

PROTOTYPE DE GAMIFICATION D'UN PARCOURS SPORTIF

Travail de Bachelor 2019

Professeur : Andres Perez-Uribe

Auteur : Lechaire Thomas

École : HEIG-VD

Cursus : Formation en emploi – Informatique logiciel

Date : Août - Septembre 2019

Table des matières

| | | |
|-------|--|----|
| 1 | INTRODUCTION | 4 |
| 1.1 | Résumé | 5 |
| 1.2 | Réflexion | 5 |
| 1.3 | Concept de gamification imaginé | 9 |
| 2 | MATÉRIEL | 12 |
| 3 | RECHERCHE DES TECHNOLOGIES | 16 |
| 3.1 | Internet of Things et Web of Thing | 17 |
| 3.2 | Node-red [3, 4] | 17 |
| 3.3 | MQTT & Broker | 20 |
| 3.4 | React-Native | 22 |
| 3.5 | Firebase[11] | 23 |
| 3.6 | Redux[19] | 24 |
| 4 | DEVELOPPEMENT DU PROTOTYPE | 26 |
| 4.1 | Introduction | 26 |
| 4.2 | Application Mobile | 27 |
| 4.2.1 | Concept | 27 |
| 4.2.2 | WireFrame[20] | 27 |
| 4.2.3 | Base de données | 30 |
| 4.2.4 | Développement | 33 |
| 4.3 | Bornes interactives | 41 |
| 4.4 | Installation du modem/module 4G | 41 |
| 4.5 | Le serveur de jeu | 44 |
| 4.6 | Gameplay | 46 |
| 4.6.1 | Zone de départ | 49 |
| 4.6.2 | Première énigme | 49 |
| 4.6.3 | Seconde énigme | 55 |
| 4.6.4 | Dernière énigme | 59 |
| 4.7 | Parcours et prise d'information sur le terrain | 63 |
| 4.8 | Problématiques et enjeux | 65 |
| 4.9 | Tests & Améliorations | 67 |
| 5 | CONCLUSION | 70 |
| 6 | REMERCIEMENTS | 71 |
| 7 | GLOSSAIRE | 72 |
| 8 | BIBLIOGRAPHIE | 74 |
| 9 | ANNEXES | 78 |
| 9.1 | Cahier des charges | 78 |
| 9.1.1 | Contexte et définition du projet | 78 |

| | | |
|--------|--|----|
| 9.1.2 | Objectif du projet | 78 |
| 9.1.3 | Périmètre du projet..... | 78 |
| 9.1.4 | Description fonctionnelle des besoins..... | 78 |
| 9.1.5 | Enveloppe budgétaire | 78 |
| 9.1.6 | Délais de réalisation | 78 |
| 9.2 | Analyse primaire et réflexion concernant la gamification du parcours sportif de la ville d'Yverdon | 79 |
| 9.2.1 | Application Mobile : | 79 |
| 9.2.2 | Les Bornes iBeacons ou NFC..... | 80 |
| 9.2.3 | Les capteurs de mouvements | 80 |
| 9.3 | Communication Bluetooth | 81 |
| 9.4 | Beacon Estimote | 81 |
| 9.4.1 | Procédure..... | 81 |
| 9.4.2 | Lecture et formation sur la technologie Beacon et Estimote | 81 |
| 9.4.3 | Initialisation des Beacons..... | 82 |
| 9.4.4 | Mise en service de l'application et d'Android Studio | 83 |
| 9.4.5 | Tests..... | 84 |
| 9.4.6 | Conclusion et résultats | 84 |
| 9.5 | Capteurs de mouvements[74] | 85 |
| 9.6 | Capteur de distance | 85 |
| 9.7 | Textes audio..... | 87 |
| 9.7.1 | Zone de départ..... | 87 |
| 9.7.2 | Zone des menhirs..... | 87 |
| 9.7.3 | Mauvais code | 87 |
| 9.7.4 | Bon code | 87 |
| 9.7.5 | Zone Clef : | 87 |
| 9.7.6 | Zone de l'observatoire..... | 88 |
| 9.7.7 | Zone du noisetier..... | 88 |
| 9.7.8 | Zone du noisetier (arbre libéré)..... | 88 |
| 9.7.9 | Zone observatoire (retour) | 88 |
| 9.7.10 | Zone du gardien | 88 |
| 9.7.11 | Erreur de code..... | 88 |
| 9.7.12 | Fin et succès | 88 |
| 9.8 | Carte du jeu | 89 |

1 INTRODUCTION

Dans le cadre du travail de Bachelor et selon la demande de la ville d'Yverdon, il a été demandé à la HEIG-VD, par le biais du professeur M. Perez-Uribe Andres, de « gamifier » le parcours sportif présent aux abords de l'école à la route de Cheseaux. Le parcours est actuellement à l'abandon, les installations sont vétustes et pour certaines délabrées. Cette demande s'inscrit donc dans une volonté de la part de la ville d'Yverdon, de mettre à jour l'ensemble des installations présentes à proximité de la plage et de fournir aux citoyens de la ville un parc sportif et une offre de service correcte.

Détente, loisirs et nature

Aller à la plage, sortir son ballon, retrouver le plaisir de la pêche, observer les oiseaux de la Grande Cariçaie: Gare-Lac rapproche soudain la ville de son lac.

En lisière du plus grand parc de la ville auquel les Yverdonnois sont très attachés, le quartier offre une qualité de vie et d'espace inégalée. Les espaces de sport et de détente situés entre le lac et le nouveau quartier seront à terme aussi repensés pour en faciliter l'accès et l'attractivité.



© Perspectives : ARCHIGRAPHIE.CH

Figure 1 : « Grand-projet » article tiré du site internet de la commune d'Yverdon[1]

L'ensemble du travail de Bachelor sera porté sur la recherche, l'évaluation et la réalisation d'une preuve de concept permettant d'améliorer l'expérience sportive des citoyens en offrant un parcours sportif interactif. Le projet a pour objectif d'analyser les différents moyens, technologies et possibilités à disposition pour l'établissement et la création d'un prototype de jeu. Le but étant d'offrir à la ville d'Yverdon les résultats des tests, les recherches et les réflexions qui auront été menées tout au long du programme.

1.1 Résumé

Le thème du projet peut paraître de prime abord un peu abstrait. Il est donc important de définir le but de ce travail, ainsi que les principales étapes qui permettront d'obtenir une preuve de concept et de faisabilité. Il est nécessaire d'introduire rapidement le terme de « gamification » qui sera utilisé tout au long de ce rapport. La « gamification » est un terme anglais traduit en français par « ludification ». La gamification est basé sur l'utilisation des mécanismes du jeu ainsi que sur ces similarités, appliquées à d'autres domaines, comme le sport dans ce travail.

Le travail en lui-même n'a donc pas pour but la création et la mise en place d'un projet final et complet. Il s'insère plus facilement dans un objectif de recherche et d'évaluation. Trouver, tester et comprendre quelles technologies et outils permettraient de rendre le parcours sportif, aux abords du lac plus ludique, le tout dans une optique d'amélioration de l'expérience de l'utilisateur. Pour arriver à un prototype abouti, le projet sera divisé en plusieurs parties. L'ensemble du rapport présente la solution imaginée puis, la réalisation du prototype. La réflexion sur le jeu étant une part importante de ce projet, le chapitre 1.2 ci-dessous aborde le questionnement qui a permis de mettre en avant certains aspects problématiques du projet. Cette réflexion offre une vision des points positifs et négatifs afin d'avoir, lors d'une future intégration, toutes les informations nécessaires.

Au chapitre 9 du présent document se trouvent les rapports et autres retours d'expériences, ainsi que des tests technologiques et d'autres essais n'ayant pas abouti à des résultats suffisamment concluant.

Pour la suite, le lecteur est renvoyé vers le glossaire au chapitre 7 où l'ensemble des termes techniques est disponible et permettra une meilleure compréhension du rapport.

1.2 Réflexion

Le partage des idées et des concepts imaginés est d'une grande importance en ce qui concerne ce travail. En effet, penser et réfléchir à toutes les possibilités d'interactions dans un jeu demande la collaboration et la mise en commun des idées de plusieurs personnes. Il a donc été nécessaire de réaliser plusieurs « Brainstorming » avec des personnes de différents milieux (techniques, sportif, non sportif, géocacheur, joueur d'escape room), afin d'extraire les points manquants et de compléter la réflexion personnelle déjà effectuée sur le projet.

C'est durant ces sessions de partage et de mise en commun d'idées que des nombreuses questions ont été mises en avant. De nombreuses situations peuvent se présenter durant un jeu et, il devient difficile de trouver une solution si aucune réflexion n'a été faite en amont. Cela offre un moyen d'anticiper un grand nombre d'actions, de comportements et de réactions possibles des futurs utilisateurs. Offrir une vision globale du concept ainsi qu'un prototype de démonstration permettra alors de répondre au cahier des charges dans son ensemble, donnant ainsi une vision à 360° sur le projet et les points importants à prendre en considération lors de sa réalisation.

Imaginons maintenant que nous demandions à un utilisateur de faire une série d'exercices ou d'atteindre un certain point dans une limite de temps fixée. Comment faire en sorte que chaque utilisateur puisse utiliser l'application selon son niveau de compétence et ses aptitudes physiques ? Comment faire en sorte qu'il ne se sente pas pénalisé ou découragé par l'activité au point d'abandonner ? Comment faire pour au contraire le motiver à performer à un niveau supérieur ?

Certaines questions ont été mises en avant car elles ciblent et donnent au projet son orientation. D'autres resteront malheureusement sans réponse, mais penser à toutes les éventualités lors de la mise en production du prototype offrira une marge de manœuvre plus importante en cas de réalisation concrète du projet.

Ci-dessous se trouve une liste non exhaustive des quelques questions les plus impactantes.

- *Quel est le sens défini lorsque l'on parle de gamification et a qui s'adresse cette gamification ?*
- *Quel est le but de cette gamification et jusqu'à où cette gamification doit-elle être poussée ?*
- *Comment l'interaction s'établit entre le sportif et le terrain de sport ?*
- *Comment faire pour capter les mouvements du sportif ?*
- *Quel système de point mettre en place ?*
- *Quel est le lien entre l'application mobile et l'appareil/dispositif sur le terrain ?*
- *Est-ce que le sportif possède un profil ?*
- *Peut-il sauvegarder ses performances pour les améliorer ensuite ?*
- *Est-ce que le sportif se mesure à d'autres ou contre lui-même ?*
- *Comment mettre en place un jeu modulable ?*
- *En cas d'utilisation de bornes, où est-il possible de disposer ces dernières ? Existe-t-il un risque de dégradation en cas de pose de matériel sur le terrain ?*
- *Quel est l'état actuel du parcours et peut-on utiliser les machines déjà présentes ?*
- *Comment faire pour éviter que les gens ne trichent ?*

Pour continuer la réflexion, il est important de répondre à ces questions afin de comprendre quels sont les points qui pourraient bloquer la mise en place d'un tel système et son développement.

- ***Quel est le sens défini lorsque l'on parle de gamification et a qui s'adresse cette gamification ?***
Est-ce le fait d'avoir des énigmes ou le fait d'avoir un classement sportif ? Cette question se lie à la suivante. Faut-il privilégier le sport ou le jeu ? Quel est le juste équilibre entre les deux. Trop de sport pourrait décourager certains préférant le jeu. Inversement, trop de jeu découragerait les sportifs qui cherchent le défi et/ou la performance. Prendre en compte cette dualité donne une ligne à suivre au projet. Le jeu et la gamification seront privilégiés durant la réalisation. Le thème central n'étant pas le sport mais bien la gamification.
- ***Quel est le but de cette gamification et jusqu'à où cette gamification doit-elle être poussée ?***
Est-ce un jeu sportif ou un sport ludique ? Est-ce que la base est le sport auquel est ajouté une touche de jeu afin de motiver les gens ou est-ce un jeu d'énigme comme géocaching auquel on ajoute une partie sportive ? Lié à la question précédente, il est important de trouver l'équilibre entre le sport et le jeu. L'ajout de la gamification au parcours sportif démontre cependant qu'il s'agit de sport dans lequel on ajoute une part de jeu et non l'inverse. Le but final est donc d'amener le joueur à faire un effort physique. Cependant, l'intégration du jeu devient le moteur. C'est donc sur la motivation par la gamification qu'il faut se focaliser afin d'atteindre le but qui lui est sportif.

- **Comment l'interaction s'établi entre le sportif et le terrain de sport ?**

Renforcer le lien entre le sportif et l'environnement permet de motiver le sportif. Les moyens à disposition ne permettront pas de tester des technologies couteuses, comme des capteurs de mouvements ou la reconnaissance par le biais de caméra. Mais un système de borne « maison » sera testé afin d'augmenter l'interaction avec l'utilisateur. Le lien entre ces bornes et le joueur sera amélioré grâce à l'utilisation d'une application mobile et un ensemble d'énigmes formant une quête. Cette partie est abordée plus en détail lors de la présentation du concept au chapitre 1.3.

- **Comment faire pour capter les mouvements du sportif ?**

Il existe à l'heure actuelle plusieurs moyens de capter le mouvement d'un sportif. Plusieurs systèmes seront testés durant le travail, afin de comprendre s'il est possible de capter de manière performante un déplacement ou d'autres types d'exercices. Ajouté à cela, dans le cas où la vérification de la pratique sportive n'est pas concluante, d'autres capteurs avec des boutons, des lecteurs de carte RFID, etc..., seront testés. Les capteurs seront utilisés pour augmenter la dimension du jeu grâce aux interactions entre l'utilisateur, les bornes interactives et les énigmes à résoudre.

- **Quel système de point mettre en place et est-ce réellement utile ?**

Le système de point est la base de la gamification, mais ce dernier engendre des questions supplémentaires. Par exemple, faut-il donner plus de point aux gens rapides ou à ceux qui ont résolu l'énigme du premier coup ? Est-ce qu'une personne qui a fait plus de km (distance), récolte plus de point, etc... Toutes ces questions demandent une plus grande réflexion et devront sûrement être discutées par la suite en cas de réelle implémentation d'un produit final. Pour le moment le postulat sera d'additionner le temps total, la distance et un certain nombre de points pour la résolution du jeu. Chaque énigme fait gagner des points et plus le joueur met longtemps plus il en perd. L'intérêt d'un classement peut être aussi abordé. Souhaite-t-on augmenter la compétition ou est-ce que le jeu est plutôt de type « escape room » en extérieur avec un temps donné pour résoudre la quête en entier. Un business plan et une étude de marché pourraient aider à trancher cette question.

- **Quel est le lien entre l'application mobile et l'appareil/dispositif sur le terrain ?**

De manière certaine, l'utilisation du téléphone portable sera indispensable. Il faut maintenant penser au moyen le plus rapide de communiquer entre les différents appareils. Le Bluetooth sera testé, mais cela implique une synchronisation du téléphone avec les bornes et cela risque de poser certaines limitations. Des avantages sont aussi présents. En imaginant devoir transformer le prototype en projet rentable, l'utilisation du Bluetooth permettrait de verrouiller les bornes uniquement aux appareils autorisés. Il serait alors possible de vendre des abonnements ou alors des parties et demander une contribution financière en retour. En test aussi, l'utilisation du réseau 4G ainsi que des Beacons, afin de voir s'ils pourraient avoir une utilité sur le terrain. Mais la communication 4G reste la plus rapide et sera privilégiée durant la réalisation de ce prototype.

- **Est-ce que le sportif possède un profil ?**

Le sportif devra posséder un profil dans l'application mobile, c'est cela qui permettra de faire l'historique de ces performances et le cas échéant, de faire un classement entre les différents joueurs. Il peut aussi être un moyen pour vérifier l'accès ou les abonnements/parties achetées en cas de produit final. Il est donc indispensable.

- **Peut-il sauvegarder ses performances pour les améliorer ensuite ?**

Le but du profil est de sauvegarder ses performances ainsi que de garantir un suivi de la pratique sportive et de l'évolution du jeu.

- ***Est-ce que le sportif se mesure à d'autres ou contre lui-même ?***

Il pourrait être possible de développer les deux. Si le temps le permet et que la création d'un classement est intégrée à l'application, ce dernier pourrait être une source de motivation pour certains. Quant aux autres, ils pourront sans problèmes tenter de battre leurs précédents records. Ce n'est cependant pas le point central de ce travail.

- ***Comment mettre en place un jeu modulable ?***

La principale question du projet est celle-ci, comment faire en sorte que le jeu puisse facilement s'adapter ? Comment faire en sorte de pouvoir ajouter, supprimer ou modifier le parcours ou encore les énigmes ? Comment faire pour que le joueur qui a déjà fait le parcours, et donc les énigmes, puisse jouer à nouveau ? Si le joueur possède déjà les réponses aux énigmes il pourra améliorer son temps et gagner plus de points, le tout sans améliorer sa performance physique. Utiliser un serveur central permet de récupérer l'ensemble des informations provenant des diverses bornes interactives et de traiter ces informations dans ce dernier. La logique du jeu se trouve donc à un seul endroit et il est donc plus facile de modifier le jeu/parcours en cas de besoin. Les énigmes sont cependant figées. D'abord car elles sont liées à une borne spécifique, mais aussi à un environnement spécifique du terrain.

- ***En cas d'utilisation de bornes, où est-il possible de disposer ces dernières ? Existe-t-il un risque de dégradation en cas de pose de matériel sur le terrain ?***

Comment faire pour disposer des bornes sur le terrain pour les utilisateurs de l'application sans que ces dernières ne soient volées ou casées ? Cela crée des couts supplémentaires comme la création de structures de protection. De plus, il faut trouver un moyen d'alimenter en électricité ces appareils et avoir quelqu'un qui puisse intervenir physiquement en cas de problème. Le risque de dégradation, aux vues des installations actuellement présentes sur le parcours, est plutôt élevé. Certains postes du parcours sportif sont tagués ou cassés ce qui tend à répondre qu'il ne sera pas possible de laisser des bornes directement sur place. Pour la réalisation du prototype des bornes amovibles seront utilisées. Elles devront être mises en place avant le début du jeu et récupérées à la fin.

- ***Quel est l'état actuel du parcours et peut-on utiliser les machines déjà présentes ?***

Le parcours sportif actuel est à l'abandon et l'utilisation des postes sportifs n'est pas envisageable. De plus, il faudrait pouvoir trouver un moyen de vérifier que la personne effectue réellement les exercices demandés. Faut-il faire confiance à la personne ou, étant donné qu'il s'agit d'un jeu, vérifier que le joueur respecte les règles ? Afin d'éviter l'ajout de problèmes liés à une quelconque vérification, aucun exercice à répétition ne sera demandé pour le moment. Le seul sport pratiqué sera la course à pied.

L'ensemble de ces questions recueillies lors de discussions et brainstorming avec des personnes auxquelles le projet a été présenté, pose la base de la réflexion et pose les premières briques du concept de jeu imaginé et présenté au point suivant. Tous ces gens se sont dit intéressées et ont proposé de tester l'application, mais une remarque est souvent revenue : « Comment faire avec les tricheurs ? ».

- ***Comment faire pour éviter que les gens ne trichent ?***

Dans ce projet, le postulat et qu'il n'est pas demandé d'implémenter des mécanismes contre les tricheurs. Ce genre de personne existe et existera toujours, mais l'enjeu du travail est la réalisation d'un prototype de démonstration viable et d'une application mobile permettant la synchronisation du profil et des données de jeu. Le temps, ainsi que le nombre de personnes travaillant au développement étant des facteurs à prendre en compte, la lutte contre les tricheurs ne sera abordée que de manière sporadique dans le rapport, donnant quelques pistes possibles pour empêcher cela.

1.3 Concept de gamification imaginé

Ici est présenté le premier concept du parcours imaginé et non le prototype final réalisé.

Afin que le lecteur comprenne les tenants et aboutissant du projet, quelques explications sont nécessaires concernant l'idée globale imaginée pour cette gamification.

Le parcours sportif de la ville d'Yverdon n'est pas en très bon état, les installations sont vétustes et complètement abandonnées, mais la zone du bord du lac est très agréable et beaucoup de coureurs profitent de l'ombre des arbres pour faire du sport. La ville d'Yverdon avait mis à disposition un certain nombre d'infrastructures permettant aux habitants de pratiquer une activité physique de manière gratuite. Actuellement ces infrastructures sont vétustes, mais leur réaménagement est, comme vu au chapitre 1 ci-dessus, un des projets envisagés par la ville pour cet espace.



Figure 2 : Installations actuelles dégradées

Le jeu imaginé se déroule de la manière suivante :

C'est aidé de son téléphone, sur lequel une application dédiée est installée, que l'utilisateur se connecte. Il s'inscrit afin de créer un profil permettant à l'application d'enregistrer l'ensemble de ses sessions et garder l'historique des entraînements ou des parties jouées. Le joueur possède un profil, avec un résumé de ces points et d'autres données importantes selon les besoins de l'application. Une liste avec les précédentes parties sera visible sur le profil. Il sera possible de voir un résumé d'une session faite précédemment ainsi que les informations concernant cette dernière, dont une carte du parcours effectué.

Lorsque le joueur décide de lancer une partie, il peut alors jouer de deux manières différentes. S'il est seul, il pourra lancer une partie en mode « solitaire ». Les interactions en rapport avec les énigmes apparaissent sur le téléphone et il entre les divers codes et autres solutions du jeu sur son appareil portable.

L'autre mode de jeu offre une dimension supplémentaire. Grâce à un système de bornes physiques, l'utilisateur interagit non plus avec son portable, mais avec un dispositif physique. Le problème de cette

solution est qu'il est nécessaire d'être plusieurs, cela afin que quelqu'un puisse positionner les bornes aux endroits prévu avant le début du jeu. Compte tenu du temps à disposition, seule la partie avec les bornes sera réalisée pour le projet. Si le temps le permet, la seconde partie sera réalisée.

Pour jouer, une fois la partie lancée sur l'application, le joueur doit se rendre dans la zone de départ.



Figure 3 : Exemple du plan de jeu. Zone de départ et 1^{ère} énigme

La zone de départ s'affiche sur une carte visible sur le téléphone mobile. Les zones de jeux sont délimitées par des cercles de plus ou moins grands diamètres que l'on nommera dans la suite du rapport « geofence ». Une fois le joueur détecté dans la zone de départ, il déverrouille et lance la première énigme. Il doit alors se rendre le plus rapidement possible (en courant) dans cette seconde zone pour résoudre l'éénigme n°1. La rapidité ayant un impact sur le nombre de points que l'utilisateur gagne en fin de partie. Le joueur doit prendre en compte ce facteur lors de sa performance. Toutes les indications relatives au jeu et à la progression sont données par audio au joueur.

Cette gamification, qui permet de gagner plus ou moins de points, permet à un utilisateur souhaitant refaire le parcours de s'améliorer. Le but imaginé, est de résoudre l'ensemble du parcours sportif et des énigmes en un minimum de temps. Chaque énigme devra vraisemblablement avoir des difficultés différentes, des activités physiques à faire, ou encore des données à collecter sur le terrain afin de pouvoir résoudre le jeu dans son ensemble.

Des niveaux de difficultés différents permettraient au joueur de pouvoir rejouer, s'il le souhaite, sans avoir déjà les réponses aux précédentes énigmes. L'éénigme possède un temps limite, comme un compte à rebours durant lequel sa réalisation est possible. Si le joueur n'arrive pas, dans le temps à disposition à résoudre l'éénigme, ou s'il ne trouve pas la bonne solution, ce dernier devra effectuer, sur le même principe qu'au biathlon (où la pratique du *sprint* demande la réalisation d'un tour de 150 mètres

pour chaque cible manquée durant le tir) un aller-retour permettant de relancer l'éénigme. Cela demande à l'utilisateur premièrement, de chercher les indices dans la zone d'éénigme rapidement et, deuxièmement, en cas d'échec, de devoir par le biais du sport et d'un effort physique, réinitialiser l'éénigme. Ci-dessous se trouve un exemple d'affichage de la zone de réinitialisation. Tant que le joueur ne s'est pas rendu dans cette dernière, il lui est impossible de tenter de résoudre l'éénigme à nouveau.

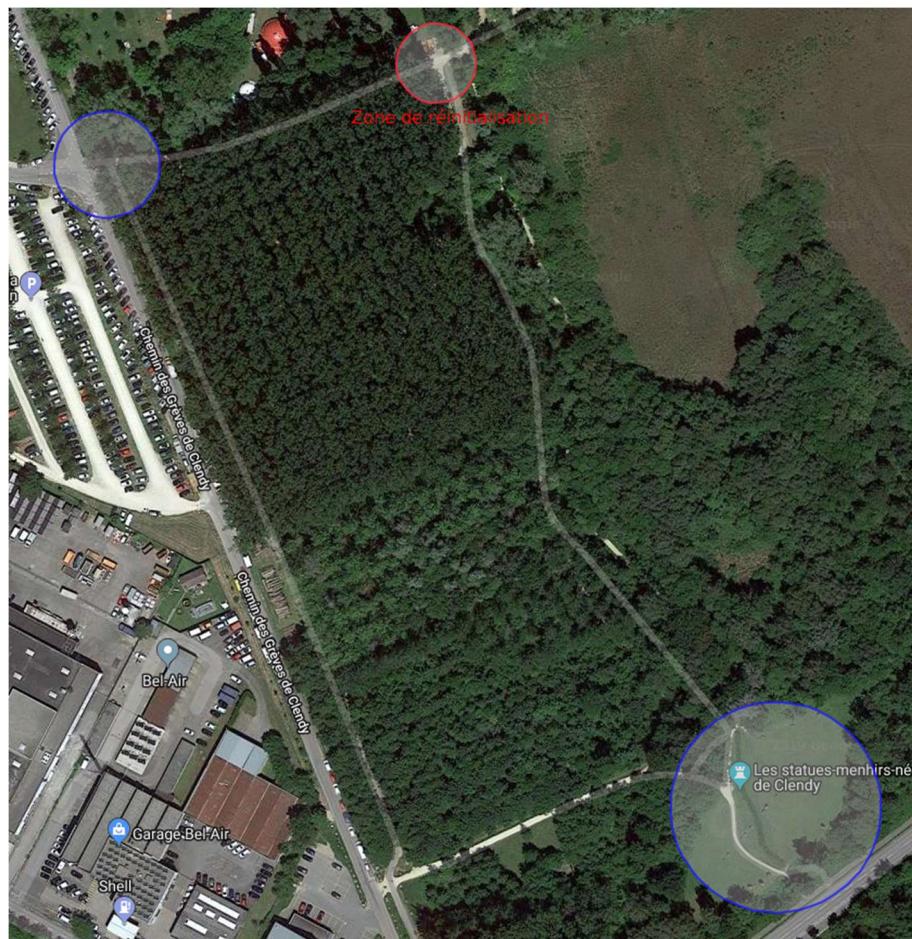


Figure 4 : Zone de réinitialisation

Une éénigme résolue a pour effet de déverrouiller l'éénigme suivante. Une nouvelle zone (geofence) s'affiche sur la carte et l'utilisateur peut s'y rendre et continuer le jeu. Résolvant l'éénigme suivante et ainsi de suite, jusqu'à la fin du parcours et du jeu.

A la fin de l'aventure, le joueur se voit attribuer un certain nombre de points en lien avec les performances de sa partie. Le système de points n'est pas demandé dans le cahier des charges, mais il pourrait être intéressant de trouver un moyen de récompenser les joueurs pour leurs performances ou selon un classement par exemple. Pour faire un classement, il faut pouvoir attribuer des points à chaque action. Selon la rapidité, la distance parcourue, ou encore le nombre de tentatives pour résoudre une éénigme et réfléchir à une solution permettant aussi de prendre en compte un niveau de difficulté des éénigmes. Comme vu précédemment, il est possible d'imaginer des niveaux de difficultés différents, concernant les éénigmes, mais aussi concernant la partie sportive, en augmentant les distances entre les zones de réinitialisation et les éénigmes par exemple. Le joueur parcourt ainsi une plus grande distance et peu alors progresser.

Pour résumer, il est demandé au sportif de progresser rapidement sur le terrain, récolter des indices qui sont disposés dans une ou plusieurs zones délimitées par un cercle sur la carte (comme illustré sur la Figure 4) et de résoudre les éénigmes pour terminer le parcours.

2 MATÉRIEL

Pour mettre en place l'ensemble du projet, une quantité importante de matériel a été nécessaire. Ci-dessous la liste non exhaustive du matériel utilisé et mis à disposition pour le travail. L'ensemble des composants électroniques est fourni par la HEIG.

- **3 Raspberry Pi 3**

Ces petites cartes possèdent la puissance d'un petit ordinateur. Ce format, leur permet d'être facilement transportable et peu énergivore. En effet une powerbank de 4000mh permet d'alimenter pendant environ 5h la Raspberry Pi. Plusieurs ports USB sont à disposition, un port Ethernet ainsi qu'un ensemble de pin GPIO. Ces différents ports permettront d'interagir avec divers capteurs et autres boutons.



Figure 5 : Raspberry Pi 3

- **Divers capteurs**

HC-Sr05 – capteur de distance utilisant les ultrasons pour déterminer la distance.



Figure 6 : Capteur de distance

HC-SR501 Infrarouge PIR détecteur de mouvement



Figure 7: Capteur de mouvement

Connecteur GPS et modem 4G



Figure 8 : Module 4G et GPS

- **Un module RFID de lecture écriture**
Fourni avec une carte et un badge



Figure 9: module RFID avec deux badges

- ***Un clavier digital***



Figure 10: Example de clavier numérique

- ***3 Powerbank Xtrom 4000mah***

Ces trois powerbank permettent d'alimenter les Raspberry lorsqu'elles sont disposées, en extérieur, sur le terrain.



Figure 11 : Powerbank Xtrom

- **Ensemble de composant électroniques**
Divers câbles, des résistances, une planche de prototypage, des boutons, des leds, etc...

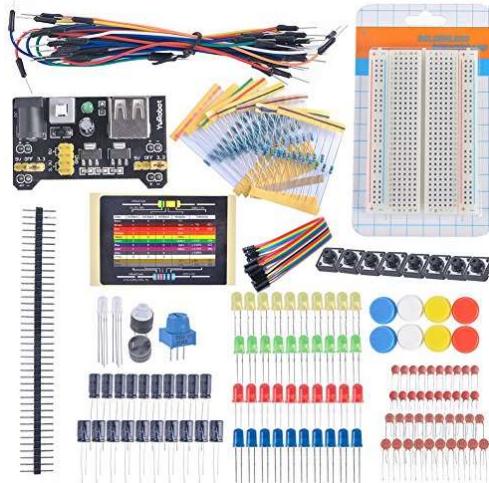


Figure 12 : Kit de composant électronique

- **Server de jeu**
Machine virtuelle Ubuntu installée sur un serveur privé.
- **Visual Studio Code**
Environnement de développement utilisé pour le développement de l'application mobile et des scripts. Il possède une bonne intégration des plugins et des librairies node Js
- **Fritzing**
Pour la création des schémas électriques des bornes.
- **Android Studio**
Environnement de développement pour Android (Java), utilisé pour émuler des téléphones mobiles.
- **Téléphone Mobile Android**
- **Ordinateur**
Un ordinateur pour toute la partie de programmation et de rédaction
- **Internet**

3 RECHERCHE DES TECHNOLOGIES

Afin de réaliser les différentes parties qui composent ce projet, du temps a été dédié à la recherche de diverses technologies. Dans le cadre d'un défi sportif il est en effet indispensable de ne pas attendre sur le matériel et d'utiliser des technologies qui permettent de répondre et transmettre rapidement les données. Il est aussi nécessaire que chaque appareil puisse communiquer avec les autres cela permet d'assurer le suivi et la progression de l'utilisateur dans l'application.

Dès le départ, une grande importance a été donnée à la réflexion et l'implémentation d'une solution adaptable/modulable au mieux. Un nombre conséquent de questions a été abordé dans la partie de réflexion et la partie de recherche des technologies se devait de trouver un moyen de communication rapide et simple entre tous les appareils afin de pouvoir déployer des changements ou des corrections rapidement sur les appareils.

Malgré le coût plus important des modules, la connexion de chaque boîtier au réseau téléphonique grâce à un module 4G et la communication par un serveur central semblait la solution la plus adéquate pour ce projet.

