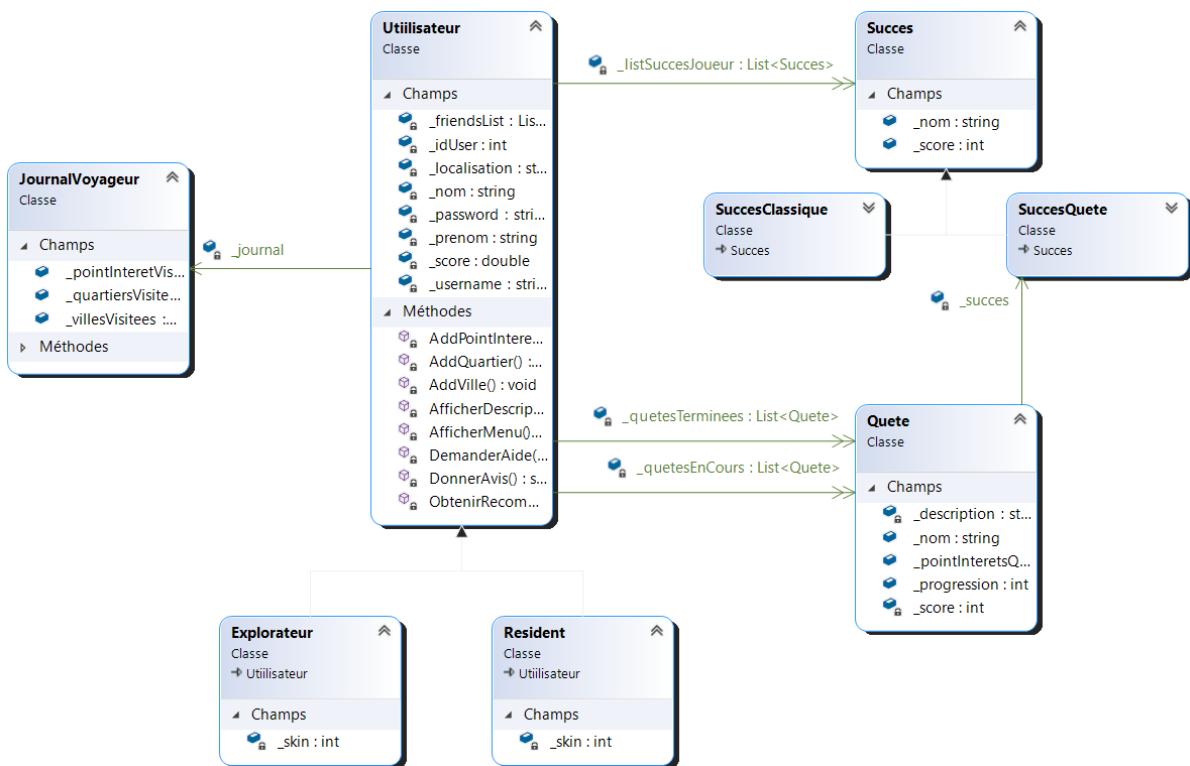


Figure 1: Diagramme de classe de l'application Baroudeurs partie

### Utilisateur

La classe **Utilisateur** regroupe toutes les informations lui appartenant comme son identifiant, son mot de passe, sa localisation ou encore sa liste d'amis. Les classes **Explorateur** et **Resident** sont des classes filles de la classe **Utilisateur**. Nous avons fait ce choix de conception car les résidents d'une ville auront une apparence différente des explorateurs de cette même ville dans l'application. De plus, il est important de différencier ces deux types d'utilisateurs car les points d'intérêts affichés sur la carte ainsi

que les quêtes ne seront pas les mêmes pour les résidents que pour les explorateurs.

A la classe Utilisateur est reliée la classe JournalVoyageur. Cette classe sert d'historique en enregistrant les villes, les points d'intérêts et les quartiers visités spécifiques à l'utilisateur. Cette classe servira aussi à l'élaboration d'une fonctionnalité permettant aux utilisateurs de tenir un carnet de bord de voyage.

