



# **Projet Hunting Town**

## **Année scolaire 2018-2019**

### **Rapport d'Axel Devaux**



## Table des matières

Introduction.....	3
Genèse du projet .....	3
Le client .....	3
L'équipement technique actuel .....	3
L'équipement des joueurs .....	3
L'équipement du Game-Master .....	4
L'état des lieux .....	4
La problématique .....	4
L'équipe .....	4
Cahier des charges .....	5
Les besoins principaux .....	5
Les besoins secondaires .....	5
Les besoins tertiaires .....	5
Etudes préparatoires.....	6
Partie individuelle .....	9
G.P.S.....	9
<i>UART</i> .....	10
Récupération des données G.P.S .....	10
Interface Homme Machine.....	10
Bluetooth .....	12
Choix .....	12
Module.....	12
nRF8001 .....	12
nRFgo Studio.....	13
.....	13
Améliorations .....	13
Code javascript.....	13
Conclusion.....	14

# Introduction

Pour le commencement de ce projet toute l'équipe a travaillé sur le même sujet, donc le début de ce rapport sera identique à celui du rapport de groupe. Nous avons commencé à faire deux équipes de deux personnes au moment de la réception des modules G.P.S et 4G. C'est donc à partir de ce moment-là qu'une différence entre les deux rapports apparaîtra.

## Genèse du projet

Ce projet est réalisé dans le cadre du Master 1 « Chef de Projet systèmes Embarqués » au sein de l'école *Campus YNOV Bordeaux*. La mise en contact de l'entreprise *Hunting Town* et de son directeur M. Thibaut Giuseppi avec l'école a été faite par un élève, Jean-Gabriel Massicot. Lui-même a été mis en relation avec la société grâce sa compagne y ayant travaillé.

À la suite d'explications sur le fonctionnement technique du jeu, sa constatation a été que le système mis en place est complexe, présente des difficultés techniques impactant le Gameplay, et que la conception d'un objet connecté serait à la fois intéressante pour l'entreprise, passionnant à mettre en place et entrant dans le cadre des compétences de l'Ecole, de la formation et du projet.

C'est après avoir demandé confirmation à M. Pierre Aubry, responsable de la formation, et après avoir effectué un premier rendez-vous téléphonique où étaient présents Jean-Gabriel Massicot, M. Pierre Aubry, et M. Thibaut Giuseppi, que la décision d'organiser ce projet a été pris.

## Le client

Thibaut Giuseppi est le directeur de la société *Hunting Town*, société proposant des Escapes-Games en plein air. Il s'est adressé à l'école *YNOV Aéronautique et Systèmes Embarqués* afin de répondre à une problématique technique qu'il rencontre.

Marketé sous le nom « Escape-Game dans la ville », *Hunting Town la mission* est une chasse au trésor 2.0 dans la ville. Les aventuriers qui participent par groupes de 2 à 6 joueurs disposent d'une heure pour se rendre sur les différents checkpoints dans la ville et y résoudre les énigmes qui leur seront soumises. Gérés à distance par le Game Master qui leur envoie des vidéos selon leur avancée, le gameplay est organisé avec le téléphone des participants en leur proposant de donner la réponse aux énigmes par SMS.

## L'équipement technique actuel

### L'équipement des joueurs

- Une GoPro filmant leur aventure
- Une montre pour gérer le chrono

- Un portable dissimulé dans l'armature de la GoPro, en communication avec le portable du Game Master, et lui permettant d'entendre les réflexions des joueurs en direct
- Un téléphone portable accroché au poignet d'un des participants et lié à l'ordinateur du Game Master de sorte à shooter les bonnes vidéos (hébergées dans la galerie du portable en question) au bon moment.
- Utilisation du téléphone personnel d'un participant pour répondre par message aux énigmes.

## L'équipement du Game-Master

- Un téléphone pour réception de l'appel et envoi des sms
- Un ordinateur qui prend la main sur le téléphone au poignet des participants pour diffuser les vidéos au bon moment et suivre les déplacements des candidats sur une carte grâce au GPS.