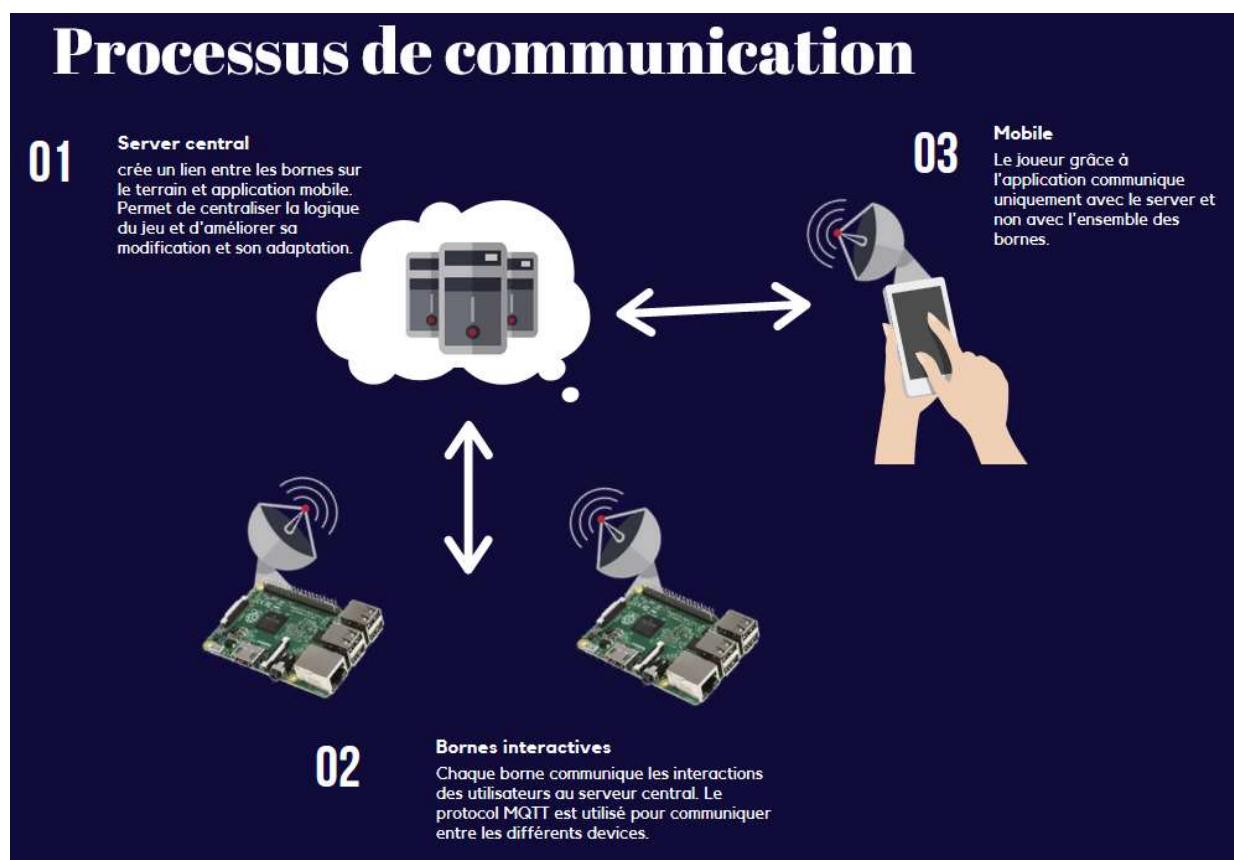


Figure 13 : Processus de communication - [2]

En effet, pouvoir centraliser les communications permet à l'application mobile de n'avoir à recevoir/communiquer qu'avec un seul appareil. Faire une seule requête lors de la demande/envoi d'information réduit considérablement l'utilisation du téléphone ainsi que des bornes. Permettant d'économiser les données utilisées par le modem 4G, réduisant ainsi les coûts liés aux cartes SIM. Cela permet aussi aux Raspberry Pi d'éviter d'implémenter la logique du jeu. Ces dernières se contentent d'envoyer au serveur central les interactions de l'utilisateur de manière non traitée.

Le serveur central est alors configuré pour traiter les requêtes des trois bornes. Il vérifie les entrées, communique avec la base de données et renvoie les résultats à l'application mobile et au joueur. De ce fait, si un changement est nécessaire, il suffit de le faire dans le serveur central sans devoir reconfigurer chaque borne interactive.

Cela introduit, cependant, une dépendance à un nœud unique. Ne posant pas de problèmes pour le développement du prototype, mais demandant la mise en service de serveurs de secours pour la redondance du système en cas de mise en production.

3.1 Internet of Things et Web of Thing

L'internet des objets ou IoT, pour internet of Thing, est une définition utilisée lorsqu'on parle d'objets physiques avec lesquels il est possible d'interagir via une communication réseau. Ces mêmes objets, selon leurs interfaces, peuvent se connecter à internet et il devient alors possible de récupérer leurs informations. On parle alors de Web of thing (wot) ou de l'internet des objets lorsqu'il est question d'utiliser le réseau internet pour communiquer à distance avec ces appareils électroniques.

Ces deux notions sont importantes et même si elles ne sont que des concepts englobant d'autres technologies, elles font partie intégrante du projet. En effet, toutes nos bornes interactives peuvent être, et seront, considérée comme IoT. La principale difficulté sera de mettre en place plusieurs technologies, provenant de différents fabricants et constructeurs afin de créer un boîtier fluide et réactif pour l'utilisateur.

3.2 Node-red [3, 4]

Basé et développé sur la technologie Node Js, Node-red est un outil de programmation visuel permettant la programmation basée sur un flow. Node-red s'installe depuis un script disponible sur leur site qu'il suffit de coller dans un terminal linux. Cela permet l'installation de tous les composant ainsi que de l'interface web.

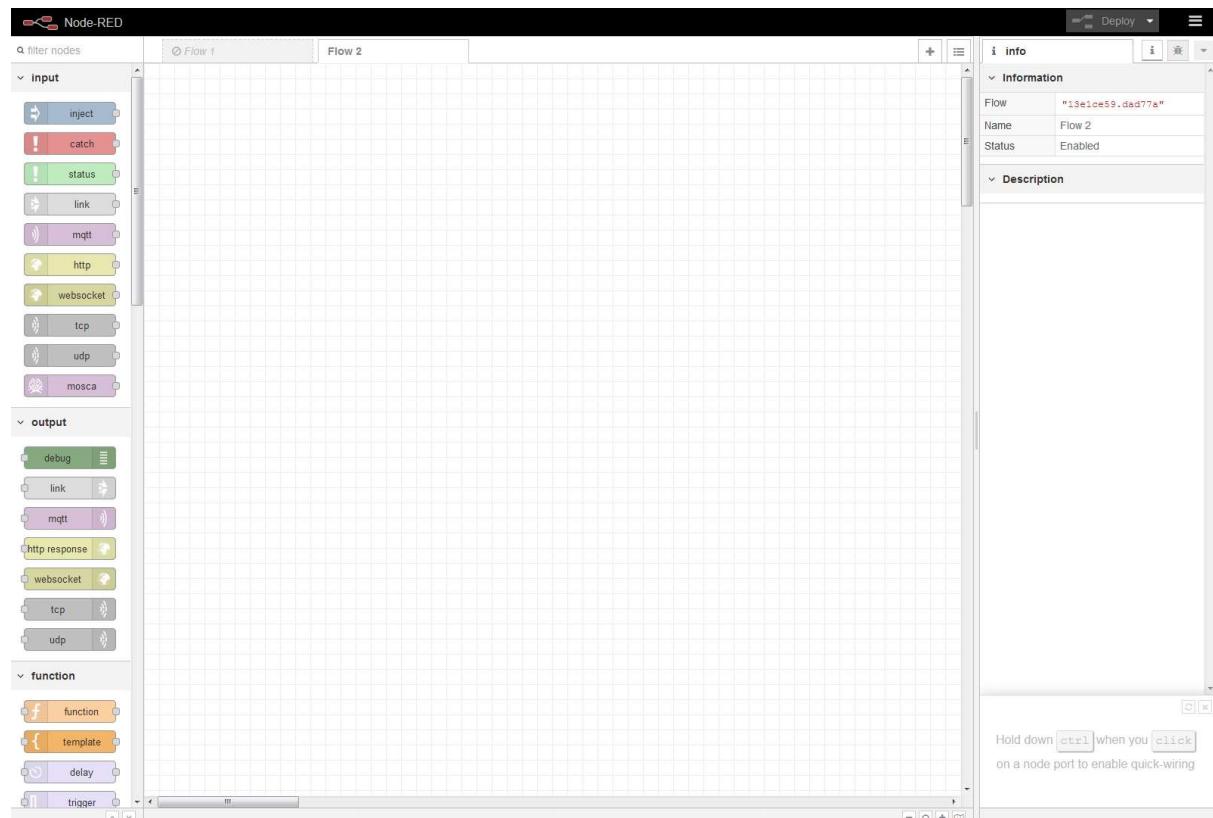


Figure 14 : Interface web de Node-red

Ce type de programmation permet d'utiliser des nœuds que l'on relie entre eux, donnant ainsi le sens de la programmation. Ce genre de programmation est très visuelle et permettrait à une personne non familière avec la programmation une rapide prise en main, dans le cas d'éventuelles modifications et de continuation ultérieure du projet.

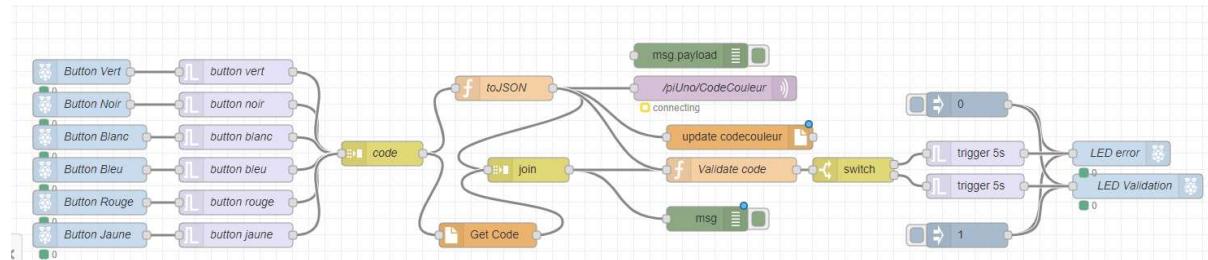


Figure 15 : Exemple de flow sous Node-red avec plusieurs nœuds

L'avantage de cette solution est qu'elle permet, lorsqu'elle est installée sur une Raspberry Pi, d'avoir accès à l'ensemble des pin GPIO et de les programmer de manière distante. Chaque pin GPIO peut être assigné directement depuis l'interface. Comme le montre l'image ci-dessous. Il suffit pour cela de prendre un nœud GPIO dans Node-red (1), de sélectionner le pin (2) à relier. On peut ensuite dire s'il s'agit d'un pin GPIO d'entrée ou de sortie (output/input) et même définir son état lors du démarrage (3).

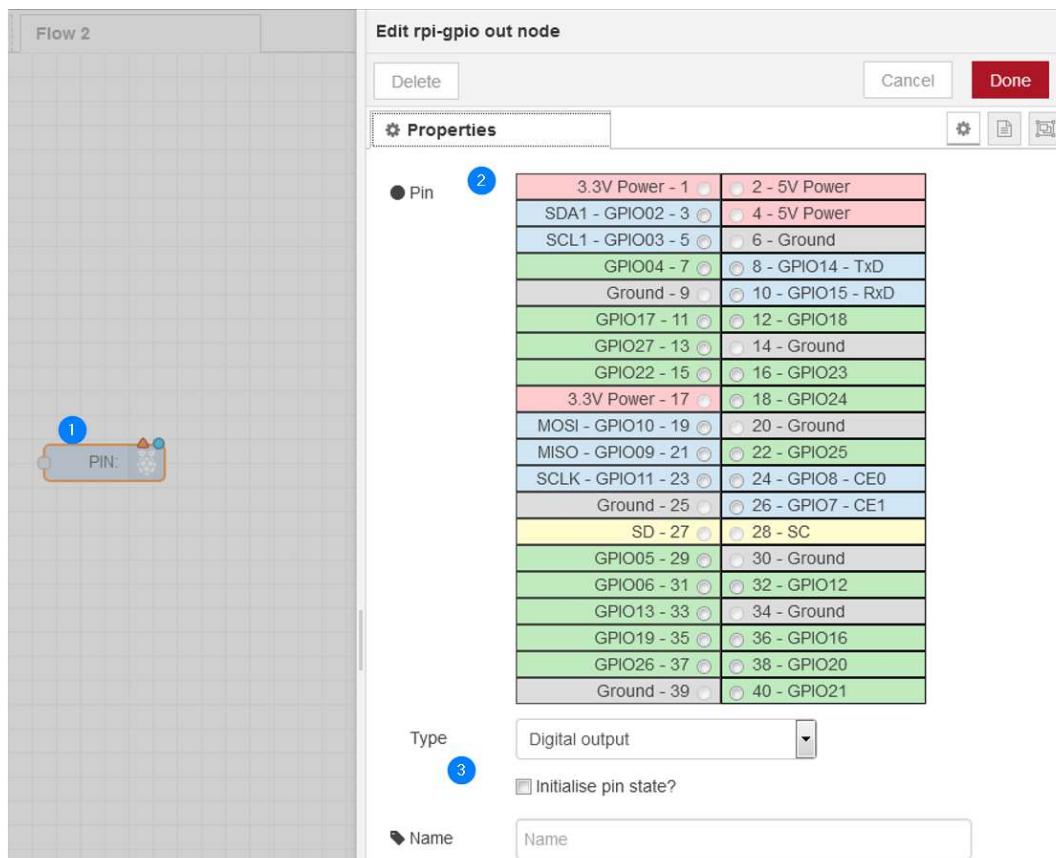


Figure 16: Exemple de configuration d'un PIN GPIO sous Node-red

Pour communiquer, Node-red utilise la technologie MQTT, présentée au point suivant, qui permet d'envoyer ou de recevoir des messages grâce à un broker ainsi que deux nœuds spécifiques. Ces deux nœuds implémentent le principe de publication et de souscription.

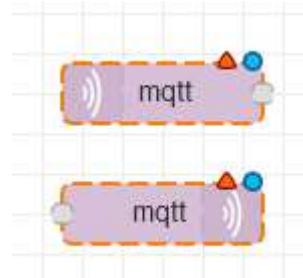


Figure 17 : Nœud de réception et d'envoi

Le site <https://flows.nodered.org> [5] permet de trouver de nombreux flow réalisé par d'autres développeurs ou des nœuds pour augmenter les interactions possibles. Ci-dessous quelques nœuds les plus téléchargés, comme un nœud pour interagir avec twitter, envoyer un mail ou encore accéder à des services google.

Node-RED Library

Add a flow

Find new nodes, share your flows and see what other people have done with Node-RED.

Search library

flows nodes 2137 of 3471 things

Sort by:
 recent
 downloads
 rating

| | | |
|---|---|--|
| node-red-dashboard A set of dashboard nodes for Node-RED v2.15.5 18482 4.7 (12) node | node-red-node-twitter A Node-RED node to talk to Twitter v1.1.5 13116 node | node-red-node-email Node-RED nodes to send and receive simple emails v1.6.2 12973 5.0 (1) node |
| node-red-node-rbe A Node-RED node that provides report-by-exception (RBE) and deadband capability. v0.2.4 11591 5.0 (2) node | node-red-node-sentiment A Node-RED node that uses the AFINN-165 wordlists for sentiment analysis of words. v0.1.4 11276 node | node-red-node-feedparser A Node-RED node to get RSS Atom feeds. v0.1.14 11140 node |
| node-red-node-tail A node to tail files for Node-RED v0.0.3 10312 node | node-red-node-random A Node-RED node that when triggered generates a random number between two values. v0.1.3 5224 node | node-red-node-twilio A Node-RED node to send SMS messages via the Twilio service. v0.1.0 3541 node |
| node-red-contrib-play-audio A node-red node for playing audio in the browser v2.3.2 3402 node | node-red-node-serialport Node-RED nodes to talk to serial ports v0.8.3 2304 5.0 (3) node | node-red-contrib-ibm-watson-iot Connect to IBM Watson Internet of Things Platform as a Device or Gateway v0.2.8 1979 node |
| node-red-contrib-cos Node-RED Nodes for IBM Cloud Object Storage service. v0.0.16 1950 node | node-red-node-google A set of Node-RED nodes to access various Google services v0.1.4 1894 node | node-red-node-ping A Node-RED node to ping a remote server, for use as a keep-alive check. v0.0.16 1658 node |

Figure 18 : Liste non exhaustive de nœuds Node-red sur le site flows

3.3 MQTT & Broker

MQTT [6] est l'abréviation de **Message Queuing Telemetry Transport**. C'est un protocole de transmission de données utilisé principalement dans le monde de l'internet des objets. Il est extrêmement léger et fonctionne avec un système de publication / souscription très simple. Il suffit d'utiliser un nœud pour publier un message (1) et remplir le champ « Topic » (2) avec le nom désiré.

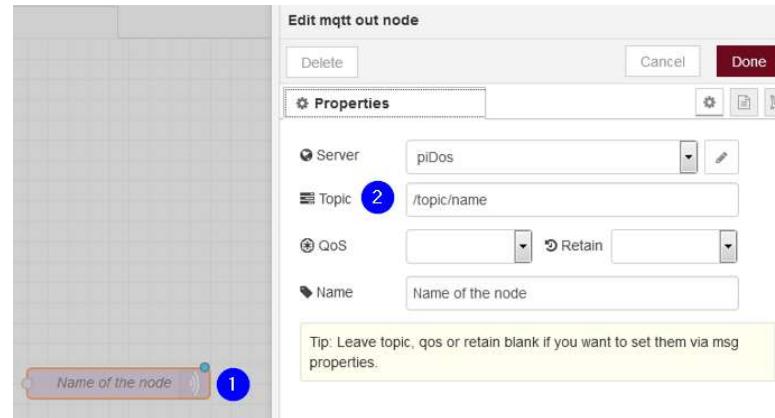


Figure 19 : Exemple de création d'une publication et définition du topic

La création d'une publication est relativement simple, il faut, cependant, utiliser un broker pour transmettre ce protocole. Eclipse Mosquitto [7] est un broker MQTT qui permet l'envoi des messages. Le broker est l'intermédiaire qui permet de gérer le modèle d'envoi des messages type *publication/subscription*. En effet, de lui-même Node-red ne peut pas envoyer de message MQTT, il a besoin d'utiliser le **broker MQTT** pour le faire. La case server (2) dans l'image ci-dessous, lors de l'ajout d'un nœud MQTT, permet de renseigner le broker à utiliser. Pour s'inscrire à un flux de données et recevoir les informations, il suffit d'utiliser le nom donné dans la case topic sur un nœud de réception (1).

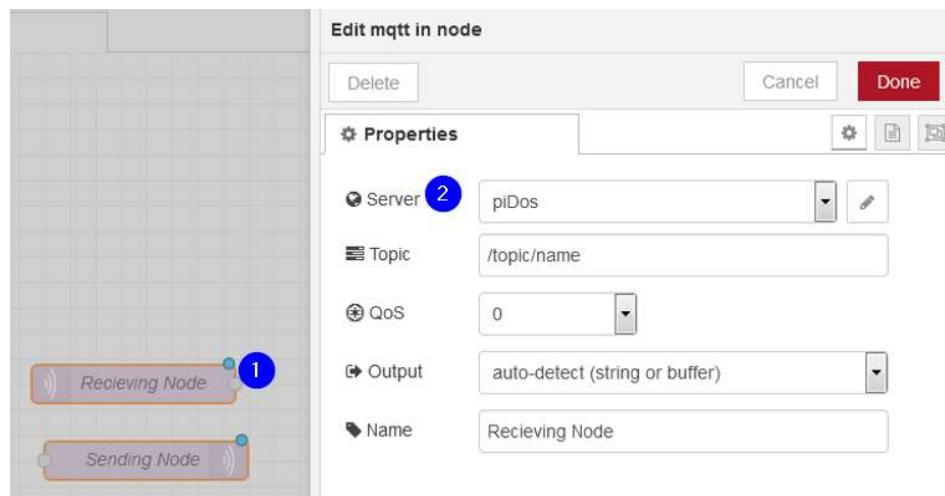


Figure 20 : Souscription au topic « /topic/name » permettant de recevoir les informations émises

C'est le server indiqué dans le champ « Server » (1) qui relayera les messages correctement entre l'émetteur du message et tous les récepteurs ayant souscrit au topic du même nom. Ainsi, lorsqu'une émission aura lieu sur « /topic/name », tous les inscrits recevront cette information. Il est, cependant, nécessaire de respecter exactement le même nom. En effet, « /topic/name » est différent de « topic/name » ou « /Topic/Name » et aucun d'entre eux ne communique avec les autres.

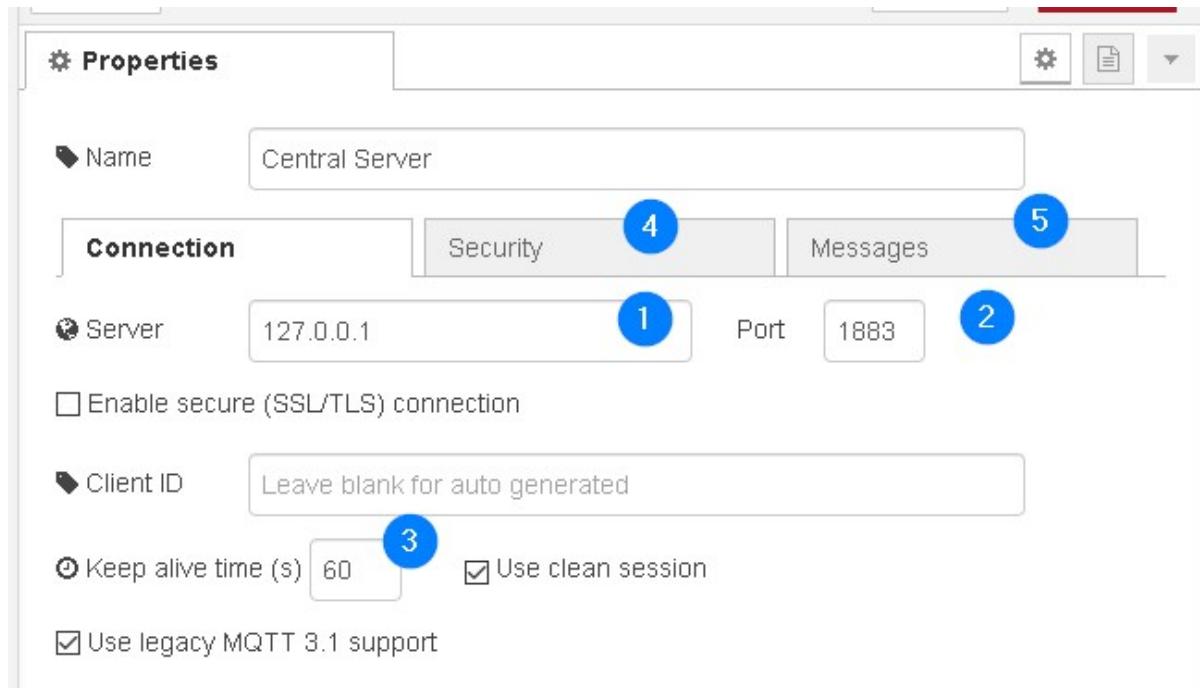


Figure 21: Configuration du server

Il est donc possible de définir les paramètres du server, comme le port d'écoute (2), le temps d'activité du nœud ou keep alive (3), ainsi que la sécurité (4) pour définir un mot de passe. Il est même possible d'envoyer un message à chaque connexion/déconnexion (5).

3.4 React-Native

React-Native [8, 9] est une technologie développée par Facebook. C'est un framework basé sur Node Js qui permet de développer des applications mobiles pour les plateformes comme Android ou iOs. Cette technologie se base elle-même sur une autre technologie pour le développement web : React [10].

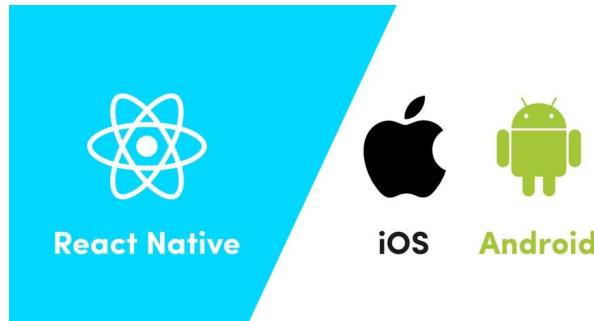


Figure 22 : react-native offre une solution pour développer sous iOS et Android

React, développé aussi par Facebook, est un framework Node Js permettant le développement d'application web intuitive et de manière rapide. Le développeur peut facilement communiquer avec un serveur pour récupérer des informations et les afficher dans une page web. La logique de programmation se base sur un concept de base appelé « composant ». Un composant devient alors une brique de l'application qui peut être appelée et réutilisée sur l'ensemble de l'application web. Un composant peut être quelque chose de complexe comme un formulaire, mais il est souvent plus simple d'utiliser des composants plus petits. Les développeurs React ont donc tendance à subdiviser leur formulaire, pour continuer avec cet exemple, en sous composants. Un bouton, un champ de texte, une image, sont quelques exemples de composants imaginables. Ces derniers sont alors rassemblés dans ce qui est couramment appelé un « container » et qui englobe tous les composants nécessaires au formulaire ainsi que sa logique. Par sa logique ont entend ici la validation des champs, son contrôle ou encore son envoi. Chaque composant possède un état ou « state » dans lequel il est possible de stocker des données comme une couleur, des variables, du texte ou d'autres format. En continuant avec l'exemple du formulaire, l'état (state) du composant formulaire peut stocker l'ensemble des erreurs faites par l'utilisateur, comme un mauvais email par exemple, et les renvoyer au moment de la validation du dit formulaire.

L'avantage de React est sa facilité de prise en main et le nombre important de librairies développées. Fort de son succès, c'est donc logiquement qu'une version de React pour mobile est apparue : « React-Native ».

React-Native possède l'avantage qu'il n'utilise pas, comme d'autres framework de développement multiplateforme, du langage html pour générer les vues. La puissance de React-Native lui permet de tourner en processus d'arrière-plan et d'envoyer des messages afin de manipuler des vues natives. Une vue native ou application native est une application qui est développée spécifiquement pour une plateforme. Par exemple : Android avec le langage JAVA, sur Android studio et iOs avec Objective-C, sur XCode. Contrairement à d'autres technologies multiplateformes, React-native n'est pas basé sur des vues html mais bien sur des vues natives, rendant l'application plus performante.

3.5 Firebase[11]

Firebase est un ensemble de services proposés par Google afin d'améliorer le développement d'application mobile/web. La gamme de services fournis permet aux développeurs de bénéficier d'hébergements pour des base de données, la gestion d'utilisateurs et des méthodes d'authentification (email, Facebook, Google, etc...). Ainsi qu'une communication en temps réel, en passant par des algorithmes gérant la scalabilité d'un projet en cas de fortes demandes ou de connexions importantes.

Tous ces services permettent de s'affranchir de la partie serveur backend et offrent la possibilité de se concentrer sur l'utilisateur et l'expérience qu'il aura avec l'application. De plus, la majeure partie de ces services est gratuite (jusqu'à un certain degré d'utilisation) ce qui permet d'utiliser cela dans notre partie prototypage sans se soucier des possibles frais.

Le service d'authentification sera utilisé pour gérer les connexions et les inscriptions des utilisateurs. Il permet, si l'on souhaite, d'activer l'authentification avec des services comme Google, Facebook, Twitter et d'autres, pour faciliter l'utilisateur.

Le service de base de données (Firestore[12–18]) sera utilisé pour sauver les utilisateurs, leur profil ainsi que leurs sessions. La base de données sera aussi utilisée pour l'enregistrement des différentes informations nécessaires au déroulement du jeu, comme par exemple les résultats attendus aux énigmes. Il s'agit d'une base de données type noSql. Par opposition à une base de données traditionnelle qui possède un schéma relationnel, les bases de données noSql sont dites sans schéma. Il existe évidemment des manières de structurer les données, mais cela ne correspond pas aux standards utilisés dans une base MySql par exemple. Firestore gère donc des « collections, des documents et des champs », s'imbriquant les uns dans les autres. Non liés à un schéma type, deux utilisateurs pourraient avoir des champs totalement différents en base de données. Dangereux en ce qui concerne l'implémentation, mais possible.

3.6 Redux[19]

Redux est une librairie Node.js qui facilite la gestion de l'état d'un composant. Redux offre une abstraction de cet état à un niveau supérieur pour l'ensemble de l'application. En effet, comme mentionné précédemment, React travaille avec des composants. Chaque composant possède son propre état, plus couramment appelé « state ». Si un composant enfant a besoin d'une information de son parent, il faut alors, dans le composant parent, passer cette information au composant enfant.

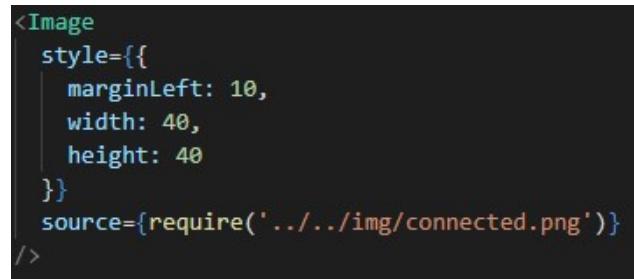


Figure 23: Exemple composant React avec passage d'information à l'enfant

Dans l'exemple ci-dessus le composant image donne à son enfant, les styles qu'il devra appliquer ainsi que la source de l'image à afficher. Chaque paramètre passé ainsi, s'appelle dans React une « prop » (diminutif de properties). Ainsi dans le composant enfant, on reçoit l'ensemble des « properties » dans une variable se nommant « props ». L'accès à la source de l'image, dans l'enfant, est alors possible en utilisant la variable « props.source ».

Redux généralise cette approche et permet alors d'avoir -en plus d'un état ou « state » par composant- un état *global* pour l'ensemble de l'application. Cela devient utile, dans le cas de la gestion d'un utilisateur ou lorsque le nombre de composant devient important.

Dans l'exemple de l'utilisateur, il est nécessaire pour l'application d'avoir sur chaque page ou vue, l'état et les informations concernant cet utilisateur. Cela permet de savoir si l'utilisateur est connecté et s'il possède les droits d'accès à certaines pages ou non. On souhaite aussi que lorsqu'un composant modifie cet utilisateur, ce changement se répercute vers les autres composants qui utilisent les données de l'utilisateur. Sans Redux, il faudrait passer à chaque sous composant l'utilisateur présent dans l'état du parent à travers les « props » et faire attention de travailler toujours sur un utilisateur à jour. Cela est possible, mais demande une grande attention et beaucoup de travail, car il faudrait passer l'utilisateur jusqu'au dernier composant de la chaîne.

Redux simplifie cette gestion et permet d'avoir un état/state d'application et de récupérer ce dernier dans n'importe quel composant React-native. Il possède, en plus, l'avantage de pas uniquement gérer l'état/state de l'application. Redux est capable de centraliser un ensemble de fonction qui sont alors accessibles depuis n'importe où dans l'application. Dans l'exemple de l'utilisateur le développeur pourrait avoir besoin de son nom pour l'afficher dans son profil, de son identifiant unique pour savoir s'il possède l'accès à une page ou encore de devoir accéder à la fonction de déconnexion, car le joueur peut quitter l'application depuis n'importe quel page ou vue.

Grâce à Redux, je peux accéder à ces informations dans n'importe quel composant. Il suffit juste de connecter mon composant à l'état Redux appelé « store » et récupérer ce qui m'intéresse.

La Figure 24 montre la manière de connecter un composant au store Redux. Ici, il s'agit du profil de l'utilisateur. Plusieurs variables ont été récupérées, comme « me » ou « avatar », qui représentent les informations du profil utilisateur ainsi que son image de profil. Mais aussi des fonctions comme « logout » ou « updateAvatar » qui permettent à l'utilisateur de se déconnecter ou de modifier son image de profil.

Le composant se connecte grâce à la méthode `connect()`. Cette méthode reçoit en paramètres, les variables `mapStateToProps` et `mapDispatchToProps`. De manière plutôt explicite, `mapStateToProps` permet de récupérer les attributs dans l'état/state global que nous souhaitons et `mapActionToProps` permet lui de récupérer les fonctions utiles au composant.

```
const mapStateToProps = state => ({
  me: state.auth.me,
  avatar: state.auth.avatar,
  geofencesLoaded: state.game.geofencesLoaded,
  //mqtt state
  mqttConnected: state.mqtt.mqttConnected,
  broker: state.mqtt.broker,
});

const mapDispatchToProps = {
  logout: logout,
  showToast: showToast,
  updateAvatar: updateAvatar,
  getGeofencesFromDb: getGeofencesFromDb,
  //mqtt reducer
  onMqttConnected: onMqttConnected,
  onMqttMessage: onMqttMessage,
  registerMqttClient: registerMqttClient
};

export default connect(
  mapStateToProps,
  mapDispatchToProps
)(Profile);
```

Figure 24: Connexion d'un composant au store Redux

De manière simple, on récupère, dans le code, le user complet en utilisant la variable `props.me`, son image avec `props.avatar`. Si l'on souhaite faire appel dans notre composant à la fonction `logout()`, on utilisera `props.logout()` au moment de la déconnexion du joueur.