Chaque utilisateur peut également compléter des quêtes, c'est pourquoi la classe Quete est directement reliée à la classe Utilisateur par le biais de deux listes : \_quêtesTerminees et \_quêteEnCours. Ayant ajouté un attribut *progression* à la classe Quete, nous aurions pu nous contenter d'instancier une seule liste dans la classe Utilisateur et regarder quelles quêtes étaient complétées en regardant dans la liste les quêtes dont l'attribut progression était égal à 100. Cependant, les quêtes représentent une partie importante de notre application tant elles incitent l'utilisateur à explorer qu'elles rendent la visite des villes plus attrayante. Nous prévoyons donc d'en développer un grand nombre, et dans le cas où un utilisateur complète de nombreuses quêtes, parcourir une unique liste serait moins optimisé que diviser les quêtes en cours des quêtes terminées en deux listes.

Enfin, chaque utilisateur peut acquérir des succès, c'est pourquoi la classe succès est directement reliée à la classe utilisateur par le biais d'une liste. Nous avons défini deux types de succès :

- Les succès obtenus par le biais de quêtes
- Les succès obtenus en jouant (nombre de pas, nombre de villes visitées..)

Il nous a donc paru logique de définir deux classes filles SuccesClassique et SuccesQuetes

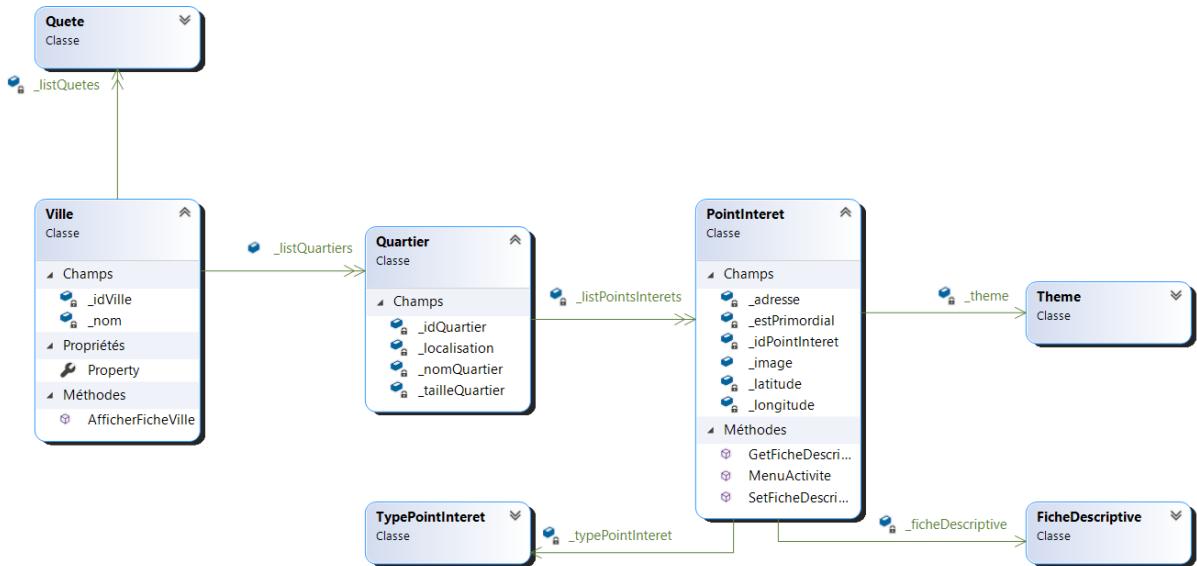


Figure 2: Diagramme de classe de l'application Baroudeurs partie Carte

La classe Ville est caractérisée par un Id et par un nom. Comme défini dans cahier des spécifications, lors de l'arrivée dans une nouvelle ville. L'utilisateur doit pouvoir avoir accès à une présentation de la ville sous forme de fiche ou d'animation. D'où la méthode AfficherFicheVille. Une ville est également caractérisée par les quêtes qu'il est possible d'y compléter ainsi que la liste des quartiers qui la composent. Nous avons fait le choix de séparer les villes sous formes de "zones" (quartiers) pour modifier l'affichage de la ville selon l'avancement de l'exploration du joueur à la manière des jeux vidéos.



figure : Carte incomplète dans le jeu Zelda - Breath of the Wild

En effet, pour afficher totalement la topographie d'un quartier, l'utilisateur devra en avoir visité la majeure partie des points d'intérêts. Sinon, la carte sera affichée sous forme primaire.

