

FOURNIER Jérémy, ROBERT Nicolas, SBAI Hatim, PAVY Killian

Rapport SAE 301

Espace Game



IUT Réseaux et Télécommunications
2022-2023

I) Présentation

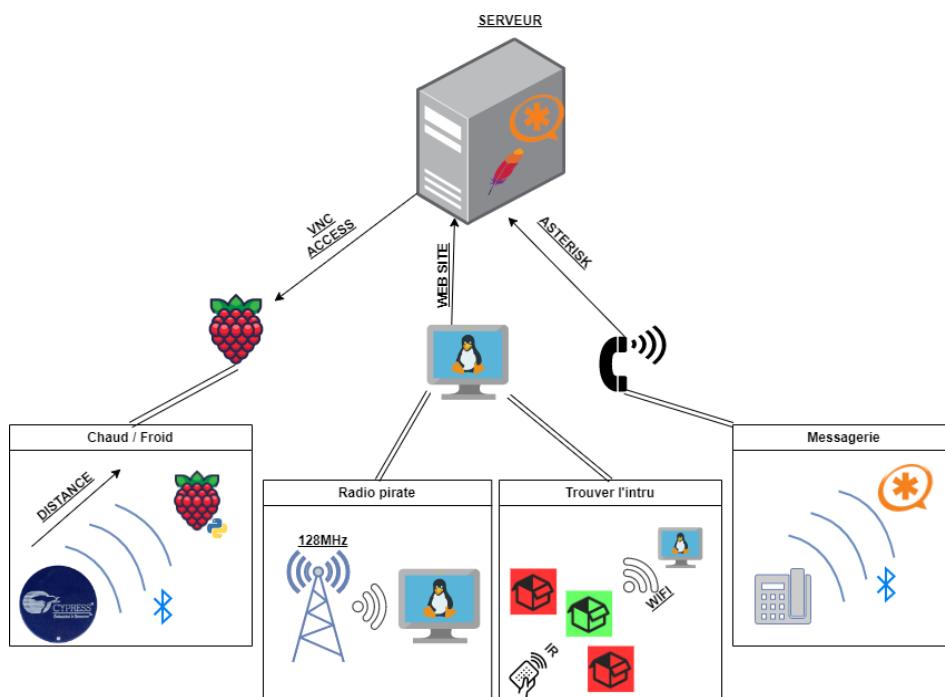
Le but de cette SAE est de mettre en place un escape game avec une suite de mini jeux utilisant différents systèmes de transmissions comme :

- **La radio**
- **L'infrarouge**
- **Le Bluetooth**
- **Le wifi**

Les différents mini-jeux sont reliés car il est nécessaire de valider le mini-jeu d'avant pour passer au prochain. L'escape game se joue sur un site hébergé sur une machine Linux.

Le but final étant d'arriver à la dernière étape afin d'obtenir une récompense.

Voici un schéma explicatif des différents jeux qui sont mis en place :



Une [vidéo](#) est disponible présentant l'ensemble de l'escape game et une démonstration des mini-jeux.

II) Différentes épreuves

L'escape game est principalement constitué de 4 mini-jeux où le joueur à partir d'une consigne sur le site du jeu arrive à trouver la solution. Voici un présentation de ces différents jeux qui le constitue :

- 1) **Chaud / Froid** : Le joueur est équipé d'un Raspberry Pi avec un mini-écran intégré ainsi qu'un récepteur bluetooth qui capte via un programme python qui tourne en fond les trames envoyées par l'émetteur caché dans la salle. Le joueur reçoit une seule information : La distance entre le récepteur et l'émetteur à partir de la puissance du signal reçu. Toutes les X secondes le programme renvoie cette distance et il reste plus qu'à réussir à se diriger pour trouver le fameux émetteur BLE caché. Une fois l'émetteur trouvé, le joueur obtient une télécommande qui lui servira pour la prochaine épreuve
- 2) **Trouver l'intru** : Le joueur se retrouve face à une dizaine de boîtes hermétiques et doit utiliser une télécommande IR qui une fois passer devant la boîte active l'une d'entre elle qui se met à allumer un LED à l'intérieur. Une fois trouvé il l'ouvre et trouve un numéro de fréquence en MHz pour le jeu suivant
- 3) **Radio Pirate** : Le joueur possède une fréquence qui lui est renseignée et doit réussir à se mettre sur cette fréquence via gqrx afin de l'écouter. Une voix robotique enregistrée est diffusée en boucle et le joueur doit prendre note du message qu'il réussit à entendre. Le message transmit correspond à un numéro de téléphone qui lui servira par la suite
- 4) **Messagerie Asterisk** : Le joueur via son téléphone passe un appel au numéro relié à la messagerie du serveur Asterisk. Un fois l'appel passé il tombe sur la messagerie, qui pendant une dizaine de secondes émet un code en morse, les joueurs doivent le décoder pour trouver la réponse finale

III) Déroulement de l'épreuve

Le joueur devra être équipé de plusieurs équipements primordiaux sinon il ne pourra pas jouer au jeu. Voici la liste des équipement et logiciels associé que doit avoir le groupe :

- **Un raspberry Pi avec écran intégré + recepteur BLE branché**
- **Un téléphone (n'importe lequel)**
- **Un poste linux avec le logiciel GQRX**
- **Une clé DVB RTL + antenne**
- **Une télécommande**

Il n'y a pas de points à proprement parler mais il s'agit du premier arrivé qui gagne.