## L'état des lieux

- Le système en place n'est pas optimal. Il est très dépendant des fluctuations des réseaux internet de l'ordinateur du Game Master et du portable remis aux participants. En cas de coupure, le gameplay est directement affecté : les vidéos ne se lancent plus. Elles sont envoyées via des liens You Tube.
- Le micro, nécessaire pour le jeu et les interactions avec les joueurs, nécessite un portable supplémentaire sur l'armature de la GoPro

## La problématique

*Concevoir un objet connecté capable d'assurer la diffusion des vidéos déclenchées à distance par le MJ, de capter le son des joueurs et de le transmettre au MJ en temps réel, de renvoyer sa position GPS au MJ, le tout dans un boîtier résistant à l'eau, devant avoir au moins 2h d'autonomie.*

## L'équipe

L'équipe sera composée de 5 élèves de Master 1 de l'école à savoir :

- Axel Devaux
- Nicolas Gibaud
- Clément Lavergne
- Antoine Venier

# Cahier des charges

## Les besoins principaux

\*Les [Étiquettes] sont des références pour le travail de groupe

- [Transmission] Choix du mode de transmission (Joueur → MJ) sur la zone géographique prédéfinie.
  - La connexion ne doit pas impacter le gameplay, si déconnexion alors il faut une reconnexion automatique
- [GPS] Traçage GPS, avec une précision d'environ 5m, communiqué en temps réel 5s (messenger) au MJ
  - Choisir un module GPS
- [Vidéo] Afficher des vidéos avec son, pré-enregistrées sur le boîtier, et lancées par le M.J.
  - Choix d'un écran d'environ 5.2" pas en dessous de 4" en environ 720p (mais la qualité n'est pas un enjeu majeur)
  - Choix de Haut-Parleurs. Le son doit être d'environ 60 dB (pour 6 personnes, environnement bruyant, discret pour les passants)
  - Choix des modules son/vidéo (out)
- [Appel] Streamer le son des joueurs (micro).
  - Module son (in)
  - Choix d'un micro
- [Boîtier] Système de fixation.
  - Poids d'environ 200 grammes support compris
  - IP04M ? IP14M ? IP24M ? [Indice de Protection (eau)] [xsolide] [xliquide] [Info complémentaire] (Waterproof (averse)).
- [Alim] Autonomie 2h minimum.
- [Adressage] Système d'adressage
  - Le MJ doit pouvoir s'adresser à plusieurs boîtiers distants différents en cas de changement de boîtier.

## Les besoins secondaires

- [Minuteur] Minuteur 1h (compte à rebours)
- [Messagerie] Système de messagerie

## Les besoins tertiaires

- [Map] Carte de bordaux
- [Direct] Vidéo de la GoPro® transmise en direct

## Etudes préparatoires

- Tutoriel sur l'utilisation de GIT → Présenté par Antoine, tutoriel collaboratif
- Recherche d'un moyen de communication Joueur-MJ (Sigfox, 4G, LoRa, autres)

	LoRaWAN	Sigfox	4G (LTE?) Exemple de <a href="#">module datasheet</a>
Portée	1Kmenville-20Kmenzone rurale plane	Nationale (zones reculées, la creuse, en particulier) Plus de 28968,192 kms	
Débit besoin = 10kbit/s	entre 0.3 et 50 Kbits/s	Fixé à 100 bits/s	4G LTE Cat M1 : 300Kbps 4G Cat NB1 : 32Kbps
Réseaux tants	OUI	Oui	Oui.
Stabilité	chiffrement des données. Possibilité de récupérer les données sur un broker MQTT. Pas adaptée à la transmission radio	bande fréquences étroites donc difficile d'être perturbé	Stable, surtout avec le SRVCC (basculement sans coupure avec 3G/2G)
Mise en vre	- ça à l'air compliqué : bibliothèque Arduino existante → on peut s'en inspirer pour utiliser le Lora	Abonnement, pas de carte SIM	
Consomm ion	Emission(mA) Min : 18 mA (7dBm) Max: 125mA(20dBm)		moyenne : 3.8V, 70mA
Bande de ences	bande de 867 - 869 MHz		ça dépend vachement 180kHz-20Mhz
Prix des dules	entre 10 et 20 euros		

