



**Botella--Broc  
Yohann**

# Ecran-TTPA

*Table Tennis Performance Analyser*  
2019

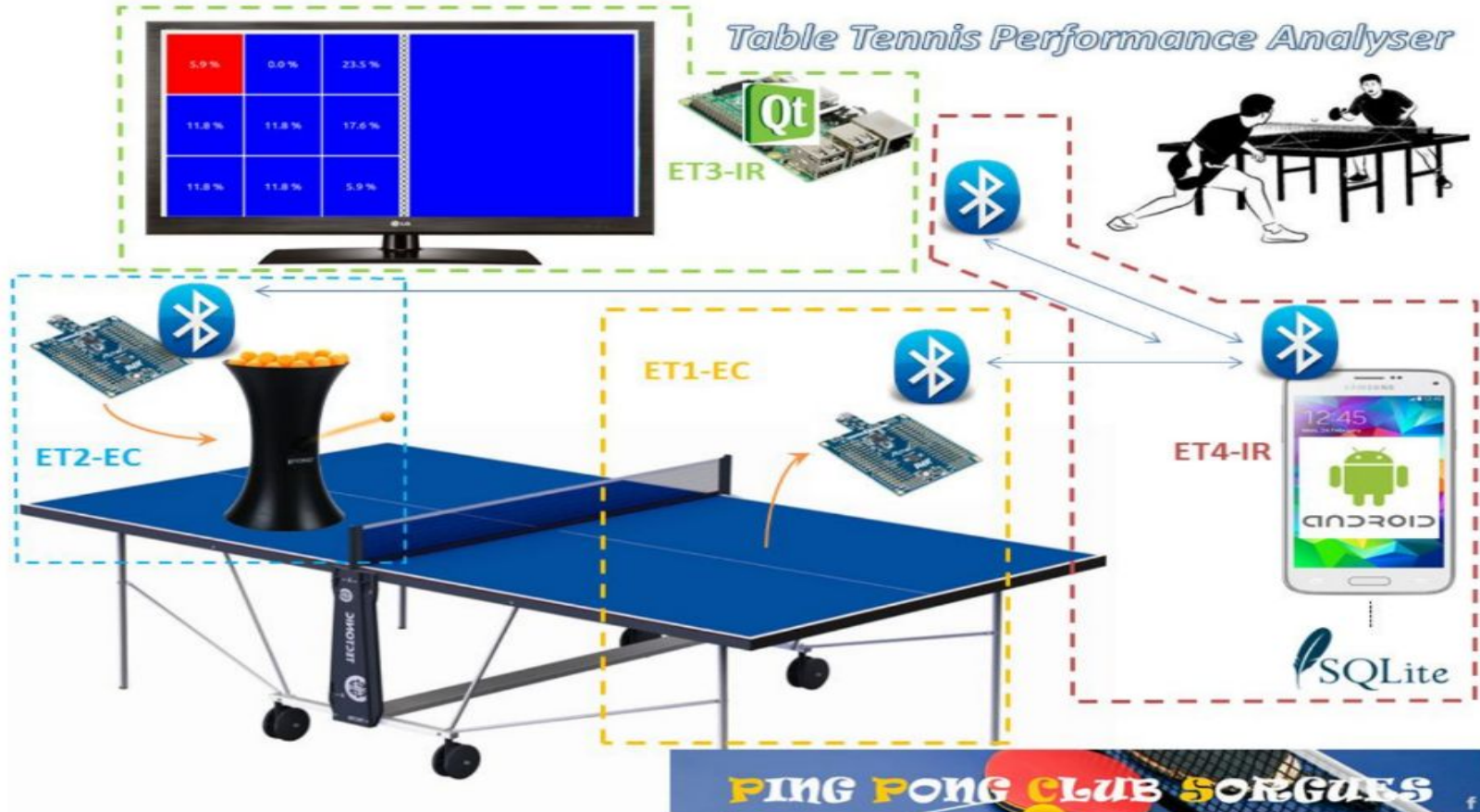
IR : HAMMOUMA Y. - BOTELLA--BROC Yohann

EC : CANONNE L. - MONTEIRO G.