Pour commencer à jouer et à comprendre dans quoi, le groupe de joueur s'embarque, ils accéderont au site web hébergé via Apache sur le serveur de jeu auquel tout leur sera expliqué et ils pourront commencer les différents mini-jeux.

Un système de logs nous permettra de tenir au courant de qui en est où enfonction de leur position dans la salle et ainsi leur IP.

En cas de difficulté de certains groupes, un bouton indice est mis à disposition sur chaque page des mini-jeux, il est gratuit car on veut que ça reste fun et que les joueurs puissent avancer assez rapidement si ils sont perdus.

IV) Mise en place des mini-jeux

1) Chaud / Froid (Cypress BLE)

Nous avons à notre disposition un kit de chez CYPRESS nommé : "**CYALKIT-E02 SOLAR POWERED BLE SENSOR RDK**" :



Il est équipé de 2 choses :

- Un **émetteur** (rond bleu) qui en fonction de la lumière qu'il reçoit envoie plus ou moins rapidement des trames BLE contenant des informations sur la température et l'humidité.
- Un **récepteur** à brancher sur un port USB, qui via l'application Cypress-BLE-Beacon offre une visualisation des données reçues sous forme de graphique.

Une application est aussi disponible sur téléphone mais nous ne l'utiliserons pas ici.

La technologie BLE transmet des trames dans laquelle un champ "RSSI" nous est très utile car elle correspond à la puissance en dB réceptionné. Il existe une formule afin de trouver la distance entre l'émetteur et le récepteur en fonction de cette valeur:

- **Distance**= $10^{**}((\text{measured_power}-\text{RSSI})/(10^*\text{N}))$
- **Mesured_power** = Puissance du signal reçu en dB
- **RSSI** = Valeur du champ RSSI en binaire
- **N** = Indice de l'environnement, range de 2-4

Il est impossible d'automatiser l'extraction des trames BLE directement via le logiciel **Cypress-BLE-Beacon** donc on peut utiliser **Python** afin de traiter les données.

En effet vu que le récepteur est connecté en USB à notre machine il est possible via Python d'écouter toutes les trames réceptionné sur ce port où il est brancher et d'extraire les trames BLE et faire ressortir la distance à partir des données qu'elle contient.

Le programme ble fonctionne de cette manière :

- 1) Ecoute sur le port où le récepteur est connecté
- 2) Analyse de toutes les trames qui transitent avec correspondance BLE
(0201041AFF4C0002150005000100001...)
- 3) Extraction du champ RSSI dans chaque trame sélectionnées (2 derniers caractères hexadecimal)
- 4) Conversion avec compléments à deux pour passer du format hexadécimal au binaire
- 5) Déduction et affichage de la distance par rapport à cette valeur.

Le programme fonctionne uniquement si le port est libre et qu'aucune autre application lis sur le même port en même temps.

! Il faut impérativement modifier la valeur du port sur lequel est branché votre récepteur, pour cela vérifiez dans votre **gestionnaire de périphérique** pour trouver le port

correspondant au **KitProg USB-UART**



Sur le Raspberry Pi, vu qu'il s'agit d'une distribution différente de Windows, il faut renseigner le chemin vers ce port, pour le trouver faut aller voir dans le répertoire suivant :
/dev/serial/by-id/

Normalement un fichier correspondant au port est présent avec un nom du style :

```
usb-Cypress_Semiconductor_Cypress_KitProg_BLE0A190B1702016400-if02
```

Il suffit de renseigner le chemin complet dans la variable **port** de la fonction **lecture_port()**

! Le reste du programme est disponible sur [Etulab](#) et dans l'archive avec lequel vous avez trouvé ce fichier, son nom est : **BLE.py**

Configuration du Raspberry :

1) Installer paquet pyserial

```
python3 -m pip install pyserial
```

2) Désinstaller impérativement serial, sinon conflit de nom de paquet

```
python3 -m pip uninstall serial
```

Afin de rendre l'accès au programme plus simple pour les joueurs nous avons fait un **script.sh** qui lance le fichier caché du programme : **.BLE.py**

Pour cela on change juste le nom du fichier

```
mv BLE.py .BLE.py  
chmod +x .BLE.py
```

Voici le contenu du script:

```
#!/bin/bash  
python3 ./BLE.py
```

Le joueur n'a plus qu'à exécuter le script présent sur le bureau pour lancer le scan d'émission BLE.