4 DEVELOPPEMENT DU PROTOTYPE

4.1 Introduction

Le lecteur est invité à utiliser la carte présente dans les annexes au chapitre 9.8 afin d'avoir une meilleure vision du jeu.

Pour la phase de développement, le projet a été séparé en deux parties. Premièrement, le développement de l'application mobile et deuxièmement, la création et la réalisation des bornes interactives. Ces deux parties sont liées par le gameplay qui est constitué de la logique du jeu et des énigmes à résoudre. Une partie plus spécifique est consacrée au gameplay dans le chapitre 4.6.

La réalisation du prototype a demandé quelques journées de recherche sur les technologies et les capteurs possibles à utiliser durant le projet. Ces recherches ont permis de gagner du temps lors de la phase de développement. Certaines technologies ont été testées avant, mais pas retenues, non pas car elles n'étaient pas performantes, mais plutôt car elles étaient inadaptées au concept imaginé.

En effet, des technologies pour capter les mouvements ou détecter une présence ont montré des failles dans la détection des objets. De plus, détecter le passage d'une personne dans un environnement ou d'autres personnes transitent, n'est pas idéal. En effet, il n'y a aucun moyen de vérifier, s'il s'agit bien du joueur ou d'un chien, d'un oiseau ou d'un autre passant. De ce fait, les énigmes ont été pensées et réalisées avec des technologies et des capteurs offrant une réelle interaction avec le joueur, comme des boutons ou encore un clavier tactile.

Sans avoir participé au jeu depuis le début, il est difficile de deviner ce qu'il faut faire avec les bornes mises en place. Il est donc indispensable de commencer le jeu du début, de prendre le temps pour résoudre les énigmes et d'écouter correctement la voix-off qui guide l'utilisateur durant la quête. Le concept imaginé (Cf. Chapitre 1.3) était réfléchi pour limiter le temps de résolution total ainsi que le temps des énigmes. Après discussion, cette contrainte c'est avérée plus limitante qu'utile. Le prototype a pour but de montrer le potentiel de l'application. Il est donc souhaitable qu'il convienne et puisse être testé par un maximum de personnes dans le cas d'une potentielle suite de production.

De plus, qu'est-ce qu'apporte réellement la contrainte de temps ? Elle permet, certes, à une personne de se dépasser physiquement et de tenter de compléter le jeu rapidement. Mais elle peut aussi, au contraire, décourager des gens qui sont peu/moins sportifs. Attirer des gens peu sportifs et les faire apprécier l'effort physique par le jeu est la base de la gamification. Un sportif quotidien n'aura pas de mal à trouver la motivation d'aller faire du sport. Le jeu est donc pensé pour motiver des gens à progresser et non pas les décourager. Dans cette optique, l'idée de base d'un temps imparti a été mise de côté. Cependant, rien n'empêche d'imaginer dans des niveaux de difficultés supérieurs d'intégrer cette contrainte en sus ou d'annoncer au joueur qu'il s'agit d'une escape room extérieur et qu'il dispose, comme dans une partie traditionnelle, d'une heure pour résoudre la quête en totalité.

4.2 Application Mobile

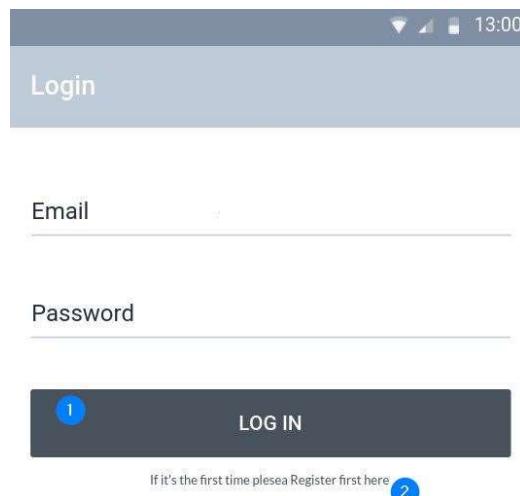
Pour ce chapitre les termes « vues » ou « pages » sont utilisés et font références à chaque écran. Par exemple : La page ou vue « login » correspond à l'écran de connexion. (Cf. Figure 25), alors que la Figure 26, elle, représente la vue « inscription » ou « registrer ». Le glossaire au chapitre 7 donne une explication des termes techniques en cas de besoin.

4.2.1 Concept

L'application mobile joue un rôle central dans la réalisation du prototype. Au départ, il était prévu de réutiliser une application existante utilisant déjà les mécanismes de calcul de distance, vitesse, etc... Il est très rapidement apparu que l'application allait subir des modifications et de ce fait devoir être codée à nouveau. Cela n'était pas défini, au commencement, dans le cahier des charges, mais sans cette application le concept du jeu et de la gamification perd tout son sens. De plus, le prototype à réaliser dépend lui de cette application, il est donc indispensable de repartir de zéro pour obtenir quelque chose de concret et de développer cette application.

Pour terminer le prototype, l'application devra être fonctionnelle, mais le design devra être mis quelque peu de côté afin de pouvoir respecter les délais tout en concentrant les efforts sur les fonctionnalités pour fournir au joueur la meilleure expérience de jeu possible. Tout le développement se fera en React-native afin d'obtenir la meilleure compatibilité avec Node-red et possiblement faciliter le développement d'une application pour iPhone, en cas de prolongation du projet.

4.2.2 WireFrame[20]



La création d'une maquette offre une dimension plus réelle au projet et permet d'imaginer le fonctionnement et les interactions entre l'utilisateur et le téléphone. Il s'agit bien sûr d'une maquette et non d'un design final, même si ce dernier sera fortement inspiré, certains changements peuvent apparaître.

Premier écran de connexion. L'utilisateur peut se connecter (1) et arriver sur son profil ou dans le cas d'une première visite, il peut aussi s'enregistrer (2). Il arrive alors sur l'écran ci-dessous (Cf. Figure 26) lui permettant de renseigner les informations nécessaires pour son profil.

Figure 25 : Écran de connexion

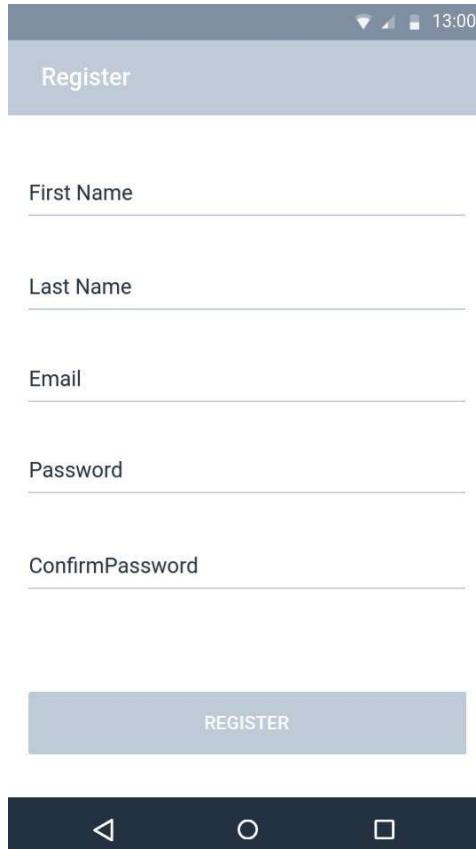


Figure 26 : Écran d'inscription



Figure 27: Vue du profil utilisateur

Le formulaire d'inscription est basique. Il demande à l'utilisateur, nom, prénom, email et mot de passe qu'il doit vérifier.

Une fois la connexion/inscription réussie, le joueur arrive sur la page de son profil. Cette vue « profil », donne l'historique des sessions effectuées. Ces informations seront sauvegardées dans une base de données grâce au service Firebase (vu précédemment) qui permettra de gérer, en plus des informations du profil, les informations du jeu, comme le nombre de points, les diverses sessions, etc...

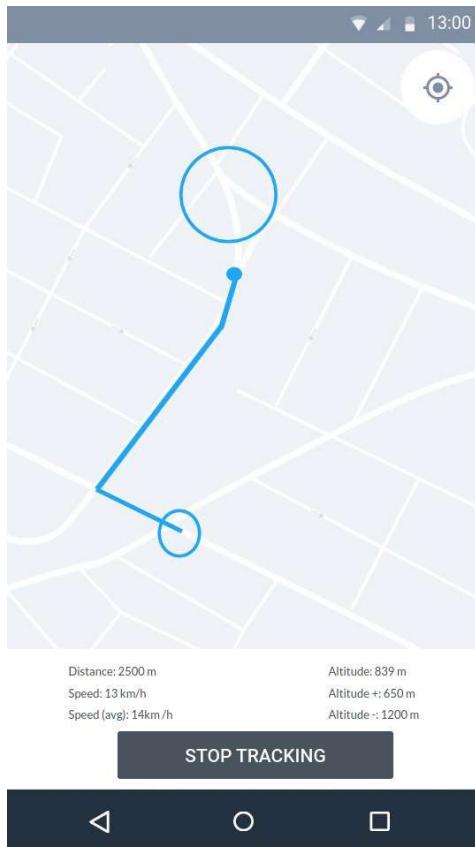


Figure 28: Ecran principal du jeu



Figure 29: Affichage du résumé d'une session

Ce bouton lance la fenêtre qui sera considérée comme la fenêtre principale du jeu (Cf. Figure 28). Cette vue offre au joueur plusieurs informations en temps réel, comme la distance qu'il a parcourue, sa vitesse, son altitude ou encore la dénivelée positive ou négative. D'autres éléments pourront bien sûr s'ajouter selon les besoins du développement.

Sur le reste de la vue est affiché une carte qui localise le joueur et suit ses déplacements. Un tracé bleu permet de voir le chemin parcouru. Le joueur reçoit les informations par audio.

Un système d'indice sera mis en place. En effet, il faut prévoir diverses éventualités. Le joueur peut ne pas être attentif ou ne pas avoir bien compris/entendu les instructions. Pour cela, des indices seront disponibles tout au long du parcours selon l'avancée du joueur. Sur cette vue se trouve aussi les informations concernant les zones (geofences) où sont situées les énigmes.

Le joueur sera donc guidé par une voix-off et devra se rendre dans les zones délimitées sur la carte. L'ensemble des zones n'est pas affiché dès le départ, mais celles-ci se dévoilent au fur et à mesure. En entrant dans une zone, le joueur déclenche une série d'événements qui le feront progresser dans le jeu. Par exemple, entrer dans la zone de départ permet d'avoir certaines informations audios sur le jeu et déverrouille la zone de la première énigme qui s'affiche alors sur la carte. Toute la partie concernant le déroulement du jeu est expliquée plus en détails dans la suite du rapport.

L'application offre la possibilité au joueur de voir ou plutôt revoir ses sessions précédentes. L'application peut donc aussi être utilisée, si on le souhaite, simplement pour courir, faire du vélo et enregistrer ses performances. Elle agira non plus comme une application de jeu, mais plutôt comme une application sportive de suivi de parcours et de performance.

Cette vue du résumé très simple donne les informations récupérées de la session et l'affichage d'une carte avec le parcours effectué ce jour-là. Le design est basique et ne sera pas travaillé plus que nécessaire étant donné qu'il s'agit d'un simple affichage d'information n'apportant pas réellement de plus au projet.

4.2.3 Base de données

Utilisant déjà les services Firebase de Google pour la gestion et l'authentification des utilisateurs, il a été choisi de continuer dans cette direction et de travailler avec Firestore, un autre service de Google. Ce service fait partie de la suite Firebase et permet la gestion d'une base de données. Ce choix a aussi été motivé par la haute disponibilité de la base de données.

Lorsqu'on pense à une base de données, les premières qui nous viennent à l'esprit sont les bases de données relationnelles comme MySql. Dans ce projet la base de données est différente. Premièrement, car Google n'offre pas de possibilités, dans la suite Firebase, de stocker une base de données relationnelle et deuxièmement, car le base de données NoSql[21] permet une gestion facilitée avec React-Native et Node JS. De plus, le base de données NoSql ne demande pas l'écriture de requête comme le langage SQL. De plus, en cas de forte demande, les services de googles sont scalables.[22]

Pour ces aspects-là, ainsi que pour la grande disponibilité des services Firebase de Google, la base de données est de type NoSql. Firestore possède sa propre interface. Pas de tables comme avec MySql/SQL mais des collections, des documents et des champs.

Le service permet donc la gestion des données grâce à trois types de structure.

- 1) Les collections : Une collection est un ensemble de documents. On peut voir une *collection* comme le dossier parent dans une arborescence de fichier ou aussi comme une catégorie. Ici au point 1, une *collection* pour le jeu (game) et une pour les utilisateurs (users).
- 2) Les documents : Une collection comprend un ensemble de documents. A noter, chaque document peut-être totalement différent. Une *collection* n'est donc pas un ensemble de *documents* identiques, même si cela est fortement conseillé. L'exemple de la catégorie illustre bien cela. En effet, une *collection* nommée « animaux » pourrait contenir des reptiles, des mammifères, etc... chacun avec des caractéristiques différentes.
- 3) Les champs : Un *document* contient un ensemble de *champs*. Le point 3 montre les données sauveées pour un utilisateur. Dans ces *champs* il est possible d'ajouter ce que l'on souhaite. Un *document* peut aussi contenir une nouvelle *collection*.

The screenshot shows the Firestore interface with three panels:

- Left Panel:** Shows a collection named "users" with a single document named "XCFzM13CeeURTGxdA0cvilh9cNP2".
- Middle Panel:** Shows the fields of the "XCFzM13CeeURTGxdA0cvilh9cNP2" document. It includes:
 - A "Commencer une collection" button (1).
 - An "Ajouter un document" button (2).
 - A "Commencer une collection" button (3).
 - A "Ajouter un champ" button.
 - Fields: displayName: "Thomas Léchaire", email: "thomas@gmail.com", level: 1, photoURL: "https://upload.wikimedia.org/wikipedia/commons/thumb/7/79/262-man-running-1.svg/512px-262-man-running-1.svg.png", sessions: [{"altitudeDown: 0, altitudeUp: 0}], totalDistance: 1000, totalPoint: 1000000, uid: "XCFzM13CeeURTGxdA0cvilh9cNP2".
- Right Panel:** Shows the full document structure with the same fields and values as the middle panel.

Figure 30: Interface Firestore pour la gestion de la base de données

Afin d'avoir quand même une représentation schématique, un diagramme de la base de données de notre jeu est proposé ici et se présente comme suit :

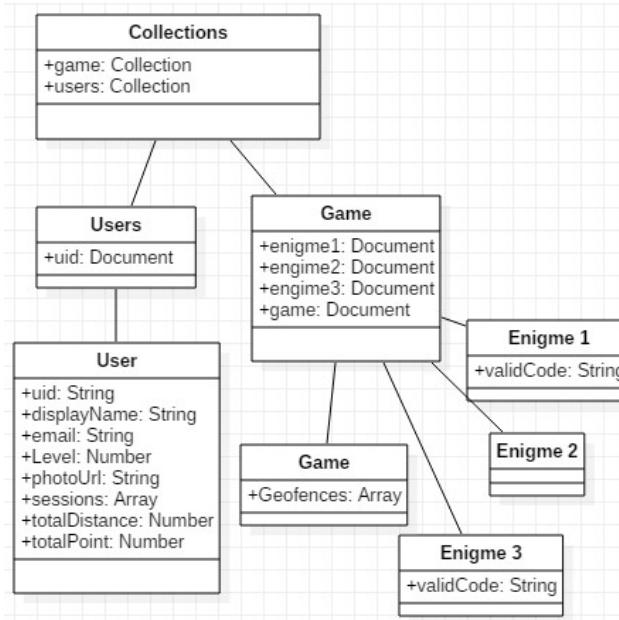


Figure 31: Schéma NoSQL de la base de données Firestore

Il ne s'agit bien évidemment pas d'un schéma relationnel, mais d'un schéma explicatif. Deux *collections* qui sont, « game » et « users », et qui contiennent un ensemble de *document*.

La *collection* « users » contient des *documents* qui sont identifiés par l'UID (1 et 3). L'UID du joueur représente son identifiant unique dans le jeu. D'autres *champs* (2) comme le nom, l'email, la photo du joueur, mais surtout un tableau de l'ensemble des sessions faites par le joueur. Toutes les informations utiles au profil sont sauveées dans un « user » (utilisateur). (Cf. voir Figure 31).

| users | |
|----------|---|
| 1 | kFwVOFzLgBMr4rjK7n4CL3EtAw73 |
| 2 | displayName: "Jean Michel" email: "jm@gmail.com" level: 1 photoURL: "http://eref-qrga.org/wp-content/grey.jpg" |
| 3 | sessions totalDistance: 0 totalPoint: 0 uid: "kFwVOFzLgBMr4rjK7n4CL3EtAw73" |

Figure 32: Document « users » et champs d'un utilisateur

Il faut, cependant, s'attarder sur le *champ* « sessions » qui est un tableau composé de l'ensemble des sessions. Chaque session est composée des champs suivants :

The screenshot shows a MongoDB document structure. The document contains the following fields:

- `altitudeDown: 0`
- `altitudeUp: 0`
- `avgSpeed: 86.2125015258789`
- `date: 10 septembre 2019 à 14:08:31 UTC+2`
- `distance: 354.81852133237885`
- `key: "2"`
- `routeCoordinates` (highlighted with a blue circle labeled 1)
 - `0 {latitude: 46.7805983, lon...}`
 - `1 {latitude: 46.7805983, lon...}`
 - `2 {latitude: 46.7805983, lon...}`
 - `3 {latitude: 46.7828983, lon...}`
- `totalPoint: 317`
- `totalTime: "00:37"`

To the right of the document, a list of items is provided:

- L'altitude positive
- L'altitude négative
- La vitesse moyenne
- La date et heure
- La distance parcourue
- Le champ `key`, utile, pour l'affichage dans l'application
- Un tableau `routeCoordinates` de longitude, latitude, permettant l'affichage sur la carte du chemin parcouru par l'utilisateur
- Le total de point pour la session
- Le temps total pour la session

A note on the right side states: "Le tableau `routeCoordinates` est un tableau qui stocke, après chaque déplacement du joueur, la nouvelle position du GPS sous forme d'une latitude et d'une longitude. C'est ce tableau qui permet de dessiner le tracé sur la carte."

Figure 33: Contenu d'une session

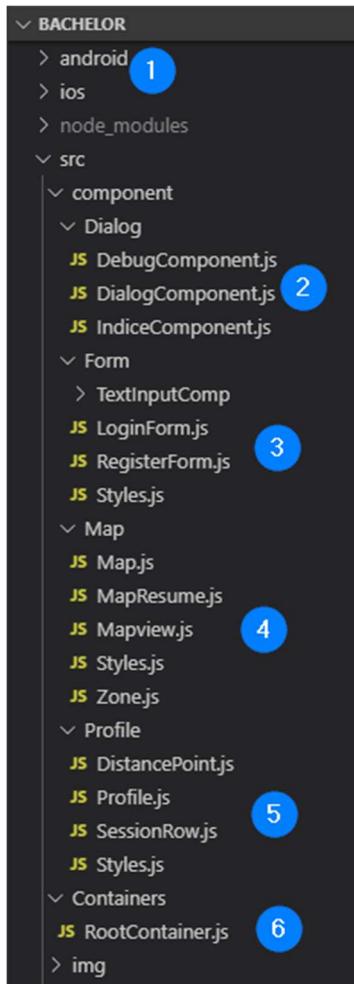
Le *collection* « game », contient quatre documents, un pour chaque énigme et un pour le jeu (game). Il est important de faire la différence entre la *collection* « game » et le *document* « game ». La Figure 31 permet de mieux se représenter la différence. Le *document* contient un tableau de geofence. C'est ce tableau qui est utilisé pour afficher les geofences, leur position ainsi que leur taille dans l'application. Une geofence (2) possède des attributs comme un id, une latitude et une longitude définissant le centre avec un radius. Comme une geofence définit une zone sur la carte, une variable `firstTimeIn` permet de savoir si le joueur est déjà entré ou non dans la geofence en question.

The screenshot shows a MongoDB document structure. The document contains the following fields:

- `+ Ajouter un document` (highlighted with a blue circle labeled 1)
- `enigme1`
- `enigme2`
- `enigme3`
- `game`
- `+ Commencer une collection`
- `+ Ajouter un champ`
- `geofences` (highlighted with a blue circle labeled 2)
 - `0`
 - `firstTimeIn: false`
 - `id: "depart"`
 - `isVisible: true`
 - `latitude: 46.782878`
 - `longitude: 6.653347`
 - `radius: 7`
 - `validate: false`
 - `1 {firstTimeIn: false, id: "..."}`
 - `2 {firstTimeIn: false, id: "..."}`

Figure 34: Le document « game » et ses champs

4.2.4 Développement

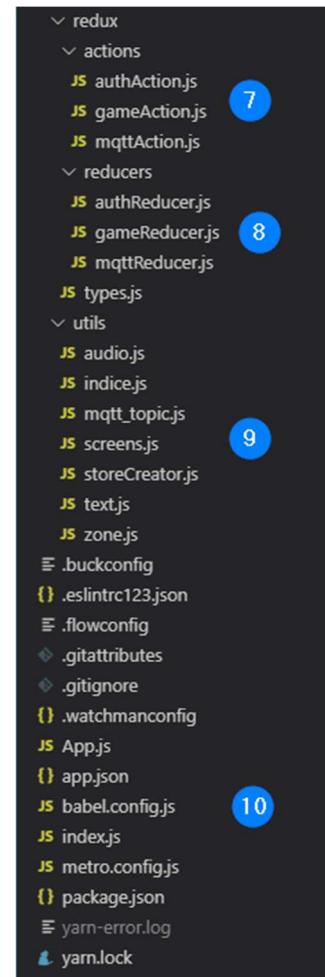


Le développement de l'application en React-native offre l'avantage d'un développement rapide sur Android, mais aussi en cas de besoin, d'avoir la possibilité de réutiliser le même code pour fournir une version de l'application sous iOS.

React-native se base sur Node Js et de ce fait permet l'importation de librairies. Chaque librairie s'importe grâce à la commande '*npm install nom_de_la_librarie*'. Cela facilite l'intégration de nouvelles fonctionnalités développées par d'autres. En revanche, cela ouvre potentiellement des failles de sécurité, car pas toutes les librairies ne sont pas codées par des professionnels.

L'arborescence de notre projet React-native se présente comme suit :

- 1) **Android, ios et node_modules** : Ces trois dossiers sont gérés par react-native et node js: presque aucunes interventions n'est nécessaire. Android et ios contiennent les sources pour la compilation des applications mobiles. Node_modules contient lui les sources des toutes les librairies installées.
- 2) **Dialog** : Le dossier Dialog est contenu dans les sources de notre projet. C'est un dossier contenant -comme vu précédemment- des composants. Ces trois composants sont utilisés pour afficher les indices dans le jeu et à des fins de contrôle durant les tests.
- 3) **Form** : Contient un composant se nommant *TextInputComp*. C'est le composant, champ texte, utilisé dans le formulaire d'inscription par exemple. Puis, nous avons deux « container ». Un pour le formulaire de login (*LoginForm*) et un pour celui d'inscription (*RegisterForm*). Le dernier fichier est un fichier de style.



- 4) **Map** : Ce dossier contient tous les composants avec une carte ou en rapport avec la carte. Comme la carte pour le jeu (*Map & MapView*) et la carte pour le résumé de session (*MapResume*).
- 5) **Profile** : Contient les composants nécessaires à l'affichage du profile, dont *SessionRow* qui est utilisé pour afficher la liste des sessions.
- 6) **Containers** : Contient un fichier *RootContainer* qui est utile pour gérer la navigation entre les pages/vues.
- 7) **Redux / actions** : Dans ce dossier se trouvent toutes les actions sous forme de méthodes/fonctions qui sont utilisées dans le jeu. Trois fichiers permettent de distinguer à quel genre de fonction nous avons à faire. Des fonctions pour l'authentification, pour le jeu ou pour la communication mqtt avec le server central.
- 8) **Redux / reducers** : Ce dossier contient les reducers des trois fichiers vus au point 7. Lorsque l'action est appelée, elle va envoyer des informations au reducer du même type. Le reducer s'occupe alors de modifier l'état/state général de l'application en conséquence.
- 9) **Utils** : Dossier contenant l'ensemble des utilitaires. Comme le nom de vues, des audios, l'ensemble des textes ou encore le nom des topics pour la communication vers le server de jeu.
- 10) **Dossier racine** : Il contient les fichiers créés par default lors de l'initialisation d'un projet React-native.

Pour gérer les sessions utilisateurs, l'application utilise Firebase, comme mentionné précédemment. En utilisant la vue « login » (Cf. Figure 36), il est possible de se connecter à l'application et d'arriver sur son profil.

```
componentWillMount() {
  firebase.auth().onAuthStateChanged(user => {
    let screen = screens.login;
    if (user) {
      screen = screens.profile;
      this.props.getUserFromDb(user);
    } else {
      this.props.logout_success();
    }
    this.props.navigation.navigate(screen);
  });
}
```

Figure 35: Méthode vérifier le statut du joueur

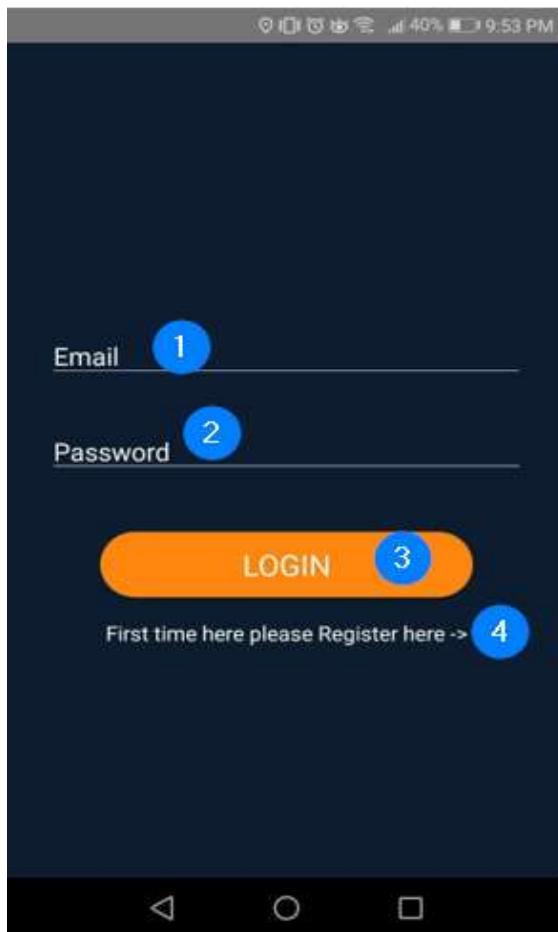


Figure 36 : Vue login dans l'application

Toute cette partie de gestion des utilisateurs s'articule avec les services de Google. Pour créer le lien vers Firebase, il faut un compte Google, puis se connecter à la console Firebase et créer un nouveau projet. Une fois le projet démarré il suffit d'activer les services que l'on souhaite utiliser et de télécharger le fichier au format json nécessaire pour autoriser l'application (mobile ou web) à communiquer avec Firebase. Ce lien avec le service de Google est aussi utilisé lorsque l'on souhaite accéder à la base de données Firestore. Pour plus d'information, un guide est disponible sur le site web hébergeant les sources du projet <https://github.com/thomaslc66/Bachelor-2019>. Ce guide permet la réinstallation et la mise en service de l'application dans le cadre d'un suivi ou d'un nouveau projet.

Dans React-Native on peut alors utiliser la librairie React-native-firebase[23] et utiliser les méthodes d'accès à notre disposition. La documentation de la librairie[24] est très complète et il existe un grand nombre de tutorial sur le sujet que l'on peut facilement trouver en ligne.[25]

Cette librairie offre une méthode `onAuthStateChanged` (Cf. Figure 35) qui nous permet de connaître et surtout d'obtenir les modifications sur l'état courant de l'utilisateur (connexion, déconnexion). La Figure 35 montre cette méthode qui permet, si l'utilisateur est connecté, de le rediriger vers sa page de profil, tout en chargeant les données le concernant.[26] Lorsque ce dernier se déconnecte ou n'est pas connecté, cette même fonction est appelée et redirige l'utilisateur vers la page de login.

Sur la vue « login », deux champs pour la connexion sont disponibles. Le champ email (1) et mot de passe (2) qui permettent à l'utilisateur de se connecter. La vérification et l'autorisation d'accès se fait grâce au bouton pour valider le login (3), qui gère aussi la vérification en cas d'erreur dans le formulaire.

Si l'utilisateur n'est pas enregistré, il lui faut d'abord s'inscrire. Pour ce faire, il doit cliquer sur le texte en dessous du bouton « Login » (Cf. Figure 36 point 4).

Il arrive alors sur la vue « Register » ou « inscription » qui lui permet d'entrer ses informations et de s'enregistrer. Pour rendre l'enregistrement plus fluide, il a été nécessaire d'écrire une fonction gérant le défilement jusqu'à au champ sélectionné, le tout de manière automatique.

La fonction `_scrollToInput(reactNode)` prend en paramètre un élément du formulaire, nommé ici `reactNode` et va faire défiler la page jusqu'à cet élément.

```
    _scrollToInput(reactNode) {  
      this.scroll.props.scrollToFocusedInput(reactNode);  
    }
```

Figure 37: Méthode gérant la fluidité du formulaire

Sur l'image ci-dessous, on visualise mieux le problème[27, 28] qui apparaît au moment où le clavier du téléphone vient masquer une partie de l'écran.

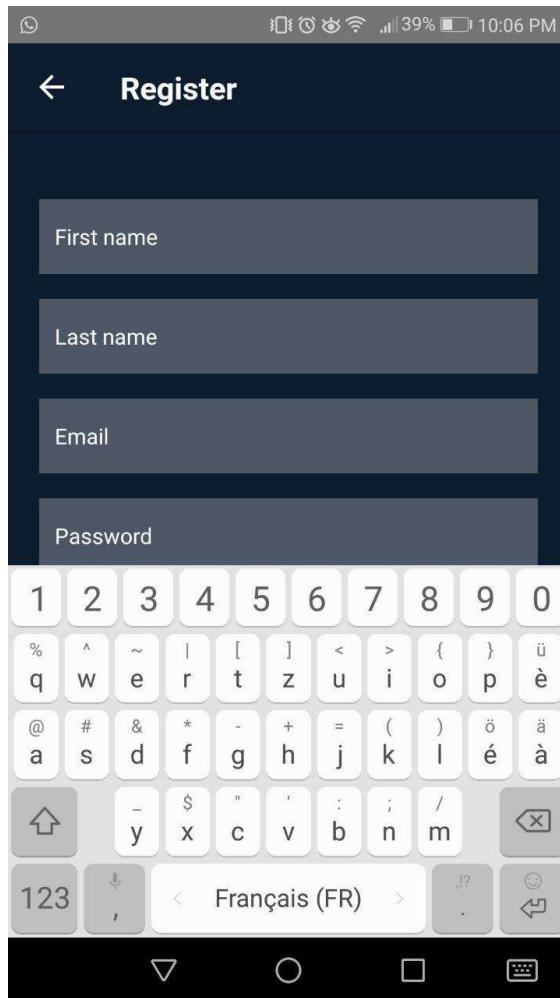


Figure 38: Clavier masquant les champs

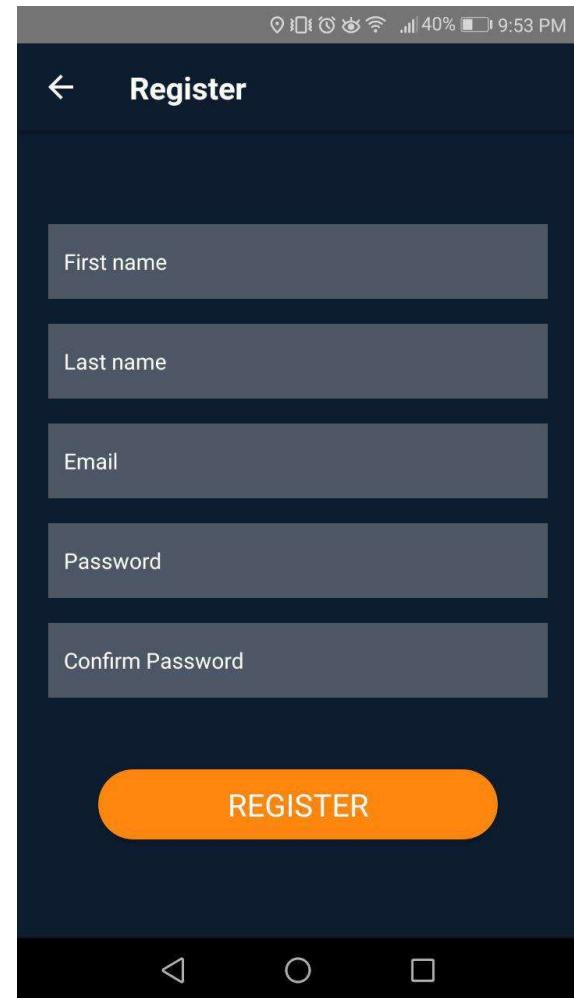


Figure 39: Ecran d'inscription dans l'application

La validation du formulaire est gérée principalement par Firebase, sauf la vérification des deux mots de passe afin qu'ils soient identiques et la vérification que l'ensemble du formulaire soit rempli. Ces derniers champs ont été vérifiés avant l'envoi du formulaire. Si l'ensemble des champs est correctement rempli, l'utilisateur est alors créé dans la base de données et l'application le redirige vers sa page de profil. Le profil étant nouveau, le joueur ne possède pas d'historique de session, ni de points ou distance parcourue.

```
me: {  
  uid: '-1',  
  level: 1,  
  sessions: [],  
  displayName: 'No Name',  
  totalPoint: 0,  
  totalDistance: 0,  
  email: 'Add an email',  
  photoURL:  
    | //Free image from pixabay.com  
    | 'https://cdn.pixabay.com/photo/2016/08/08/09/17/avatar-1577909\_960\_720.png'  
},
```

Figure 40: Attributs du profil du joueur.