La classe Quartier est donc caractérisée par sa localisation, son nom et sa taille. Elle est également logiquement composée d'une liste de points d'intérêts.

La classe PointInteret implémente donc les points d'intérêts affichés sur la carte. Un point d'intérêt est défini par sa localisation avec les attributs latitude et longitude et adresse, s'il est considéré comme primordial (un lieu à voir par tous les touristes d'une ville) et une image du lieu. Un point d'intérêt possède également une fiche descriptive qui affiche les différentes informations du lieu telles que son histoire ou bien les activités qu'il est possible d'y faire etc.. Enfin un point d'intérêt est caractérisé par un thème : *religieux, art, point de vue, histoire, restauration, sciences, commerce..* et un type : musée accueillant une exposition ou sculpture par exemple dans le cas d'un point d'intérêt ayant le thème art.

### III.3. Programmation du prototype

Dès le lancement du projet, nous avions pour objectif de coder une partie de l'application. Cependant, nos connaissances en React Native étant principalement basées sur un auto-apprentissage et la durée de ce projet étant courte, nous avons fait le choix de nous concentrer uniquement sur certaines fonctionnalités et classes du diagramme de classe présenté précédemment.

### Outils utilisés

Pour réaliser notre prototype, nous avons utilisé les outils suivants :

- L'[API Google Maps](#)
- L'[API Styling Wizard](#)
- La librairie [react-native-maps](#)

### Exigences fonctionnelles

Nous avons également défini certaines exigences fonctionnelles auxquelles devait répondre notre prototype qui sont les suivantes :

Référence	Description	Exigence remplie
EF_1	Le prototype doit être programmé en React Native	
EF_2	Le prototype doit fonctionner sur IOS et sur Android	
EF_3	Le prototype doit utiliser l'API Google Maps	
EF_4	Le prototype doit afficher la carte Google Maps selon le style de notre choix à l'aide de l'API Styling Wizard	
EF_5	La position de l'utilisateur doit être affichée sur la carte	
EF_6	Le prototype doit suivre la position du joueur et	

	mettre à jour la carte selon sa position	
EF_7	Différents points d'intérêts doivent être affichés sur la carte	
EF_8	Les points d'intérêts doivent être personnalisés en fonction du thème qu'ils représentent	
EF_9	L'utilisateur doit pouvoir cliquer sur les points d'intérêts. Le point d'intérêt se transforme en fiche descriptive et affiche : - Le nom du lieu - Une image correspondant au lieu - Une courte description du lieu	
EF_10	L'utilisateur doit pouvoir cliquer sur la fiche descriptive d'un point d'intérêt. Il est alors redirigé vers une fiche plus détaillée de ce lieu.	
EF_11	L'utilisateur doit pouvoir décider d'un thème. Seuls les points d'intérêts représentant ce thème seront alors affichés.	
EF_12	L'utilisateur doit pouvoir afficher la carte en Dark/Light Mode	
EF_12	Les données concernant les points d'intérêts doivent être stockés dans une base de données Firebase	

Toutes les exigences n'ont pas pu être remplies ou alors partiellement remplies. En effet, l'application est bien codée en React-Native, cependant certaines fonctionnalités telles que la géolocalisation fonctionnent sur IOS mais pas sur Android. Nous n'avons donc pas pu remplir l'exigence fonctionnelle n°2 et partiellement les exigences n°5 et 6 puisqu'elles fonctionnent uniquement sur IOS. Cela est dû au fait que certaines fonctions issues de la librairie React-Native-Maps fonctionnent sur IOS mais pas sur Android. L'exigence 11 n'a également été que partiellement accomplie, l'utilisateur peut en effet sélectionner un thème, cependant l'affichage des marqueurs ne s'actualise pas. Enfin, nous avons manqué de temps pour transférer les données de nos marqueurs vers une base de données. Nous pensions que nos connaissances acquises à l'ENSC seraient suffisantes pour implémenter une base de données utilisable par une

application mobile. Cependant, la tâche était plus compliquée que nous ne le pensions et nous avons manqué de temps pour la réaliser.