	LTE Cat 1	LTE Cat 0	LTE Cat M1 (eMTC)	LTE Cat NB1 (NB-IoT)	EC-GSM-IoT
<b>3GPP Release</b>	Release 8	Release 12	Release 13	Release 13	Release 13
<b>Downlink Peak Rate</b>	10 Mbit/s	1 Mbit/s	1 Mbit/s	250 kbit/s	474 kbit/s (EDGE) 2 Mbit/s (EGPRS2B)
<b>Uplink Peak Rate</b>	5 Mbit/s	1 Mbit/s	1 Mbit/s	250 kbit/s (multi-tone) 20 kbit/s (single-tone)	474 kbit/s (EDGE) 2 Mbit/s (EGPRS2B)
<b>Latency</b>	50-100ms	not deployed	10ms-15ms	1.6s-10s	700ms-2s
<b>Number of Antennas</b>	2	1	1	1	1-2
<b>Duplex Mode</b>	Full Duplex	Full or Half Duplex	Full or Half Duplex	Half Duplex	Half Duplex
<b>Device Receive Bandwidth</b>	1.4 - 20 MHz	1.4 - 20 MHz	1.4 MHz	180 kHz	200 kHz
<b>Receiver Chains</b>	2 (MIMO)	1 (SISO)	1 (SISO)	1 (SISO)	1-2
<b>Device Transmit Power</b>	23 dBm	23 dBm	20 / 23 dBm	20 / 23 dBm	23 / 33 dBm

MODULE 4G	Prix	Mode Communication	Taille	Type de 4G	Autres protocoles supportés (3G, 2G, etc...)	Débit	Service de Localisation	Consommation	Liens	Connexion Beagle Bone
EM7565 LTE	128€	USB3.0, 2.0 high speed APIs, AT Commands	42x30x2.3 mm	LTE	3G	download 600Mbps upload 150Mbps	Galileo, Glonass, GPS, Beidou		<a href="#">click me</a>	
Skywire 4G (plugable avec des headers)	154€	UART ou USB	29x33x10.5 mm	LTE CAT 3	3G, 2G	download 100Mbps upload 50Mbps	GPS, Glonass	29mA	<a href="#">click me</a>	<a href="#">click me</a>
DIGI XBEE3	60€	UART, SPI, USB	24.38 mm x 32.94 mm	LTE-M	NB-IoT (nul)	up to 375 kb/s	X	235mA	<a href="#">link</a>	
E series module (CECI EST UNE BOARD)	hardware : \$49 soft : \$2.99/months				3G, 2G				<a href="#">ClicKé</a> (full datasheet)	
Botletics SIM7000E → Linux drivers	75€	I2C x1 GPIO x5 ADC x1 UART x2	24 x 24 x 2.6mm	LTE CAT-M1	EDGE	300kbps	GPS, GLONASS	200mA	<a href="#">Amazon</a> <a href="#">Git-VRAIMENT-complet</a>	
EM7355 MPCIE-L210 PCI	38.50€	USB	30.0 x 51.0 x 3.7 mm	LTE Cat 4	4G	download 150Mbps upload 50Mbps			<a href="#">click me</a>	

LARA-R211(VoLTE)	170 €	UART, USB, HISC, SDIO		LTE cat 1	3G, 2G	10.3 Mbit/s DL 5.2 Mbit/s UL			<a href="#">Lien</a>	

Résultat : choix de la communication par le réseau 4G. Le facteur limitant pour les autres moyens de communication est la transmission du son en direct. Les autres solutions ont un débit trop faible qui ne permettent pas le passage d'assez de données. Cependant il existe beaucoup de sortes de 4G et différents modules, avec des différences qui semblent assez importante en termes de bande passante, de consommation d'énergie et autres.