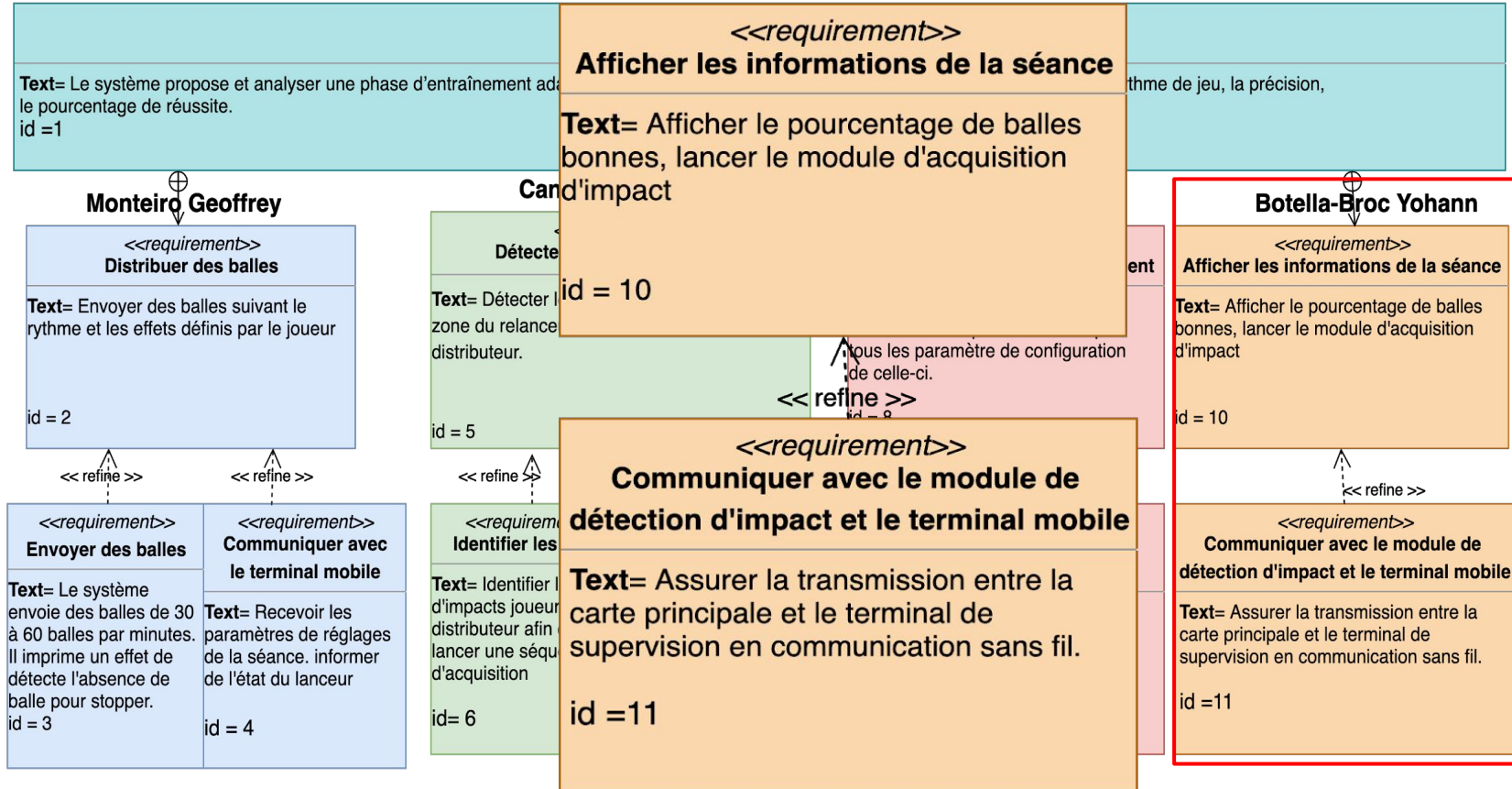


Systeme supportant le projet

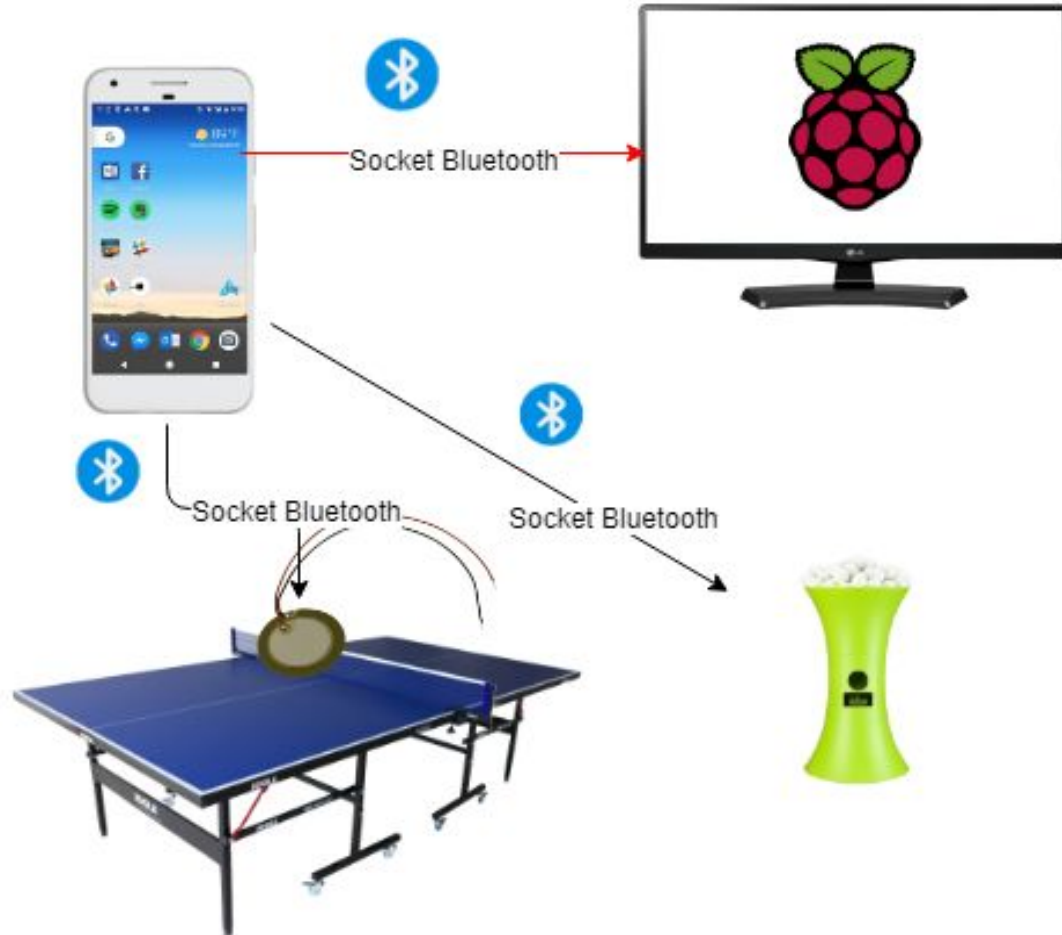
# Présentation général



# Diagramme des exigences

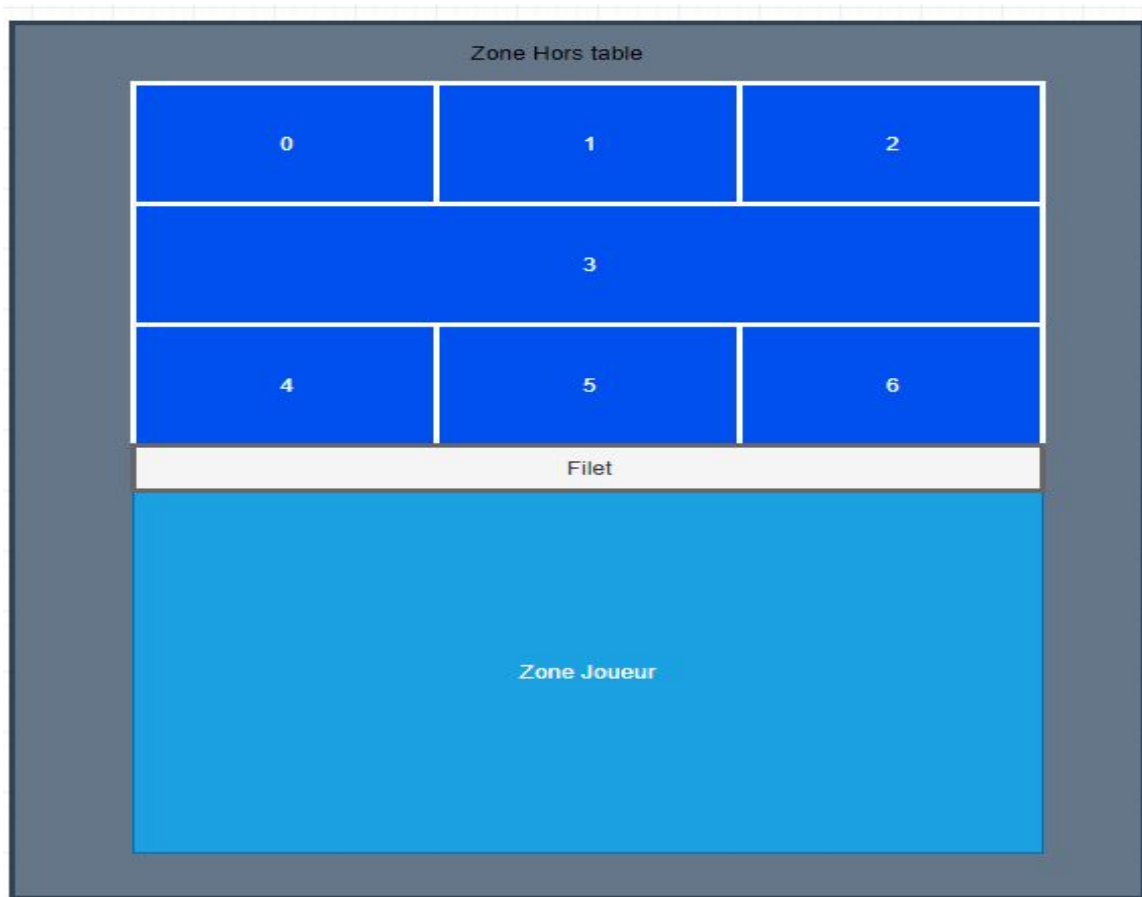


# Synoptique du système



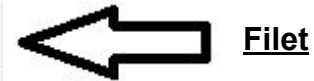


# Les Zones de la table TTPA



**-Le haut(zone 0 à 6)** est la partie de la table comptabilisant les zones

**-Le milieu (zone 3)**



**-Le bas la zone joueur**

# Répartition des tâches (IR)

## Botella-Broc Yohann (IR 1)

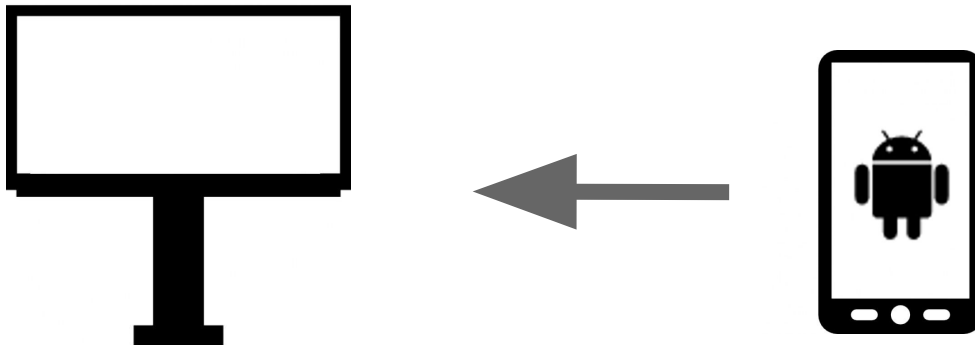
### *Module de visualisation de performance*

- Afficher un écran d'accueil
- Visualiser l'impact de balles dans les zones de la table
- Visualiser les données de la séance en temps réel et en fin de séances
- Dialoguer avec le terminal mobile

## Hammouma Youssef (IR 2)

### *Module de gestion de séance*

- Créer un profil de joueur
- Paramétrer une séance d'entraînement
- Enregistrer les données des séances
- Consulter l'historique des séances d'un joueur
- Purger les séances
- Dialoguer avec les modules



# Organisation commune au sein du projet

- Utilisation de [Subversion](#) pour les codes sources
- Utilisation d'un espace de stockage commun ([NAS](#) et Google [Drive](#)) pour tous les documents ressources
- Utilisation d'un [protocole de communication](#)



# Planification des tâches

The image displays a Kanban board for task planning, organized into three columns: 'Chose à faire' (To Do), 'En cours' (In Progress), and 'Terminer' (Completed). Each column contains task cards with descriptions, progress indicators, and due dates.