## Résultats

Comme expliqué précédemment, nous nous sommes concentrés sur certaines fonctionnalités de l'application. Nous avons donc mis en place la carte (composante Map), certains marqueurs correspondant à différents points d'intérêt (composante CustomMarker), une fiche descriptive qui s'affiche lorsque l'on clique sur un marqueur (fonction FicheDescriptive) et une fenêtre détails qui s'ouvre lorsque l'on clique sur la fiche descriptive (fonction Details). Nous avons aussi ajouté une fonctionnalité permettant de mettre la carte en mode nuit ou jour, et une fonctionnalité permettant de filtrer les marqueurs affichés sur la carte en fonction de leur thème.

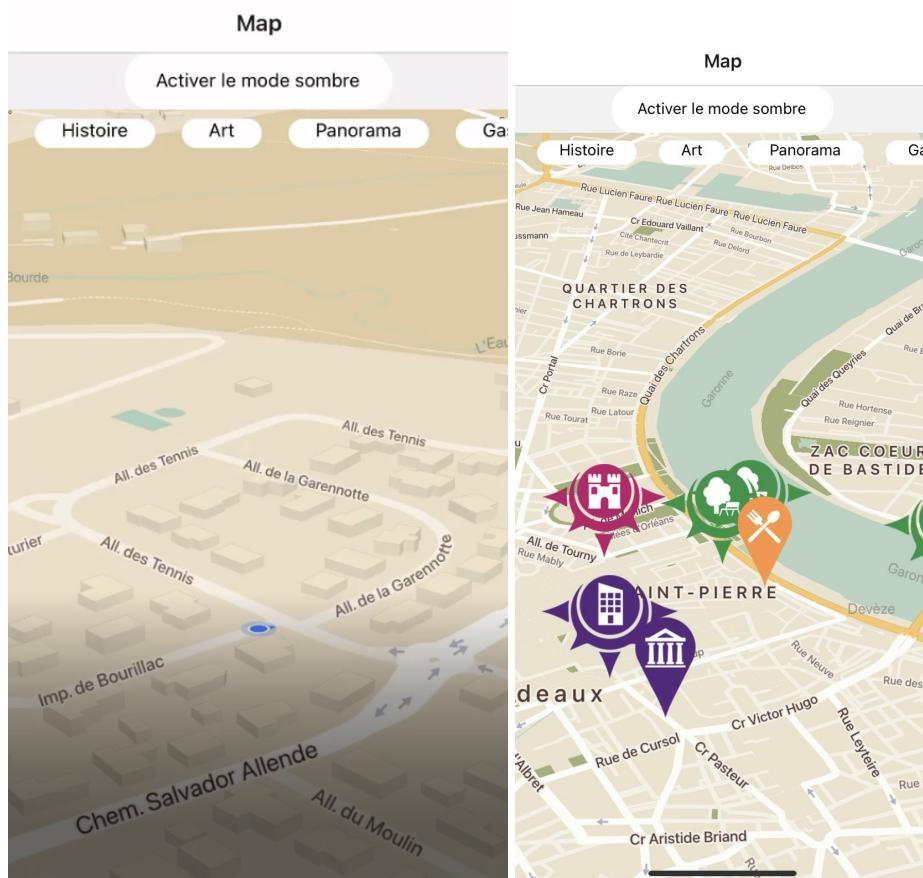


Figure : Localisation et Carte

Cliquez sur chaque image pour voir une démonstration

Sur la capture d'écran de droite, nous pouvons voir la carte en mode jour avec différents marqueurs correspondant notamment à la place de la bourse, au pont de pierre ou encore au grand théâtre.

Nous avions prévu de faire appel à une base de données afin de stocker les différentes informations correspondantes à chaque marqueur. Mais l'apprentissage de react native ayant pris du temps, nous n'avons pas eu le temps de nous y intéresser. Nous avons ainsi créé un fichier contenant les différentes informations des six marqueurs.

Sur la capture d'écran de gauche, nous pouvons voir la localisation de l'utilisateur sur la carte.