Certains modules ont les 3, certains ont le GPS intégré.

Protocoles intéressants : LTE Cat M1, LTE Cat 12, VoLTE. Avec le protocole SRVCC qui permet de basculer sur 3G/2G sans coupure.

Nous avons trouvé plusieurs modules 4G qui correspondent à nos besoins (avec GPS intégré). Ensuite nous nous sommes intéressés aux cartes de développement linux qui intègrent un module 4G, un GPS et un écran d'à peu près 5 pouces. A la suite de ça nous avons constaté qu'il existait peu de cartes qui correspondaient à tous nos critères, il y a donc de forte chance qu'il faille rajouter des modules à la carte que nous aurons choisie.

Après ces recherches, nous avons compris :

- Que la 4G est le meilleur moyen de communication
- Que le GPS est souvent inclus dans les modules 4G
- Le choix d'un module est très lié au choix d'un SBC (Single Board Computer).

Nous avons donc étudié diverses SBC pour définir la plus adaptée :

Carte	Prix	Ecran	Taille poids	4G	GPS	Mémoire fixe et volatile fréquence de calcul	Audio(jack)	Liens	Architecture
BBONE-BLACK	50-60\$	<a href="#">Block</a> ou "externe"	86mm x 53mm 116 g	<a href="#">Skywire 4G</a> (déjà vu) et plein d'autres à pluger		1 GHz 4 GB RAM	No mais : <a href="#">cape</a>	<a href="#">Mouser</a>	ARM Cortex A8
SBC-EC8800	74,83€	Connecteur LCD IF	0.062 kg	No	No	4GB (possibilité de SD) et environ 1GB	No	<a href="#">Farnell</a>	ARM Cortex A-9
OEM android embedded	177,31€	MPEG4 / H.264: 720p @	47.8*53.5*3.5 mm 100g	oui	oui	ram: 1GB LPDDR3	oui	<a href="#">Pondelectronics</a>	



		30fps				En option 2 Go LPDDR3) rom : 8Go de mémoire CMEM			
FRDM-K64F Freedom Board	40€			module additionn el	module addition el	1 MByte flash 256 KByte de RAM	No	<a href="#">start kit</a>	Cortex@- M4
HummingBoa rd	170€	pas trouvé		<a href="#">petits modules disponibles</a>		2GB RAM 8GB eMMC	Suivant les version	<a href="#">click</a>	NXP i.MX6 ARM A9
Tinker Board	65€	Sortie mini HDMI (4K)	55 g	Rien trouvé		support SD - ram 2 Go	1 entrée- sortie Hdmi = sortie	<a href="#">Site constructeur</a>	Mali T764 basé sur ARM
Orange Pi	80€ (aliexpress e)	inclus		Inclus FDD LTE LTE Cat...	inclus	8GB flash 1GB DDR3	1 prise jack I/O 1 micro	<a href="#">Site constructeur</a>	ARM Cortex- A53

NB : La taille de l'écran et par conséquent celle de la carte ne doit pas dépasser 5  
pouces(diagonale) soit 12.7 cm

Site pour les modules : [Site modules](#) en fait ce site répertorie aussi les boards  
<http://simcomm2m.com/En/module/detail.aspx?id=86>