En utilisant Redux, l'état/state global de l'application stocke un utilisateur par défaut. Puis, lorsque l'utilisateur valide le formulaire, l'état/state est modifié avec les informations entrées dans le formulaire (nom, prénom, email, etc...), l'utilisateur par défaut est mis à jour puis sauvegardé en base de données. Lors des prochaines connexions, c'est le service Firebase qui vérifie au login si le profil existe ou non et qui nous retourne l'information lors du login de l'utilisateur.

La connexion ainsi qu'une inscription validée ont pour effet de modifier l'état de l'utilisateur de « déconnecté » à « connecté ». Cela déclenche la fonction vue précédemment `onAuthStateChanged` (Cf. Figure 35) et redirige l'utilisateur vers son profil.

Sur cette page/vue du profil, plusieurs informations sont disponibles.

- 1) L'état du serveur de jeu.
- 2) Un lien pour se déconnecter
- 3) L'image du joueur, son nombre de points et de kilomètres au total.
- 4) Sa liste résumant les sessions qu'il a déjà faites.[29–31]
- 5) Un bouton pour démarrer le jeu.

Il peut alors se déconnecter, changer l'image de son profil, visualiser une ancienne session ou jouer une partie.

React-native propose un outil de navigation[32] de base dans sa librairie. Ce système permet de passer d'une page ou vue à l'autre facilement. C'est ce qui est utilisé pour changer d'écran. Il est aussi possible de gérer le comportement lors de l'appui sur le bouton retour[33, 34] en bas de l'écran. C'est le cas, par exemple, lorsque le jeu a commencé : on ne souhaite pas que l'utilisateur puisse quitter la partie sans confirmation de sa part qu'il s'agit d'un geste voulu. Lorsque le joueur presse le bouton retour du téléphone, la boîte de dialogue de la Figure 43 s'affiche lui demandant s'il souhaite réellement quitter le jeu ou non.

Lorsqu'un joueur est connecté et qu'il et qu'il appuie sur le bouton jouer (5), l'écran de jeu (Cf. Figure 42) s'affiche et la partie débute.

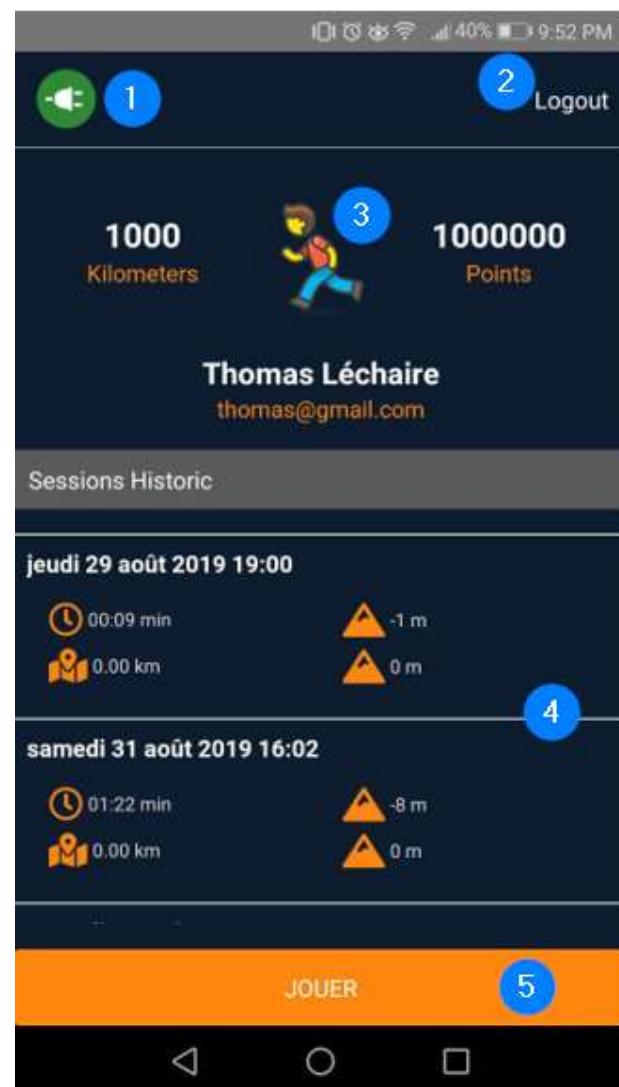


Figure 41: Vue du profil utilisateur

La partie débute et le jeu se présente ainsi :

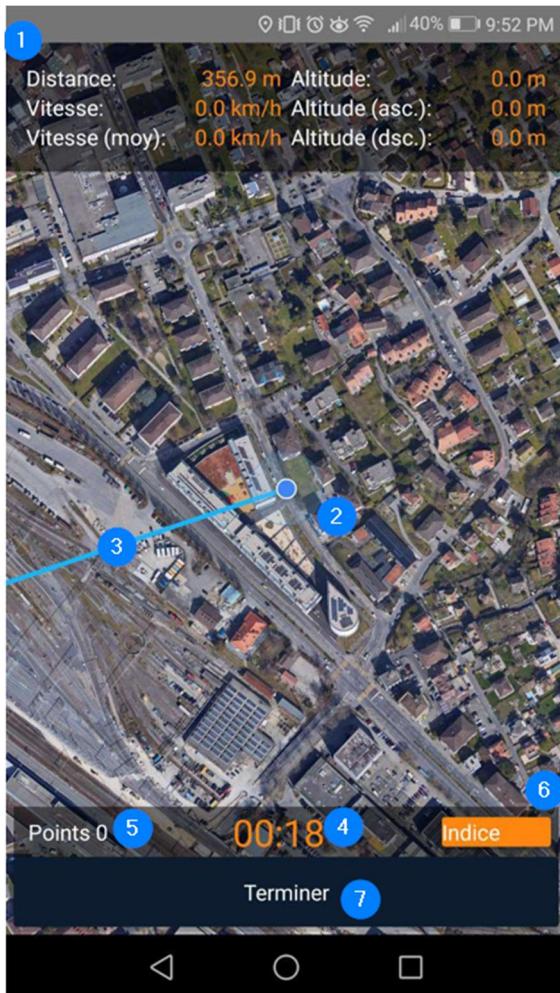


Figure 42: Ecran de jeu

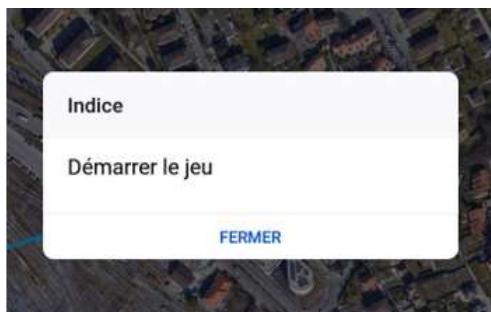


Figure 44: Exemple d'un indice

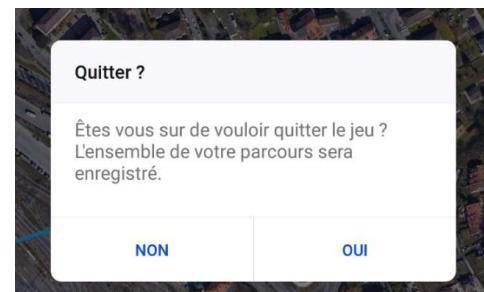


Figure 43: Boîte de dialogue de fin de jeu

La programmation du jeu se base sur les déplacements du joueur qui est localisé sur la carte par un cercle bleu (1). La librairie React-native-location offre une méthode d'écoute des déplacements en temps réel. Cette fonction, écoute et reçoit les mises à jour de position initiée par le GPS du téléphone puis, déclenche simultanément la mise à jour des informations pour le jeu (1) ainsi que la boucle du jeu (Cf. Figure 45). A chaque nouvelle réception de position cette dernière est enregistrée dans le tableau `routeCoordinates` vu au chapitre 4.2.3 Base de données et permet l'affichage du tracé bleu représentant le parcours du joueur.

Lors d'un déplacement, la boucle du jeu va vérifier si la nouvelle position reçue se trouve dans une des zones définies. Si elle détecte que la nouvelle position se trouve bien à l'intérieur, cela veut dire que notre joueur est entré dans une des zones (geofences). Une série d'actions est alors déclenchée en commençant par la lecture du fichier audio et la mise à jour de l'indice (ligne 93,118,123) en rapport avec cette zone. Puis, différentes actions, comme la validation de l'éénigme si le code entré est correct (ligne 109), puis l'activation de la zone (geofence) suivante (ligne 89, 112) ou l'attente du résultat s'il s'agit d'une éénigme (comme pour l'éénigme n°1 ou n°3).

```
82  export function gameFlow({ id, secondTimeIn, firstTimeIn }) {
83    return (dispatch, getState) => {
84      console.log('show id', id);
85
86      switch (id) {
87        case zone.depart.id:
88          // depart zone is reached so we activate the next zone
89          dispatch(activateEnigme1());
90          break;
91        case zone.enigme1.id:
92          // enigme 1 zone is reached we play audio
93          dispatch(updateIndiceAndPlayAudio(zone.enigme1.enter));
94          //then we wait for the user to enter the right color code
95          break;
96        case zone.enigme1.reset:
97          // When user is in the reset zone -> unlock the enigme 1 box
98          console.log('show zone', zone.enigme1.reset);
99          dispatch(unlockEnigme1());
100
101        break;
102        case zone.enigme2.id:
103          dispatch(setIndice(zone.enigme2.id));
104          // Check if the user enter for the second time.
105          if (!firstTimeIn && !secondTimeIn) {
106            //second time in, set it to true
107            dispatch(updateSecondTimeIn(id, true));
108            //play second audio and activate last enigme
109            dispatch(validateEnigme(zone.enigme2.id));
110            dispatch(activateEnigme3());
111          } else {
112            dispatch(activateNoisetierZone());
113          }
114        break;
115        case zone.enigme2.noisetier.id:
116          // When user is in the Noisetier zone
117          //PLAY AUDIO ENTER THE NOISETIER ZONE
118          dispatch(updateIndiceAndPlayAudio(zone.enigme2.noisetier.enter));
119          // wait for the user to pass the key on the box and unlock
120          break;
121        case zone.enigme3.id:
122          // When user is in the enigme 3 zone
123          dispatch(updateIndiceAndPlayAudio(zone.enigme3.enter));
124          // Then wait user to enter the digital code
125          break;
126        default:
127          break;
128      }
    }
```

Figure 45: Présentation de la boucle de jeu

Lorsque la partie est terminée ou interrompue, l'ensemble du parcours ainsi que les données comme la distance, les points des énigmes ainsi que le temps sont ajoutés dans les sessions du joueur et sauvegardées en base de données. Cela permet de garder la trace des efforts du joueur et lui permet, s'il le souhaite, de revoir une partie précédente. Pour cela, il suffit de la sélectionner dans la liste sur son profil (Cf. point 4 Figure 41).

Il arrive alors sur cette vue :



Figure 46: Vue résumé d'une session

Il est alors possible de visionner le résumé du parcours effectué sur la carte (1) ainsi que l'ensemble des données obtenues durant le parcours (2).

Le design de cette vue est basique, mais cette dernière remplit la demande du cahier des charges assurant le suivi des performances d'un utilisateur. Son design pourra être revu dans le cadre d'une continuation du projet.

L'application mobile comprend donc : une page de connexion et d'inscription, un système d'authentification et de gestion des utilisateurs, une page de profil avec un résumé des performances du joueur, des informations le concernant, ainsi qu'une liste de ses anciennes sessions desquelles il peut obtenir les informations résumées et une image du tracé effectué ce jour-là. Il lui est aussi possible de rejouer une partie ou d'utiliser l'application comme un traquer d'activité sportive.

L'ensemble du cahier des charges semble être rempli et la réflexion faites au départ du projet a permis d'obtenir un résultat concernant l'application mobile agréable au niveau des fonctionnalités et du design. L'ensemble du projet de développement et le code sont consultables sur le site <https://github.com/thomaslc66/Bachelor-2019>

4.3 Bornes interactives

La seconde partie de la phase de prototypage, concerne les bornes interactives. Lorsqu'il est question de bornes interactives, il faut penser à quelque chose similaire à l'illustration ci-dessous :

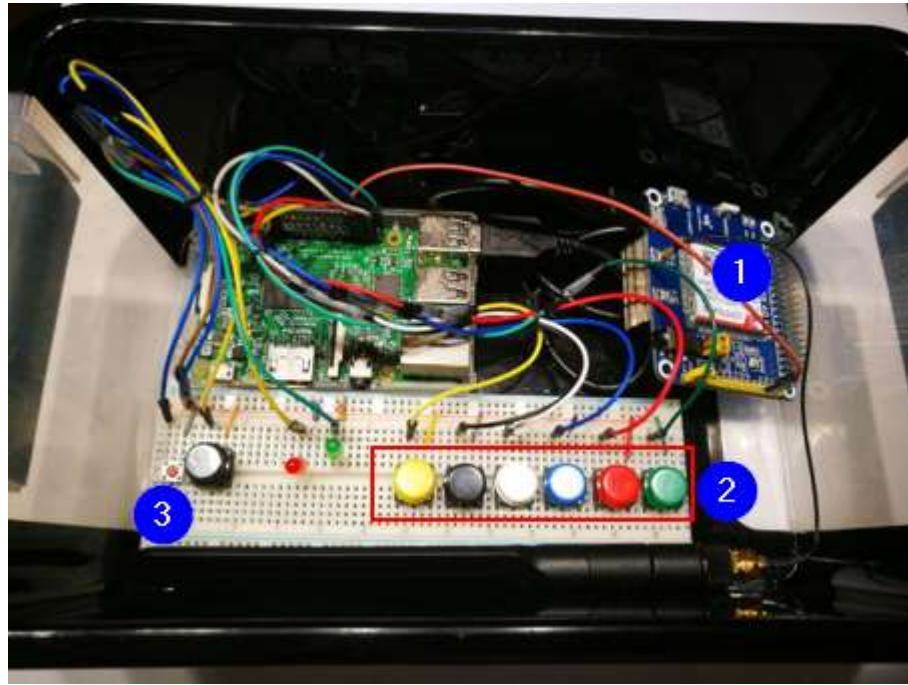


Figure 47 : Exemple de borne interactive, ici la borne n° 1

La borne se compose d'un ensemble de pièces électroniques avec lesquelles l'utilisateur interagit et résoud une série de problèmes posés, durant le jeu.

Concernant la partie physique/matériel, chaque borne possède un modem 4G (1), une série de composants avec lesquels une interaction est possible afin de résoudre l'éénigme (2) et deux boutons, pour la reconnexion au réseau 4G et le test du lien avec le server, tous connectés à une Raspberry pi 3b et alimentés par une batterie externe.

Pour la partie installation logiciel, un server Node-red ainsi que mosquitto et un broker mqtt installé, ces trois technologies ont été présentées dans le chapitre sur les technologies utilisées (Cf. Voir chapitre 3.2 et 3.3). Afin de faciliter le déploiement d'une future borne, une image type avec l'ensemble des scripts et logiciels est disponible sur le site <https://github.com/thomaslc66/Bachelor-2019>.

Chaque borne a donc reçu la dernière image Raspbian stretch 9 (Système d'exploitation Raspberry pi), le kernel 4.19.42-v7[39], puis l'installation de plusieurs librairies pour la gestion du modem[40, 41] et enfin un script de connexion tiré d'une librairie github [42] modifié pour fonctionner avec les cartes SIM utilisées et le modem.

Les trois bornes ont été nommées piUno, piDos et piTres.

4.4 Installation du modem/module 4G

Cette partie du travail est la plus technique et instable à l'heure actuelle. [40, 42, 43]

Le modem 4G est présent sur chaque borne. C'est un composant enfichable qui est ajouté et qui sert de modem à l'aide d'une carte SIM et d'une connexion 3G/4G. Chaque boîtier communique avec le serveur central du jeu par le biais de ce modem 4G. Ces modules sont des SIM 7600-H pour la version européenne. Ils sont alimentés par USB et peuvent être utilisés comme modem externe pour la connexion internet de la Raspberry pi sur le terrain, rendant alors possible la communication des bornes

avec le server central. Cette partie est importante, car sans ces modules, la connexion au server n'est pas possible et l'utilisation des bornes est impossible.

Un script de connexion spécifique a été réalisé afin de pouvoir démarrer les bornes directement en utilisant ce modem 4G. Le module prend du temps à s'initialiser au démarrage, c'est pour cela que le script de connexion est lancé deux minutes après le démarrage des Raspberry Pi. Ainsi, tous les modules sont prêts, initialisés et le script de connexion s'exécute sans erreurs de module introuvable.

Pour configurer le script il faut posséder le nom du réseau mobile de son fournisseurs de carte SIM. Deux librairies utilisées dans le script de connexion possèdent des bugs qui les rendent incompatibles et bloquent la connexion au modem 4G, ainsi que sa connexion au réseau mobile.[44] Il s'agit des modules dhcpcd dans sa dernière version et de la librairie libqmi-utils présente de base sur Raspbian. Pour éviter cela il faut installer et compiler une version plus ancienne de dhcpcd (6.11.5) et de libqmi (1.16.0). Puis utiliser un script permettant d'éviter les nombreuses erreurs rencontrées, en utilisant libqmi et qmi-network manager. [45–55]

```

28 function helpmsg {
29     echo "usage: $0 {start|stop|restart|status}"
30     exit 1
31 }
32 #boot GPIO
33 function qmi_boot {
34     if [ ! -d /sys/class/gpio/gpio18 ]; then
35         echo "File exists."
36         echo "18" > /sys/class/gpio/export
37         sleep 2
38     fi
39     echo "out" > /sys/class/gpio/gpio18/direction
40     sleep 1
41     echo "0" > /sys/class/gpio/gpio18/value
42     echo "1" > /sys/class/gpio/gpio18/value
43     sleep 15
44     # your cdc_wdm modem location
45     CDC_WDM=/dev/cdc-wdm0
46     qmi_start
47 }
48 # Start connection to 4g network
49 function qmi_start {
50     $COMMAND_PREFIX $LSUSB | grep Qualcomm
51     #check status
52     STATUS=$($COMMAND_PREFIX $QMI_NETWORK $CDC_WDM status | sed -n 's/.*Status: //p')
53
54     if [ $STATUS == 'connected' ]; then
55         echo "Network is connected, only need to ask for an IP"
56     else
57         echo "---- Reseting qmi lib ----"
58         $COMMAND_PREFIX $QMICLI -d $CDC_WDM --wds-reset
59         echo "Setting data format to raw-ip"
60         $COMMAND_PREFIX $IP link set $WWAN_DEV down
61         $COMMAND_PREFIX $QMICLI -d $CDC_WDM --set-expected-data-format=raw-ip
62         $COMMAND_PREFIX $IP link set $WWAN_DEV up
63         echo "start network..."
64         $COMMAND_PREFIX qmicli -d $CDC_WDM --wds-start-network="apn=$APN,ip-type=$TYPE" --client-no-release-cid
65     fi
66
67     if [ $? -ne 0 ]; then
68         echo "your wwan device may be RFKilled?"
69         exit 1
70     fi
71
72     $COMMAND_PREFIX $DHCPD -q -f -n -i $WWAN_DEV
73     $COMMAND_PREFIX $IFCONFIG -v wwan0
74     $COMMAND_PREFIX $IP r s
75 }
```

Figure 48: Script de connexion au réseau en utilisant le modem externe

Le script, modifié, offre l'utilisation de plusieurs commandes comme le montre la ligne 29 de l'image ci-dessus. La commande « boot » permet au script d'initialiser/réinitialiser le modem lorsque ce dernier n'est pas allumé. Cela arrive parfois, car les modules ont de la peine lorsqu'ils sont alimentés avec les powerbanks. Ils affichent souvent un message de sous voltage et ne fonctionnent pas correctement dû à cela. Les powerbanks utilisés délivrent en effet 2.1 ampères alors que la Raspberry en consomme à elle seul (sans le modem) 2.5A.

Ces modules 4G sont très sensibles et peuvent ne pas fonctionner pour de nombreuses raisons. Durant mes recherches, il est apparu que les ports USB des Raspberry sont sur le même bus/hub électrique que la carte réseau et donc la prise réseau. Plusieurs fois le boîtier 4G ne démarrait pas car une prise réseau était connectée. En débranchant la prise et en redémarrant le boîtier, le problème est souvent rentré dans l'ordre.

Un des boîtiers avait de réels problèmes de démarrage et de connexion au réseau. Grâce à internet, j'ai pu trouver certains problèmes présents dans les librairies utilisées spécialement pour le système d'exploitation Raspberry. En effet, il faut savoir que le système d'exploitation Raspbian se base sur un noyau linux, mais présente certaines différences.

Après avoir réussi à configurer un des boîtiers de manière qu'il se connecte automatiquement avec les bons drivers et les bonnes librairies, et après avoir perdu du temps à tenter de faire la même installation depuis le départ sur les autres deux bornes, il a été décidé faire un clone du boîtier fonctionnel en clonant la carte SD et de réinstaller ce clone sur les deux autres boîtiers.

De cette manière chaque boîtier est configuré exactement de manière identique et tout problème concernant ces boîtiers serait alors non plus lié aux divers modules et librairies mais bien à des problèmes hardware.

4.5 Le serveur de jeu

N'ayant pas la possibilité de payer pour un serveur virtuel dans un data center, il a été pris la décision d'utiliser un serveur personnel. La machine est assez puissante pour gérer un nombre conséquent de requêtes et peut, le cas échéant, être modifiée (allocation de plus de mémoire) car le server est virtualisé.

Sur un système d'exploitation Ubuntu, Node-red ainsi qu'un server Mosquitto et un broker Mqtt ont été installés. Puis, ne possédant pas d'IP fixe, un service de DNS dynamique a été utilisé permettant d'utiliser une url pointant toujours sur l'adresse IP du routeur privé. Ainsi, même si l'adresse IP de ce dernier, change, le jeu n'est pas impacté, car l'url utilisée pour se connecter au server de jeu sera mise à jour afin de pointer vers la nouvelle adresse IP du routeur.

Pour accéder au server de jeu, l'ouverture de port a été nécessaire dans le routeur. Ceci permet aux bornes de pouvoir s'adresser au bon service du réseau. Les ports 1880 pour l'interface web et 1883 pour la communication Mqtt sont ouverts et laissent passer le trafic.

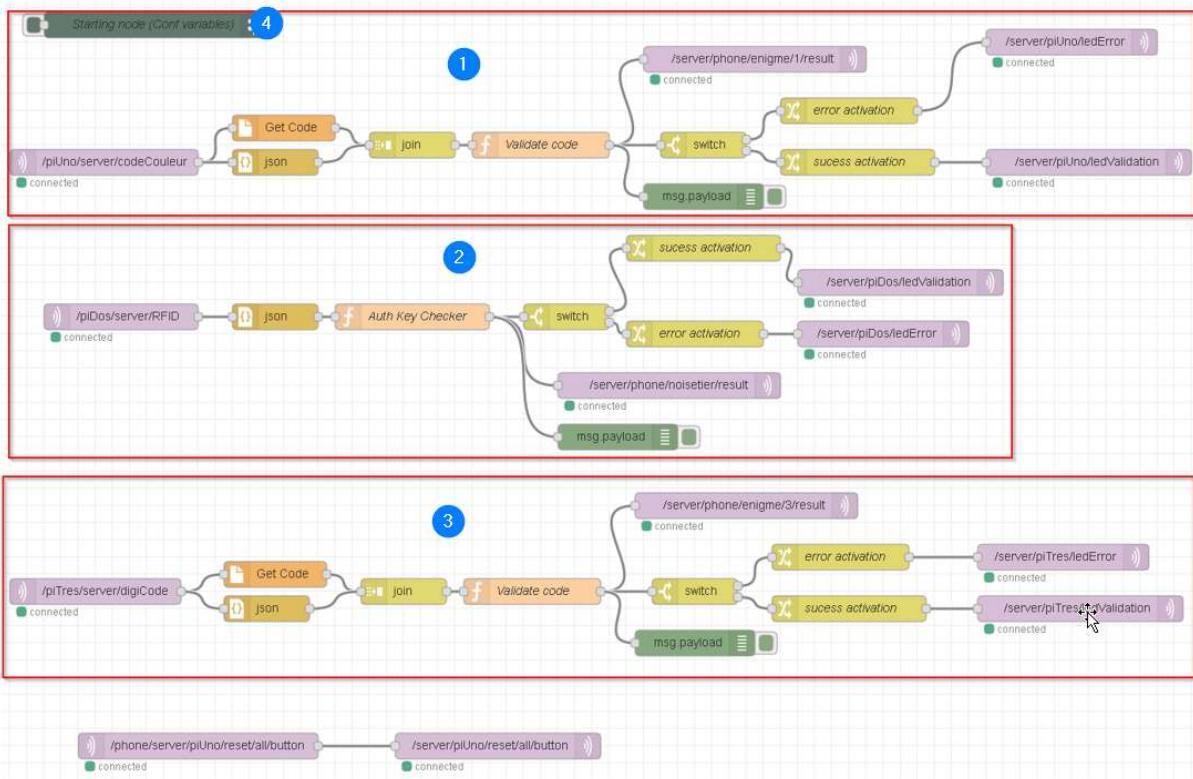


Figure 49: Server de jeu central avec les trois énigmes

Dans la Figure 49 ci-dessus, l'on trouve le flow créé dans le serveur central. Il permet de gérer les trois énigmes. La première énigme attend l'entrée faite par le client. Les nœuds Mqtt de publication et de souscription possèdent une nomenclature spécifique. Ils commencent tous par le caractère « / », puis vient le nom de l'émetteur, suivi du nom du destinataire et le sujet de ce qui est transmis dans le message. Ainsi « /server/phone/enigme/3/result » permet de comprendre qu'il s'agit du résultat de l'énigme trois qui est envoyé du server central au téléphone. Cela donne une structure et facilite le debugage en cas d'erreur.

Pour un nœud Mqtt de réception, le système est identique « /piUno/server/codeCouleur ». On devine alors que c'est la borne n°1 (piUno) qui transmet au server le code couleur entré par l'utilisateur dans une tentative de résolution de l'énigme n°1.

Une fois le code couleur reçu, la demande code valide est envoyée à la base de données. Le résultat passe dans une fonction permettant de comparer si le code reçu est correct ou non. De ce fait, si l'on souhaite changer le code, il suffit uniquement de changer le code attendu dans la base de données. Selon les résultats, on retourne une réponse valide au téléphone ainsi qu'au boîtier afin que l'utilisateur sache si son code est correct. Dans le cas contraire un résultat « faux » est envoyé au joueur et le led rouge du boîtier s'affiche. Cela donne deux indications au joueur, une audio et une lumineuse. Le principe est exactement le même pour l'énigme trois, le code est reçu, comparé à celui de la base de données puis le résultat est envoyé au téléphone du joueur et au boîtier.

La seconde énigme, elle compare l'identifiant de la clé passée sur le module RFID avec l'identifiant de la clé définie comme valide. Sur la Figure 50 au point 1, on remarque un appel à `global.get('validKeyUid')` cet appel permet d'utiliser le stockage interne de Node-red. Ce stockage est présent sur la Figure 49 au point numéro 4. Ce nœud permet de stocker des variables qui sont initialisées au démarrage du serveur et qui persistent en mémoire. En ouvrant le nœud (4) on peut alors voir que l'UID de la clé attendue a été enregistré ici.

```

1 var result = true;           i
2 var validKeyUid = global.get('validKeyUid').UID;
i 3 var uid = msg.payload.uid
4 var len = uid.length;
5
6 for(i = 0; i < len; i++){
7   if(uid[i] != validKeyUid[i]){
8     result = false;
9     break;
10  }
11 }
12
13
i 14 msg.payload = {"result": result}
15
16
17 return msg;

```

Figure 50: Fonction Auth Key Checker



Figure 51: valid Key Uid stocké sur Node-red

Pour changer la clé, il suffit de connaître son UID et de le remplacer dans ce nœud. Il aurait été possible de sauver cette valeur dans la base de données, mais il a été choisi ici de montrer les possibilités offertes par la technologie Node-red.

4.6 Gameplay

Le gameplay est la partie qui consiste à créer le jeu et le chemin qui sera suivi par le joueur. Il définit aussi les règles, contraintes ainsi que les résultats des interactions de l'utilisateur. Il a été important de réfléchir sur cette partie avant de commencer le développement. Une attention particulière a été apportée aux énigmes et à leurs liens avec le terrain.