2) Radio pirate

Le principe est de diffuser sur une fréquence radio précise, un long message préenregistré.

Coté organisateur

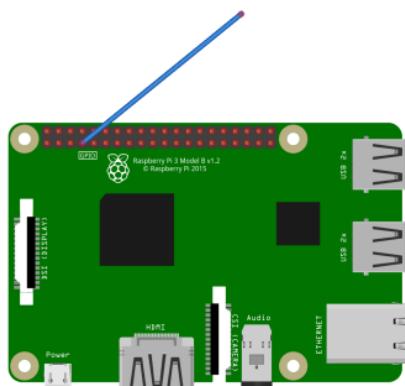
Nous avons besoins de 3 équipements :

- **Raspberry Pi** : Logiciel d'émission
- **Cable** : Emission physique

Nous avons aussi besoin d'un **fichier audio** que nous faisons nous-même. Pour cela, on peut soit utiliser **audacity** pour s'enregistrer et s'amuser à faire des choses ou utiliser un site qui génère une voix robotique à partir d'un texte.

Tout se passe sur le **Raspberry Pi** en réalité, son installation est assez simple. Voici les étapes résumées de l'installation complète :

- 1) Brancher le Raspberry Pi à un écran, clavier, souris et antenne



- 2) Connecter le Raspberry au réseau (peu importe)
- 3) Télécharger le logiciel d'émission radio **pi_fm_rds**, pour cela on a besoin de git

```
apt-get install git
apt-get install libsndfile1-dev
git clone https://github.com/ChristopheJacquet/PiFmRds.git
cd PiFmRds/src
make clean
make
```

4) Lancer le programme pi_fm_rds pour émettre

```
pi_fm_rds -ps escapeFM -audio enigme.wav -freq 107M -rt "message" -ppm  
10000000
```

Explication des options :

- **ps** : Nom de la station (8 caractères max)
- **audio** : Nom du fichier audio qui sera émis en boucle, obligatoirement en .wav
- **freq** : La fréquence sur laquelle on émet, prendre une de libre entre **76** et **108**
- **rt** : Message personnalisé diffusé
- **ppm** : Vitesse de diffusion, 1 million sinon lenteur

Coté joueur

De son côté le joueur a besoins de 3 équipements :

- **Un poste linux** : Avec **GQRX** pour capter les ondes radio
- **Clé -T + Dab + FM + RTL**: Réception sur poste de la radio

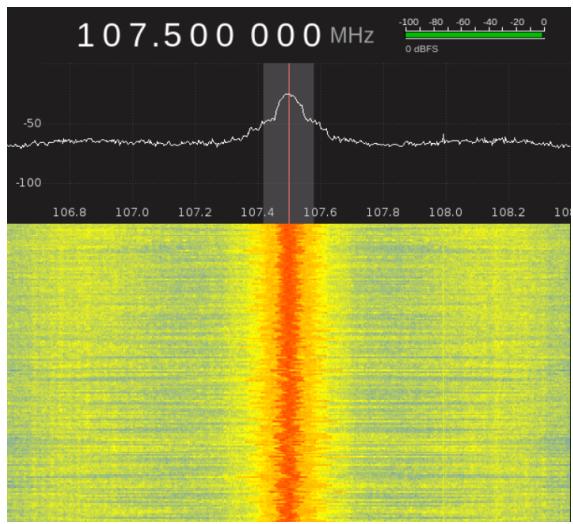


L'installation est très simple et voici les différentes étapes :

- 1) Brancher cle FM sur port USB
- 2) Lancer application GQRX

```
apt-get install gqrx-sdr  
gqrx
```

- 3) Activer RDS : View/RDS/Enable RDS
- 4) Sélectionner la bonne fréquence



Voilà toutes les informations nécessaires pour jouer au jeu mais, il s'agit d'un résumé rapide. Pour plus de précision Mr. Février à mis à disposition un guide avec plus de détails si vous avez des interrogations.