### Chose à faire

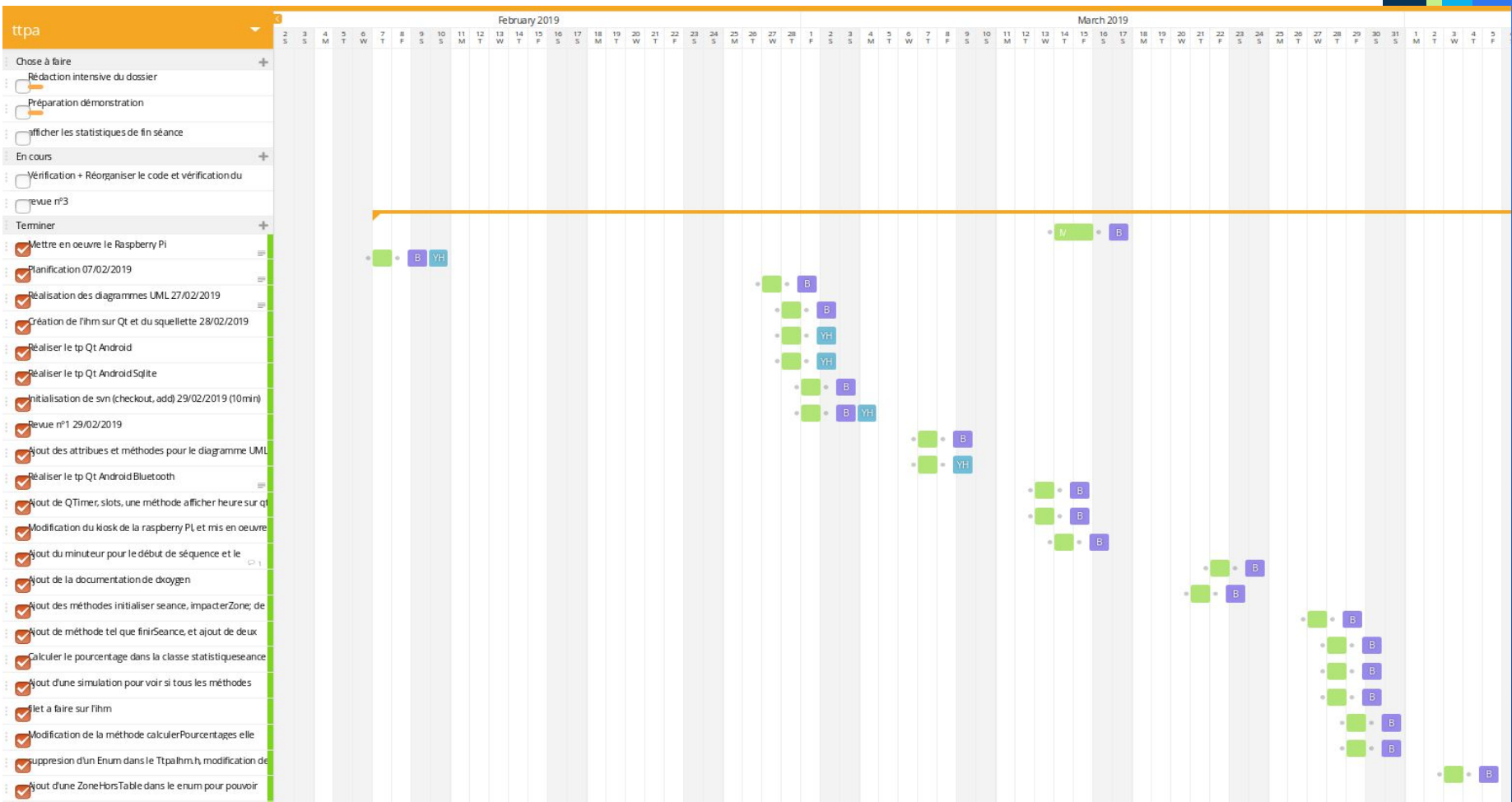
- Rédaction intensive du dossier (Progress: 10%, Due: B)
- Préparation démonstration (Progress: 10%, Due: B)
- afficher les statistiques de fin séance (Progress: 10%, Due: B)
- + Ajouter une autre carte

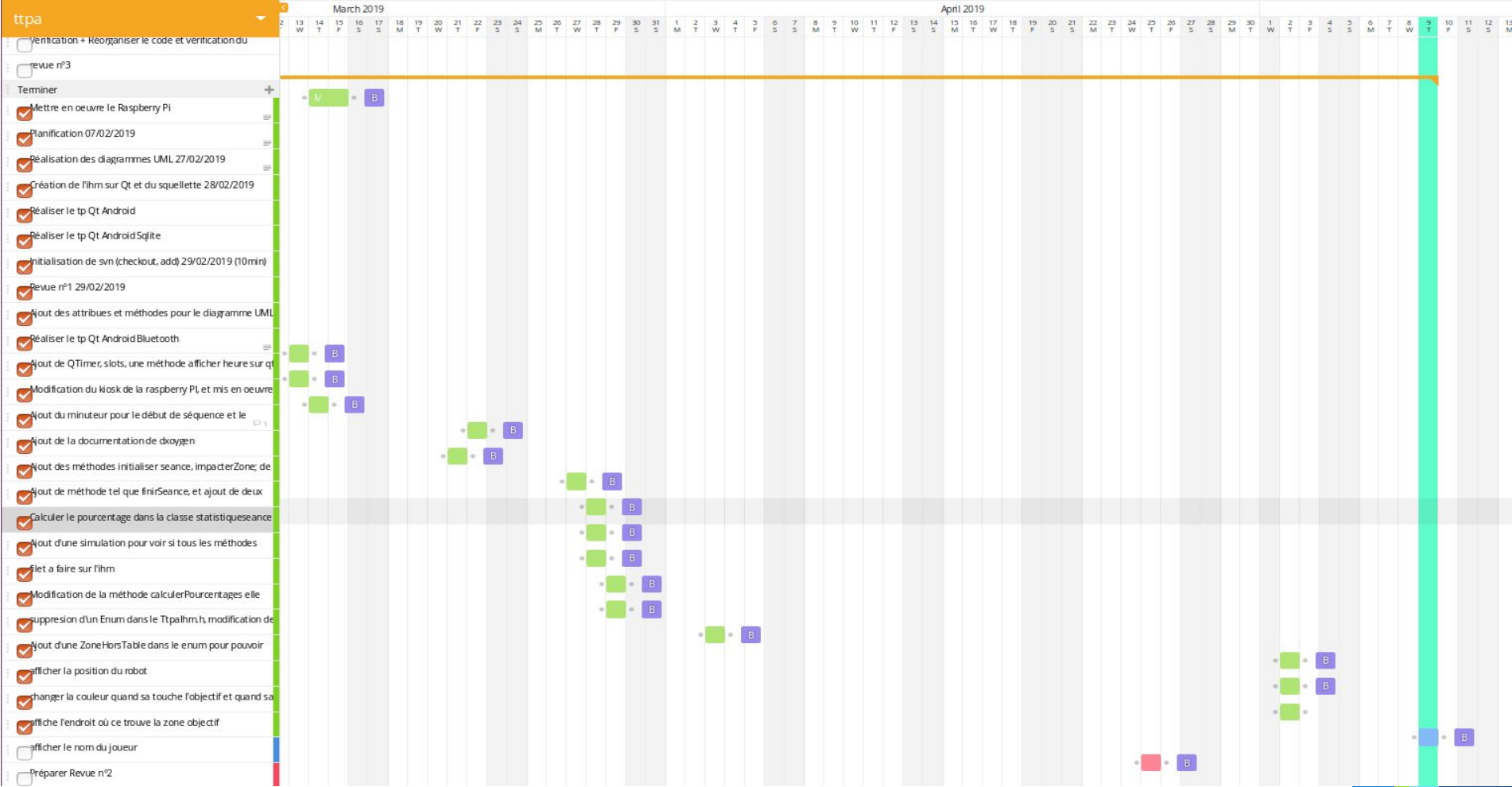
### En cours

- Vérification + Réorganiser le code et vérification du protocole (Progress: 10%, Due: B)
- revue n°3
- + Ajouter une autre carte