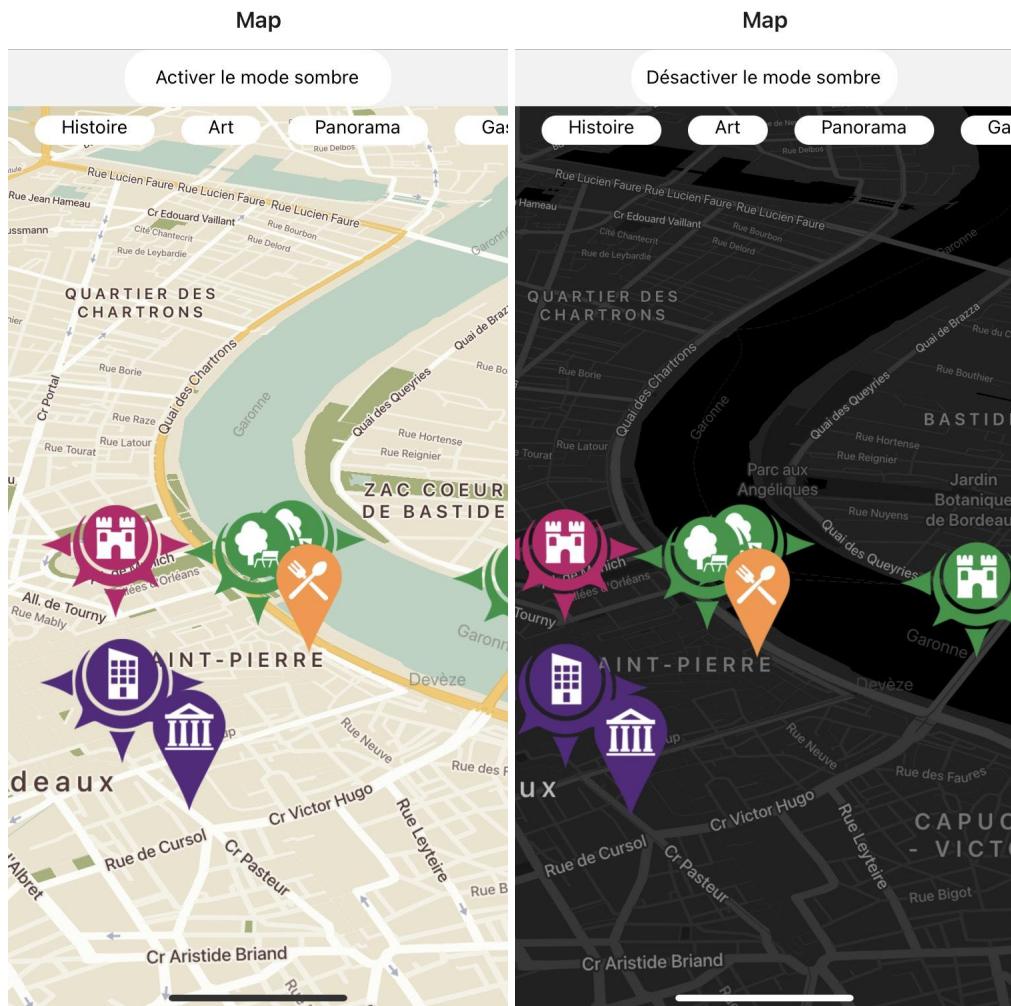


figure: Mode light/dark

Grâce à un bouton situé au-dessus de l'affichage de la carte, l'utilisateur peut choisir d'utiliser la carte en mode sombre ou en mode clair.

Pour cela, nous avons enregistré dans deux fichiers distincts les deux modes en les exportant depuis le site <https://mapstyle.withgoogle.com/>.

Ainsi le bouton déterminant si le mode sombre est activé renvoie un booléen (vrai s'il est activé, faux sinon). Si le booléen vaut vrai, le component MapView est affiché avec le style sombre, et sinon il est affiché avec le style clair.