## Partie individuelle

### G.P.S

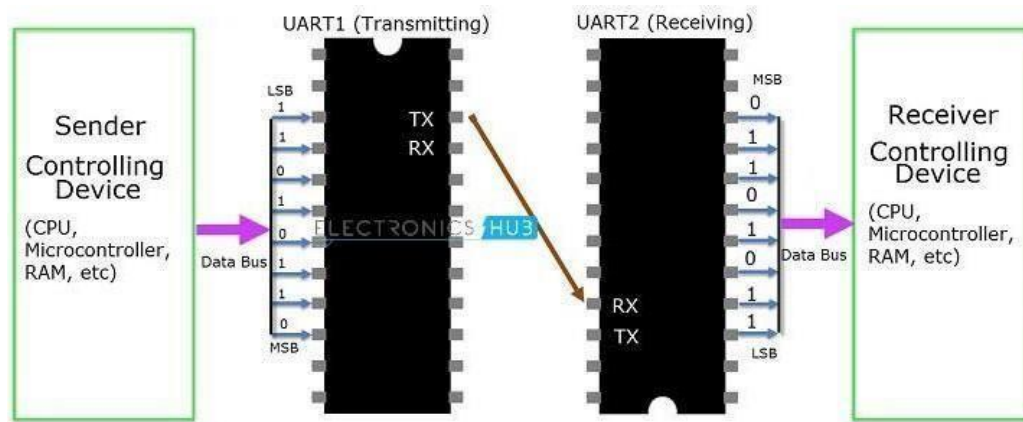
Je suis chargé de la partie G.P.S, c'est-à-dire de récupérer les informations du  
module.

G.P.S en ligne de commande pour le moment et par la suite sur une I.H.M.

Pour la récupération de la position nous avons choisi le module NANO GPS click. Ce module  
comprend le module Nano Hornet d'OriginGPS. C'est le plus petit module G.P.S (10x10x3.8mm)  
comprenant une antenne patch. Malgré sa petite taille il possède une forte sensibilité avec une  
TTFF (Time to first fix qui est une mesure du temps requis par un appareil de navigation GPS pour  
acquérir les signaux satellite et les données de navigation et calculer une solution de position.)  
inférieure à une seconde avec une précision d'environ un mètre.

Ce module est également protégé du bruit (perturbation électromagnétique). Le module  
G.P.S communique en UART.

## UART



<https://www.electronicshub.org/basics-uart-communication/>

L'UART est une liaison série permettant l'envoi et la réception de données. Elle a une architecture assez simple, qui se résume à une alimentation (VCC et GND), une pin d'émission (TX) et une pin de réception (RX).

L'émetteur envoie des commandes au récepteur (TX => RX) et l'opération est inversée pour que le récepteur réponde à l'émetteur (RX => TX).

## Récupération des données G.P.S

Lors de la réception du module G.P.S, je devais mettre en place la liaison UART mais comme j'ai eu une suite d'entretiens, Antoine avait bien avancé l'établissement de la liaison, je me suis donc tourné vers la mise en place de l'Interface Homme Machine.

## Interface Homme Machine

Maintenant que le module G.P.S est mis en place et que l'on reçoit la longitude et la latitude il faut que le Game Master puisse visualiser ces informations sous forme d'un point sur une carte. Nous avons donc pensé à la programmer en python. Nous avons recherché dans les librairies existantes, cartopy semblait une bonne librairie pour nos besoins.

### *Cartopy*

Après avoir installé les extensions nécessaires nous avons pu essayer d'afficher les cartes développées sous Numpy. Malheureusement ces cartes n'étaient pas ce qui nous fallait. Nous avons donc dû rechercher d'autres solutions.

## Selenium

La deuxième solution à laquelle nous avons pensé était d'actualiser la page de Google Maps en faisant une recherche pour les coordonnées récupérées. Nous avons donc fait un programme python permettant utilisant le module **Selenium** permettant d'interagir avec un navigateur web. Cela fonctionnait correctement, nous avons cependant rencontré un problème. Le problème est que Google Maps et la plupart des sites affichant des cartes interactives arrondissent la longitude et la latitude au centième près, ce qui engendrait un léger décalage de la position. A cause de ce décalage nous n'étions plus dans les conditions fixées par Thibaut.