### Terminer

- suppression d'un Enum dans le Ttpalhm.h, modification des noms d'attributs correction du français, ajout de QDebug dans statistiquesséance.cpp pour la simulation (Progress: 100%, Due: 29 mars, B)
- Ajout d'une ZoneHorsTable dans le enum pour pouvoir aussi l'afficher et calculer le pourcentage (Progress: 100%, Due: 3 avr., B)
- afficher la position du robot (Progress: 100%, Due: 2 mai, B)
- changer la couleur quand sa touche l'objectif et quand sa touche la mauvaise zone (Progress: 100%, Due: 2 mai, B)
- affiche l'endroit où se trouve la zone objectif (Progress: 100%, Due: 2 mai)
- afficher le nom du joueur (Progress: 100%, Due: 9 mai, B)
- Ajout de la simulation random pour la zone robot et la zone objectif, ajout de décoder trame dans communicationbluetooth (Progress: 100%, Due: 9 mai)
- + Ajouter une autre carte

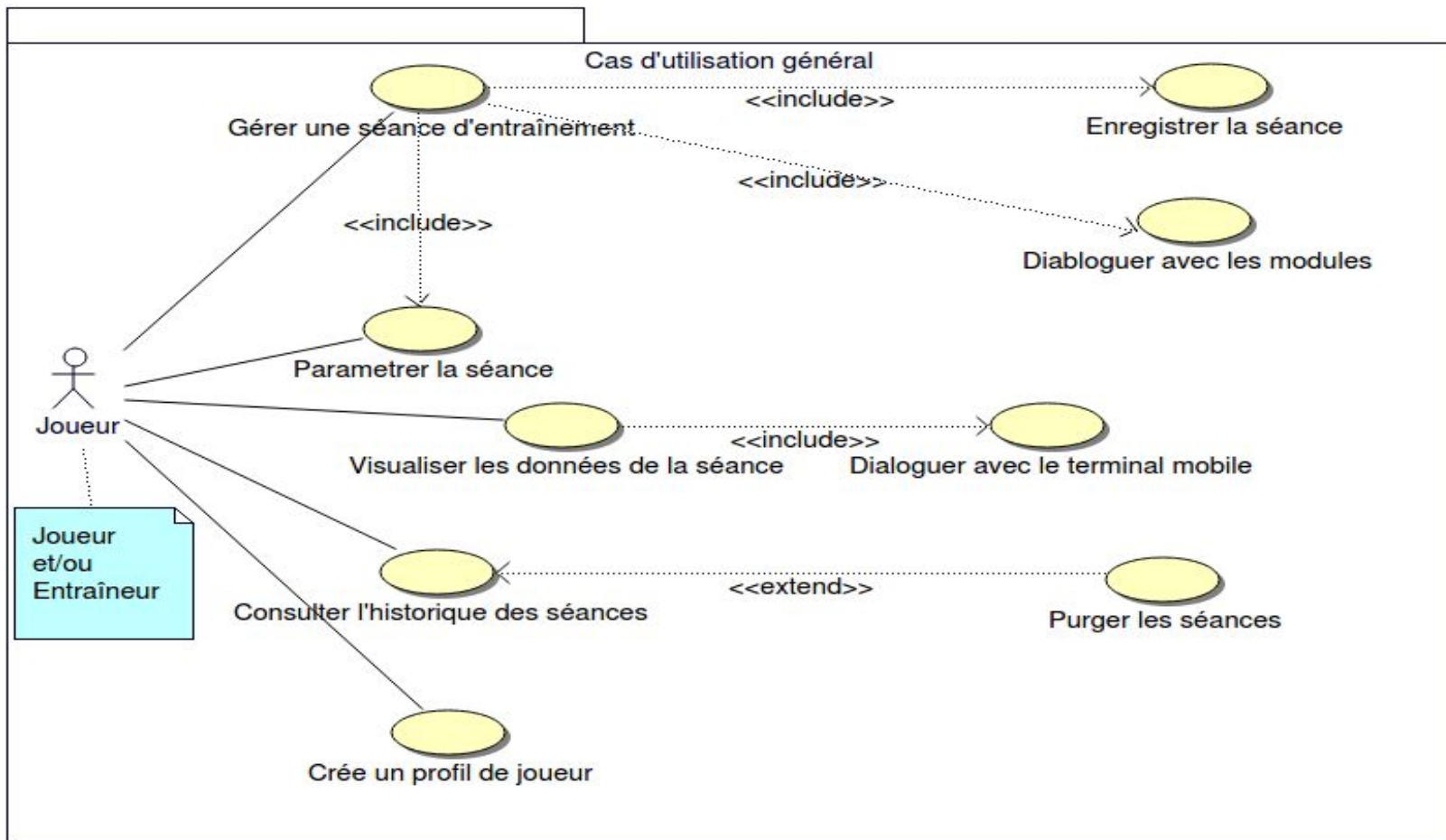




# Outils de développement

<i>Description</i>	<i>Version</i>
Système d'exploitation du poste de développement	Linux 4.8.0(Ubuntu 16.04)
Planification	Trello gantt + Trello
Journal de bord	RoadBook (Google Drive)
Diagrammes UML	BOUML 7.8
Gestion de versions	RiouxSvn (Subversion) 1.9.3 (r1718519)
Documentation de code	Doxygen 1.8.11
Langage utilisé	C++ / avec le framework Qt 5.11.2
Raspberry PI 3	Linux version 4.14.84-v7+