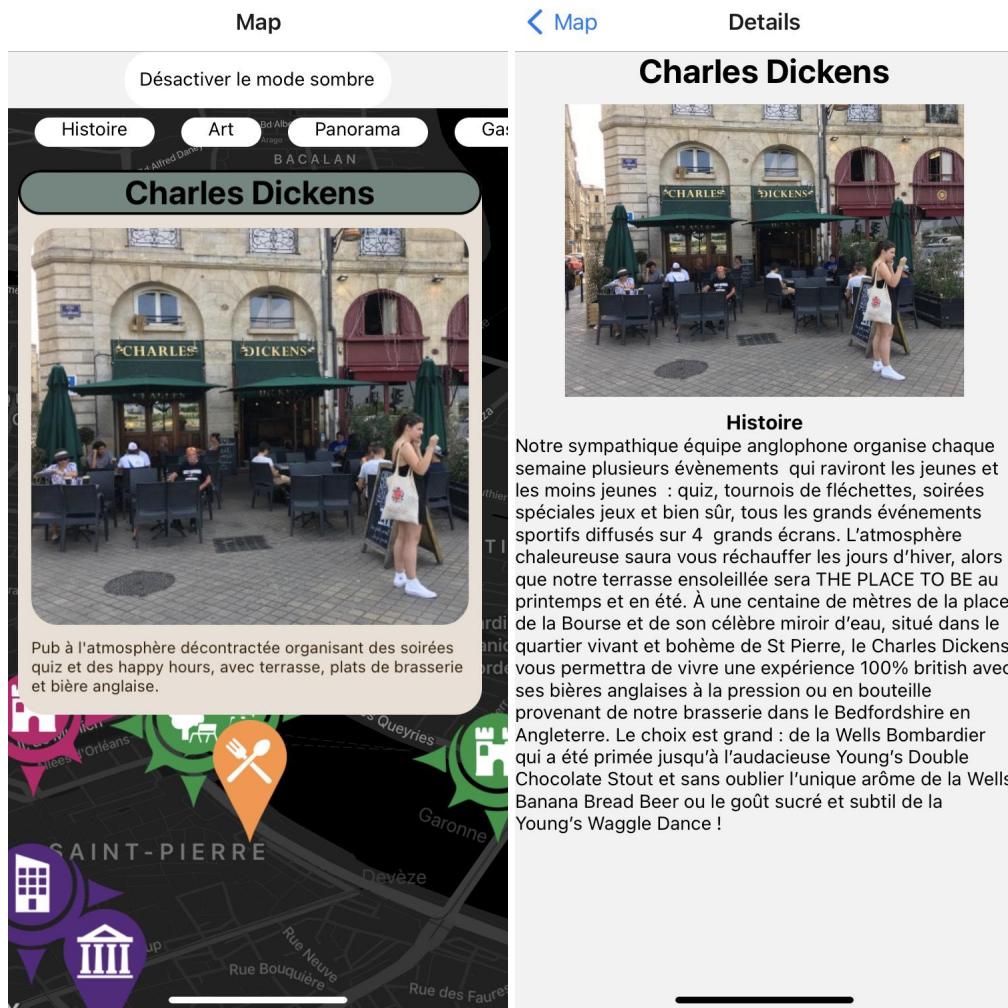


figure : Fiche descriptive marqueur et détail

Lorsque l'utilisateur clique sur un marqueur, une fiche descriptive apparaît au-dessus de celui-ci (capture d'écran ci-dessus à gauche).

Pour ce faire, le composant CustomMarker fait appel à la fonction FicheDescriptive dès que l'utilisateur vient presser le marqueur. Cette fiche est propre à chaque marqueur.

Puis si l'utilisateur veut avoir plus de détails sur le point d'intérêt, celui-ci peut cliquer sur la fiche descriptive afin d'ouvrir une nouvelle page qui donnera des informations supplémentaires (capture d'écran ci-dessus à droite).

Afin de pouvoir ouvrir une nouvelle page depuis une autre déjà ouverte, nous avons utilisé la bibliothèque React Navigation. Cette bibliothèque a permis ainsi de pouvoir, au clic de l'utilisateur sur la fiche, d'ouvrir la page détail tout en transmettant les informations sur le marqueur souhaité.