3) Trouver l'intru

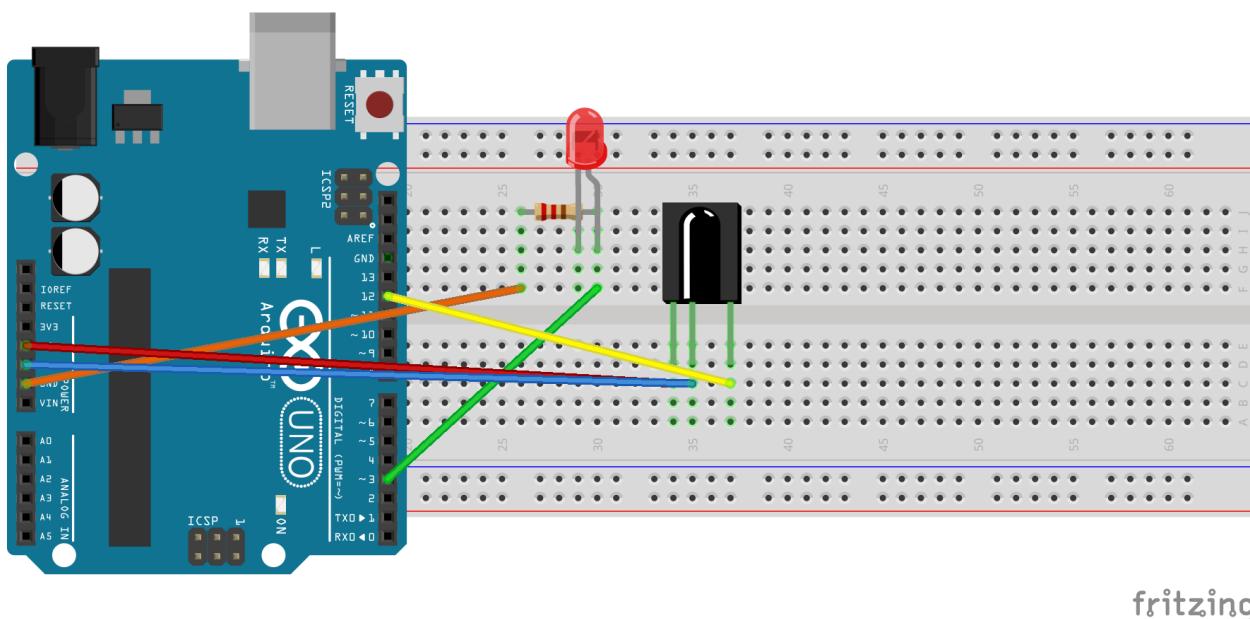
Le but de ce mini-jeu est de cacher dans une armoire un ensemble de plusieurs boîtes, mais seulement l'une d'entre elle possède l'indice pour passer à l'étape suivante.

Dans une des boîtes nous avons disposés un Arduino wifi rev 2 avec un câblage bien particulier.

Il est disposé d'un capteur infrarouge qui possède une sensibilité assez faible donc il faut être assez proche pour que la diode à l'intérieur s'active. Le joueur doit via une télécommande qu'il aura trouvé passer devant chacune d'elles et voir laquelle réagit aux infrarouges. Nous aurions aimé qu'une fois la diode allumée l'Arduino envoie des données wifi qui seraient affichées sur le page de l'épreuve. Par manque de temps, seul la technologie infrarouge est fonctionnel

Vu que l'Arduino est dans une boîte hermétique sans accès extérieur, nous avons besoin d'une batterie externe afin d'auto alimenter l'Arduino pendant la durée de l'épreuve.

Voici le schéma du câblage à faire sur le Arduino :



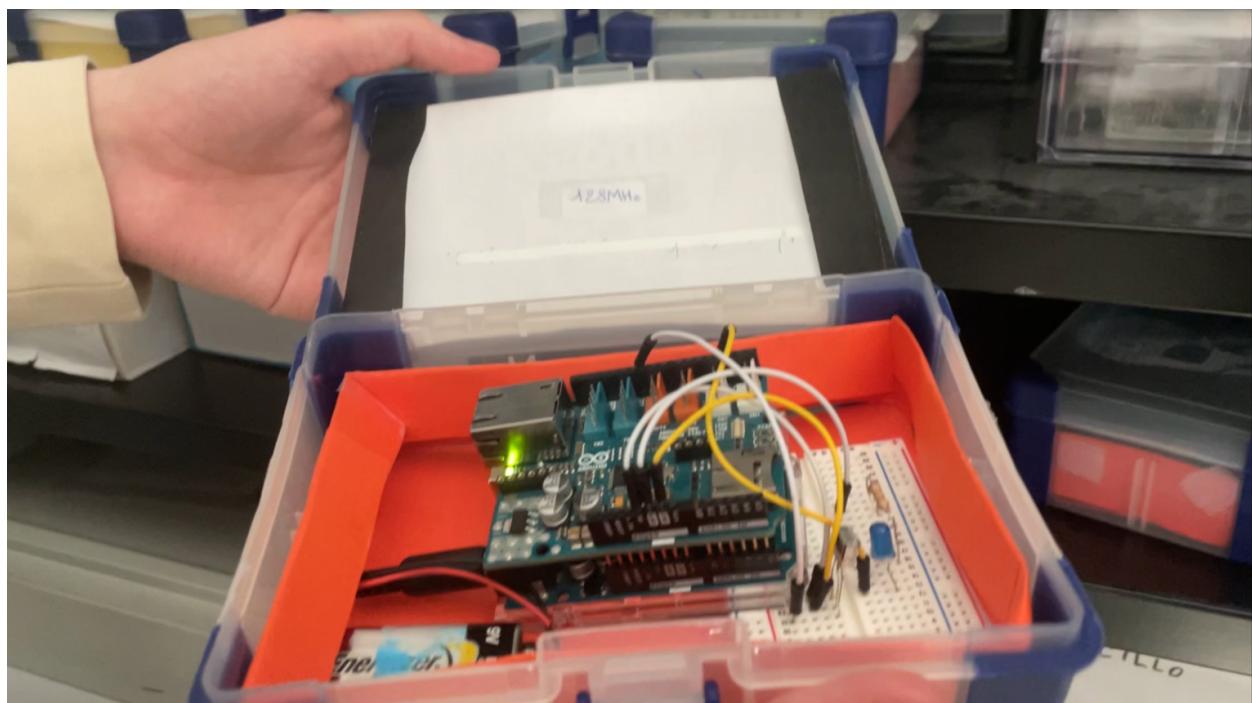
Le capteur noir représente le capteur infrarouge(pas trouvé d'équivalent sur le logiciel)