# Diagramme des cas d'utilisation

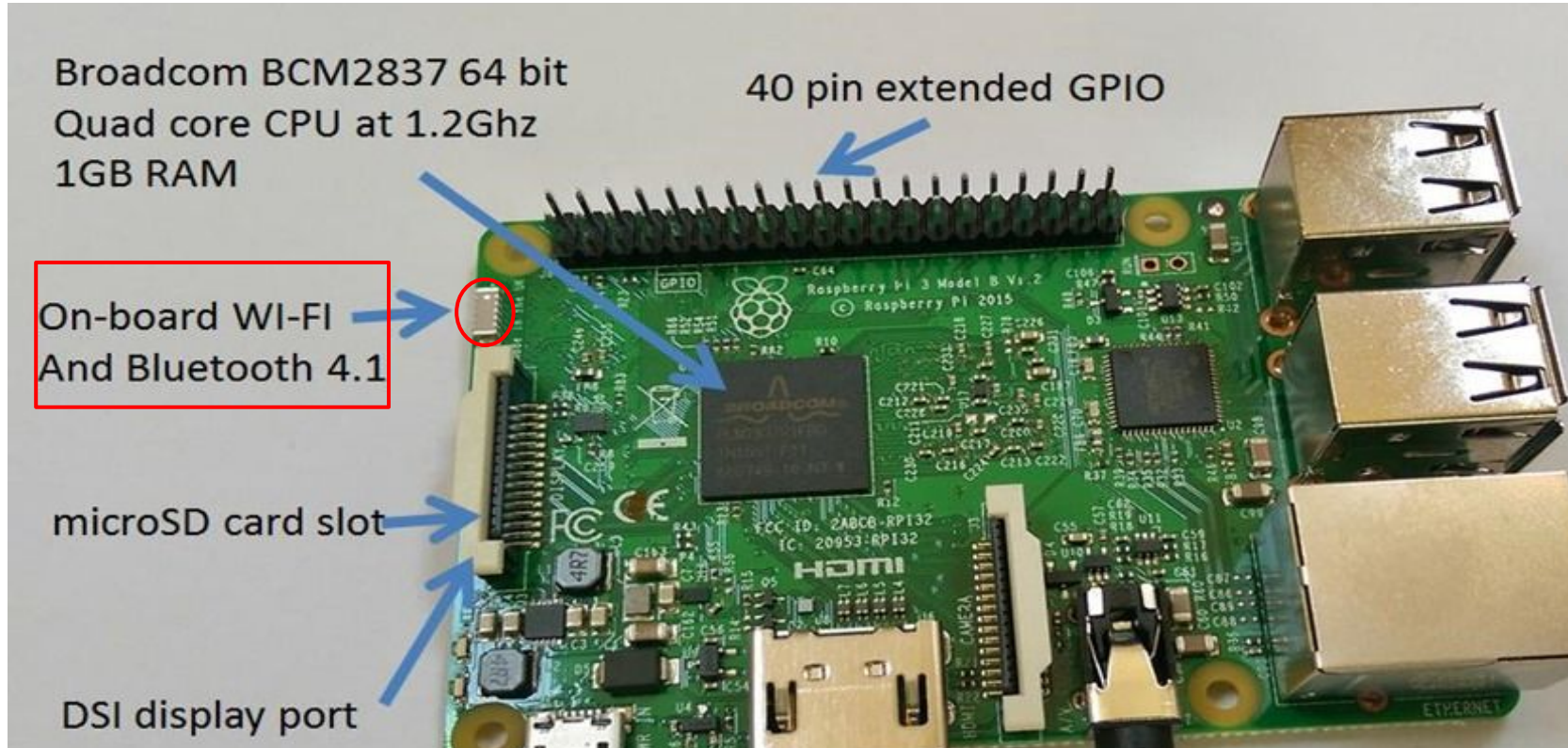


# Spécification du Raspberry Pi3

<i>Raspberry PI 3B</i>	<i>Caractéristiques</i>
Système d'exploitation	Raspbian GNU/Linux 9.1 (stretch)
Processeur	Quad Core 1.2GHz Broadcom BCM2837 ARMv8 CPU RAM1GB
Stockage	Micro SD
Sans-Fil	BCM43438 LAN et "Bluetooth Low Energy" (BLE)
Connectiques	40-pin GPIO, 4 ports USB 2, HDMI, CSI, DSI

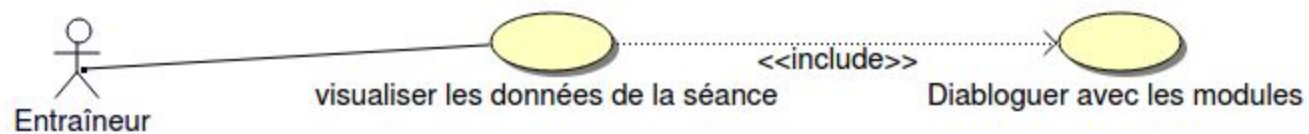


# Le Bluetooth de la raspberry Pi 3 B v1.2

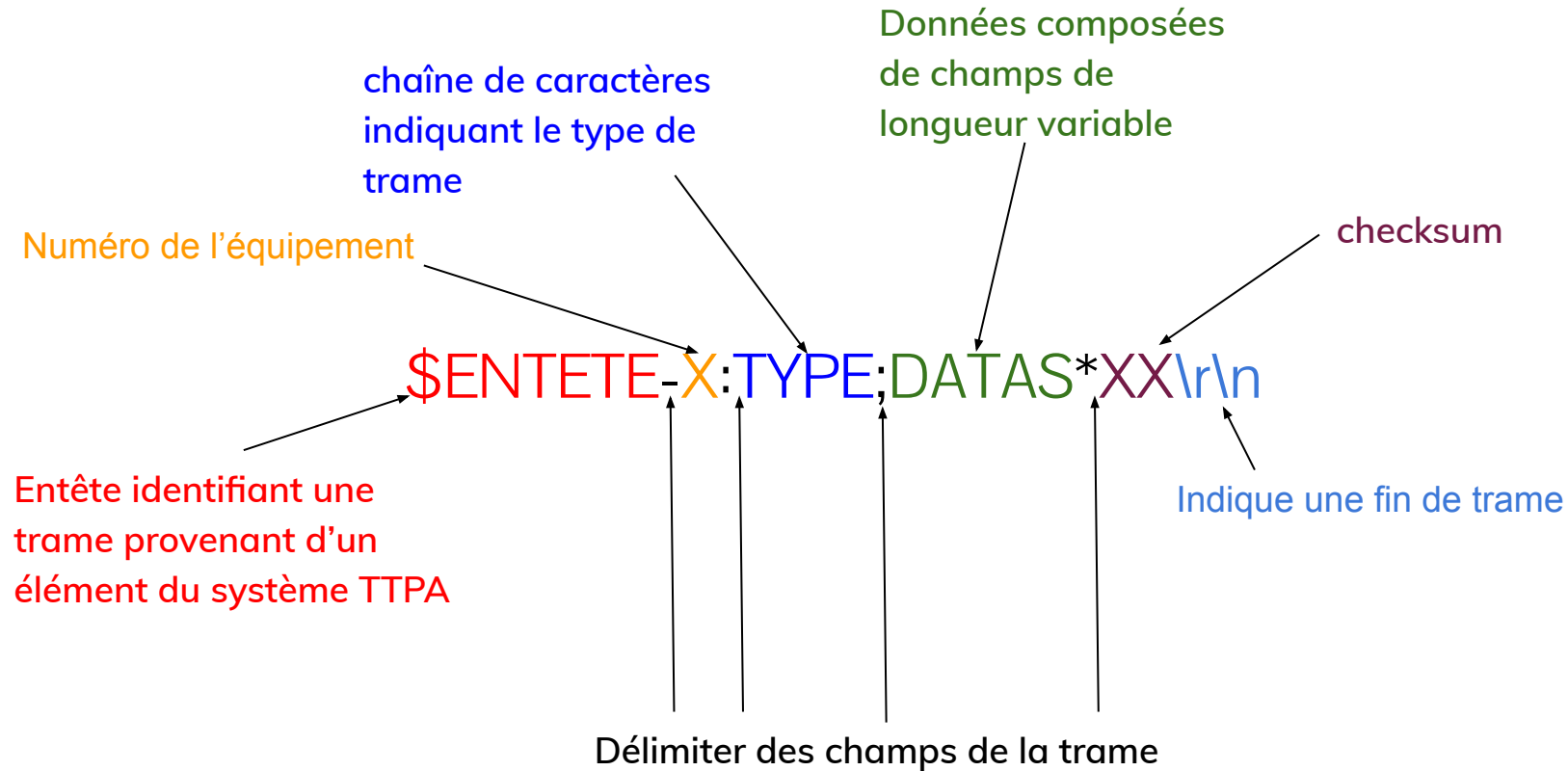




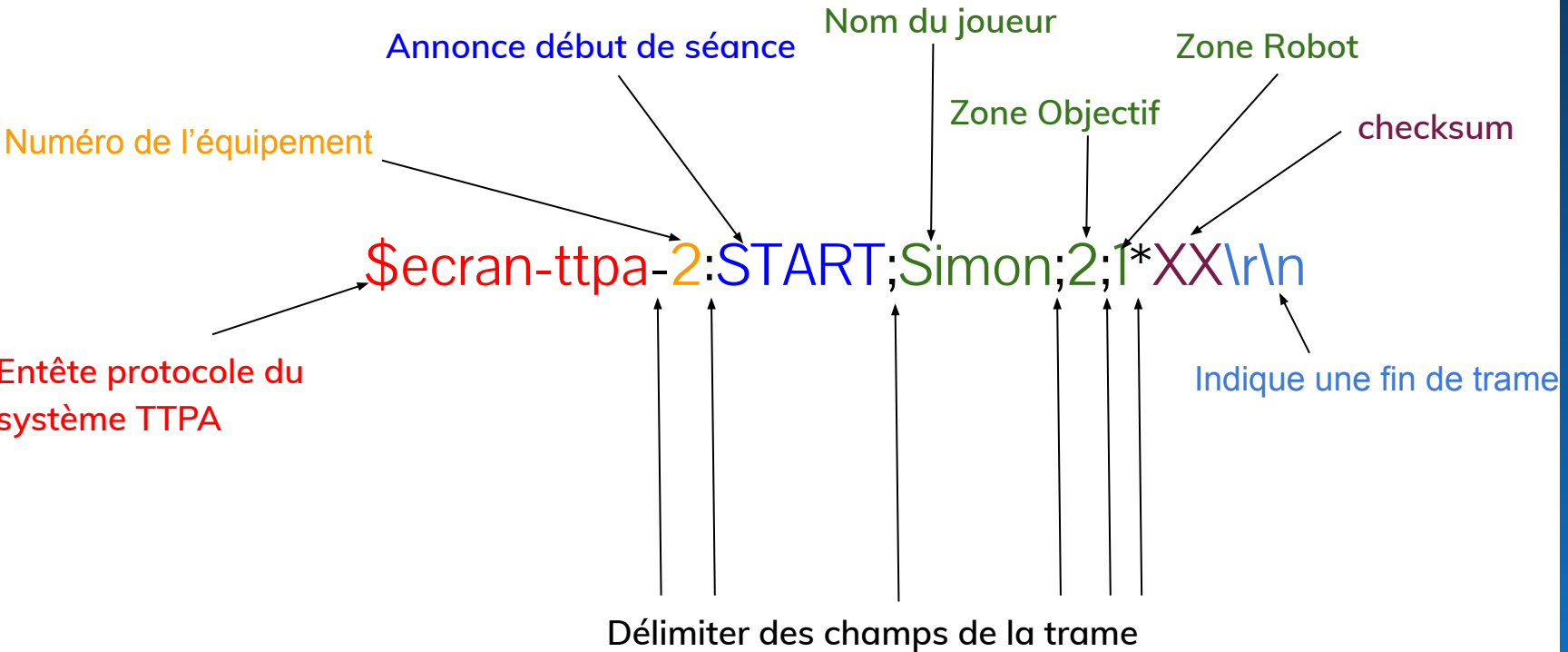
## Module de visualisation de performance