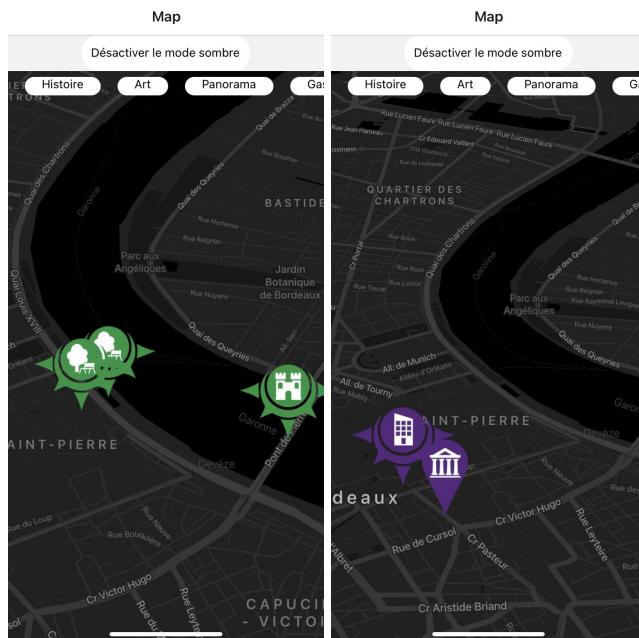


figure : Affichage des points d'intérêt de thème Panorama/Histoire

```

class CustomMarker extends React.Component {
  constructor(props) {
    super(props)
    this.themeChoisi = "Histoire"
  }
  render() {
    const { navigation } = this.props;
    return (
      <View>
        {MARKERS_DATA.map((marker) => (
          marker.theme != this.themeChoisi ? (null) :
          <Marker
            key={marker.id}
            //pinColor = {marker.color}
            icon={marker.imgmarker}
            coordinate={{
              latitude: marker.latitude,
              longitude: marker.longitude,
            }}
          >
            <Callout tooltip>
              <CalloutSubview onPress={() => {
                navigation.navigate('Details', { marker: marker });
              }}>
                {FicheDescriptive(marker)}
              </CalloutSubview>
            </Callout>
          </Marker>
        )))
      </View>
    )
  }
}

```

figure : Recherche par thème fonctionnelle depuis le code ligne surlignée

En entrant un thème choisi dans le composant CustomMarker, nous sommes arrivés à créer un système de filtrage des points d'intérêt. Il est donc possible d'afficher les points d'intérêt appartenant à un thème en particulier en cliquant sur un thème, puis en réactualisant toute l'application manuellement.

Cependant, ce système ne fonctionne pas en cliquant simplement sur les boutons visibles en haut de l'écran. En effet, le composant CustomMarker ne s'actualisant pas à chaque clic, les marqueurs affichés ne changent pas directement. Il faut donc forcer le component CustomMarker à se ré-actualiser.

## Conclusion et Acquis pédagogiques

La partie développement de ce projet nous a permis d'acquérir diverses compétences.

Grâce à notre phase de formation, nous avons pu apprendre à programmer en React Native. Nous pensions que, grâce à ce langage, nous pourrions aisément rendre notre application efficace à la fois sur Androïd et sur iOS. Cependant, des fonctionnalités importantes telles que la localisation ne marchent pas sur Android, et nous avons donc élaboré un prototype fonctionnant uniquement sur iOS.

Nous avons également découvert comment modéliser une application selon des besoins prédéfinis (déterminés par les résultats de l'enquête UX), et comment créer un diagramme de classe, nécessaire au bon développement du prototype.

Développer ce prototype nous a également permis d'apprendre à s'organiser lors d'un travail collaboratif, en s'intéressant à la bonne gestion du groupe. Nous avons notamment exploité les avantages qu'offrent github pour coder à plusieurs sur le même programme.

Si les résultats de notre prototype diffèrent légèrement des exigences fonctionnelles que nous avons fixées au départ, nous avons tout de même réussi à nous adapter face aux difficultés rencontrées et à produire un prototype présentant les principales fonctions de notre projet application.

## Conclusion

Pour tous les membres du groupe, le travail sur l'application "Baroudeurs" a été motivant et formateur. Les membres de chaque pôle ont développé de nouvelles compétences, et ce dans le cadre d'un projet pour lequel tous portaient un intérêt.