## Mapbox

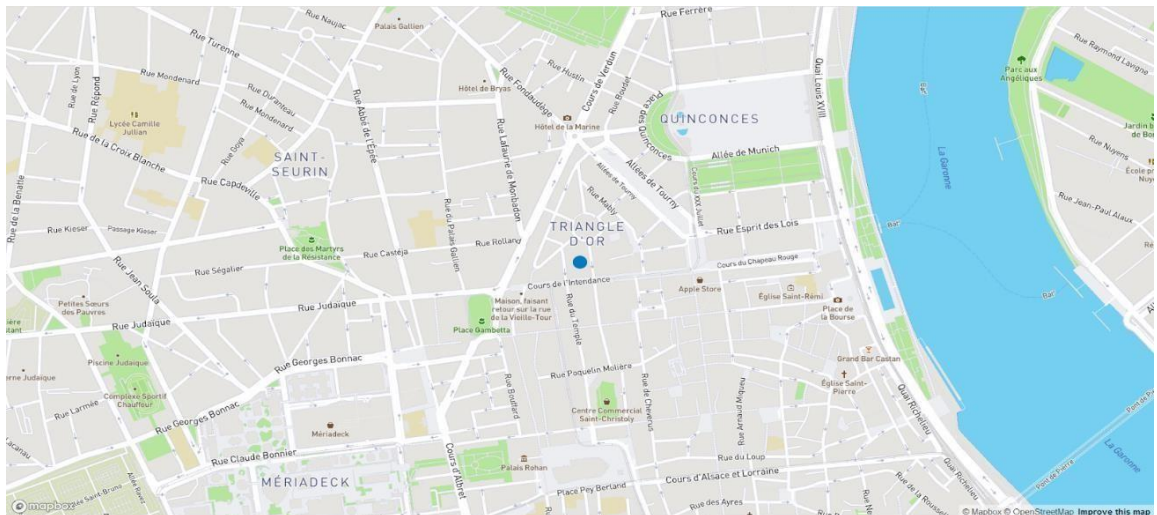
Nous nous sommes finalement tournés vers l'API Mapbox. Le seul problème c'est qu'elle est codée en javascript. Après une légère autoformation en javascript nous avons pu exploiter cette API.

L'avantage de celle-ci, comparée aux autres, c'est qu'elle n'arrondit pas la longitude et la latitude, nous obtenons donc une position exacte.

À la suite de la réalisation du programme qui affiche un point sur une carte nous avons intégré le programme écrit également en javascript qui récupère les coordonnées G.P.S envoyées par le module.

Nous arrivons actuellement à envoyer des GET sur l'API d'Hologram et à récupérer les données envoyées par le module G.P.S pour les afficher sur une page internet. Voici un exemple de ce qu'on peut obtenir sur la page internet. Nous récupérons les données sur l'A.P.I d'Hologram environ toutes les 7 secondes. Au début nous actualisions les données en continu, mais Hologram nous a envoyé un mail pour nous dire qu'il fallait au minimum deux secondes d'intervalle entre deux get.

Nous avons rencontré un second problème, c'était de pouvoir actualiser la position du point sans rafraichir la page. Nous avons finalement trouvé une solution. Notre point s'actualise sans que la page web n'est besoin de se rafraichir. Nous avons donc un affichage relativement fluide de la position du G.P.S.



Après une démonstration du prototype à monsieur Giuseppe peu concluante nous avons décidés de changer de module G.P.S. nous aurons donc potentiellement à modifier notre code.

# Bluetooth

Pour la suite nous avons distribuées les tâches restantes. J'ai donc choisi la partie Bluetooth. Cette partie consiste à envoyer un ordre de lecture de vidéo depuis la stm32 à la beagleboneblack. Pour faire cela, j'ai recherché s'il existait des technologies adaptées au modules que l'on a. En effet il existe nrfgo studio développé par Nordic Semiconductor.

## Choix

Nous avons besoin d'une connexion entre la STM32 et le BeagleBone, cette connexion n'a une utilité assez simple : transférer les ordres du maître du jeu reçu par 4G concernant le lancement des vidéos.