# Protocole de communication TTPA

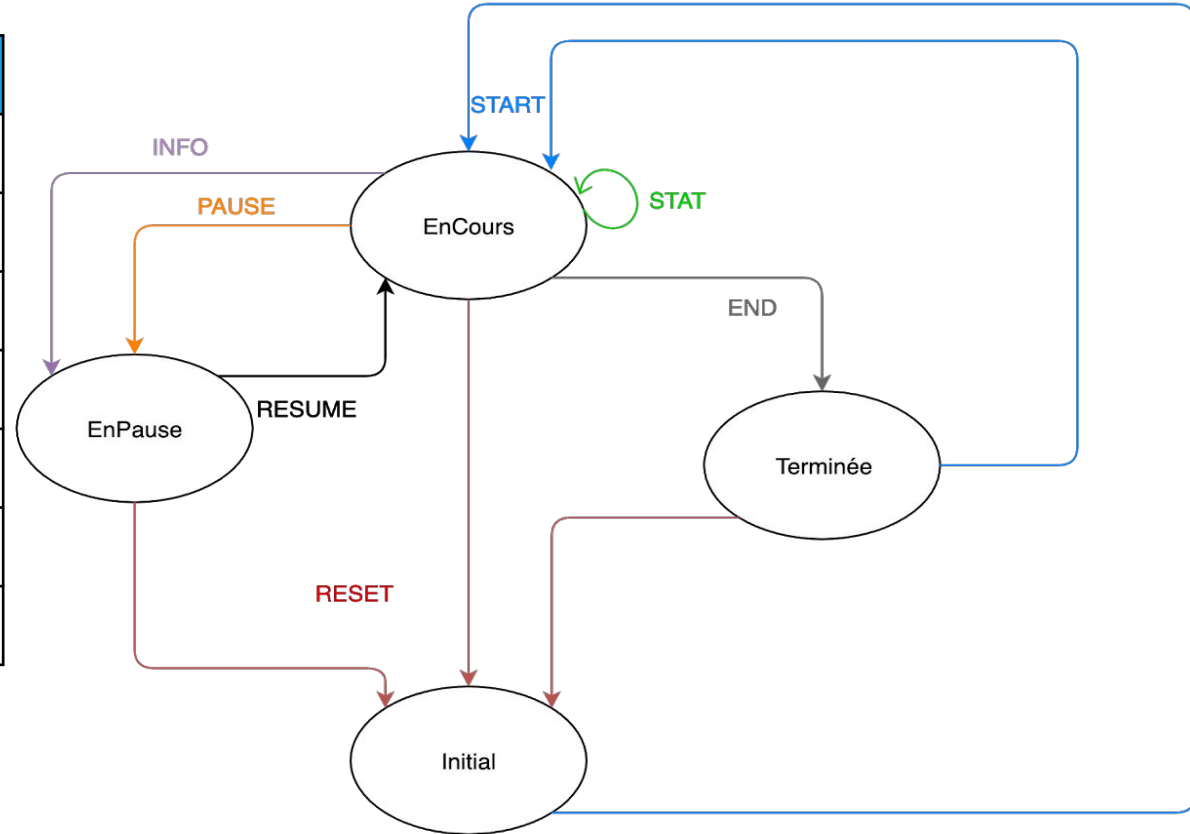


# Protocole TTPA



# État d'une séance

Trame
\$ecran-tpa-2:START
\$ecran-tpa-2:STAT
\$ecran-tpa-2:PAUSE
\$ecran-tpa-2:RESUME
\$ecran-tpa-2:END
\$ecran-tpa-2:RESET
\$ecran-tpa-2:INFO



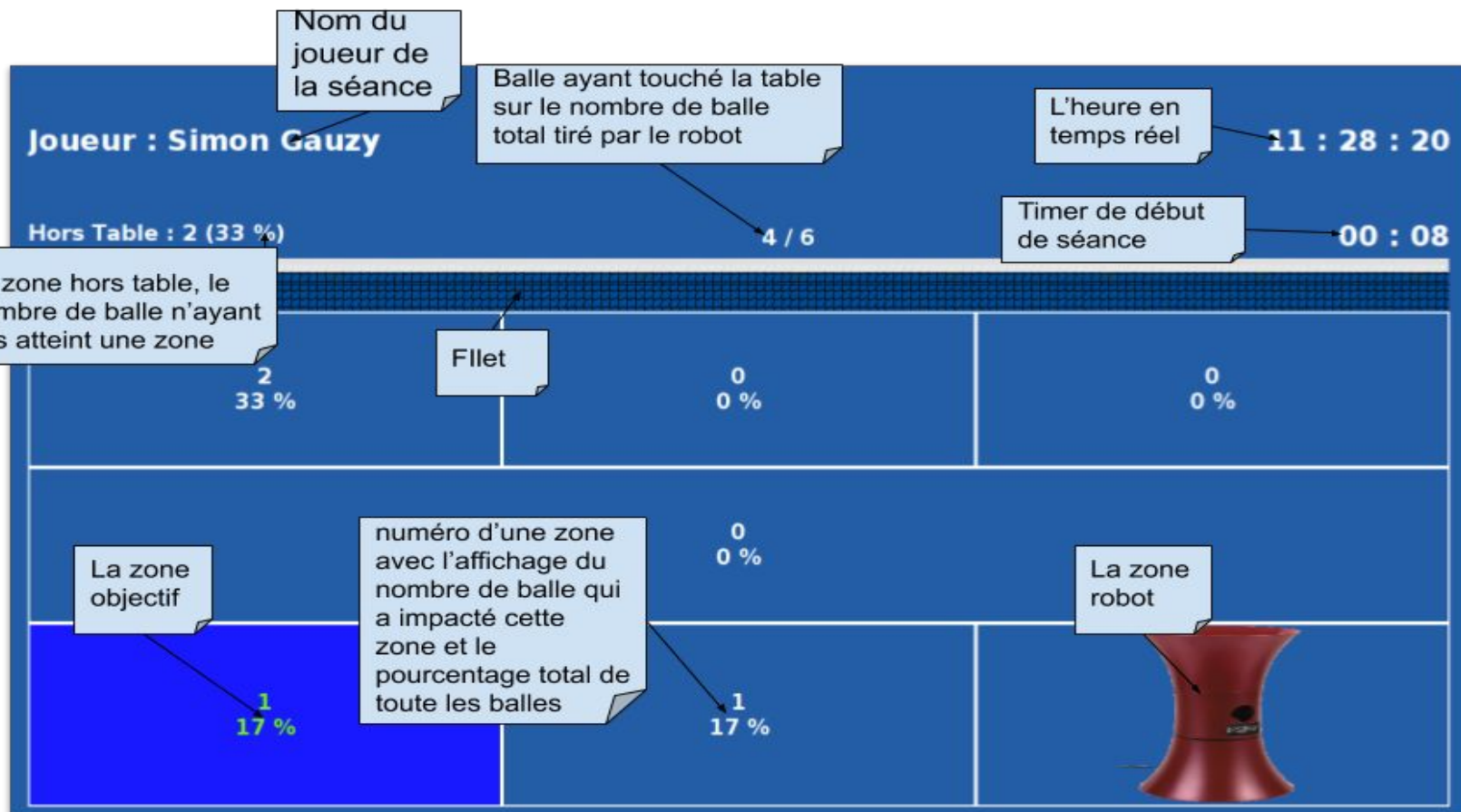


## L'écran d'attente

En attente de connexion

L'application  
attend la  
trame START

# L'écran principal




Joueur : Simon Gauzy

12 : 17 : 24

Hors Table : 0 (0 %)