Il suffit juste de déposer l'arduino alimenté via sa batterie ou pile dans une boîte avec le câblage et l'histoire est jouée.

Il y a un code arduino qui va avec qui nous permet de savoir quand le récepteur reçoit des informations de la télécommande, malheureusement pour des raisons qu'on ignore, les valeurs sont inconsistantes, mais toujours présentes(on appuie, on reçoit bien). Dans notre cas c'est largement suffisant pour notre jeu, car peu importe la valeur de la touche appuyé la LED s'allumera si le joueur appuie.

Le code Arduino est aussi disponible dans le répertoire déposé sur Ametice sous le nom de : **intru.ino**

! Il est important de noter qu'il faut au préalable transverser le programme sur l'Arduino sinon ça ne marchera pas !



4) Messagerie Asterisk

L'objectif ici va être de connecter un smartphone en Bluetooth à notre serveur asterisk pour mettre en place une messagerie personnalisée (ici un message personnalisé en morse) lorsqu'un appelant extérieur téléphone au smartphone connecté au serveur.

Configuration du serveur Asterisk sur le Raspberry :

- 1) Installer les paquets

```
sudo apt install asterisk  
sudo apt install asterisk-mobile
```

- 2) Récupérer l'ID (souvent c'est "hci0") et l'@MAC de notre contrôleur Bluetooth et du téléphone

```
sudo hcitool dev  
root@raspberrypi:~# sudo hcitool dev  
Devices:  
        hci0      DC:A6:32:F3:88:DD  
root@raspberrypi:~#
```

- 3) Récupérer l'adresse MAC et le nom du téléphone

```
bluetoothctl  
scan on  
devices  
[Xiaomi 11T]# devices  
Device 70:5F:A3:C3:8B:20 Xiaomi 11T  
Device 6C:D8:1C:10:E2:61 6C-D8-1C-10-E2-61
```

- 4) Modifier fichier modules.conf

```
sudo nano /etc/asterisk/modules.conf
```

5) Remplir le fichier avec ces paramètres

```
[modules]
autoload=yes
noload => pbx_gtkconsole.so
noload => pbx_kdeconsole.so
noload => app_intercom.so
noload => chan_modem.so
noload => chan_modem_aopen.so
noload => chan_modem_bestdata.so
noload => chan_modem_i4l.so
noload => chan_capi.so
load => res_musiconhold.so
noload => chan_alsa.so
load => chan_console.so
noload => cdr_sqlite.so
noload => app_directory_odbc.so
noload => res_config_odbc.so
noload => res_config_pgsql.so
load => chan_mobile.so
noload => chan_oss.so
load => chan_alsa.so
[global]
```

6) Modifier fichier extensions.conf

```
sudo nano /etc/asterisk/extensions.conf
```

7) Remplir le fichier avec ces commandes

```
[default]
exten => s,1,Answer()
exten => s,2,Playback(tt-monkeys)
exten => s,3,Wait()
exten => s,4,Hangup()
exten => _0.,1,Dial(Mobile/telephone/${EXTEN},42)
[incoming-mobile]
exten => s,1,Answer
exten => s,2,Playback(tt-monkeys)
exten => s,2,MorseCode(victoire) ; Solution de l'enigme
```

! Remplacez "telephone" par le nom du votre, il faut qu'il concorde avec les autres noms dans les autres fichiers

! [incoming-mobile] est utilisé lorsque un appel est reçu, on peut y ajouter des lignes pour personnaliser une messagerie avec un son custom et des sous-menus pour ajouter des sons .gsm c'est dans **/usr/share/asterisk/sounds/en**

8) Modifier fichier chan_mobile.conf

```
sudo nano /etc/asterisk/chan_mobile.conf
```