Pour le choix de la technologie nous nous sommes naturellement tournés vers le Bluetooth Low Energy pour deux raisons :

La BeagleBoneBlack Wireless est équipé d'un module Bluetooth supportant le BLE

Le BLE permet une communication simplifiée qui consomme peu d'énergie

## Module

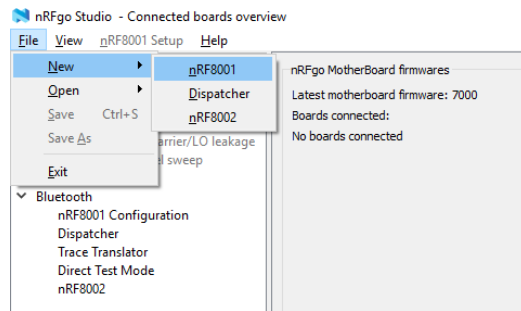
Nous avons choisi d'utiliser le module BLE P click conçu par MikroElektronika, comme pour les autres modules nous favorisons les modules de ce constructeur car nous pouvons facilement brancher les modules proposés aux "shield" présent à l'école.

Le module est équipé du nRF8001, composant de gestion du Bluetooth. Le module est "plug and play" tout est opérationnel grâce notamment à l'antenne intégrée.

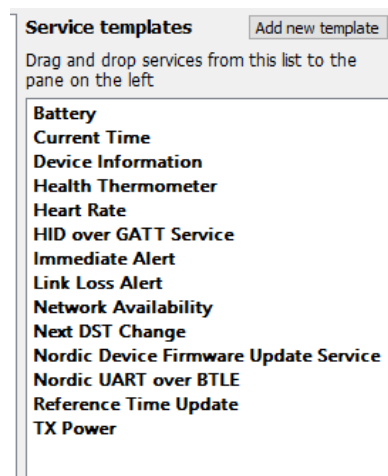
## nRF8001

Ce module est spécifique au BLE, il possède un protocole unique : "Application Controller Interface" (ACI). Ce protocole définit comment nous allons communiquer avec le module depuis la carte STM32, basé sur le protocole SPI nous communiquons à l'aide de commandes et d'événements propre à cette famille de module.

## nRFgo Studio



Pour le paramétrage du logiciel il suffit de créer un nouveau fichier « nRF8001 ».



Une fois ce fichier créé, le logiciel nous présente des templates préexistants. Ceux-ci ne me convenant pas je suis parti dans l'optique de créer le miens. Ce qui m'a pris beaucoup de temps. Je n'ai pas réussi à le réaliser dans des délais raisonnables, Clément a donc repris cette partie.

## Améliorations

### Code javascript

Nous avons changé de technologie pour la réception de la position G.P.S, en effet nous utilisons dorénavant un serveur sftp. Un fichier avec les coordonnées G.P.S sera téléchargé sur le pc du maitre du jeu. Avec le code javascript il faudra ouvrir ce fichier, découper les différentes informations contenues dans celui-ci pour ne garder que la longitude et la latitude. Nous ne communiquons plus avec l'api d'Hologram. J'ai donc dû réécrire le code. En prévision d'un jeu avec plusieurs groupes j'ai également dû dissocier le cadrage de la carte de la position du point. Pour ce faire j'ai réécrit le code en créant des fonctions que le code appellera au fur et mesure de son exécution. Cette réécriture sera bénéfique si nous avons besoin de reprendre ce code pour rajouter des onglets de choix de groupe sur la carte par exemple. Nous avons donc maintenant un centrage de la carte sur la zone de l'escape game et non sur un seul point. J'ai également rajouté un outil pour changer l'orientation de la carte et un outil de zoom.

## Conclusion

Ce projet m'a permis, dans un premier temps de mettre en place une gestion de projet mais également d'expérimenter de ce côté-là ce qui nous conviendrait le mieux en termes d'organisation. Ensuite il m'a permis de développer des compétences personnelles en rapport avec les systèmes embarqués mais également d'acquérir des compétences transversales comme la communication avec les membres d'une équipe mais aussi la communication avec un client. Ce qui implique de savoir vulgariser son propos, de savoir écouter mais aussi donner son point de vue.

Pour résumer ce projet m'a beaucoup apporté techniquement qu'en termes de communication et de gestion de projet.