Comme dit précédemment, le pôle UX a pris du retard sur la phase finale, ce qui a impacté les livrables prévus. Au-delà d'une gestion du temps

critiquable sur la fin du projet, le déroulement s'est fait sans encombres, avec une bonne communication et une dynamique générale très positive. L'aventure commune "Baroudeurs" a atteint la majorité de ses objectifs, mais il reste encore beaucoup à faire pour que l'application voit le jour. Le projet n'est donc pas fini et le flambeau n'attend qu'à être repris.

# Bibliographie

## Développement

### Phase d'apprentissage

- [Tutoriel Openclassroom](#)
- [Styles de la carte](#)
- [React-native-maps](#)

## État de l'art

<https://www.newworld-map.com/#/>

<https://unbound.com/books/virtual-cities/>

<https://www.canadiangeographic.ca/article/inside-intricate-world-video-game-cartography>

<https://www.vgmaps.com/>

[https://www.afjv.com/news/10193\\_le-level-design-des-jeux-en-monde-ouvert-part-3-5.htm](https://www.afjv.com/news/10193_le-level-design-des-jeux-en-monde-ouvert-part-3-5.htm)

<https://www.usgamer.net/articles/the-20-best-in-game-maps>

<https://www.eurogamer.net/articles/2018-12-12-the-brilliance-of-video-game-maps>

<https://neocarto.hypotheses.org/12644>

<http://www.escapad.io>

<https://baludik.fr/>

<https://www.parisenigmes.com/>

<https://www.waynote.fr/>

<https://www.fairtrip.org/fr>

<https://www.france-explorateurs.fr/index.php?page=carte>

<https://explorama.app/>

<https://www.plages.tv/>

<https://citymapper.com/bordeaux>

<https://hidden.place/>

<https://www.google.com/travel/>

[Map Markers: An Explorer's Guide to Open World Video Games](#)

Typologie des joueurs de Marczewski [typologie des joueurs](#)

## UX

[UX Mapping Methods Compared: A Cheat Sheet](#)

[Experience map, parcours utilisateur ou carte de l'expérience utilisateur](#)

[Game Design Advance – Home of New York's Intellivisiongentsia](#)

[Game UI Database | Welcome](#)

[Game Developer](#)

[Player pleasure: exciting rewards](#)

Tondello et al. "28 heuristiques pour l'évaluation experte d'un système gamifié"

[Designing for meaningfulness - Player freedom and prototyping explorat](#)

Méthodes de design UX, 30 méthodes fondamentales pour concevoir des expériences optimales

Carine Lallemand, Guillaume Gronier. 2018, Collection Design Web

## Annexes

 **Cahier des charges.pdf**

 **Entretiens \_ Grille.pdf**

 **Entretiens \_ Synthèse.pdf**

## Dépot Github - Code source du prototype

Le dépôt Github étant public et donc visible par tous, le fichier app.json a été modifié afin de supprimer les clés de l'API Google Maps pour des raisons de sécurité. Si vous souhaitez lancer l'application, le code du fichier app.json doit être remplacé par le code suivant :

```
{
  "expo": {
    "name": "Baroudeurs",
    "slug": "Baroudeurs",
    "version": "1.0.0",
    "orientation": "portrait",
    "icon": "./assets/icon.png",
    "splash": {
      "image": "./assets/splash.png",
      "resizeMode": "contain",
      "backgroundColor": "#ffffff"
    },
    "updates": {
      "fallbackToCacheTimeout": 0
    },
    "assetBundlePatterns": [
      "**/*"
    ],
    "ios": {
      "supportsTablet": true,
      "config": {
        "apiKey": "AIzaSyDX1AemurEn2jZVQL0vmAe7LduLhLb01bw"
      }
    },
    "android": {
      "adaptiveIcon": {
        "backgroundImage": "res://backgroundImage"
      }
    }
  }
}
```

```
        "foregroundImage": "./assets/adaptive-icon.png",
        "backgroundColor": "#FFFFFF"
    },
    "config": {
        "googleMaps": {
            "apiKey": "AIzaSyDX1AemurEn2jZVQL0vmAe7LduLhLb01bw"
        }
    },
    "web": {
        "favicon": "./assets/favicon.png"
    }
}
```

[Questionnaire](#)

[Maquettes](#)

[Démonstration Vidéo Maquette](#)