9) Remplir les lignes suivantes à l'intérieur

```
[adapter]
id=hci0 ;Modifiable
address= FF:FF:FF:FF:FF:FF ;Modifiable
context=incoming-mobile
;forcemaster=yes
;alignmentdetection=yes

[telephone] ;Modifiable
address=FF:FF:FF:FF:FF:FF ;Modifiable
port=3
context=incoming-mobile
adapter=hci0 ;Modifiable
```

10) Modifier fichier sip.conf

```
sudo nano /etc/asterisk/sip.conf
```

11) Remplir les lignes suivantes

```
[telephone] ;Modifiable
type=friend
address= FF:FF:FF:FF:FF:FF ;Modifiable
groupe=1 ;À modifier
context=Bluetooth
```

12) Redémarrer service Asterisk

```
sudo systemctl restart asterisk
```

13) Lancer Asterisk

```
sudo asterisk -rvvvvvvvv
```

```
root@raspberrypi:~# asterisk -rvvvvvvvvvvvvv  
Asterisk 16.28.0-dfsg-0+deb11u1, Copyright (C) 1999 - 2021, Sangoma Technologies Corporation and others.  
Created by Mark Spencer <markster@digium.com>  
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.  
This is free software, with components licensed under the GNU General Public  
License version 2 and other licenses; you are welcome to redistribute it under  
certain conditions. Type 'core show license' for details.  
=====  
Connected to Asterisk 16.28.0-dfsg-0+deb11u1 currently running on raspberrypi (pid = 2003)  
raspberrypi*CLI> module load chan_mobile.so  
Loaded chan_mobile.so  
[Jan 11 11:24:30] WARNING[2280]: chan_mobile.c:3166 sdp_register: Failed to connect sdp and create session.  
== Registered channel type 'Mobile' (Bluetooth Mobile Device Channel Driver)  
== Registered application 'MobileStatus'  
== Registered application 'MobileSendSMS'  
Loaded chan_mobile.so => (Bluetooth Mobile Device Channel Driver)  
-- Bluetooth Device xiaomi has connected, initializing...  
-- Bluetooth Device xiaomi initialized and ready.  
raspberrypi*CLI>
```

14) Vérifier que le module chan_mobile.so est chargé :

```
module load chan_mobile.so
```

! Si vous avez une erreur à ce moment, c'est qu'il faut déconnecter le smartphone et redémarrer (il faut charger le module avant de connecter le smartphone). Ensuite, il ne reste plus qu'à connecter en Bluetooth le smartphone avec pair et connecté comme montré en début de section. Vous pouvez maintenant essayer d'appeler le téléphone connecté.

Connexion du téléphone en Bluetooth :

1) Connexion au Bluetooth avec le téléphone depuis le terminal:

```
bluetoothctl  
show #permet d'afficher l'état du contrôleur  
discoverable yes  
pair @MAC  
connect @MAC
```

2) Accepter l'appairage si c'est demander en tapant yes sur le terminal

Vous pouvez vous connecter directement en passant par votre téléphone et en sélectionnant votre appareil dans lequel tourne votre asterisk

Maintenant, vous pouvez passer un appel et il sera réceptionné par Asterisk.

V) Serveur de jeu

Le serveur de jeu a pour rôle d'hébergeur du site web sur lequel le joueur va pouvoir rentrer les réponses et avancer dans le jeu.

Pour cela, on utilise le service Apache2 qui va nous permettre de l'héberger le rendre accessible à partir de l'IP de la machine host.

Configuration apache2

Son installation est très simple et se fait quelques étapes :

- 1) Installer le paquet apache2

```
apt-get install apache2
```

- 2) Activer apache à chaque démarrage

```
systemctl enable apache2
```

- 3) Placer tous les fichiers du site (.html,.css) dans **/var/www/html/**

```
mv rep_fichier_site/* /var/www/html/
```

- 4) Modifier fichier de configuration **/etc/apache2/site-available/000-default.conf**

```
<VirtualHost *:80>
    ServerName escape.com
    ServerAlias www.escape.com
    DocumentRoot "/var/www/html/"
    ErrorLog /var/log/apache2/error.escape.com.log
    CustomLog /var/log/apache2/access.escape.com.log combined
</VirtualHost>
```

- 5) Activer site

```
a2ensite 000-default.conf
```

Le site est accessible à partir de l'ip de la machine ex : 172.16.11.20, il suffit d'aller sur un navigateur via une machine dans le même réseau et de rentrer

```
http://172.16.11.20
```