5 / 5

00 : 27

0 0 %	0 0 %	0 0 %
5 100 %	0 0 %	

La zone objectif à été  
touché donc s'affiche  
d'une différentes  
couleur



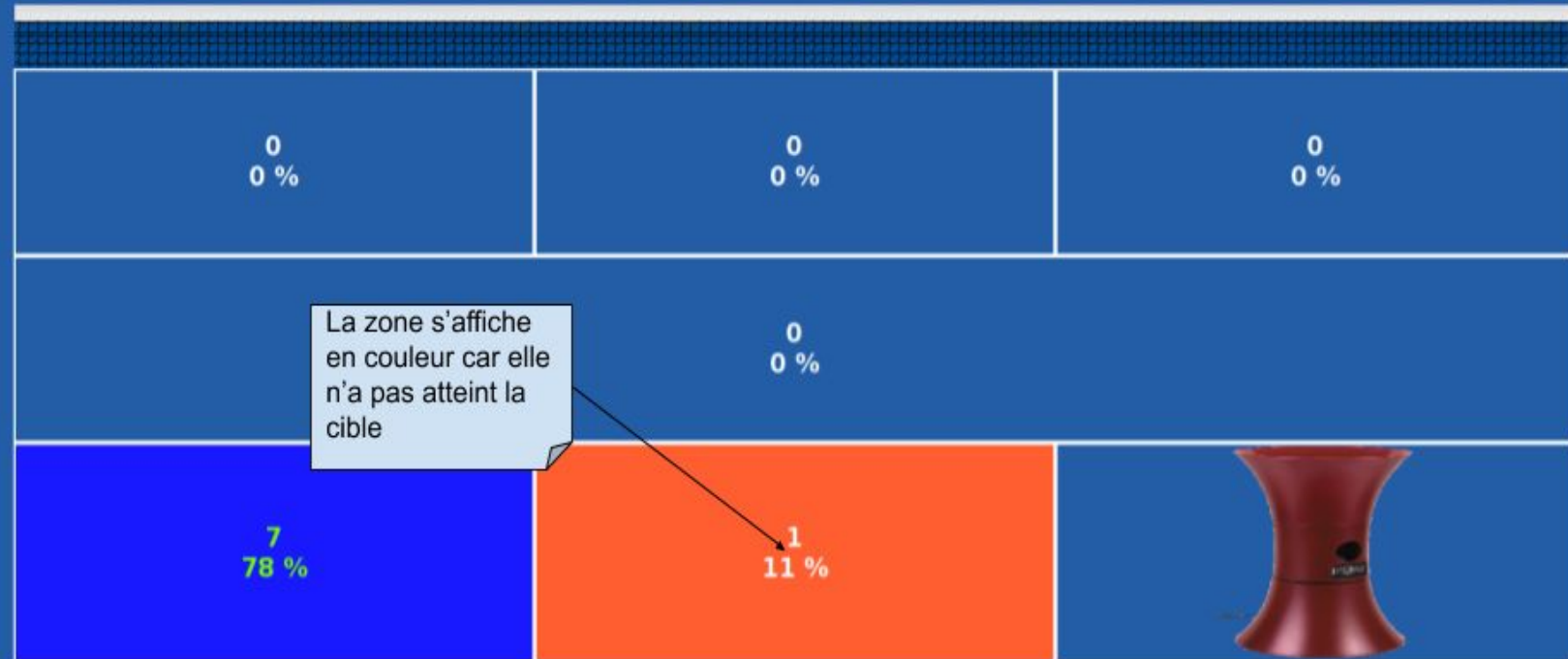
**Joueur : Simon Gauzy**

**12 : 17 : 59**

**Hors Table : 1 (11 %)**

**8 / 9**

**01 : 02**



# L'écran résultats

**Joueur : Julien**

**06 : 37 : 52**

## Fin de la Séance

**Durée de la séance**

**00 : 56**

**Balles ayant atteint l'objectif**

**5 / 9**

**56 %**

**Balles Hors de la Table**

**2 / 9**

**22%**

**Nombre série de balles sur l'objectif**

**3 / 5**

**60%**

**Joueur : Julien**

**06 : 41 : 03**

## **Fin de la Séance**

**Durée de la séance**

**00 : 41**

**Balles ayant atteint l'objectif**

**1 / 7**

**14 %**

**Balles Hors de la Table**

**4 / 7**

**57%**

**Nombre série de balles sur l'objectif**

**1 / 1**

**100%**

# Diagramme de classes

## CommunicationBluetooth

```

- monBluetooth : QBluetoothLocalDevice
- serveur : QBluetoothServer
- socket : QBluetoothSocket
- serviceInfo : QBluetoothServiceInfo
- nom : QString
- adresse : QString
- etatConnexion : bool
- connexionErreur : bool

+ CommunicationBluetooth(inout parent : QObject = nullptr)
+ ~CommunicationBluetooth()
+ demarrer() : void
+ arreter() : void
- decoderTrame(in trame : QString) : void
+ nouveauClient() : void
+ socketConnected() : void
+ socketDisconnected() : void
+ socketReadyRead() : void
+ socketErreur(in error : QBluetoothSocket::SocketError) : void
+ deviceConnected(in adresse : QBluetoothAddress) : void
+ deviceDisconnected(in adresse : QBluetoothAddress) : void
+ error(in erreur : QBluetoothLocalDevice::Error) : void
+ connecte() : void
+ erreur() : void
    
```

## StatistiquesSeance

```

- numeroZoneObjectif : int
- numeroZoneRobot : int
- <<QVector>> nblImpactsZone : int
- nbBallesTotales : int
- nbBallesTotalesTable : int
- nbBallesTotalesHorsTable : int
- balleToucheTable : bool
- ballePrecedenteToucheTable : bool
- balleToucheRobot : bool

+ StatistiquesSeance(inout parent : QObject = nullptr)
+ ~StatistiquesSeance()
+ impacterZone(in numeroZone : int) : bool
+ finirSeance() : void
+ initialiserStatistiques() : void
+ getBallesTotal() : int
+ getBallesHorsTable() : int
+ getballesPrecedentToucheTable() : int
+ getballesToucheTable() : int
+ calculerPourcentage(in nblImpactsZone : int) : QString
+ afficherStatistiquesZones() : void
+ affichageZone(in numeroZone : int, in nblImpacts : int, in pourcentage : QString) : void
    
```

# Utilisation des classe Qt : QString

QString : Classe Qt pour gérer une chaîne de caractères

## Besoins :

- Lire la longueur de la chaîne → **length()**
- Vérifier la présence de caractères en début de chaîne → **startsWith(QString)**
- Vérifier la présence de caractères en fin de chaîne → **endsWith(QString)**
- Tester si la chaîne est vide → **isEmpty()**
- Découper la chaîne en fonction d'un délimiteur → **section()**
- Lire le caractère de la chaîne situé à une position donnée → **at(int)**