Il se présente comme suit :

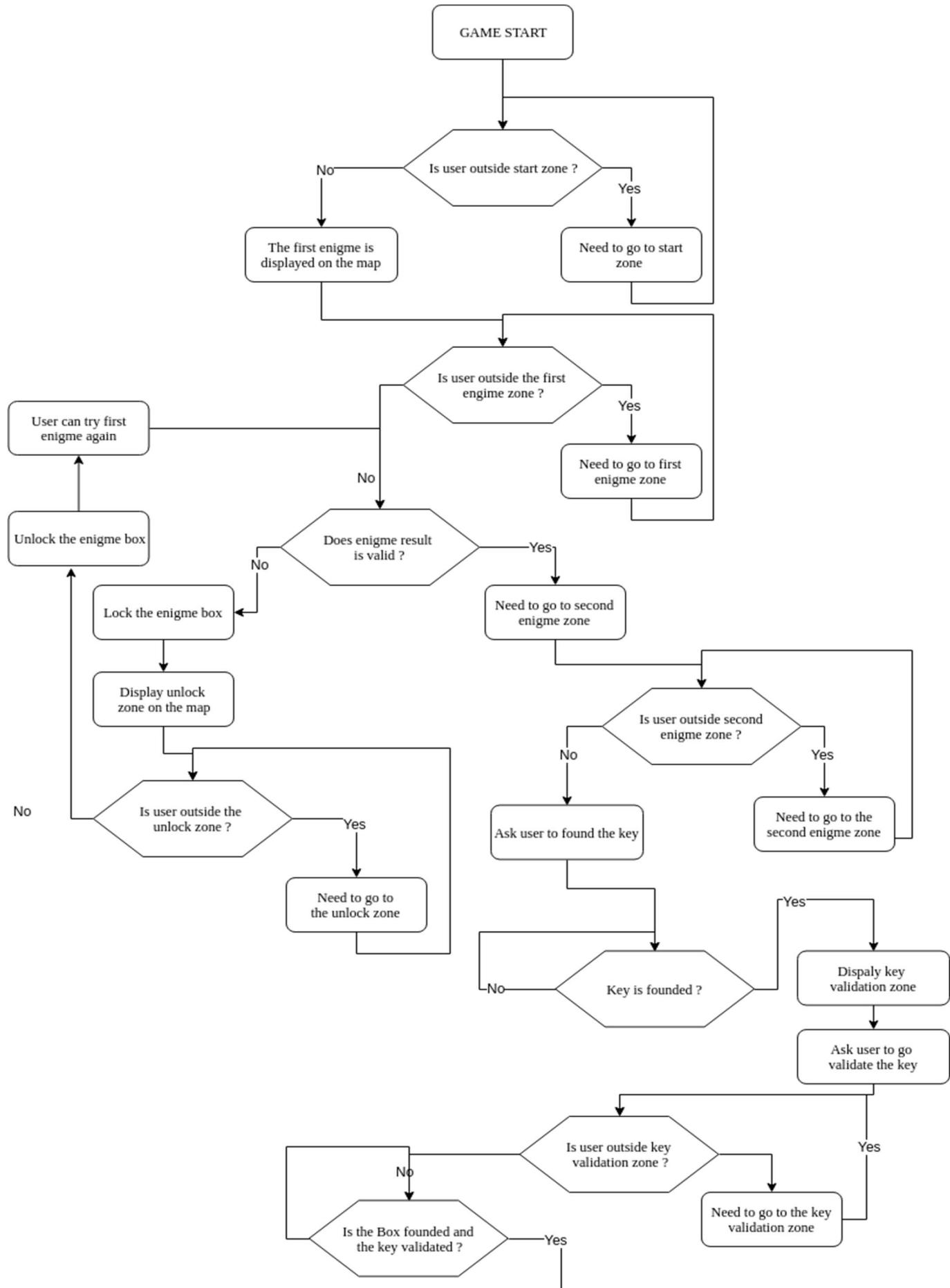
- 1) *Le joueur, s'inscrit ou se logue avec son compte*
- 2) *Il accède à son profil.*
- 3) *Il démarre le jeu avec le bouton prévu à cet effet*
- 4) *Il suit le flow du jeu schématisé (Voir schéma du flow Figure 52)*
- 5) *Il termine le jeu où quitter le jeu*
- 6) *Les points sont calculés et sa progression est sauvegardée dans son profil.*
- 7) *Il peut visualiser sa séance.*

Dans ce chapitre seront présentés, le prototype final, le jeu implémenté avec les modifications par rapport au concept imaginé au point 1.3, ainsi que la présentation des énigmes et leur implémentation. Cette partie de présentation des bornes interactives aurait pu se retrouver au point 4.3 mais beaucoup d'informations auraient dû être répétées dans cette portion du rapport.

En s'inspirant du concept imaginé, le jeu est devenu une quête. Le gardien c'est fait kidnapper par une bande de ravisseur souhaitant s'emparer du secret de la grande-cariçaie. Pour ce faire, ils ont kidnappé le gardien afin de lui soutirer des informations. Pendant ce temps, personne ne surveille la réserve naturelle. La quête du joueur est donc de retrouver le gardien, protéger la réserve et restaurer l'équilibre des lieux. Pour ce faire, il doit suivre les traces laissées par les ravisseurs sur leur chemin, résoudre les énigmes qui se présentent devant lui et libérer le gardien.

Afin de mieux comprendre ce chapitre, une carte détaillée se trouve dans les annexes (Cf. chapitre 9.8) et offrira une aide visuelle aux explications.

Le jeu comprend trois énigmes, chaque énigme est délimitée par une ou deux zones. Chaque zone du jeu est une geofence de diamètre différent. L'entrée dans une zone a pour effet de déclencher la suite du jeu ainsi que la lecture d'un fichier audio pour aider le joueur, lui donnant ainsi des indications sur l'énigme à résoudre, ou pour continuer l'aventure. Pour commencer, ci-dessous se trouve le flow du gameplay qui démontre les possibilités et interactions qui s'offre au joueur durant une partie.



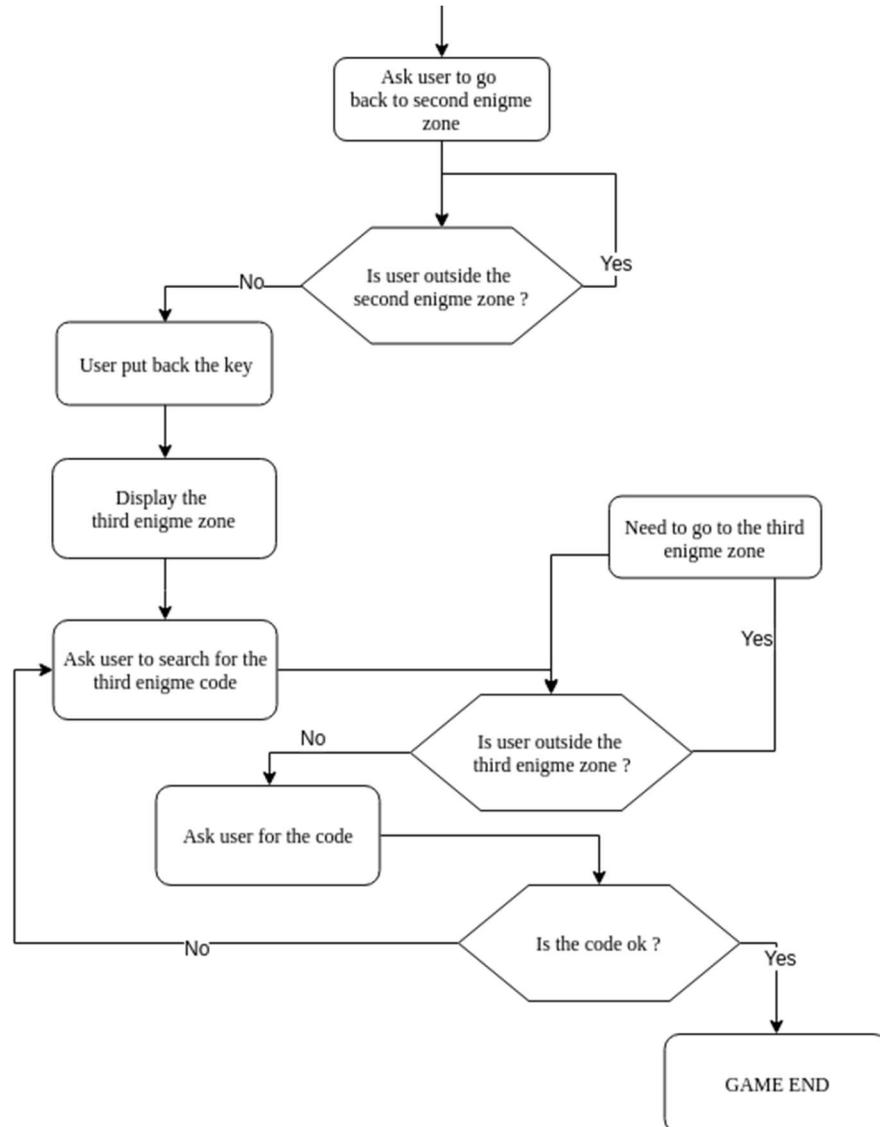


Figure 52: Flow du jeu complet.

4.6.1 Zone de départ

Lorsque le joueur démarre la partie, seule la zone de départ est affichée. Il doit alors s'y rendre pour démarrer sa quête. En pénétrant dans cette zone, il est détecté par la boucle de jeu (Cf. Figure 45). Les indications concernant son aventure lui sont jouées par audio. (Il est conseillé d'utiliser des écouteurs, afin d'éviter de sortir sans cesse le téléphone et pour une immersion totale).

C'est ici, que l'utilisateur prend connaissance de sa quête (Cf. Texte audio 9.7.1). Le gardien de la zone étant kidnappé, c'est à lui que revient la tâche de le retrouver et de protéger la réserve pendant son absence. Pour cela il doit suivre les traces laissées par les ravisseurs qui n'ont pour unique but que de s'emparer du secret de la réserve de la grande caricaie. Il n'y a bien évidemment pas de réel secret, mais cela donne une dimension fantastique à l'aventure. Tout au long de l'aventure, la voix-off aide le joueur est possède un rôle d'assistant. C'est d'ailleurs cette voix qui indique à l'utilisateur où se rendre. En démarrant le jeu et en prenant connaissance de la quête à mener, le joueur reçoit l'indication que les ravisseurs ont pris la direction de la zone des menhirs ou est situé la première énigme.

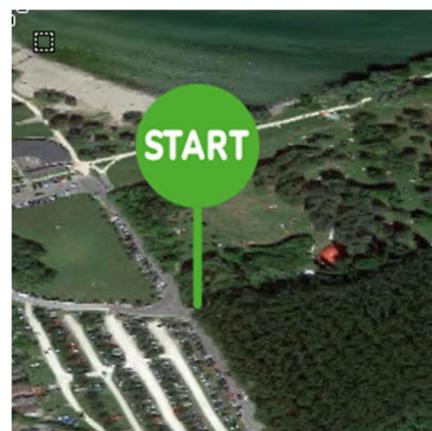


Figure 53: Zone de départ

Le départ de cette zone est modulable. Il a été mis ici, car il correspond au départ de l'ancien parcours sportif.

4.6.2 Première énigme

La première énigme se trouve vers la zone des menhirs de Clendy (2) :



Figure 54 : Zone de départ et zone de la 1^{ère} énigme

Quarante-cinq menhirs sont présents dans cette zone aux abords de la plage d'Yverdon et seront utilisés pour la première énigme du jeu. L'utilisateur est donc sur la trace des ravisseurs. Il entre alors dans la zone des menhirs. La voix-off lui annonce (Cf. Texte audio 9.7.2) que les ravisseurs sont en train de sonder le sol à la recherche du « secret » de la grande-cariçaie. Une des tâches du joueur étant la protection de la réserve en l'absence du gardien, c'est au joueur qu'il revient de trouver les six sondes cachées et de les désactiver.



Figure 55 : Menhirs de Clendy

Il doit alors explorer la zone complète et trouver ces sondes de couleurs différentes. Ces sondes ont été cachées aux abords des menhirs. La voix-off indique, d'ailleurs, au joueur qu'il doit chercher la première sonde de couleur aux abords du plus grand des menhirs de la place. Il doit alors comprendre, pour résoudre l'énigme, que les six couleurs des sondes doivent être classées selon l'ordre de grandeur des menhirs.



Figure 56: Exemple des points de couleurs à trouver

Pour représenter les sondes, des grosses visse à boulons ont été peintes avec les mêmes couleurs que la borne et ont été disposés dans cette zone derrière les menhirs sélectionnés pour l'énigme.



Figure 57: Exemple des supports de couleurs.
Certains supports ont été enlevés par le personnel de la voirie lors de l'entretien du gazon.

Un fois les six couleurs trouvées, le joueur est invité à utiliser la borne à disposition pour entrer le code, désactiver et stopper l'exploration du sol.



Figure 58 : série de boutons permettant d'entrer le résultat

Si l'ordre des couleurs est respecté et que le code entré par l'utilisateur est correct, cela déclenche l'activation de la seconde énigme ainsi que la voix-off qui félicite notre joueur d'avoir désactivé les six sondes en stoppant les plans des ravisseurs (Cf. Texte audio 9.7.4). En cas d'erreur, le joueur est informé qu'il s'est trompé. La voix-off lui rappelle qu'il doit commencer par le plus grand des menhirs (Cf. Texte audio 9.7.3). Il doit cependant se rendre dans la zone de réinitialisation, car la borne s'est bloquée. Il ne peut alors plus rentrer de code tant qu'il n'est pas aller chercher la clef dans cette zone. Il ne s'agit pas d'une clef physique, mais l'entrée dans cette zone a pour effet de déverrouiller la boîte et permet au joueur de retenter sa chance.

Tant que le joueur n'est pas capable d'entrer le code de six couleurs, trouvé dans la zone des menhirs, correctement, cette zone s'activera à chaque échec et le joueur devra se rendre dans cette zone de réinitialisation, lui faisant perdre du temps dans sa quête et en lui faisant pratiquer du sport.



Figure 59: Position 1ere énigme et zone de réinitialisation

A première réflexion, cette énigme était censée disposer d'un temps maximum pour sa réalisation. Après plusieurs tests et questions, il a été décidé de ne pas rajouter de temps limite. En effet, il s'agit en premier lieu de donner envie aux utilisateurs de jouer. De plus sur quel critère faut-il fixer le temps à disposition pour ne pas décourager certaines personnes moins sportives ? Une modification pourrait, cependant, être d'utiliser le temps total du joueur dans l'attribution de ces points. Attribuant ainsi des pénalités ou points négatifs si le temps total défini pour résoudre l'ensemble du jeu est écoulé.

Certaines autres idées ont été envisagées, lors de la réflexion, pour la création d'un second niveau. Disposer les pics de couleurs de façon que chaque groupe de menhirs possède une couleur distincte. Ce n'est plus la hauteur des menhirs qui fera fois pour l'ordre du code, mais le nombre de menhirs dans chaque groupe. Cela s'est avéré plus difficile que prévu. En effet, certains groupes de menhirs possèdent le même nombre total, d'autres rangées sont coupées en deux par le lit d'une ancienne rivière. Pour le moment un seul niveau a été imaginé en entier.

La programmation de la borne n°1 avec Node-red se présente comme suit :

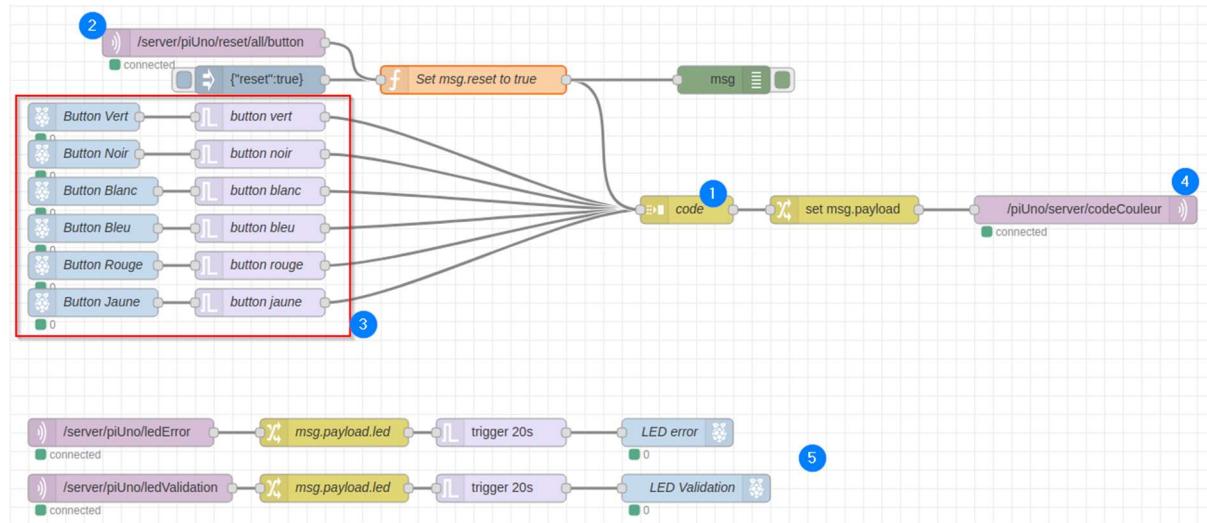


Figure 60: programmation Node-red de la borne n°1

Chaque bouton est lié à son pin GPIO (3). Le nœud `code` (1), attend d'obtenir six entrées, soit six valeurs provenant des boutons avant de continuer. Il est donc possible, grâce à ce nœud, d'attendre le code couleur complet, entré par l'utilisateur, avant de l'envoyer.

Les nœuds du dessus (2) permettent une réinitialisation du compteur. En effet, durant nos tests, il s'est avéré que des boutons étaient pressés par mégarde. Cela ayant pour but d'initialiser le compteur du nœud « `code` » (1), rendant la première saisie du joueur fausse. Pour éviter cela, le nœud « `code` » (1) reçoit une information de réinitialisation à zéro au moment où le joueur entre dans la zone.

Une fois que le code complet est reçu, il est transmis au server central par un nœud de publication MQTT (4). Le server traite le code, puis retourne le résultat au téléphone (au joueur) ainsi qu'à la borne (5) qui allume la led rouge ou verte selon le succès ou l'échec de l'opération. L'envoi du résultat au téléphone se fait directement depuis le serveur de jeu. (Cf. Figure 49)

Ci-dessous le schéma de câblage de la borne n°1 :

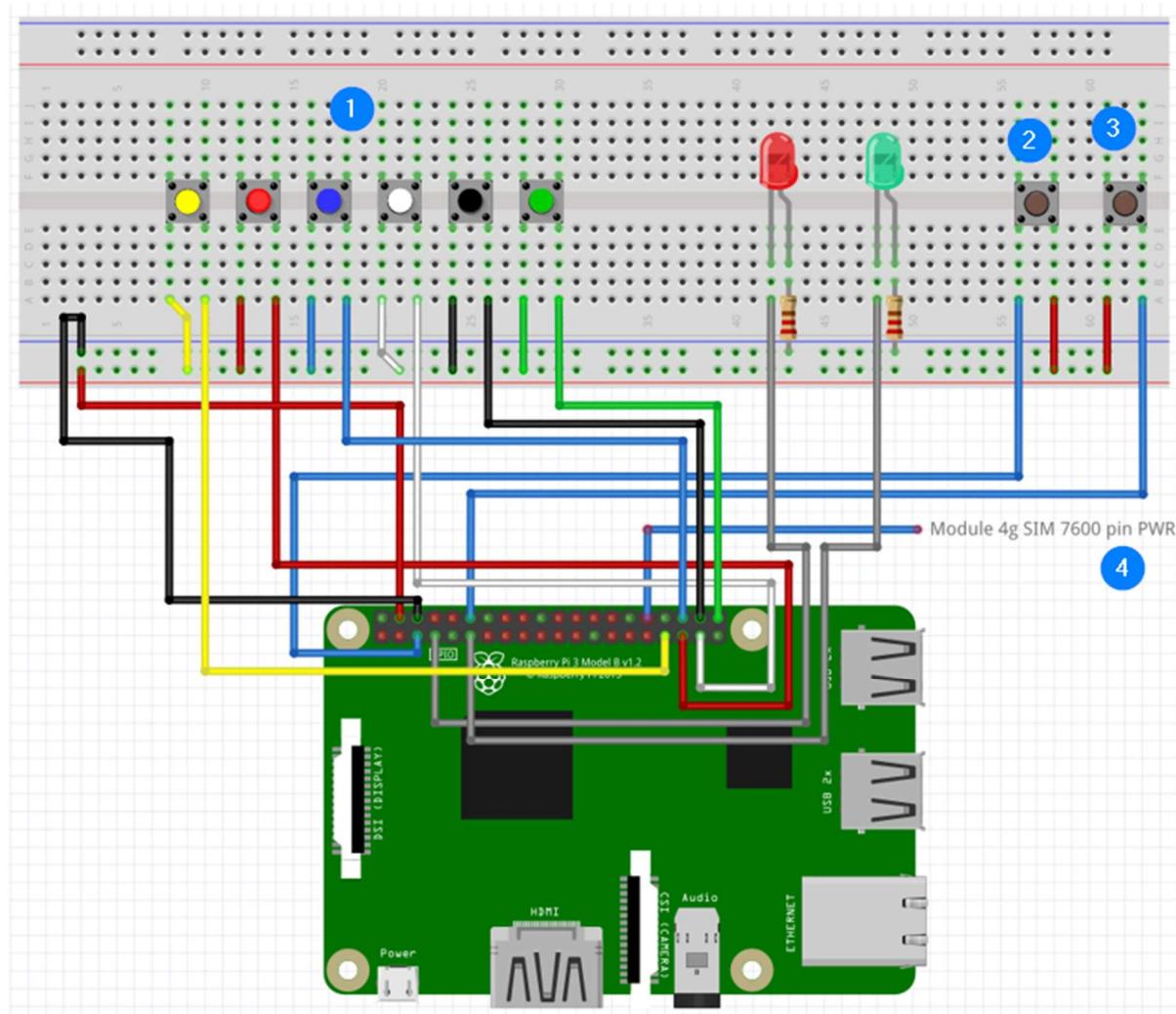


Figure 61: Câble et schéma de la borne n°1. Fait avec le logiciel Fritzing [56]

- 1) Ensemble de boutons pour entrer le code couleur
- 2) Bouton permettant de réinitialiser une connexion en cas de coupure du modem 4G
- 3) Bouton de vérification de connexion
- 4) Lien vers le modem 4G sur le pin PWR du module.

4.6.3 Seconde énigme

La seconde énigme se présente comme suit :



Figure 62: Situation sur la carte de l'énigme 2

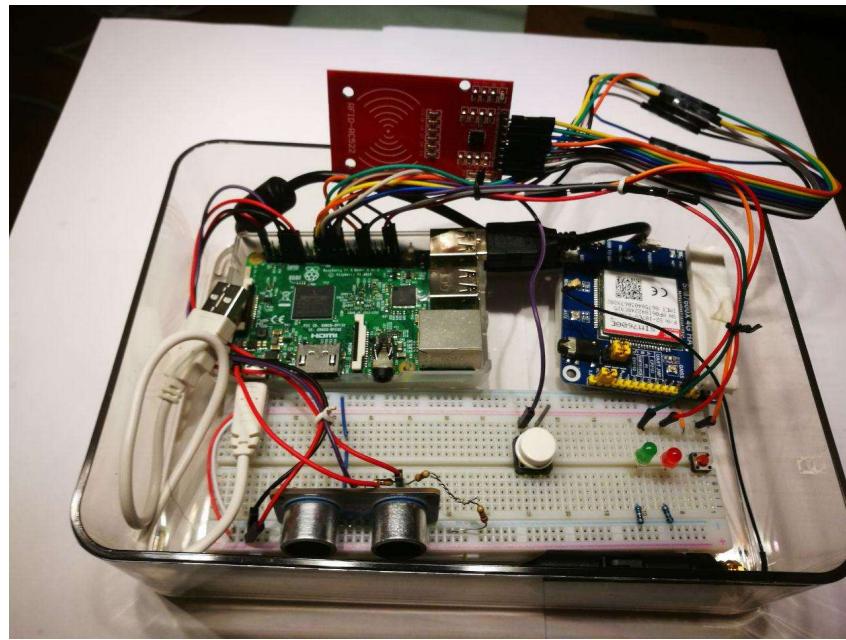


Figure 63 : Borne interactive pour l'énigme n°2

Toujours sur la trace des ravisseurs et après la résolution de la première énigme des menhirs avec succès, la voix-off indique au joueur de se rendre dans la zone de l'observatoire (Cf. Texte audio voir chapitre 9.7.6). Cette zone apparaît sur la carte du jeu. Le joueur arrive alors à l'observatoire de la grande carrière. (Cf. Figure 66).



Lors de son déplacement en direction de l'observatoire, le chemin du joueur croise un arbre « magique » (pour les besoins du jeu). Un noisetier comme indiqué sur le panneau se trouvant devant (Cf. Figure 65). Il ne le sait pas encore, mais cet arbre est utile pour la suite de l'aventure car il a vu passer les ravisseurs.

Le joueur entre dans la zone de l'observatoire. La voix-off lui indique alors avoir vu un arbre ligoté et demande à l'utilisateur de trouver une clef (Cf. Texte de l'audio, voir chapitre 9.7.6) Il s'agit d'un badge RFID.



Figure 64: Clef / Badge RFID

Figure 65: Le noisetier qui cache la seconde borne entre les troncs

La zone du noisetier s'affiche alors sur la carte (Cf. point 2 ou icône de l'arbre de la Figure 62). C'est dans cette zone que l'utilisateur utilisera le badge.



Figure 66: Enigme 2 - Observatoire de la grande cariçaie



Figure 67: Panneau explicatif qui cache la clef

Le joueur prend possession de la clef aimantée/cachée (Cf. Figure 64) sous le panneau (Cf. Figure 67), se dirige avec l'aide de la carte vers la zone du noisetier à présent visible dans le jeu et valide la clef sur la seconde borne qui se trouve au milieu des nombreux troncs de l'arbre. (Cf. Figure 65)

Lorsque la clef RFID est lue est validée, le noisetier, précédemment ligoté, est libéré (Cf. Texte Audio, voir chapitre 9.7.7 et 9.7.8). Remerciant le joueur il lui annonce la suite de sa quête. Il lui est alors demandé de retourner à l'observatoire (Point (2) de la Figure 62).

Ce retour à l'observatoire permet deux choses :

- 1) Demander au joueur de reposer la clef.
- 2) Faire chercher au joueur le code pour la troisième et dernière énigme.

La seconde entrée dans la zone de l'observatoire valide la totalité de l'énigme n°2 et permet de continuer l'aventure en direction de l'énigme n°3 et la fin du jeu. Cette phase du jeu étant à cheval entre l'énigme n°2 et l'énigme n°3 la suite est présentée au point suivant.

Le flow Node-red de la borne se présente comme suit :

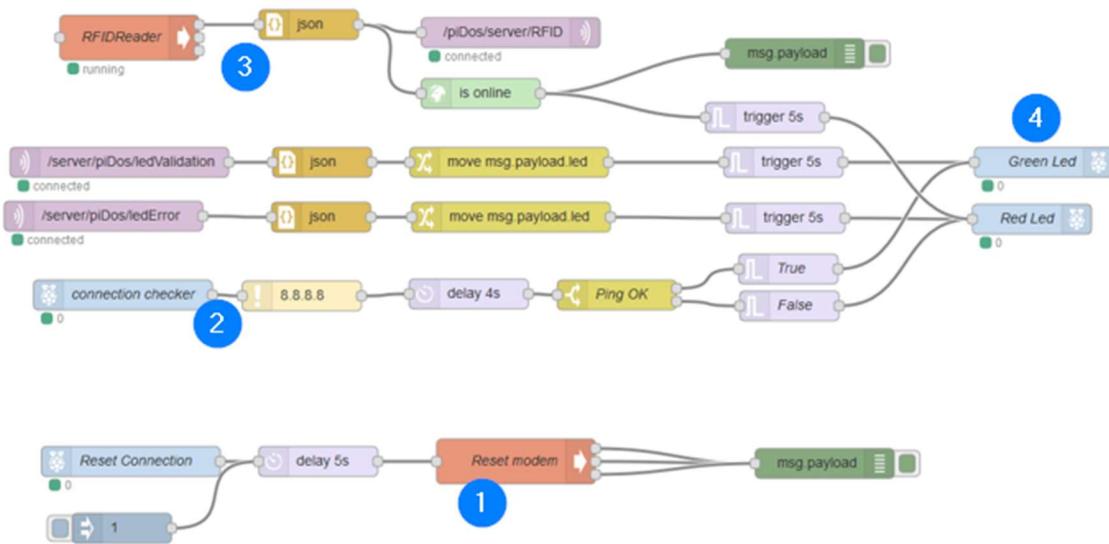


Figure 68: Image du flow de la borne n°2

3 points sont importants. Comme dans chaque borne, on retrouve :

- 1) Un code pour réinitialiser la connexion au modem
- 2) Un bouton pour vérifier la connexion au server.
- 3) Le script de lecture de la carte et l'envoi au server central.
- 4) La gestion des leds sur la borne

Le flow utilise un script de lecture pour lire les clef RFID (3) et vérifie que le tag de la clef passée sur le lecteur correspond bien au TAG de la clef autorisée. Le script est présent dans les sources disponibles sur le site : <https://github.com/thomaslc66/Bachelor-2019>. Cela permet en cas du scan d'une mauvaise clef de refuser et de ne pas libérer le noisetier. Contrairement à l'énigme n°1, il n'y a pas ici de zone de réinitialisation, car on demande déjà à l'utilisateur de faire un aller-retour entre ces deux zones.

Schéma de la borne numéro 2 :

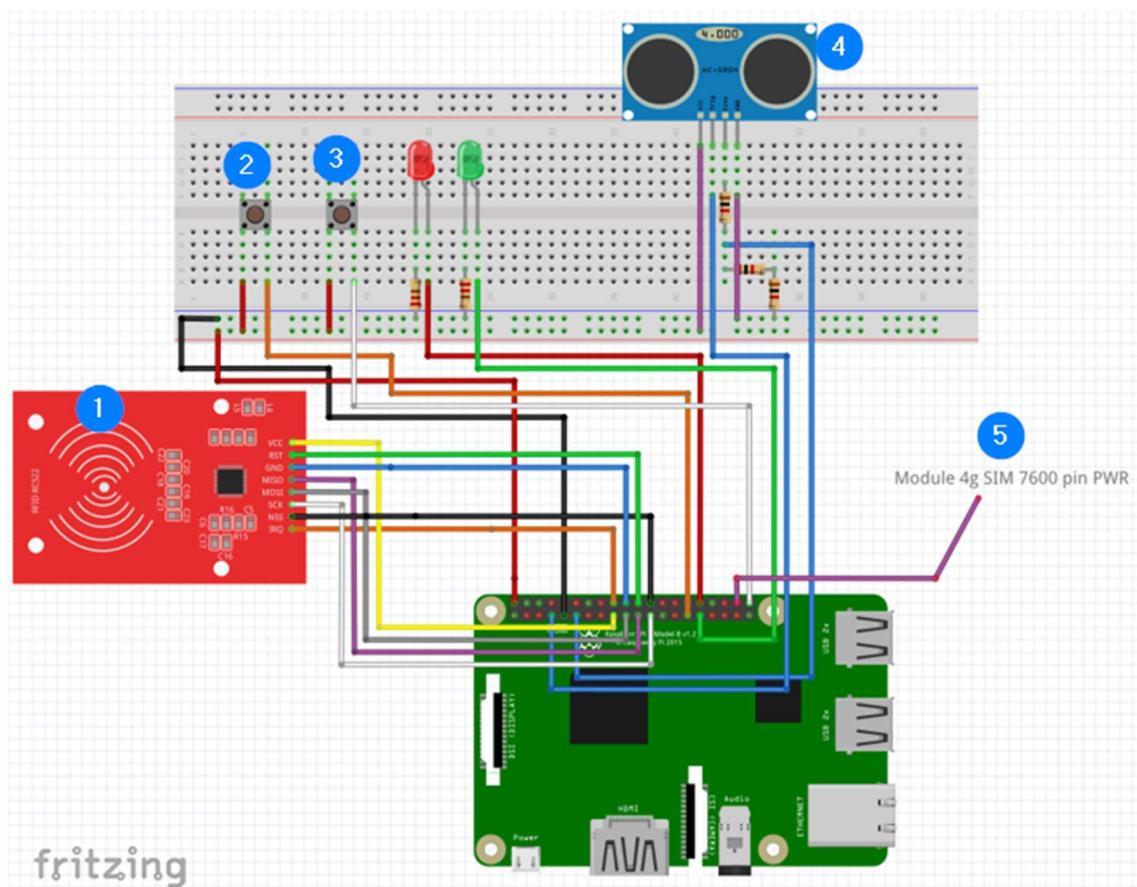


Figure 69: Câblage et schéma de la borne n°2. Fait avec le logiciel Fritzing [24]

- 1) Lecteur de carte RFID
- 2) Bouton de vérification de connexion
- 3) Bouton permettant de réinitialiser une connexion en cas de coupure du modem 4G
- 4) Capteur de distance (voir chapitre 9.6)
- 5) Lien vers le modem 4G sur le pin PWR du module.

4.6.4 Dernière énigme

Le joueur vient de libérer l'arbre « magique ». Il retourne à l'observatoire comme demandé par le noisetier et repose la clef. De retour dans cette zone de l'observatoire, la voix-off lui indique avoir vu les ravisseurs prendre la direction du port, laissant derrière eux un code de six chiffres (Cf. Texte audio 9.7.9). Ce retour a aussi pour effet la validation de l'énigme n°2 et l'affichage sur la carte de la zone de la dernière énigme n°3.

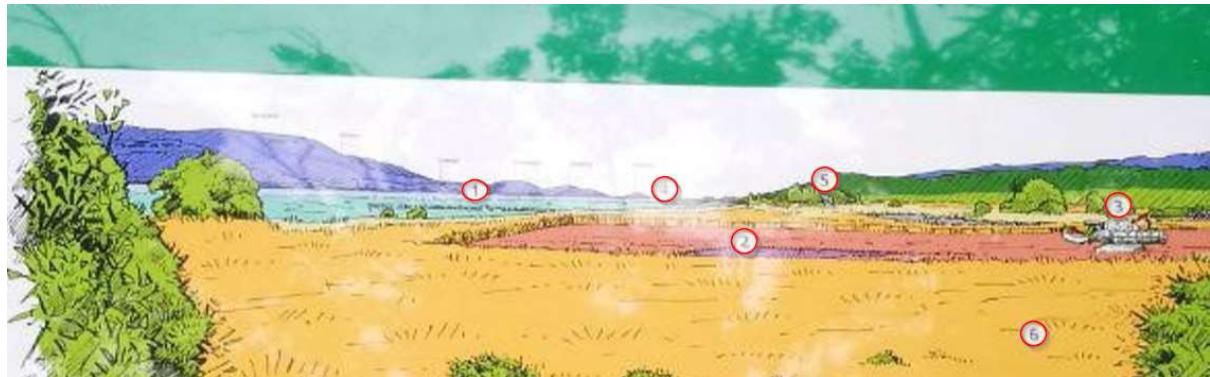


Figure 70: La voix-off annonce du lac à la moissonneuse batteuse le code apparaît.