Gestion des logs

De ce fait, en notant l'ip de chaque groupe de joueurs, nous pouvons déterminer sur quelles pages ils sont ainsi que leurs actions.

```
root@debian:/var/log/apache2# cat access.log
root@debian:/var/log/apache2# cat access.log
127.0.0.1 - - [15/Jan/2023:00:39:39 +0100] "GET / HTTP/1.1" 200 3380 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
127.0.0.1 - - [15/Jan/2023:00:39:39 +0100] "GET /favicon.ico HTTP/1.1" 404 487 "http://localhost/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
127.0.0.1 - - [15/Jan/2023:00:39:40 +0100] "GET / HTTP/1.1" 200 3379 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
root@debian:/var/log/apache2# █
```

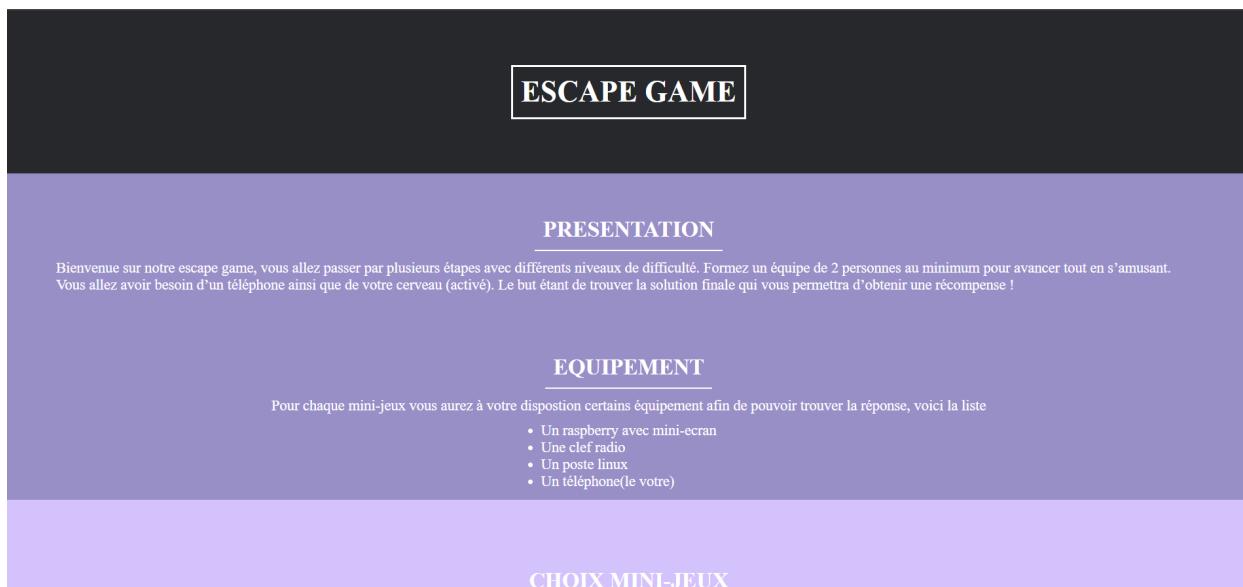
De ce fait, en notant l'ip de chaque groupe de joueurs, nous pouvons déterminer sur quelles pages ils sont ainsi que leurs actions.

Interface du site

Le site web est divisé en 6 pages principales :

- 1) **index.html** : Page d'accueil du site avec la présentation du contexte, la liste des équipements à avoir ainsi que l'accès + présentation rapide des mini-jeux
- 2) **mini_jeux_X.html** : Page du mini jeu sélectionné avec une consigne générale permettant de mettre le joueur sur la voie + champ d'entrée pour la réponse à l'énigme + bouton indice
- 3) **final.html** : Accessible uniquement après la résolution de la dernière énigme avec écran de fin, félicitant le joueur !

Page d'accueil :



The home page features a dark header with the text "ESCAPE GAME" in a white box. Below it is a purple section containing the title "PRESENTATION" and a welcome message. A list of equipment is provided, followed by a "CHOIX MINI-JEUX" section.

PRESENTATION

Bienvenue sur notre escape game, vous allez passer par plusieurs étapes avec différents niveaux de difficulté. Formez un équipe de 2 personnes au minimum pour avancer tout en s'amusant. Vous allez avoir besoin d'un téléphone ainsi que de votre cerveau (activé). Le but étant de trouver la solution finale qui vous permettra d'obtenir une récompense !

EQUIPEMENT

Pour chaque mini-jeux vous aurez à votre disposition certains équipement afin de pouvoir trouver la réponse, voici la liste

- Un raspberry avec mini-écran
- Une clé radio
- Un poste linux
- Un téléphone (le vôtre)

CHOIX MINI-JEUX



The interface shows four colored boxes labeled "ETAPE 1", "ETAPE 2", "ETAPE 3", and "ETAPE 4". The first box is dark blue, the second is dark red, the third is maroon, and the fourth is light red. To the right of the boxes is a dark blue column with the heading "ETAPE 1" and descriptive text. At the bottom, there is a blue footer with contact information.