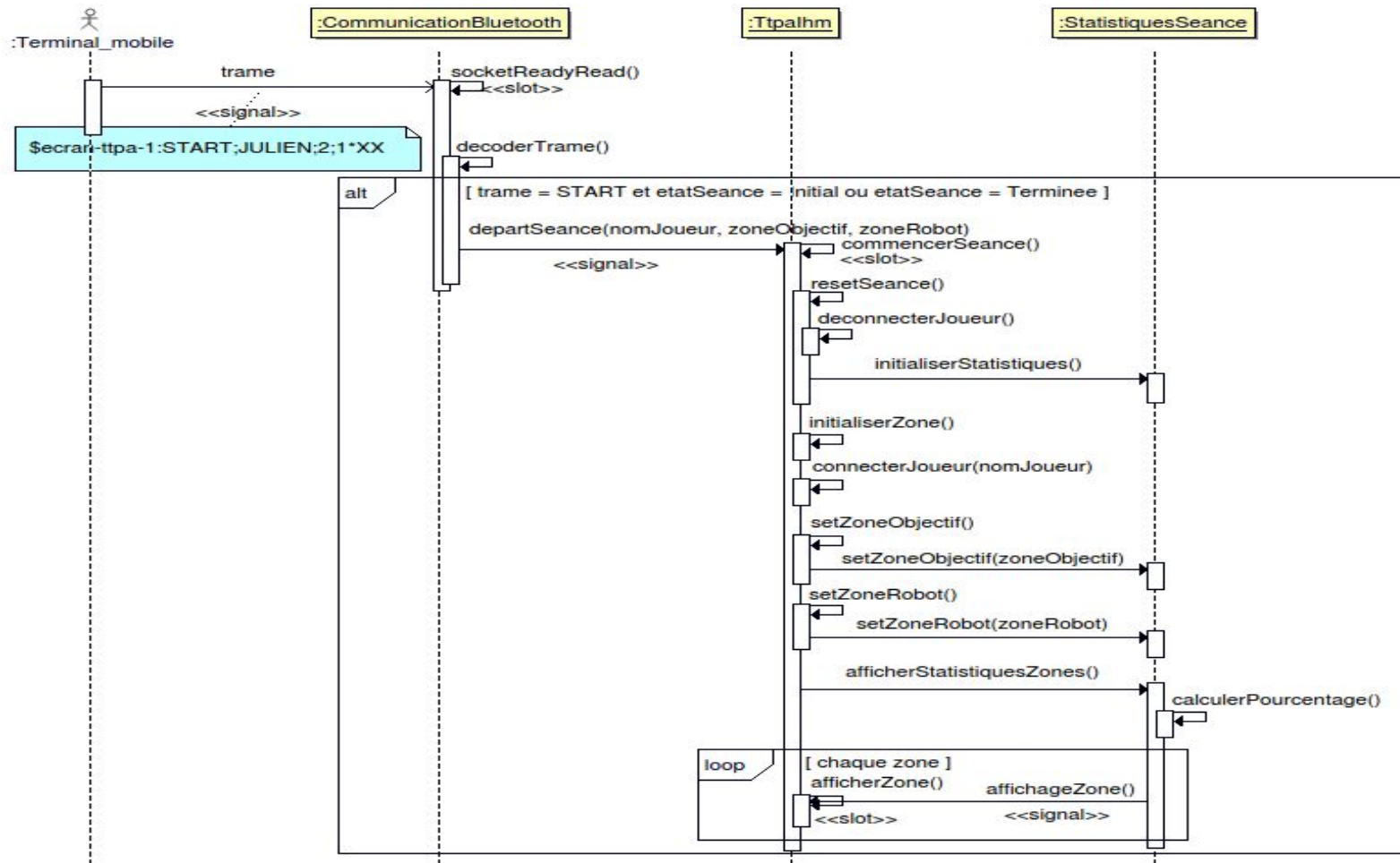
# Décodage trames

```
QStringList trames;  
trame.remove("\r\n");
```

```
const QString typeTrame = "ecran-ttpe";  
QString donnees;  
QString nomJoueur;  
QString zoneRobot;  
QString zoneObjectif;
```

```
if(trame.startsWith("$" + typeTrame)) //On retrouve startsWith  
{  
    donnees = trame.section(':', 1, 1); // START;JULIEN;2;1*XX //On retrouve section  
    if(trame.contains("START") && (etatSeance == EtatSeance::Initial || etatSeance ==  
EtatSeance::Terminee))  
    {  
        etatSeance = EtatSeance::EnCours;  
        nomJoueur = donnees.section(';', 1, 1);  
        zoneObjectif = donnees.section(';', 2, 2);  
        zoneRobot = donnees.section(';', 3, 3).at(0); //On retrouve at  
        emit departSeance(nomJoueur, zoneObjectif, zoneRobot);  
    }  
}
```

# Scénario du démarrage d'une séance

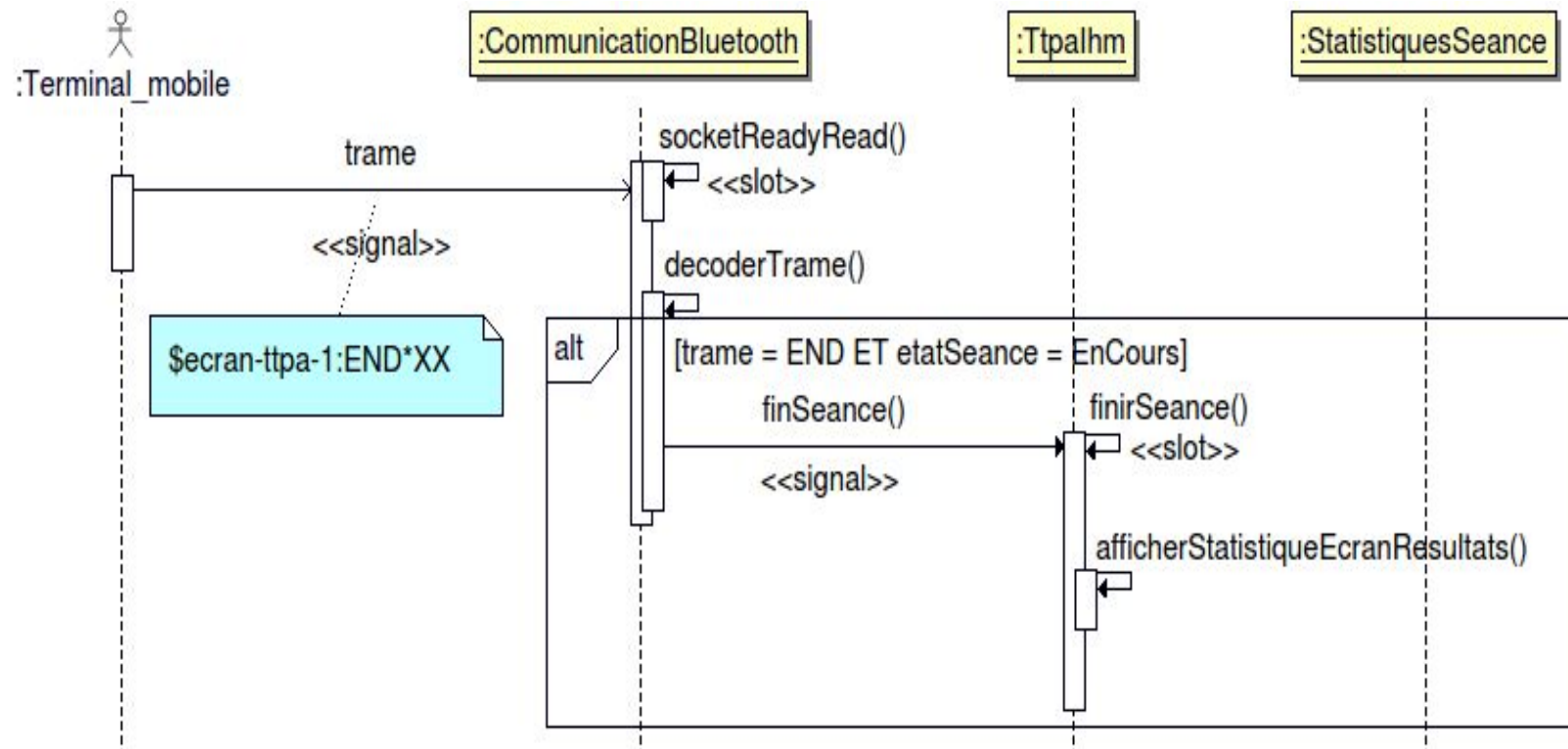




# Scénario d'impact de balle dans une zone



# Scénario de fin d'une séance



# Tests de validation

Description	OUI	NON
Le système d'exploitation est installé et fonctionnel	X	
L'écran est configuré en mode "kiosque"	X	
La zone d'impact est identifiée et affichée en temps réel	X	
Les données de la séance ( le pourcentage de balles par zones et le nombre de pourcentage ayant été sur la bonne zone) sont affichées en temps réel	X	
Les liaisons sans fil sont opérationnelles (Dialoguer avec le terminal mobile)	X	
Les statistiques sont affichées en fin de séquence	X	

# Conclusion

## → État de l'avancement

### ◆ Améliorations possibles

- Graphiques
  - Visualisation des appareils Bluetooth
  - Informations complémentaires d'une séance
  - Option retour pour revenir à l'écran principal
- Techniques
  - Actualisation des appareils Bluetooth

## → Bilan global

- ◆ Enjeu réel : partenariat avec le club PPC Sorgues