Il est alors demandé au participant de trouver le code de six chiffres permettant la résolution de la troisième et dernière énigme du jeu. Dans cette énigme il devra mémoriser les six chiffres présents sur le panneau de la grande caricaie. La voix-off indique « Trouvez le code de six chiffres du lac à la moissonneuse batteuse ». On voit donc sur la Figure 70, le lac à gauche et la moissonneuse batteuse à droite et on peut deviner le code 1-4-2-5-6-3, qu'il faut mémoriser avant de se rendre à la dernière zone.

Cette dernière étape se déroule un peu plus loin que le reste du parcours puisqu'elle se situe aux abords de la piscine d'Yverdon, proche du lac.

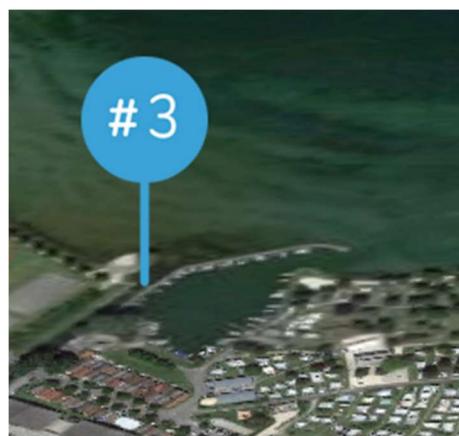


Figure 71: Disposition de la dernière énigme

Au bord du canal se trouve une statue regardant aux jumelles. Elle représente dans notre jeu le gardien qu'il faut libérer. En effet, la moitié de son corps est dans un socle en béton. Il est alors prisonnier et ne peut plus surveiller la réserve naturelle de la grande carrière.



Figure 72: Dernière étape, sauver le gardien de son socle

Emprisonné dans son socle, le joueur a pour mission de le libérer. La voix-off lui indique que le code mémorisé à l'étape précédent doit être utile ici (Cf. Texte audio 9.7.10). Le joueur doit alors utiliser sa mémoire pour se souvenir du code de la Figure 70 et utiliser le boîtier à disposition pour entrer le code.

La borne n° 3 se présente comme suit :

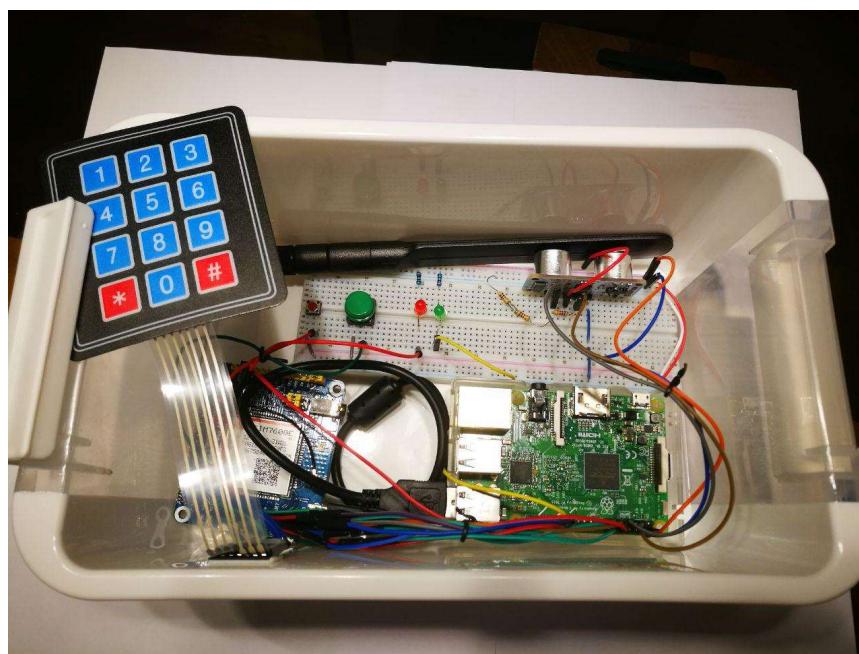


Figure 73: Borne n° 3

Si le code entré est correct, le gardien est libéré. Il s'adresse alors au joueur (Cf. Texte audio 9.7.12) le remerciant de l'avoir libéré. Il annonce aussi n'avoir pas vu ses ravisseurs et qu'ils pourraient revenir, laissant la possibilité pour créer une seconde énigme.

En cas d'erreur du code (Cf. Texte audio 9.7.11), le joueur peu tenter à nouveau d'entrer le code sans avoir besoin de se rendre dans une quelconque zone. En effet, dans le cas où le joueur a complètement oublié le code, il devra déjà faire la route pour retourner à l'observatoire, qui est conséquente.

Le flow de la borne n°3 se présente comme suit :

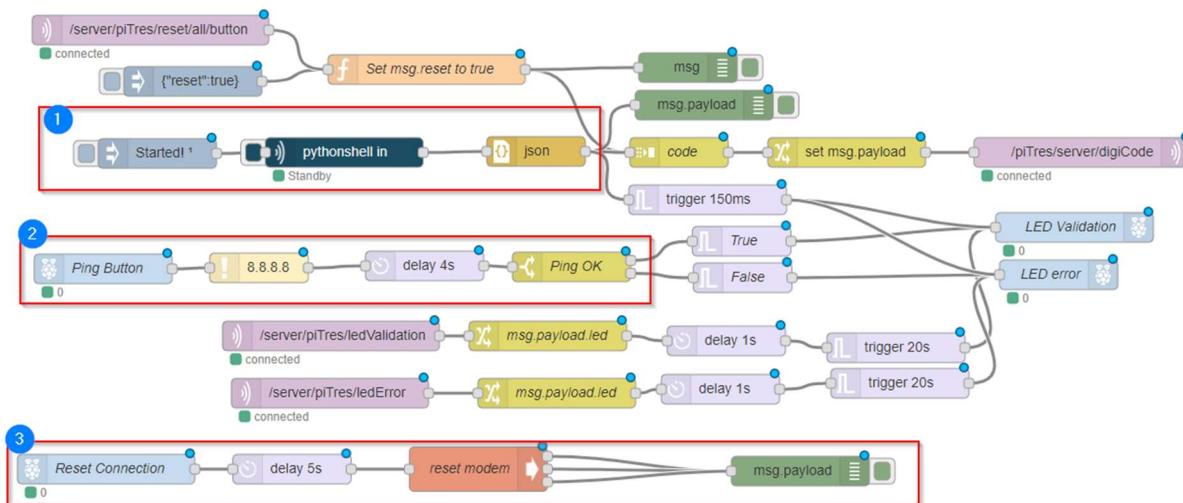


Figure 74: Flow de la borne n°3

Comme pour les autres bornes :

- 1) Le script en python qui permet de lire le code
- 2) Le bouton de test de connexion au server
- 3) Le bouton de réinitialisation du modem 4G

Le script de lecture de code est aussi présent dans les sources disponibles sur : <https://github.com/thomaslc66/Bachelor-2019>. Il gère le clavier numérique comme une matrice. Le nombre activé est celui au croisement entre la ligne et la colonne qui s'activent. Si la ligne deux et la colonne deux s'activent c'est alors le chiffre 5 qui est pressé par l'utilisateur. Le script s'occupe donc de recevoir les inputs et connaissant les colonnes et lignes il retourne ensuite le bon numéro. Ce script est tiré du site : <http://crumpspot.blogspot.com/2013/05/using-3x4-matrix-keypad-with-raspberry.html> et modifié pour les besoins du prototype. Il se base également sur une librairie python *matrix_keypad*.[57, 58]

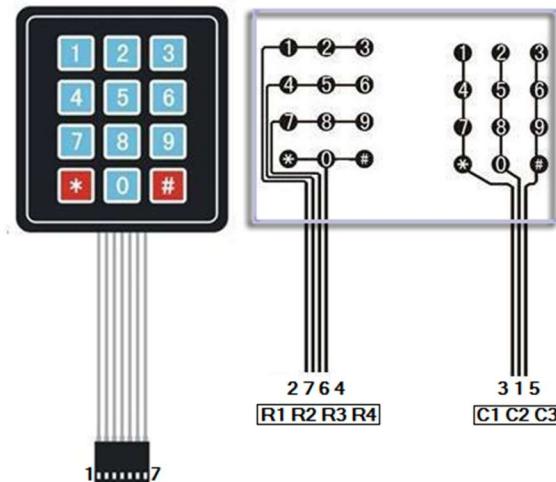


Figure 75: Schéma tiré du site <http://cliccostebelleprive.fr/>

Le Schéma électrique de la borne n° 3 se présente comme suit :

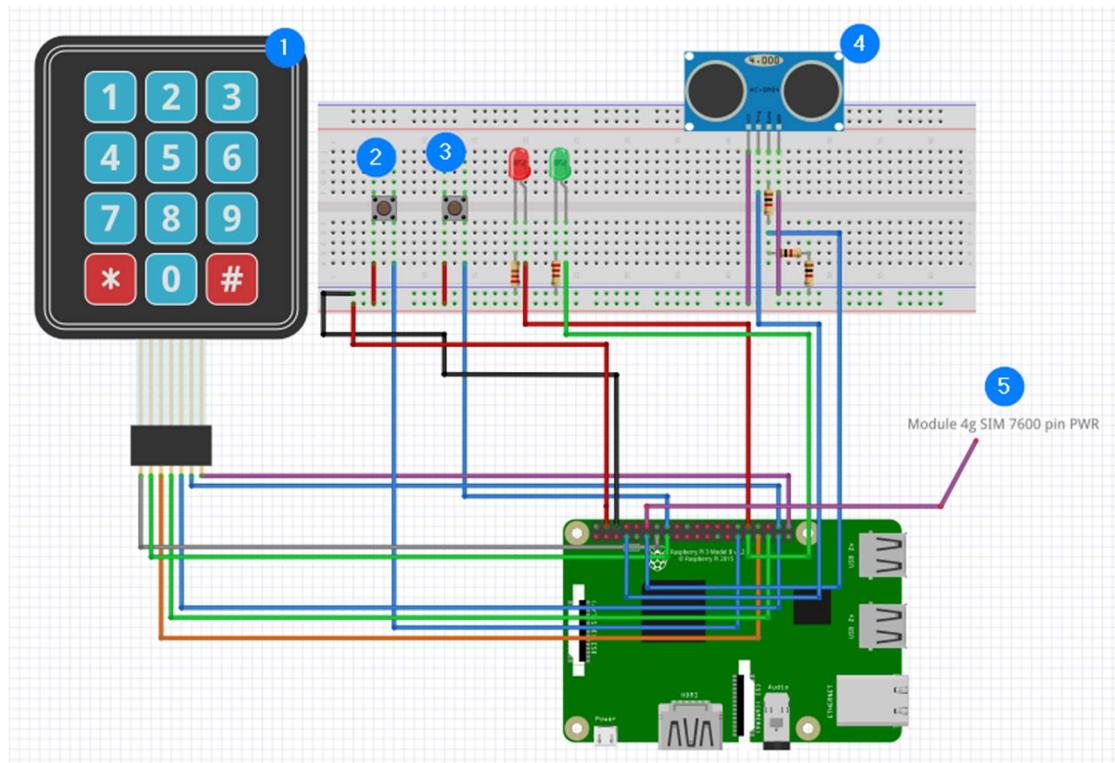


Figure 76: Schéma électrique de la borne n° 3

- 1) Clavier digital
- 2) Bouton de vérification de connexion
- 3) Bouton permettant de réinitialiser une connexion en cas de coupure du modem 4G
- 4) Capteur de distance (voir chapitre 9.6)
- 5) Lien vers le modem 4G sur le pin PWR du module.

4.7 Parcours et prise d'information sur le terrain

Afin de pouvoir mettre en place un prototype, certaines mesures et coordonnées ont été prises sur le terrain. Pour récolter ces données, le parcours sportif a été effectué à l'aide d'un traceur GPS sur un téléphone portable, en utilisant l'application GeoTracker [59], sur Android. Cette application permet d'obtenir le suivi d'un parcours sous forme de fichier « gpx ». Un fichier « gpx » est un fichier contenant une liste de latitudes et de longitudes formant un tracé complet.

```

1  <gpx version="1.0"
2  |   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3  <trk>
4  <trkseg>
5  <trkpt lat="46.7800525" lon="6.6545865"></trkpt>
6  <trkpt lat="46.7800062" lon="6.6546628"></trkpt>
7  <trkpt lat="46.78003" lon="6.6546844"></trkpt>
8  <trkpt lat="46.7802156" lon="6.6546626"></trkpt>
9  <trkpt lat="46.7805091" lon="6.654519"></trkpt>
10 <trkpt lat="46.7808554" lon="6.6543096"></trkpt>
11 <trkpt lat="46.781253" lon="6.6540756"></trkpt>
12 <trkpt lat="46.7815702" lon="6.6538706"></trkpt>
13 <trkpt lat="46.7819237" lon="6.6536809"></trkpt>
```

Figure 77: Exemple d'un fichier gpx

Plusieurs sites web permettent, non seulement la création, sur une carte, de ce genre de fichier, mais aussi la visualisation et la modification. En exportant le parcours de l'application, il a été possible d'utiliser ce fichier dans notre émulateur Android (Cf. Figure 81).

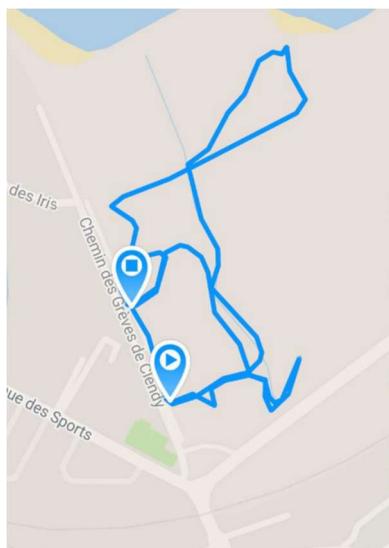


Figure 78: Résultat du tracé enregistré dans un fichier GPX

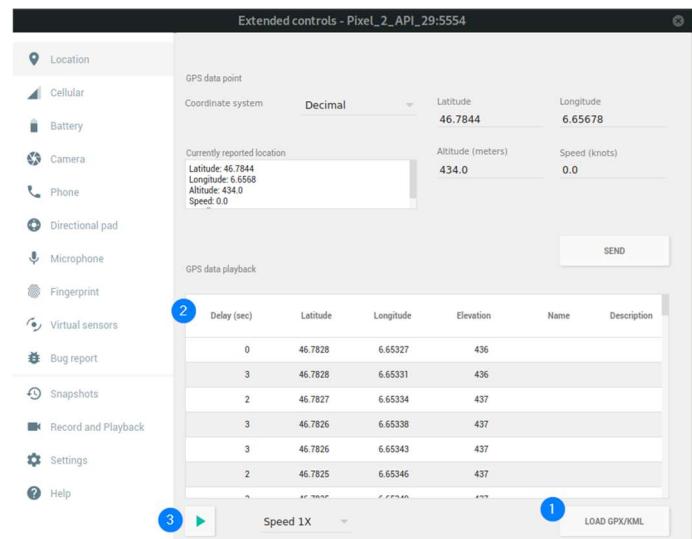


Figure 79: Importation d'un fichier gpx dans l'émulateur Android

En effet, l'émulateur Android (Cf. Figure 79), offre la possibilité d'importer des fichiers « gpx » (1). L'ensemble des points contenus dans le fichier apparaît alors sous forme de liste et il est ensuite possible de simuler les déplacements le long de ce tracé en appuyant sur le bouton « play » (3). Cette fonctionnalité rend les tests dans l'émulateur très pratiques.

L'espace de jeu est plutôt vaste et il aurait été contraignant de devoir sans cesse se déplacer afin de réaliser les tests. L'importation du fichier « gpx » a permis d'économiser beaucoup de temps.



Figure 80: Étendue du jeu

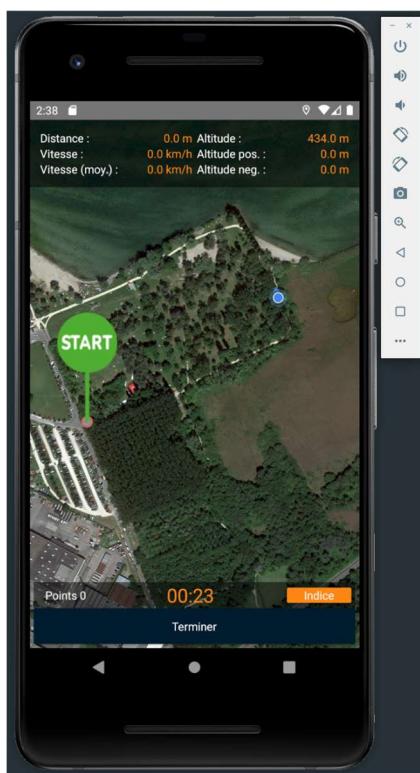


Figure 81: Emulateur Android

Le tracé du parcours sportif est plus ou moins long selon le sens dans lequel on débute la partie. Certains raccourcis existent, tout au long du parcours et l'utilisateur désireux de les emprunter pourra le faire sans problème. Cela importe peu, car l'utilisateur suivra le parcours selon les énigmes ainsi que les informations sur la carte. Ne connaissant que l'énigme suivante, il ne peut pas anticiper ses déplacements. Il lui est juste possible, lorsqu'une nouvelle zone apparaît, de prendre le chemin le plus court pour s'y rendre.

Lors des tests, certaines vidéos ont également été réalisées. Ces vidéos seront éditées, dans l'optique de réaliser un montage explicatif du jeu. Par le biais de cette vidéo, le prototype pourra être présenté lors de la défense du projet. Ce montage aura pour seconde utilisation être mise à disposition de M. Andrés Perez-Uribe pour une conférence sur le mouvement organisée par la ville d'Yverdon.

4.8 Problématiques et enjeux

C'est en développant le jeu que certaines problématiques sont apparues. Même si beaucoup ont été pensées avant, certaines limitations sont apparues causées par les technologies utilisées.

La première problématique a été l'utilisation des capteurs. Quels capteurs étaient les plus pertinents dans la création de ce prototype ? Pourquoi utiliser ce capteur plutôt qu'un autre ou encore comment intégrer ce module dans l'ensemble du jeu et du parcours ? En amont de la réalisation du travail de Bachelor et de la création du prototype, une analyse primaire a été réalisée. Cette analyse sommaire est disponible dans les annexes au chapitre 9.2. Le sujet n'étant pas clairement défini au moment de la rédaction de cette réflexion, ce document est présent à titre indicatif, permettant au lecteur de comprendre la démarche et l'analyse menant aux choix faits concernant les énigmes et le prototype final.

Si au début il était question d'intégrer des capteurs de distance ou de détection de mouvements, il est très vite apparu que cela ajouterait un ensemble de problèmes concernant la détection des faux positifs, des mouvements parasites ou les conditions météos. L'utilisation ici des détecteurs de mouvements basiques provenant de Chine, même si achetés sur des sites Suisse, ne donne pas les mêmes résultats qu'avec des modules professionnels. Les capteurs de distances sont, par exemple, moins performants en cas temps nuageux ou de vent. Il est donc difficile de demander à des composants de cette qualité d'offrir des performances et des rendus de niveau supérieur. Il a cependant été rajouté en marge du jeu un essai sur 2 bornes de ces capteurs de distance afin de démontrer une potentielle utilisation de tels capteurs. Un chapitre dans les annexes (Cf. chapitre 9.6) peut être trouvé pour plus d'informations.

La seconde problématique concerne l'éventualité du multi-joueur. Non demandé dans le cahier des charges pour le prototype, il est cependant important d'offrir une réflexion sur le sujet, étant donné que le jeu offre peu d'évolution sans cette possibilité. Comment faire alors pour mettre en place cela ? Trois propositions hypothétiques sont envisageables.

Premier cas : Les bornes physiques ne sont pas disposées sur le terrain. En effet, leur entretien coûte trop à la ville et des actes de vandalisme sont à prévoir. Notre joueur utilise donc exclusivement son téléphone mobile pour jouer. L'interaction avec le terrain est réduite, même si la résolution des énigmes demande une recherche d'indices dans les zones définies. Un mode de jeu sans l'interaction avec les bornes pourrait être développé assez facilement. Dans ce cas, l'utilisateur possédant un compte, possède de facto un identifiant unique pour ce compte qui est enregistré dans la base de données. Cet identifiant peut alors être ajouté dans chaque requête envoyée du téléphone au serveur de jeu central permettant aux multiples joueurs de ne traiter que les informations concernant leur identifiant unique de profil ; ignorant ainsi toutes les requêtes concernant d'autres joueurs et parties en cours. Peu d'investissement, aucun problème de conflit entre les requêtes, car Node-red traite les requêtes les unes après les autres et non de manière parallèle.

Second cas : Les bornes physiques sont fixées à des endroits précis, elles sont protégées de la pluie et la ville a débloqué un budget pour leur remplacement en cas de casse et de vandalisme. Plusieurs problèmes sont à relever : il faut modifier les bornes pour intégrer une plus grande batterie et un panneau solaire pour les recharger. Second problème : il faut développer un système de vérification permanent afin de connaître l'état des bornes. En effet, nous souhaitons que l'utilisateur puisse jouer quand bon lui semble. Les bornes doivent donc être monitorées pour les remplacer dès qu'une panne se présente.

Toujours dans le cadre du second cas et en prenant l'exemple de l'énigme n° 1 durant laquelle le joueur est sensé entrer le code de six couleurs trouvées aux abords des menhirs. Comment faire pour savoir qui est en train d'entrer le code ?

La détection de l'utilisateur le plus proche de la borne est indispensable. Il devrait pour cela envoyer sa position GPS au serveur central du jeu avant d'entrer son code. Mais comment déclencher cet envoi automatiquement ?

Solution n°1 : Envoi de la position du joueur à intervalles régulier. Chaque 10 secondes, par exemple, les joueurs ayant une partie en cours transmettent leurs positions au serveur central du jeu ; celui-ci calcule alors quel joueur est le plus proche de chaque borne. Une augmentation de la consommation des données mobiles, ainsi que de la batterie du joueur est à prévoir ainsi qu'une augmentation de la charge de travail demandé au serveur.

Solution n°2 : Appui sur un bouton sur la borne devant laquelle le joueur se trouve, déclenchant l'envoi d'une requête, au serveur central, récupérant la position de chaque joueur en cours de partie. L'avantage c'est qu'il y aurait l'envoi d'une seule requête et non d'une requête toutes les 10 secondes. Le problème est que le joueur ne doit pas oublier d'appuyer sur le bouton avant d'entrer le résultat de l'énigme. Tout de même, cette solution entraîne une augmentation raisonnable de la charge de travail du serveur par rapport à la solution n°1, cela étant dû à la diminution du nombre de requête. Dernier désavantage à cette solution : la problématique de la proximité. Deux joueurs sur la même énigme pourraient être les deux à équidistance de la borne et le code entré par le premier joueur pourrait être perçu comme celui entré par le second.

Dans le cadre d'un jeu de cette ampleur et comme il n'est pas attendu que cinquante joueurs soient sur la même borne en même temps. Le fait, de faire appuyer sur un bouton de validation avant l'envoi du code est envisageable. Cela augmenterai le nombre de requêtes en direction du serveur et demanderai, cependant, une machine plus puissante avec une meilleure connexion (de niveau professionnel).

Troisième cas : Dans le cadre d'un développement plus poussé du prototype actuel, la création ou l'utilisation d'appareils dédiés uniquement au jeu pourrait aussi être une solution viable. En effet, créer et utiliser des portables spécialement configurés pour le jeu permettrait une forme de contrôle supplémentaire.

Le jeu prendrait alors un aspect commercial. Chaque joueur devrait alors se rendre à la commune (ou à un stand), payer un forfait pour une partie (voir un abonnement) et ainsi récupérer un appareil (téléphones portables) préconfiguré pour le jeu. Cela permettrait de rentabiliser les appareils, d'assurer un revenu pour l'entretien de bornes ainsi que de payer des développeurs pour réaliser d'autres énigmes. De plus, le fait d'utiliser ces appareils préconfigurés permettrait une synchronisation avec les bornes. En utilisant la technologie Bluetooth (par exemple), il serait alors possible de savoir qui est en train d'introduire le code et sur quelle borne. La protection des bornes pourrait également être augmentée en verrouillant totalement leur accès (dans un coffre) par un badge RFID intégré dans l'appareil préconfiguré.

Cette dernière proposition s'insère dans une optique plus commerciale, la rapprochant d'un modèle type « Escape room ». La plus-value, de ce concept, est l'intégration du jeu dans le monde externe. Par monde externe, ici nous entendons que l'escape room ne se trouve pas dans une pièce fermée. La météo pourrait cependant influencer fortement la fréquentation, mais il n'est pas impossible de créer des énigmes uniquement solvables en temps de pluie ou lorsqu'il a neigé. Ce genre de concept s'intégrerait totalement dans une partie de paint-ball ou sur des trails de montagne afin d'offrir aux gens des alternatives aux simples randonnées.

Finalement, il est intéressant de voir que des solutions existent et permettraient une continuation du projet. La création d'un business plan et d'une étude de marché pourraient aider l'orientation future du projet.

4.9 Tests & Améliorations

La grande difficulté, de cette partie de test, reste évidemment sa dimension physique. En effet, il faut pouvoir transporter les bornes, les alimenter correctement, les disposer sur le terrain. La problématique est là, comment rendre les tests, sur le terrain, le plus pratique possible. Tous les tests virtuels, avec l'émulateur et les bornes connectées au réseau par câble, n'ont posés aucun problème. Dès la sortie sur le terrain, toutes les problématiques liées à l'environnement et à la connexion par le réseau mobile sont apparues.

Pour des raisons de proximité il n'a pas été possible d'effectuer les tests autant de fois que désiré. De plus, les tests demandent une grande organisation et le transport de beaucoup de matériel. Il est nécessaire, de prendre les trois boîtiers qui sont encombrants, avec les trois powerbanks pour alimenter les Raspberry pi, le téléphone pour l'application, l'ordinateur pour débugger et de la caméra pour filmer.

L'accès aux boîtiers étant difficile sur le terrain, il aurait été intéressant d'avoir un écran LCD portable pour les boîtiers, afin de récupérer les messages d'erreurs lors de perte de connexion et voir les erreurs directement sur le terrain. Il est en effet difficile de comprendre pourquoi un boîtier refuse de se connecter lorsqu'il n'est pas accessible par le réseau.

Durant les premiers tests certaines incohérences ont pu être observées, notamment concernant la précision des données de la localisation. L'ensemble des tests réalisés au préalable sur l'émulateur Android étaient plutôt concluants, la localisation était précise et le téléphone suivait correctement les coordonnées du fichier « gpx » récupérées du tracé enregistré (Cf. chapitre 4.7). Lors des tests sur le terrain, la première chose visible, a été le fait que le téléphone n'affichait aucun indicateur de position. [60–64]

La librairie de base, dans React-native, pour la géolocalisation présente deux défauts. Premièrement, elle ne permet pas de gérer la localisation de l'utilisateur lorsque le téléphone se verrouille. Deuxièmement, si le mode « High accuracy » a été sélectionné dans les paramètres de la librairie (Cf. Figure 82), cela ne fonctionne pas. Il faut que l'utilisateur se rende dans les paramètres de localisation de son téléphone et change le mode pour « Battery saving ». Le problème, est qu'il est difficile de vérifier dans quel mode de détection le téléphone se trouve. De plus, le mode « Battery saving » est moins précis. Finalement, pour résoudre cela, il a été choisi de changer de librairie pour utiliser « *React-native-location* »[37].

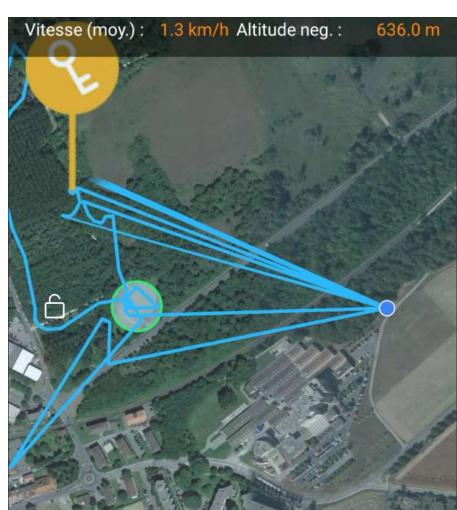


Figure 83: Déplacements parasites

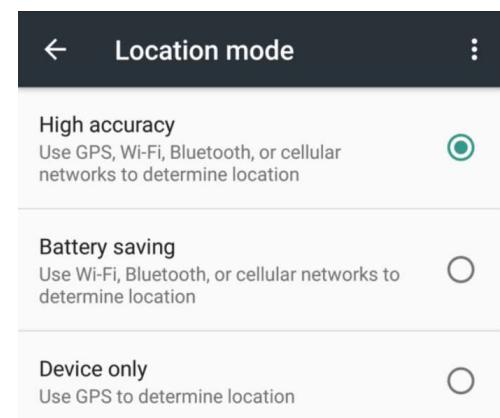


Figure 82: Mode de localisation

Cette correction et changement de librairie a permis de résoudre le problème de localisation lorsque le téléphone est verrouillé, ainsi que l'affichage de la position de notre utilisateur. Cependant, cette nouvelle librairie a montré quelques bugs également. En effet, sans savoir réellement pourquoi, d'un coup, la localisation du joueur sautait de 400 à 500 mètres avant de revenir au point réel de localisation du téléphone. Cette situation a permis lors des tests de gagner plus de 1 km sans bouger. (Cf. Figure 83).

Les paramètres de précision et de réglage de la librairie ont été modifiés et retestés, mais sans grands changements dans les résultats. Un message a même été laissé au concepteur de la librairie sur le site GitHub de cette dernière concernant le manque de précision ainsi que le réglage des divers

paramètres[65]. Les tests ont montré que, lorsque le joueur se déplace, ces problèmes ne sont pas présents. Ils arrivent lorsque l'utilisateur reste pendant quelque temps sur place. Après plusieurs tentatives et une réponse du développeur, les paramètres ont pu être affinés et lors des derniers tests aucun déplacement de ce type n'a été remarqué.

Dans un second temps, il est apparu que les premiers relevés des coordonnées GPS n'étaient pas assez précis. En effet, certaines zones, ne se trouvaient pas aux abords d'un chemin, mais carrément dans la forêt, rendant la zone inaccessible à l'utilisateur. Cela est dû au fait que les coordonnées ont été relevées avec un téléphone. Pour corriger cela, un *GPS Garmin Montana* a été utilisé, afin d'obtenir des coordonnées plus précises. Très peu de corrections à faire si ce n'est dans la base de données qui stocke les coordonnées du centre de chaque geofence.

Ces deux premiers contretemps résolus, c'est la connexion 4G qui a commencé à poser des problèmes. Chaque boîtier est contrôlé par un Raspberry Pi 3b+, qui possède son propre module 4G et permet donc de communiquer directement avec le serveur central du jeu. Cependant, ces modules 4G ne se connectent pas automatiquement et ne font aucune demande d'adresse IP de leur propre initiative. Il faut donc lancer un script au démarrage pour que la connexion s'effectue.

Le problème est survenu lors de l'utilisation des bornes. Le module 4G se branche sur les pins GPIO de la Raspberry Pi et utilise cette connexion pour s'alimenter. L'erreur est apparue, car certains pin GPIO utilisés pour le boîtier et le jeu sont en fait aussi utilisés pour le module 4G. Il est donc arrivé, sur certains boîtiers, qu'ils ne communiquent plus du tout avec la Raspberry, ni internet, ayant reçu l'information de s'éteindre, à la suite de l'activation d'un des pins GPIO utilisé pour le boîtier.