CHOIX MINI-JEUX

ETAPE 1 ETAPE 2 ETAPE 3 ETAPE 4

ETAPE 1

Vous allez utiliser la technologie BLE ainsi que de votre environnement afin de retrouver la cache secrète

[Commencer](#)

En cas de soucis technique, adressez-vous à FOURNIER Jérémie

Page mini-jeu :

The screenshot shows a dark-themed web page for an escape game. At the top, a black header bar contains the text "ESCAPE GAME" in white, bold letters, and a small "Accueil" link. Below the header, a purple section is titled "Enigme 1". It includes a welcome message and three bullet points describing the task: intercepting BLE signals, equipping a raspberry Pi, and validating the answer. A red "FAUX" message is displayed below the input field. The input field itself is empty.

Enigme 1

Bienvenue sur votre première énigme

- Nous avons intercepté des ondes dans la salle, il semblerait qu'il s'agisse de trame BLE, mais cela reste à vérifier..
- Equipez-vous du raspberry avec son mini-écran et essayez de trouver un moyen de trouver l'origine de l'émission
- Pour valider l'étape, rentrer la réponse qui vous semble la plus logique par rapport au déroulement de l'épreuve

Votre réponse

je sais pas :)

FAUX

This screenshot shows the same challenge page after an attempt. The "FAUX" message remains at the bottom. The input field now contains the word "FAUX".

Votre réponse

je sais pas :)

FAUX

This screenshot shows the challenge page again. The "FAUX" message is still present. The input field now contains "FAUX". A green button labeled "Indice" is visible on the left, with the text "Sur le bureau du raspberry :)" to its right.

Indice

Sur le bureau du raspberry :)

En cas de soucis technique, adressez-vous à FOURNIER Jérémie

Page final :

Félicitations ! Vous avez réussi l'Escape Game !

Vous avez réussi à résoudre tous les défis



Merci d'avoir joué !

[PAGE D'ACCUEIL](#)

Comportement du site

Les pages des mini-jeux sont faites que pour quand le joueur rentre une mauvaise réponse, un texte avec marqué "faux" s'affiche et "vrai" si le joueur rentre la bonne réponse .

Si la réponse est bonne, le joueur reçoit une alerte javascript lui indiquant qu'il sera redirigé vers la page du prochain jeu. Le script est assez bien fait pour qu'il puisse, peu importe l'ip que vous avez, arrive à vous renvoyer sur la page

"http://XXXXXXXXXX/mini_jeu_suivant.html"

Exemple :

Votre réponse

toto Valider Effacer

FAUX



! Tout le code des pages sont disponibles dans l'archive sur Ametice et sur [Etulab](#)

V) Problèmes et améliorations

Tous les jeux présentés fonctionnent mais certaines améliorations potentielles seraient attendues, en voici un petit listing :

- 1) **Jeu de l'intru** : Que quand la LED s'allume, une transmission wifi se fasse vers le serveur apache et affiche en temps réel s'il y a réception sur la page du jeu. La partie la plus facile étant de transmettre des données wifi via Arduino, étant donné qu'on a comme matériel un arduino wifi rev2 dans notre groupe, la bibliothèque wifi NINA peut s'en occuper. Là où les idées deviennent compliquées et pour la réception sur le serveur et surtout la modification en direct sur la page, sans doute du PHP, mais on n'a pas eu le temps d'approfondir plus que ça cette partie.
- 2) **Chaud Froid** : Nous aurions aimé développer une interface kivy fonctionnelle sur le Raspberry avec une affiche spéciale en fonction de la distance, le programme est fait et fonctionne sur Windows, mais sur le Raspberry impossible actuellement de le faire fonctionner. Le second problème est la précision du capteur, nous avons du mal à déterminer s'il s'agit de l'équipement de base qui n'est pas très précis et qui possède une distance très faible. En effet, au-delà de 5m, le programme ne renvoie qu'une seule valeur et n'arrive pas à aller plus loin, néanmoins à courte distance (1-4 m) les valeurs sont très précises.

Concernant les autres mini-jeux, nous n'avons pas grand-chose d'autre à rajouter. Le seul dernier point sur lequel on pourrait s'attarder est le design du site, mais ça reste assez subjectif et chronophage quand on veut juste pouvoir jouer et entrer des informations dessus.

VI) Conclusion

Pour conclure, ce projet fut très intéressant pour chaque membre du groupe, il nous aura permis de mettre en place, d'expérimenter et debugger différents systèmes de transmissions.

Le résultat final est plutôt concluant de notre point de vue, car nous avons atteint tous les principaux objectifs que nous nous étions fixés, avec notamment le développement de 4 mini-jeux et un site web fonctionnel pas trop moche.

Nous avons tout fait pour essayer de rendre ça le plus attractif pour les joueurs avec des équipements qu'on n'a pas trop l'habitude de voir comme le Raspberry et son mini-écran intégré, un système de jeu non-punitif où chacun peut y aller à son rythme.