Il a donc fallu séparer le module 4G du Raspberry Pi et utiliser un port USB pour les connecter l'un à l'autre. Le second bug qui semble provenir du Raspberry Pi 3b+ [66] est que le modem a de la peine à se connecter. Il semble qu'il y ait un problème de connexion, mais surtout de port USB[67]. En effet, comme le montrent les journaux système du Raspberry pi (Cf. Figure 84), on remarque que le modem est détecté et se connecte, puis sans aucune raison, il se déconnecte perdant totalement la connexion à internet (4G) et engendrant l'impossibilité de continuer le jeu sans redémarrer le boîtier. A l'heure actuelle, ce problème n'est pas entièrement résolu, car les causes peuvent être multiples. Il semblerait cependant qu'en changeant de port USB le problème ne se pose plus. Il faudrait pouvoir monitorer cela sur des durées plus importantes afin de vérifier si cela se reproduit, car à dépendance du nombre d'ampères fourni par l'alimentation ces problèmes réapparaissent et les powerbanks n'offrent que 2.1 ampères quand la Raspberry en demande au minimum 2.5A.

```

[ 8.195355] usbserial: USB Serial support registered for generic
[ 8.240097] qmi_wwan 1-1.3:1.5: cdc-wdm0: USB WDM device
[ 8.242304] qmi_wwan 1-1.3:1.5 wwan0: register 'qmi_wwan' at usb-3f980000.usb-1.3, WWAN/QMI device, ae:a9:f2:03:97:
e5
[ 8.242535] usbcore: registered new interface driver qmi_wwan
[ 8.251322] Adding 102396k swap on /var/swap. Priority:-2 extents:1 across:102396k SSFS
[ 8.268128] usbcore: registered new interface driver_option
[ 8.268169] usbserial: USB Serial support registered for GSM modem (1-port)
[ 8.268556] option 1-1.3:1.0: GSM modem (1-port) converter detected
[ 8.273228] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB0
[ 8.273562] option 1-1.3:1.1: GSM modem (1-port) converter detected
[ 8.274190] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB1
[ 8.274555] option 1-1.3:1.2: GSM modem (1-port) converter detected
[ 8.275059] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB2
[ 8.275378] option 1-1.3:1.3: GSM modem (1-port) converter detected
[ 8.281228] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB3
[ 8.282484] option 1-1.3:1.4: GSM modem (1-port) converter detected
[ 8.283458] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB4
[ 8.644001] smsc95xx 1-1.1:1.0 eth0: hardware isn't capable of remote wakeup
[ 12.585088] Bluetooth: Core ver 2.22
[ 12.583154] NET: Registered protocol family 31
[ 12.583157] Bluetooth: HCI device and connection manager initialized
[ 12.583174] Bluetooth: HCI socket layer initialized
[ 12.583183] Bluetooth: L2CAP socket layer initialized
[ 12.583209] Bluetooth: SCO socket layer initialized
[ 12.598891] Bluetooth: HCI UART driver ver 2.3
[ 12.598903] Bluetooth: HCI UART protocol H4 registered
[ 12.599061] Bluetooth: HCI UART protocol Three-wire (H5) registered
[ 12.599166] Bluetooth: HCI UART protocol Broadcom registered
[ 12.938998] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
[ 12.939007] Bluetooth: BNEP filters: protocol multicast
[ 12.939023] Bluetooth: BNEP socket layer initialized
[ 159.896686] smsc95xx 1-1.1:1.0 eth0: link up, 100Mbps, full-duplex, lpa 0xCL1E
[ 947.377983] usb 1-1.3: USB disconnect, device number 4
[ 947.378340] option1 ttyUSB0: GSM modem (1-port) converter now disconnected from ttyUSB0
[ 947.378405] option 1-1.3:1.0: device disconnected
[ 947.381471] option1 ttyUSB1: GSM modem (1-port) converter now disconnected from ttyUSB1
[ 947.381556] option 1-1.3:1.1: device disconnected
[ 947.382161] option1 ttyUSB2: GSM modem (1-port) converter now disconnected from ttyUSB2
[ 947.382215] option 1-1.3:1.2: device disconnected
[ 947.382608] option1 ttyUSB3: GSM modem (1-port) converter now disconnected from ttyUSB3
[ 947.382645] option 1-1.3:1.3: device disconnected
[ 947.383028] option1 ttyUSB4: GSM modem (1-port) converter now disconnected from ttyUSB4
[ 947.383065] option 1-1.3:1.4: device disconnected
[ 947.383322] qmi_wwan 1-1.3:1.5 wwan0: unregister 'qmi_wwan' at usb-3f980000.usb-1.3, WWAN/QMI device
[ 947.735069] usb 1-1.3: new high-speed USB device number 5 using dwc_otg
[ 947.878362] usb 1-1.3: New USB device found, idVendor=1e0e, idProduct=9001, bcdDevice= 3.18
[ 947.878376] usb 1-1.3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 947.878386] usb 1-1.3: Product: SimTech, Incorporated
[ 947.878395] usb 1-1.3: Manufacturer: SimTech, Incorporated
[ 947.878404] usb 1-1.3: SerialNumber: 0123456789ABCDEF
[ 947.880532] option 1-1.3:1.0: GSM modem (1-port) converter detected
[ 947.880887] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB0
[ 947.881685] option 1-1.3:1.1: GSM modem (1-port) converter detected
[ 947.881997] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB1
[ 947.882830] option 1-1.3:1.2: GSM modem (1-port) converter detected
[ 947.883128] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB2
[ 947.883957] option 1-1.3:1.3: GSM modem (1-port) converter detected
[ 947.884258] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB3
[ 947.885175] option 1-1.3:1.4: GSM modem (1-port) converter detected
[ 947.886197] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB4
[ 947.896016] qmi_wwan 1-1.3:1.5: cdc-wdm0: USB WDM device
[ 947.906567] qmi_wwan 1-1.3:1.5 wwan0: register 'qmi_wwan' at usb-3f980000.usb-1.3, WWAN/QMI device, ae:a9:f2:03:97:

```

Figure 84: Problème connexion modem 4G

Dans l'attente, pour contourner ce problème, un bouton permettant de relancer le script de connexion, sans avoir à redémarrer l'ensemble des modules, a été posé sur chaque boîtier. Dans la même idée, un autre bouton a été ajouté afin de tester la connexion, au serveur, des bornes. Les autres problèmes rencontrés non pas lors des tests mais lors de la mise en service du modem 4G sont présents dans le (chapitre 4.4) du présent rapport.

Les tests des énigmes ont été concluants. Un problème de timeout de session vers le serveur de jeu est apparu, mais rien de complexe si ce n'est l'augmentation de ces délais. Les délais de communication sont corrects. Lorsque la borne est connectée au réseau internet, les informations arrivent rapidement au serveur, ainsi qu'au téléphone. Le réseau vers la plage semble bon et permet une communication entre les modules acceptable. Cependant, il apparaît de manière sporadique des déconnexions de

certains boitiers, principalement la borne n°1 qui pourtant a été dupliqué à l'identique d'une borne fonctionnelle.

La dernière problématique apparue lors des tests est la pose des supports pour le jeu. Principalement ceux aux abords de zone des menhirs, qui ont été enlevés par des passants ou du personnel de voirie. Deux des six couleurs étaient manquantes et l'énigme n'était donc pas réalisable jusqu'à la fin. Il faudrait se renseigner auprès de la commune d'Yverdon et voir s'il est possible de faire quelque chose avec leur accord puisqu'il est impossible de déplacer ces supports. Dans l'attente et pour la réalisation de la démonstration ces couleurs sont positionnées lors des tests et des prises de vidéos puis enlevées. Il en est de même pour la clef RFID de l'énigme 2 qui n'a pas été mise en place de peur de la voir disparaître.

Concernant les améliorations possibles du projet, ici se trouve une liste non exhaustive :

- Amélioration de la connectivité des modules 4G
- Implémentation du mode de jeu en solitaire (sans bornes)
- Implémentation du mode de jeu gérant les joueurs en simultané
- Finalisation du design de l'application mobile
- Ajout de fonctionnalités dans l'application mobile (modification du nom, email, etc.)
- Amélioration du calcul et de l'attribution des points en cours et fin de partie
- Création d'un classement inter-joueur
- Création d'une seconde énigme ou d'un second niveau d'énigme
- Gestion du niveau de l'utilisateur
- Finalisation des bornes et création de model fixe (impression carte électronique, soudures, etc.)
- Amélioration de la procédure de test sur le terrain

Toutes ces fonctionnalités permettraient d'arriver du prototype actuel vers un produit de démonstration correct.

5 CONCLUSION

Un projet d'une telle envergure demande beaucoup d'investissement et de temps. Il faut posséder des compétences dans de nombreux domaines différents : en développement logiciel, pour la partie applicative ; en électronique, pour la partie relative aux bornes ; en administration de système, pour configurer les servers ainsi que les Raspberry Pi ; enfin en réseau, pour tout ce qui concerne la partie communication 4G ainsi que les différentes manipulations pour ouvrir les ports et communiquer avec les bons protocoles.

Le lancement a été quelque peu ralenti, par l'ajout de projet obligatoire dans le cursus de la HEIG-VD en emploi, ainsi que par la versatilité du projet. En effet, les premiers balbutiements du projet n'ont pas permis de trouver la ligne à suivre pour la réalisation du prototype. Il s'agissait davantage de tests de différentes technologies et appareils électroniques afin de savoir si ces derniers avaient un avenir au sein du prototype, puis de la recherche d'un fil rouge. Cette manière de faire a permis, certes, de focaliser le prototypage, mais a augmenté la charge de travail et diminué le résultat en qualité. Il est vrai que toutes les possibilités technologiques n'ont pas été abordées, ni testées dans ce projet, tant leur nombre est important. Cependant, beaucoup ont été rejetées par leur incompatibilité à répondre aux demandes du cahier des charges (Cf. Annexes chapitre 9.1).

Les technologies testées, mais écartées, du projet sont disponibles dans les annexes du rapport avec des explications concernant les choix de non-intégration de ces dernières. Toutes n'ont pas reçu la même attention et de ce fait certains retours sont plus complet que d'autres.

Le travail, une fois clairement défini, a permis la réalisation d'une démonstration de jeu. Il n'est pas exempt de lacunes, mais il est plus avancé dans sa conception qu'un prototype de base. En effet, il comprend un exemple complet, avec les bornes, les audios, les zones et la création des énigmes. Le tout dans une histoire en lien avec la zone sur laquelle le jeu se déroule.

Par rapport au concept imaginé au chapitre 1.3, l'application offre la possibilité de jouer uniquement avec les bornes. Après discussion avec le professeur responsable, il a été choisi de se concentrer sur un prototype. Le mode de jeu « solitaire » imaginé au début uniquement avec le téléphone n'a pas été implémenté.

Le prototype présenté dans ce rapport répond au cahier des charges, présent en annexe. Les autres attentes du travail sont satisfaites par les tests technologiques, la réflexion sur le projet et son futur ainsi que le pistes et idées apportées au lecteur. Le projet et son résultat, invitent à continuer et à perfectionner le concept. Cependant, ce travail devrait être fait, non pas par une seule personne, mais au moins par un groupe de deux. L'application à elle seule aurait pu être confiée à une personne en informatique logiciel et la configuration des bornes aurait pu, elle par exemple être attribuée à une élève en électronique. La charge de travail était suffisante et aurait permis à chaque élève de rendre deux produits de qualité supérieure. Les tests auraient aussi été facilités en ce qui concerne le transport du matériel et des bornes.

En ce qui concerne la suite, les possibilités d'énigmes et d'amélioration sont nombreuses. Enigmes à thèmes, énigmes découvertes, énigmes sportives, énigmes pour enfants, énigmes selon le temps pluie, neige, soleil etc...sont imaginables et la finalisation complète des bornes, par un câblage et des soudures définitives, donnerait une dimension plus concrète du concept et pourrait totalement faire partie d'un programme de start-up.

A titre personnel :

J'ai trouvé le travail intéressant, la réalisation du prototype m'a vraiment plu et j'ai énormément appris, surtout en électronique, durant ce projet. Cependant, j'ai eu de la peine à démarrer. Le projet était flou au départ, et j'ai dû investir beaucoup de temps afin de délimiter mon terrain d'action.

J'ai ressenti beaucoup de frustration, lorsque certaines librairies choisies n'étaient pas au point, ou avaient des comportements inadéquats lors des tests sur le terrain. De même, lorsque, sans raison, les modules 4G ont cessé de fonctionner à cause de déconnexion des ports usb, sans que je ne sache pourquoi, ni quoi faire pour résoudre le problème. La frustration provient du fait que j'ai investi beaucoup de temps, et qu'au final, certains problèmes sont liés à l'utilisation de librairies ou de matériels tiers qui empêchent un fonctionnement correct et constant du prototype. Certes, il s'agit d'un prototype, mais ayant été pris dans le développement du jeu, il me tenait à cœur de rendre quelque chose d'abouti.

Finalement, j'ai beaucoup apprécié, la confiance et la liberté qui m'ont été offertes. Cela m'a permis d'implémenter quelque chose dont je suis plutôt fier. Même si le prototype n'est pas parfait, je trouve qu'il est déjà très complet pour un simple prototype. J'espère que cela donnera une bonne ligne directrice dans le cas d'une continuation.

6 REMERCIEMENTS

Beaucoup de personnes m'ont aidé dans la réalisation de ce projet. Par leurs idées, les discussions et le partage des moments de réflexion, en passant par la réalisation des audios pour le jeu, les énigmes ou encore les aspects graphiques de l'application. Tout ce travail qui n'est pas quantifiable, mais qui donne au résultat sa profondeur.

Je tiens donc à remercier :

- Ma femme pour avoir réalisé la voix-off et avoir participé à la réflexion sur le jeu, ainsi que les énigmes.
- Mon père, pour son temps, la voix dans le jeu du noisetier ainsi que pour l'aide dans les énigmes et leur implémentation sur le terrain.
- La graphiste de mon entreprise (Practeo SA), pour avoir réalisé les logos de connexion au serveur ainsi que les markers qui s'affichent sur la carte du jeu.

Pour finir, je tiens à remercier, de manière générale, pour la liberté ainsi que la confiance qui m'ont été accordées pour la réalisation de ce projet.

7 GLOSSAIRE

A

Android

Système d'exploitation pour téléphone mobile 15, 22, 63, 67, 82, 83, 84

Android studio

Nom de l'éditeur logiciel pour développer sur Android 15, 22

B

Beacons

Petite balise utilisant le Bluetooth pour transmettre des informations 7, 82, 83, 84, 85

Broker

Service permettant l'envoi de message avec le protocole MQTT (cf. MQTT) 19, 20, 41, 44

F

Flow

un flow est un moyen de programmer et relier un ensemble de nœuds (cf. nœud) 17, 18, 46

Framework

Ensemble de composant logiciel formant la base ou une brique d'un projet plus grand 22

G

Gamification

action de gamifier (cf. gamifier) 5, 6, 7, 9, 10, 26, 27

Gamifier

rendre ludique, en y apportant les aspects du jeu 4

Géocacheur

Un géocacheur est une personne jouant au géocaching (cf. géocaching) 5

Géocaching

Le géocaching est une chasse au trésor grandeur nature 6

Geofence

Barrière virtuelle, en forme de cercle. Affichée sur la carte 10, 32, 46, 68

GPIO

port d'entrée/sortie général ou General PIN Input/OutPut 12, 18, 53, 68

I

iOs

Système d'exploitation pour téléphone Apple 22, 83

IoT

Internet of Things ou internet des objets 17

J

Java

Langage de programmation 15, 22

M

MQTT

Protocol de communication (Message Queuing Telemetry Transport) 19, 20, 82

MySQL

Base de donnée relationnelle, utilisant le langage SQL dans ces requêtes 23, 30

N

Node Js

Plateforme logicielle (libre) orientée réseau conçue en javascript 15, 17, 22

Nœud

Unité de programmation dans node-red. Facilite les interactions 17, 18, 19, 20, 53

| | | |
|--------------|---|--|
| NoSql | Type de base de données qui se distingue du standard relationnel et du langage SQL | 30 |
| O | | |
| Objective-C | Nom du langage utilisé pour code des application pour iPhone..... | 22 |
| P | | |
| Powerbank | Batterie de secours permettant d'alimenter ou charger des appareils portables | 12, 14, 43, 67, 68 |
| R | | |
| Raspberry Pi | Très petit ordinateur. Utile pour son format portable. | 12, 16, 18, 42, 68 |
| Raspbian | Système d'exploitation linux spécifique aux Raspberry pi | 41 |
| Raspian | Système d'exploitation linux spécifique aux Raspberry pi | 42 |
| RFID | Radio Frequency Frequency Identification (méthode de radio-identification) | 7, 13, 56, 57 |
| S | | |
| Scalabilité | La scalabilité permet d'augmenter les ressources machines en cas de forte sollicitation du service (application)..... | 23 |
| Scalable | voir scalabilité..... | 30 |
| Session | Unité utilisée pour mesurer un entraînement ou une partie du jeu | 5, 9, 23, 28, 29, 32, 34, 36, 37, 38, 69 |
| T | | |
| Terminal | Équivalent de la console Windows mais sous linux..... | 17 |
| U | | |
| Ubuntu | Système d'exploitation basé sur un noyau linux | 15, 44 |
| Uid | Uid est le diminutif de User Identification et représente l'identifiant unique dans la base de données | 31, 83 |
| V | | |
| Vue | Une vue sur android ou mobile, correspond à une fenêtre ou page de l'application..... | 8, 22, 24, 27, 28, 29, 34, 35, 36, 37 |
| Vue native | Est une vue (cf. vues) développée avec le langage natif du mobile et non un substitut | 22 |
| W | | |
| WireFrame | Tiré de l'anglais, correspond à la création d'une maquette | 27 |
| X | | |
| XCode | Nom de l'éditeur pour développer des applications pour les téléphones Apple | 22 |

8 BIBLIOGRAPHIE

1. Commune d'Yverdon-les-Bains - Gare-Lac - Site officiel de la Commune. <https://www.yverdon-les-bains.ch/grands-projets/grands-projets-urbains/gare-lac#c719>. Accessed 28 Jul 2019
2. Collaborate & Create Amazing Graphic Design for Free. In: Canva. <https://www.canva.com>. Accessed 27 Jul 2019
3. (2019) Node-red. <https://nodered.org/>. Accessed 23 Jul 2019
4. (2019) A visual tool for wiring the Internet of Things. Contribute to Node-red/Node-red development by creating an account on GitHub. Node-red
5. Library - Node-red. https://flows.nodered.org/?sort=downloads&type=node&num_pages=1. Accessed 29 Jul 2019
6. MQTT. <http://mqtt.org/>. Accessed 29 Jul 2019
7. (2018) Eclipse Mosquitto. In: Eclipse Mosquitto. <https://mosquitto.org/>. Accessed 29 Jul 2019
8. React Native · A framework for building native apps using React. <https://facebook.github.io/react-native/>. Accessed 27 Jul 2019
9. Getting Started · React Native. <https://facebook.github.io/react-native/>. Accessed 3 Aug 2019
10. React – Une bibliothèque JavaScript pour créer des interfaces utilisateurs. <https://fr.reactjs.org/>. Accessed 29 Jul 2019
11. Firebase. In: Firebase. <https://firebase.google.com/>. Accessed 30 Jul 2019
12. @firebase/firestore: Firestore (6.0.4): INTERNAL UNHANDLED ERROR: configureNetworkMonitoring · Issue #1824 · firebase/firebase-js-sdk. In: GitHub. <https://github.com/firebase/firebase-js-sdk/issues/1824>. Accessed 5 Aug 2019
13. Studio B (2018) Kickstart your React Native app with Firebase — Cloud Firestore (Tutorial #1). In: Medium. <https://blog.benestudio.co/kickstart-your-react-native-app-with-firebase-cloud-firestore-tutorial-1-207a8f64e4f>. Accessed 8 Aug 2019
14. Get data with Cloud Firestore. In: Firebase. <https://firebase.google.com/docs/firestore/query-data/get-data>. Accessed 5 Aug 2019
15. (2019) Kickstart your React Native app with Firebase tutorial: benestudio/fakestagram. Bene Studio
16. QuerySnapshot | JavaScript SDK | Firebase. <https://firebase.google.com/docs/reference/js/firebase.firestore.QuerySnapshot>. Accessed 5 Aug 2019
17. Retrieving Data | Firebase Realtime Database. In: Firebase. <https://firebase.google.com/docs/database/admin/retrieve-data>. Accessed 5 Aug 2019
18. Get realtime updates with Cloud Firestore | Firebase. <https://firebase.google.com/docs/firestore/query-data/listen>. Accessed 8 Aug 2019
19. Getting Started with Redux · Redux. <https://redux.js.org/>. Accessed 23 Sep 2019
20. Produle MockFlow - Wireframe Tools, Prototyping Tools, UI Mockups, UX Suite. <https://mockflow.com>. Accessed 22 Sep 2019

21. (2019) NoSQL. Wikipédia
22. Is NoSQL faster than SQL? - Quora. <https://www.quora.com/Is-NoSQL-faster-than-SQL>. Accessed 12 Sep 2019
23. React Native Firebase - Simple Firebase integration for React Native. In: React Native Firebase. <https://rnfirebase.io/docs/v5.x.x/auth/getting-started>. Accessed 6 Aug 2019
24. (2019)  React Native Firebase documentation source files: please use the website to browse the docs. - invertase/react-native-firebase-docs. Invertase
25. Authenticate with Firebase using Password-Based Accounts using Javascript. In: Firebase. <https://firebase.google.com/docs/auth/web/password-auth>. Accessed 3 Aug 2019
26. How to Connect Firebase Users to their Data - 3 Methods - YouTube. <https://www.youtube.com/watch?v=2ciHixbc4HE>. Accessed 5 Aug 2019
27. Content hides Input when keyboard visible (i.e. not keyboard-aware-scroll-view) · Issue #1418 · GeekyAnts/NativeBase. In: GitHub. <https://github.com/GeekyAnts/NativeBase/issues/1418>. Accessed 3 Aug 2019
28. scrollview - How can i style the scroll indicator in react native scrollbar. In: Stack Overflow. <https://stackoverflow.com/questions/50754284/how-can-i-style-the-scroll-indicator-in-react-native-scrollbar>. Accessed 9 Aug 2019
29. React Native: How to use FlatList | What did I learn. /2018/12/09/react-native-how-to-use-flatlist.html. Accessed 8 Aug 2019
30. ListView is deprecated and will be removed in a future release. · Issue #2398 · GeekyAnts/NativeBase. In: GitHub. <https://github.com/GeekyAnts/NativeBase/issues/2398>. Accessed 8 Aug 2019
31. Carli S (2017) React Native Basics: How to Use the ListView Component. In: Medium. <https://medium.com/differential/react-native-basics-how-to-use-the-listview-component-a0ec44cf1fe8>. Accessed 7 Aug 2019
32. Navigating Between Screens · React Native. <https://facebook.github.io/react-native/>. Accessed 3 Aug 2019
33. BackHandler · React Native. <https://facebook.github.io/react-native/>. Accessed 25 Aug 2019
34. reactjs - Disable back button in react navigation. In: Stack Overflow. <https://stackoverflow.com/questions/42831685/disable-back-button-in-react-navigation>. Accessed 3 Aug 2019
35. Ernst S (2019) Timer plugin for Moment.js that allows creation of setInterval and setTimeout-like timers.: SeverinDK/moment-timer
36. Machado G (2016) Timers in React Apps with Redux. In: Medium. <https://medium.com/@machadogj/timers-in-react-with-redux-apps-9a5a722162e8>. Accessed 26 Aug 2019
37. Park T (2019) timfpark/react-native-location
38. Lam J (2019) A React Native Popup Dialog. Highly customizable. Support multi dialogs & Support custom animation. For IOS & Android.: jacklam718/react-native-popup-dialog
39. kernel: Bump to 4.19.42 · Hexxeh/rpi-firmware@18ef673. In: GitHub. <https://github.com/Hexxeh/rpi-firmware/commit/18ef6736e42023ea4e411d36a60d3eed18fdcccf>. Accessed 3 Sep 2019

40. Accessing internet with SIM7600 4G Modem using MBIM interface in Pocketbeagle/Beaglebone. In: Elementz Engineers Guild Private Limited. <https://www.elementzonline.com/blog/Accessing-internet-with-SIM7600-4G-Modem-using-MBIM-interface-in-Pocketbeagle-Beaglebone>. Accessed 10 Aug 2019
41. 4G Hat - Raspberry Pi Forums.
<https://www.raspberrypi.org/forums/viewtopic.php?f=36&t=224355&sid=a1004814d97f2cd391f54b93da6281c5>. Accessed 3 Sep 2019
42. penguin2716/qmi_setup. In: GitHub. https://github.com/penguin2716/qmi_setup. Accessed 3 Sep 2019
43. Techship - FAQ - How to step by step set up a data connection over QMI interface using qmicli and in-kernel driver qmi_wwan in Linux? <https://techship.com/faq/how-to-step-by-step-set-up-a-data-connection-over-qmi-interface-using-qmicli-and-in-kernel-driver-qmi-wwan-in-linux/>. Accessed 16 Sep 2019
44. My DHCP lease problem - Raspberry Pi Forums.
<https://www.raspberrypi.org/forums/viewtopic.php?t=159950>. Accessed 17 Sep 2019
45. (2012) An introduction to libqmi. In: SIGQUIT. <https://sigquit.wordpress.com/2012/08/20/an-introduction-to-libqmi/>. Accessed 19 Sep 2019
46. error: couldn't create client for the "dms" service: QMI protocol error (3): "Internal" · Issue #1 · borovsky/libqmi. In: GitHub. <https://github.com/borovsky/libqmi/issues/1>. Accessed 17 Sep 2019
47. Harvey T (2017) MC7455: inconsistent QMI protocol error (14): "CallFailed" - ipv4-only-allowed
48. Williams D (2013) MC7750 libqmi
49. networking - qmicli Cannot set expected data format. In: Unix & Linux Stack Exchange.
<https://unix.stackexchange.com/questions/397863/qmicli-cannot-set-expected-data-format>. Accessed 12 Aug 2019
50. networking - libqmi - Transaction timed out. In: Unix & Linux Stack Exchange.
<https://unix.stackexchange.com/questions/360202/libqmi-transaction-timed-out>. Accessed 17 Sep 2019
51. networking - Loading kernel module *qmi_wwan* before *option* to get /dev/cdc-wdm0 automatically. In: Raspberry Pi Stack Exchange. <https://raspberrypi.stackexchange.com/questions/68978/loading-kernel-module-qmi-wwan-before-option-to-get-dev-cdc-wdm0-automatica>. Accessed 19 Sep 2019
52. Gothe M (2015) Not getting IP Address after wds-start-network
53. Morgado A (2015) QMI protocol error (26): "NoEffect" on stopping
54. qmicli - Control QMI devices - Linux Man Pages (1).
<http://www.systutorials.com/docs/linux/man/docs/linux/man/1-qmicli/>. Accessed 17 Sep 2019
55. utils/qmi-network.in - chromiumos/third_party/libqmi - Git at Google.
https://chromium.googlesource.com/chromiumos/third_party/libqmi/+/master/utils/qmi-network.in. Accessed 17 Sep 2019
56. Fritzing. <http://fritzing.org/>. Accessed 27 Jul 2019
57. Unknown (2013) Crump Projects: Python Matrix Keypad Package. In: Crump Projects.
<http://crumpspot.blogspot.com/2013/08/python-matrix-keypad-package.html>. Accessed 30 Sep 2019
58. Crumpacker C matrix_keypad: Matrix Keypad code for use with Raspberry Pi

59. Geo Tracker - GPS tracker – Applications sur Google Play.
https://play.google.com/store/apps/details?id=com.ilyabogdanovich.geotracker&hl=fr_CH. Accessed 27 Jul 2019
60. geolocation - Can I check devices location is open or close on React Native. In: Stack Overflow.
<https://stackoverflow.com/questions/38471794/can-i-check-devices-location-is-open-or-close-on-react-native>. Accessed 25 Aug 2019
61. Geolocation “enableHighAccuracy” on Android always times out · Issue #7495 · facebook/react-native. In: GitHub. <https://github.com/facebook/react-native/issues/7495>. Accessed 14 Sep 2019
62. Geolocation timed out, code 3 · Issue #12996 · facebook/react-native. In: GitHub.
<https://github.com/facebook/react-native/issues/12996>. Accessed 14 Sep 2019
63. Geolocation · React Native. <https://facebook.github.io/react-native/>. Accessed 25 Aug 2019
64. navigator.geolocation.getCurrentPosition not showing error on Android · Issue #287 · react-community/create-react-native-app. In: GitHub. <https://github.com/react-community/create-react-native-app/issues/287>. Accessed 14 Sep 2019
65. Interval, fastestInterval and maxWaitTime not working. · Issue #90 · timfpark/react-native-location. In: GitHub. <https://github.com/timfpark/react-native-location/issues/90>. Accessed 14 Sep 2019
66. Raspberry Pi 2 USB disconnect · Issue #897 · raspberrypi/linux. In: GitHub.
<https://github.com/raspberrypi/linux/issues/897>. Accessed 13 Sep 2019
67. USB Disconnect - Raspberry Pi Forums. <https://www.raspberrypi.org/forums/viewtopic.php?t=32165>. Accessed 13 Sep 2019
68. Raspberry Pi 3 bluetooth connect and disconnect · Issue #1092 · raspberrypi/firmware. In: GitHub.
<https://github.com/raspberrypi/firmware/issues/1092>. Accessed 27 Jul 2019
69. Fix Raspberry Pi 3 Bluetooth Issues • Pi Supply Maker Zone. In: Pi Supply. <https://learn.pi-supply.com/make/fix-raspberry-pi-3-bluetooth-issues/>. Accessed 27 Jul 2019
70. Bluetooth LE and Node-red with MQTT - Espruino. <https://www.espruino.com/BLE+Node-red>. Accessed 23 Jul 2019
71. Intro to Estimote APIs - Estimote Developer. <https://developer.estimote.com/>. Accessed 27 Jul 2019
72. (2019) A reference application for the Android Beacon Library: AltBeacon/android-beacon-library-reference. AltBeacon
73. Bridoux N (2019) A simple android iBeacon, AltBeacon, Eddystone and RuuviTag beacon scanner: Bridouille/android-beacon-scanner
74. HC-SR501 PIR Motion Detector - With Arduino & Raspberry Pi
75. timfpark/react-native-location. In: GitHub. <https://github.com/timfpark/react-native-location>. Accessed 14 Sep 2019

9 ANNEXES

9.1 Cahier des charges

9.1.1 Contexte et définition du projet

Dans le cadre du travail de Bachelor et d'après la demande de la ville d'Yverdon le but et d'étudier la mise en place d'un système prototypé sur un parcours mesuré de type parcours Vita, dans la zone indiquée par la commune d'Yverdon à savoir le site de Clenly, proche de la HEIG-VD.

9.1.2 Objectif du projet

Le travail de Bachelor « Gamification du parcours Vita de la ville d'Yverdon » aura pour but la proposition, après études des meilleures options, d'une version gamifiée de l'activité physique pouvant être pratiquée sur le parcours en exploitant au mieux le prototype développé précédemment.

9.1.3 Périmètre du projet

Réaliser un prototype d'application exploitant les données de l'activité physique de l'utilisateur et la communication avec les bornes externes. Par exemple, imaginer des « challenges » de vitesse où l'utilisateur doit aller d'une borne à l'autre dans un temps donné ou un certain nombre de fois, voir le plus rapidement possible. Explorer divers mécanismes de feedback, via le smartphone et via les bornes (p.ex. indicateurs lumineux ou sonores).

Réaliser le prototype d'une application Android capable de donner la localisation de l'utilisateur, sa vitesse de déplacement ainsi que d'autres variables telles que la durée écoulée de l'activité physique, distance parcourue. Faire en sorte que l'application puisse communiquer le plus rapidement possible avec les éléments extérieurs positionnés sur le parcourt. Exemple : des bornes extérieures (p.ex. modules Arduino ou Raspberry-pi, Estimote ou autre borne disponible sur le marché).

9.1.4 Description fonctionnelle des besoins

L'utilisateur doit pouvoir utiliser l'application le plus dynamiquement possible. Le but n'est pas que le sportif tienne son portable dans les mains, mais qu'il reçoive un feedback par exemple de manière audio ou sonore. Le prototype pourra être revisité, si le temps le permet, afin de pouvoir avoir un feedback en direct, sans forcément utiliser le téléphone portable. Le but dans un premier temps et de fournir un prototype permettant d'améliorer l'expérience de l'utilisateur et de rendre le parcours plus ludique qu'une simple course à pied. Le sportif assidu pourra alors suivre sa progression s'il passe plusieurs fois par le parcours

9.1.5 Enveloppe budgétaire

Le matériel est fourni par la HEIG, mais s'agissant d'une phase de prototypage et de test, un seul prototype sera mis en place afin de pouvoir conduire les tests. Il n'est pas question ici d'une production à grande échelle, du moins pour le moment.

9.1.6 Délais de réalisation

Les délais sont ceux fourni par la HEIG-VD Yverdon. Dans sa deuxième version, afin de pouvoir respecter mes responsabilités professionnelles tout en assurant la charge de travail demandée par le travail de Bachelor.

9.2 Analyse primaire et réflexion concernant la gamification du parcours sportif de la ville d'Yverdon.

Dans ce premier document, nous allons mettre en avant les différents points, tant positifs que négatifs des différentes possibilités listées ci-dessous ainsi que les fonctionnalités possibles. Cela dans le but de voir l'étendue du champ d'application et de mettre en évidence certaines problématiques qui pourraient se présenter lors du développement ou de la phase de mise en service.

De ce fait trois possibilités s'offrent à nous :

- 1) L'application mobile
- 2) Les bornes iBeacon ou NFC
- 3) Les capteurs de mouvements

9.2.1 Application Mobile :

Liste des possibles fonctions pour une application mobile :

- Proposer un entraînement de renforcement musculaire pendant le parcours sportif (cardio)
- Guidage du coureur dans les exercices par fichier audio
- Proposition d'exercices selon la difficulté choisie ou la progression du coureur
- Attribution d'un nombre de points par exercice effectué, permettant d'assurer une progression sportive
- Proposition d'exercices durant la course ainsi qu'aux différents postes présents sur le parcours.
- Calcul du temps et du parcours effectué grâce au GPS et code QR ?
- Présentation vidéo des exercices selon la position GPS du coureur et la disposition du parcourt.
- Utilisation des capteurs du téléphone pour compter les exercices effectués.
- Choix de l'objectif selon la personne (Force, Rapidité, Endurance... etc)

Avantages :

- Tout le monde possède un smartphone de nos jours
- Beaucoup de capteurs sont déjà présents dans le téléphone
- La localisation GPS permet de positionner le coureur de manière plutôt précise
- L'aide audio et vidéo donne des idées d'exercices aux coureurs lorsqu'ils s'arrêtent aux différents postes
- Le suivi et les statistiques du coureur permettent une visualisation de la progression.
- L'application peut-être facilement mise à jour à distance
- Permettre de sauver le meilleur résultat pour chaque poste et de créer une forme de challenge entre les coureurs
- Si dans un premier temps trop coûteux de développer l'application avec le gps et les capteurs ; on peut s'appuyer sur les panneaux physiques étant sur place au parcours sportif avec QR Code

Inconvénients :

- L'application n'a pas besoin du parcours sportif pour exister, elle serait alors développée uniquement pour ce parcourt. (Coût de développement vs nombre d'utilisateurs)
- Le développement final comprendra une app pour Android et une app pour iPhone
- La course avec son téléphone n'est pas forcément du goût de tout le monde
- Les gens préfèrent courir avec de la musique plutôt que des explications audios
- Le téléphone est souvent dans la poche ou attaché au bras et les interactions avec l'appareil demandent au coureur de s'arrêter et déverrouiller son appareil à chaque fois

Importance l'application ?

Dans le cas de l'application, on peut se demander où se trouve la gamification du parcours sportif. Car c'est le parcours qui doit être gamifiée. Créer une application mobile n'aurait de sens que si elle est liée au parcours sportif de la ville d'Yverdon. Il y aurait alors une forte dépendance entre le parcours en question et l'application. On pourrait bien sûr imaginer gamifier d'autre parcours sportifs par la suite, mais il faudrait alors se rendre à chaque fois sur place pour prendre les coordonnées GPS afin de pouvoir déclencher les exercices aux bons moments et devant les bons postes présents sur le parcours.

9.2.2 Les Bornes iBeacons ou NFC

Les bornes doivent être liés avec une application mobile, on retrouve donc l'ensemble des fonctionnalités et des avantages/inconvénients d'une application mobile, mais on ajoute les suivants :

Liste des fonctionnalités en sus :

- Forcer le coureur à passer devant certaines bornes comme point de contrôle ou lors d'exercices. (Sprint entre 2 bornes ou aller-retour explosifs... etc)
- Création d'exercices spécifiques ou déclenchement selon la distance entre le coureur et les bornes
- Motiver le coureur à s'arrêter aux différents postes (si le coureur passe devant une borne mais ne s'arrête pas)
- Changement du rythme de course entre deux bornes ou amélioration du temps entre deux bornes (en demandant au coureur de courir plus vite)

Avantages :

- Les bornes offrent des possibilités d'exercices supplémentaires sur le parcours
- Elles offrent un moyen de contrôle du passage du coureur additionnel
- Elles permettent le calcul du temps de course entre deux bornes

Inconvénients :

- Les bornes doivent être installées
- Possibilité de vandalisme sur les bornes
- Demande un entretien physique (piles, batteries, usure)
- Gestion dans l'application d'une panne quelconque. (Que se passe-t-il si l'utilisateur doit passer devant une borne, mais qu'elle n'est plus en fonction?).

9.2.3 Les capteurs de mouvements

Les capteurs de mouvements offrent la possibilité d'être utilisés sans téléphone portable, mais demandent de plus gros moyens de maintenance et de développement.

Fonctionnalités :

- Exercice de rapidité avec les mains en devant cacher les capteurs
- Exercice de déplacement en devant cacher les capteurs avec les pieds
- Exercice de saut en devant sauter le plus haut et cacher le capteur le plus haut possible
- Sprint entre deux capteurs
- Endurance en passant devant les différents capteurs présents le long du parcours
- Jeux par équipe en devant prendre le plus grand nombre de capteurs (chaque équipe à une couleur)

Avantages :

- Concept de gamification plus présent
- Possibilité de jouer sans être dépendant du téléphone
- Publicité touché plus large (pas besoin de télécharger une application)

Inconvénients :

- Investissement conséquent au niveau des infrastructures
- Achat de matériel
- Maintien du matériel en état
- Comment vérifier la panne de certains capteurs
- Comment centraliser les informations
- Batterie et durée de vie des capteurs
- Problème de détection parasite

9.3 Communication Bluetooth

Durant la phase de développement, la technologie utilisée, Node-red, demandait peu de temps de prise en main et permettait, grâce au protocole MQTT, de communiquer grâce au réseau. Cependant, utiliser le Bluetooth pour communiquer avec le téléphone puis utiliser le téléphone pour communiquer ensuite avec le serveur a rajouté une couche de complexité qui n'était pas prévue lors de la première partie de développement. Plusieurs jours ont été consacrés à l'exploration et aux recherches pour trouver un moyen de communiquer avec le téléphone par Bluetooth. La carte Raspberry acceptait la connexion avec le téléphone, mais déconnectait immédiatement le lien. Quelques solutions trouvées sur internet ont été testées, mais aucune n'a donné de résultats satisfaisants. [68, 69]. Cependant, une dernière solution avec un autre module Espruino n'a pas été testé, par manque de temps mais aussi parce que le module n'était pas disponible.[70]

Trouver un moyen de communiquer avec les boîtiers Raspberry permettrait d'ôter la couche de complexité et de dépendance au réseau 4G ainsi qu'aux modem externes. Cela demande cependant un temps supplémentaire pour l'implémentation de cette solution. En effet, Bluetooth demande la validation des deux appareils qui souhaitent communiquer entre eux, souvent grâce à un code partagé. Il faudrait alors permettre aux téléphones de se connecter sans vérification aux bornes du jeu.

9.4 Beacon Estimote

9.4.1 Procédure

- Lecture et formation sur la technologie Beacon et l'api Estimote
- Initialisation des Beacons
- Mise en service application et Android Studio
- Tests, expériences et les résultats
- Conclusions

9.4.2 Lecture et formation sur la technologie Beacon et Estimote

Estimote [71] est une entreprise créée par trois ingénieurs passionnés de mobile et de capteurs avec la volonté de créer un lien entre l'environnement physique et le téléphone, augmentant ainsi l'expérience utilisateur. Cette compagnie créée en 2012 possède trois succursales une à San Francisco, une à New York et une à Krakow en Pologne. Elle fournit sept types de Beacon différents et assiste plus de 60'000 clients dans le monde.

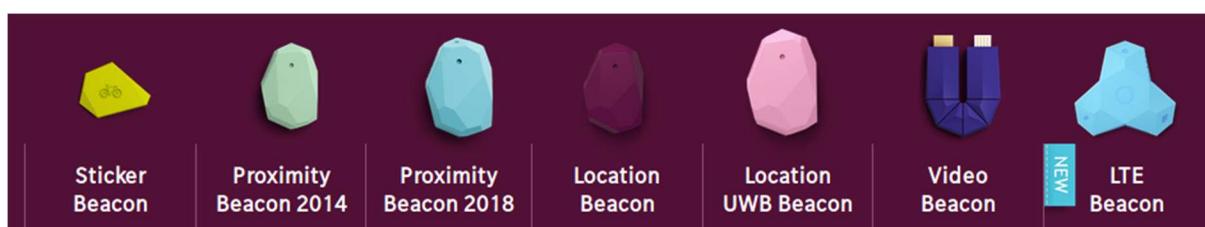


Figure 85 : Les différents types d'Estimote Beacon

Dans le cadre de ce projet les beacons utilisés pour cette phase de test sont des « Beacons Proximity » de 2014. Ils ne disposent donc pas de la dernière version du sdk Estimote, mais uniquement du Android-SDK comme le montre l'encart ci-dessous, tiré du site : <https://developer.estimote.com/android/tutorial/part-1-setting-up/>



Figure 86 : Note pour les utilisateurs des anciennes versions de Beacon

Il faut savoir que plusieurs types de paquets Bluetooth existent : les deux principaux sont iBeacon et Eddystone.

Eddystone est un protocole de communication open-source développé par Google en juillet 2015. Un Beacon avec Eddystone diffuse simplement en Bluetooth un signal avec des paquets de données à l'intérieur. Ce signal est alors détectable par un mobile ou autre équipement Bluetooth. Un paquet Eddystone contient quatre parties.

- Eddystone-UID: est un id unique.
- Eddystone-URL: est une URL au format compressé utilisable par le client.
- Eddystone-TLM: contient les informations du beacon, batterie, température, etc...
- Eddystone-EID: identique à UID mais encrypté.

A l'époque (avant décembre 2018) les téléphones Android étaient capables de recevoir des notifications beacon grâce au service NearBy de Google, mais cette fonctionnalité n'est plus supportée par Android pour des raisons de sécurité.

iBeacon: est un protocole de communication principalement utilisé sous iOS, il est similaire au protocole Eddystone, mais communique plus efficacement avec des appareils Apple qu'Android, c'est pourquoi nous n'allons pas nous intéresser à cette technologie dans cette partie du travail.

9.4.3 Initialisation des Beacons

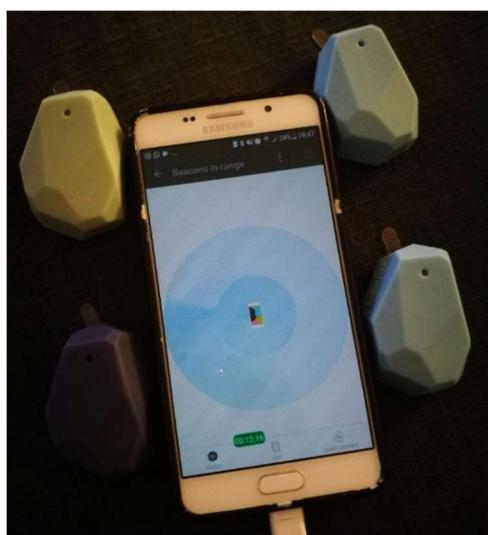


Figure 87 : Estimote non détecté

La technologie Estimote évolue rapidement et les beacons qui ont été utilisés sont des beacons de première génération de 2014. Ils ne sont donc pas compatibles avec les dernières technologies Estimote de proximité et utilisent encore l'ancien SDK Android comme mentionné précédemment. De ce fait, lors de leur initialisation, aucun d'entre eux n'était reconnu par l'application Estimote du fournisseur, car cette dernière n'est capable de reconnaître uniquement les beacons qui émettent avec le protocole iBeacon. Tous nos Estimote Beacons émettant le protocole Eddystone, aucun des quatre appareils n'apparaît donc dans le scanner.

Pour configurer les Beacons, Estimote met à disposition une application présente sur le Playstore. Elle doit être téléchargée afin de pouvoir effectuer les modifications sur les Beacons. Les informations sont ensuite envoyées dans un serveur cloud présent à l'adresse : <https://cloud.estimote.com>.

Il est donc possible de synchroniser les Beacons dans les deux sens : de l'application vers le cloud, ou du cloud vers l'application ; l'application étant l'intermédiaire (utilisant le Bluetooth) permettant l'envoi/réception des données entre les Beacons et le cloud. Afin d'utiliser cette application, nous avons dû réinitialiser les beacons afin qu'ils puissent émettre le protocole iBeacon qui était désactivé.

9.4.4 Mise en service de l'application et d'Android Studio

Dans un premier temps c'est une application mise à disposition sur le site développé par Estimote qui a permis de réinitialiser deux des beacons et de les faire passer en iBeacon (Protocole émis). Puis une application de test Beacons permettant de détecter les Beacons a été utilisée. Application provenant du site développeur de Estimote permettant de faire quelques tests [72] de la technologie, avant de voir si cela valait la peine de continuer avec ces appareils.



Figure 88 : Scan iBeacon et Eddystone



Elle peut être mise en place grâce à l'environnement de développement Android studio. Cette application basique est censée détecter un iBeacon et changer la couleur du fond de l'écran selon la couleur du Estimote IBeacon le plus proche, mais l'application ne fonctionne pas correctement, comme le montre l'image à droite qui devrait trouver le iBeacon et afficher un fond d'écran bleu clair de la même couleur que le iBeacon juste à côté.

En cherchant sur internet des applications en relation avec les beacons, je suis tombé sur une application Android sur le site: <https://github.com/Bridouille/android-beacon-scanner> [73]

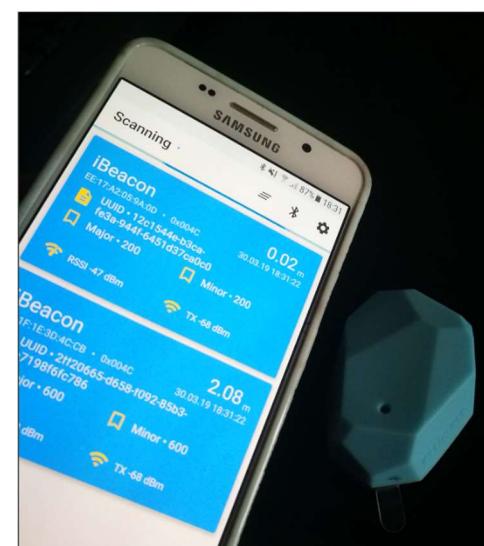


Figure 89 : Tests actualisation des distances

Cette application permet de scanner les beacons émettant les protocoles Eddystone et IBeacon. Comme le montre l'image à gauche. Cela va permettre de faire les tests que nous désirons afin de voir si la technologie Beacon est capable de répondre aux besoins de notre travail.

9.4.5 Tests

Pour les tests, deux Beacon Estimote sont disposés à environ dix mètres de distance dans des bureaux open Space. Il n'y a donc pas de murs qui viennent entraver la transmission des signaux. Dans un premier temps, la distance affichée dans l'application sera relevée. Puis nous allons relever les temps de synchronisation et les distances au fur et à mesure que nous nous déplaçons d'un beacon à l'autre.

Test du protocole iBeacon :

Dès le départ, la distance entre l'appareil et le beacon semble correcte : 0.02m cela est plutôt précis, cependant l'autre beacon qui se trouve entre 9 et 10 mètres du premier, apparaît à 2 mètres seulement selon l'application. On remarque déjà ici un gros problème des précisions quant à la distance. Cela est sûrement dû à la manière de calculer la distance, car le Beacon ne transmet pas sa distance exacte, mais plutôt sa force de signal RSSI et à partir de cette information, la distance est ensuite calculée.

En me déplaçant d'un beacon à l'autre à la vitesse du pas, j'observe des délais de mise à jour pouvant aller jusqu'à 10 secondes. J'ai fait le déplacement plusieurs fois en marchant et en trottinant comme l'aurait fait un jogger et j'ai pu attendre jusqu'à 15 secondes devant le beacon avant que le scanner mette à jour la distance du beacon le plus proche. En marchant, les délais passent entre 6 et 9 secondes, mais cela reste problématique, car le but est de pouvoir utiliser ces appareils dans un cadre sportif où les délais doivent être plutôt réactifs.

Test du protocole Eddystone :

Pour le protocole Eddystone, la précision de la distance s'améliore. J'ai pu avoir une distance entre un beacon et l'autre allant jusqu'à 4 mètres ce qui est plus raisonnable sachant qu'il y a environ 10 mètres entre les 2. Cependant en faisant le déplacement entre l'un et l'autre des beacons, le constat est le même : les temps de mise à jour sont impressionnantes, entre 8 et 10 secondes alors que les beacons émettent chaque 100ms. En trottinant à l'allure d'un jogger, j'ai réussi à faire l'aller-retour sans que le scanner ne détecte mon déplacement.

9.4.6 Conclusion et résultats

Pour les tests, un second scanner a été utilisé, car la vitesse de mise à jour aurait pu venir de l'application en elle-même et non de la vitesse des données. On remarque que le temps de mise à jour est très lent avec l'une ou l'autre des applications et même si ces tests ne sont pas très techniques/scientifiques et très poussés au niveau de leur rigueur, il n'y a pas vraiment eu besoin de faire beaucoup de recherches pour vérifier que cette technologie n'est pas adaptée au monde du sport. Il n'y a d'ailleurs aucune utilisation sportive préconisée par la marque Estimote sur leur site. Toutes les utilisations mises en avant sont des utilisations dites statiques ou l'utilisateur reste immobile pendant plus d'une dizaine de secondes, comme dans un musée ou lors d'une visite, ainsi que dans des magasins. Cela pourrait fonctionner pour une énigme ou pour donner un indice à une personne, mais les Beacons Estimote ne sont capables de fournir que quatre types d'informations comme vu au point 3.1.1. Cela voudrait dire qu'il faut que le joueur quitte notre application pour ouvrir un lien internet etc... Ce qui devient compliqué pour le joueur qui risque de se perdre en confusion. Cette technologie est très utile si l'on souhaite renvoyer quelqu'un sur une page internet en rapport avec un lieu, un tableau ou en monument, mais s'avère très limitée dans sa vitesse et dans le type d'information qu'il est possible de transmettre. Pour approfondir le sujet et voir des tests sur la rapidité de communication ce lien offrira au lecteur curieux une autre source : <https://shinesolutions.com/2014/02/17/the-beacon-experiments-low-energy-bluetooth-devices-in-action/>.

9.5 Capteurs de mouvements[74]



Figure 90: Capteur de mouvements

Ces capteurs ont été testés en même temps que les capteurs de distance. Ils pourraient être utiles pour une utilisation en intérieur. Cependant, en extérieur ils présentent des problèmes de déclenchement. La sensibilité de ces capteurs est réglable grâce à deux régulateurs (Cf. Figure 91).

Le premier, règle la distance de détection et le second le temps durant lequel le détecteur reste enclenché. Un « jumper » permet de choisir si l'on souhaite que la détection s'opère à chaque détection de nouveau mouvement ou si l'on souhaite, après un mouvement, lancer le chronomètre régler précédemment avec les régulateurs et garder le capteur actif durant cette période.



Figure 91: Régulateurs précision et délais

Une très mauvaise impression se dégage de ces boîtiers. De l'échantillon reçu deux n'ont pas l'air de fonctionner correctement. Ils sont fragiles et arrivent sans documentation. Heureusement internet permet de trouver les informations nécessaires à leur bon fonctionnement [74].

Grâce à l'aide trouvée, deux prototypes ont pu être mis en place. Très vite, la configuration s'est avérée compliquée. Entre les capteurs non fonctionnels, la précision, le délai, le jumper et la capacité de détection, ce capteur a été rapidement écarté. De plus, au même titre que les capteurs de distance, présentés dans le chapitre suivant, le problème des détections « parasites » reste présent. Il est donc difficile de trouver une utilisation sur le terrain de ce genre de capteur.

Finalement, il fonctionnera très bien pour déclencher une action en milieu contrôlé, par exemple dans une salle ou une escape room en interne après avoir par exemple passé une porte. Il est cependant à déconseillé pour une application sportive.



Figure 92: Jumper

9.6 Capteur de distance

Les capteurs de distance, ont été un des premiers choix et le capteur a été mis en place sur une borne. Ce sont des capteurs petits et permettant de récupérer la distance grâce à des ultrasons et leurs échos. Le capteur possède deux antennes, une émettrice et une réceptrice. Il est utilisé dans des projets de robotique pour la détection des obstacles. Il est donc satisfaisant au niveau des informations qu'il fournit et la précision des distances est acceptable.



Figure 93: Capteur de distance

Dans un environnement sportif, il est alors intéressant, car il permet de vérifier la proximité d'une personne ou de calculer la distance entre deux points, de manière plus ou moins précise.

Pour montrer que son utilisation est possible, ce capteur a été connecté sur deux bornes. Sur ces deux bornes, un petit programme a été codé permettant de calculer les allers et retours entre ces deux bornes. Cela permet d'avoir un exemple d'utilisation. Chaque borne est paramétrée pour détecter un mouvement à 35 cm de distance. Une fois détecté, la borne allume le led vert sur son tableau et envoie

un signal pour allumer le led rouge de l'autre borne à travers le server central du jeu. Le joueur se déplace alors vers l'autre borne et ainsi de suite. Cette partie du travail n'est pas présente dans le prototype elle est ici à titre indicative.

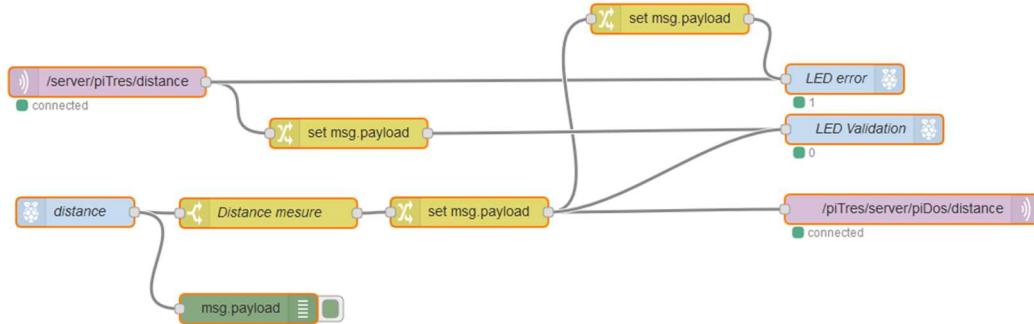


Figure 94: Flow du test pour les capteurs de distance

Le problème avec ce genre de capteur, c'est qu'il n'est pas possible de faire la différence entre le participant du jeu ou un passant, un animal ou un objet qui pourrait fausser alors la détection (détection parasite). De ce fait, il est tout à fait possible que le joueur passe devant la première borne et que quelques secondes plus tard un mouvement soit détecté sur l'autre borne empêchant le joueur de continuer le jeu ou lui permettant de tricher.

Pour cela, ces capteurs (au même titre que les capteurs de mouvements) n'ont pas été utilisés dans le prototype, car la zone de la plage d'Yverdon enregistre beaucoup de passage et il aurait été impossible de pouvoir détecter les mouvements du joueur sans avoir des parasites ou d'autres détections non dues aux déplacements de l'utilisateur.

9.7 Textes audio

9.7.1 Zone de départ

Bienvenus, la réserve de la grande caricaie et son secret sont menacés. Plus de 800 espèces végétales et près de 10'000 espèces du règne animal sont menacés.

Le gardien et protecteur des lieux a été kidnappé par des individus souhaitant s'emparer du secret de la réserve. Il est de votre devoir de le retrouver et de protéger cet endroit magique en son absence.

Pour trouver le gardien, protéger le secret de la réserve et restaurer l'équilibre des lieux, suivez les traces laissées par les ravisseurs.

Utilisez les indices et la carte en votre possession pour progresser dans votre quête. N'ayez pas peur, je vous assisterai tout au long de l'aventure.

J'ai vu les ravisseurs kidnapper le gardien et partir en direction de la zone des menhirs, je vous envoie les coordonnées qui devraient être apparues sur votre carte.

Vite, le temps nous est compté !

9.7.2 Zone des menhirs

Bravo, la zone des menhirs est là.

Quel endroit magnifique...

Regardez, les ravisseurs ont commencé l'exploration de la zone à la recherche du secret de la réserve.

Six sondes de couleurs ont été introduites à proximité de chaque groupe de menhirs. Chaque sonde possède sa propre couleur et doit être désactivée dans le bon ordre.

Je vois la première. Là-bas derrière le plus grand des menhirs. Il vous faudra trouver le reste des sondes et utiliser le boîtier à votre disposition pour toutes les désactiver. Attention n'oubliez pas, leur ordre est important et il vous faut commencer par le plus grand.

Faites attention à ne pas les toucher elles sont peut-être dangereuses. Dépêchez-vous le gardien a besoin de nous !

9.7.3 Mauvais code

Non, les couleurs des sondes ne sont pas dans le bon ordre.

Le boîtier vient de se verrouiller. Ne vous inquiétez pas, je vous envoie les coordonnées de la zone ou vous pourrez trouver la clef et déverrouiller le boîtier. Faites vite !

9.7.4 Bon code

Super, le code est bon ! Toutes les sondes ont été désactivées. Cela risque de ralentir les kidnappeurs. Il y a dans la réserve un observatoire duquel nous devrions pouvoir observer les lieux et peut-être trouver une piste.

La zone devrait s'afficher sur votre carte. Allez-y au plus vite. Le temps presse.

9.7.5 Zone Clef :

Bravo vous venez de récupérer la clef. Retournez au plus vite dans la zone des menhirs et entrez les six couleurs dans le bon ordre. Souvenez-vous, il faut commencer par le plus grand des menhirs.

9.7.6 Zone de l'observatoire

Super, l'observatoire est devant : montez au sommet.

J'ai vu en passant un arbre ligoté, peut-être a-t-il vu passer les ravisseurs et le gardien ?

Il doit y avoir une clef quelque part, pour le libérer. Il faut la trouver, elle doit-être par ici j'en suis sûre. Cherchez la clef et allez libérer le noisetier. Je vous ai affiché sa position sur la carte. Ne perdez pas de temps et surtout, ne perdez pas la clef.

9.7.7 Zone du noisetier

Super il est là, trouvez la serrure pour le délivrer de ces liens, il doit avoir des informations sinon les ravisseurs n'auraient pas pris la peine de le ligoter ainsi.

9.7.8 Zone du noisetier (arbre libéré)

Hoooo, merci de m'avoir libéré. J'ai vu qu'ils avaient pris le gardien et qu'ils l'amenaient quelque part, mais je n'ai pas vu où. Cependant, ils ont parlé d'un code dissimulé au sommet de l'observatoire.

Il faut que vous retourniez là-bas reposer la clef où vous l'avez trouvée, on ne sait jamais s'ils me ligotent à nouveau. Elle sera peut-être encore utile.

Retournez d'où vous êtes venu. La suite de votre aventure est là-bas.

9.7.9 Zone observatoire (retour)

Bravo, nous avons maintenant plus d'informations pour la suite, il nous faut reposer la clef et trouver le code.

Mais...il me semble apercevoir les ravisseurs ! Avant de partir cherchez bien le code de six chiffres qui se cache devant vous en haut de l'observatoire. La mission de conserve de la grande caricaie pourra vous aider.

Du lac à la moissonneuse batteuse vous verrez le code de six chiffres apparaître devant vous. Mémorisez le bien, il doit être plus qu'utile pour la suite.

J'ai noté et affiché sur votre carte la dernière position des ravisseurs. Avant de partir, n'oubliez pas de reposer la clef avant de partir et de mémoriser le code.

9.7.10 Zone du gardien

Le gardien est là ! Non il est prisonnier de son socle. Il faut l'aider. Trouvez le moyen de le libérer. Je suis sûr que le code de l'observatoire sera utile ici.

9.7.11 Erreur de code

Non, ce n'est pas le bon code. Le gardien est toujours prisonnier. Es-vous sur de l'avoir mémorisé comme il faut à l'étape précédente ?

9.7.12 Fin et succès

Gardien : Merci, grâce à vous la réserve et son secret sont sain et sauf. Mais les ravisseurs étaient masqués, je n'ai pas eu le temps de les voir. Mais je sais que je dois faire attention maintenant, ils pourraient revenir. Encore merci pour votre aide, je vous suis très reconnaissant.

9.8 Carte du jeu

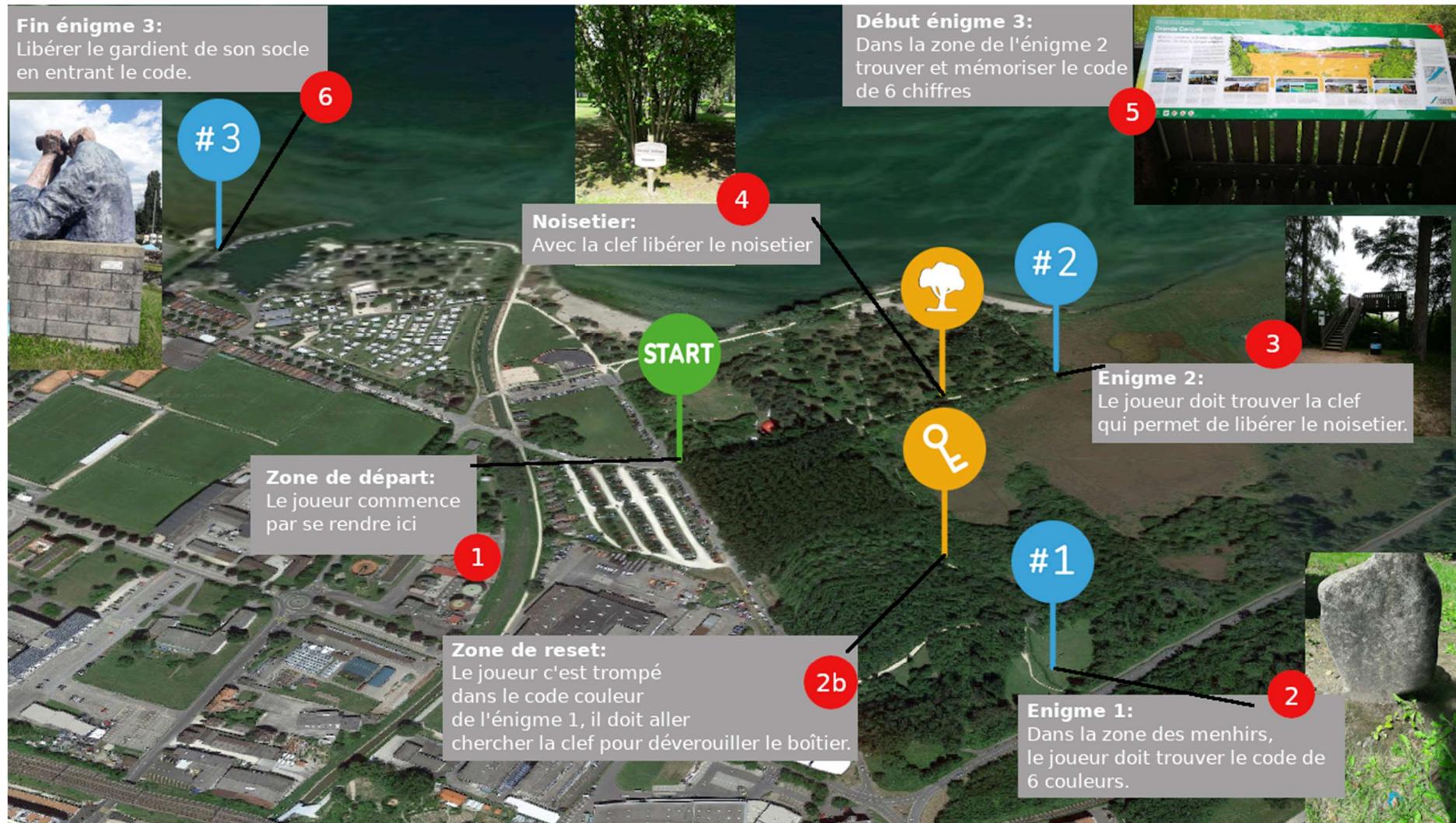


Figure 95: Plan et carte du jeu