



Table Tennis Performance Analyser

Projet BTS SN Lasalle - 2018

VIDAL Damien (EC), GRENOD Pierre (EC), RACAMOND Adrien (IR), **SMANIOTTO Nathan** (IR)

Terminal Mobile
SMANIOTTO Nathan



Revue finale

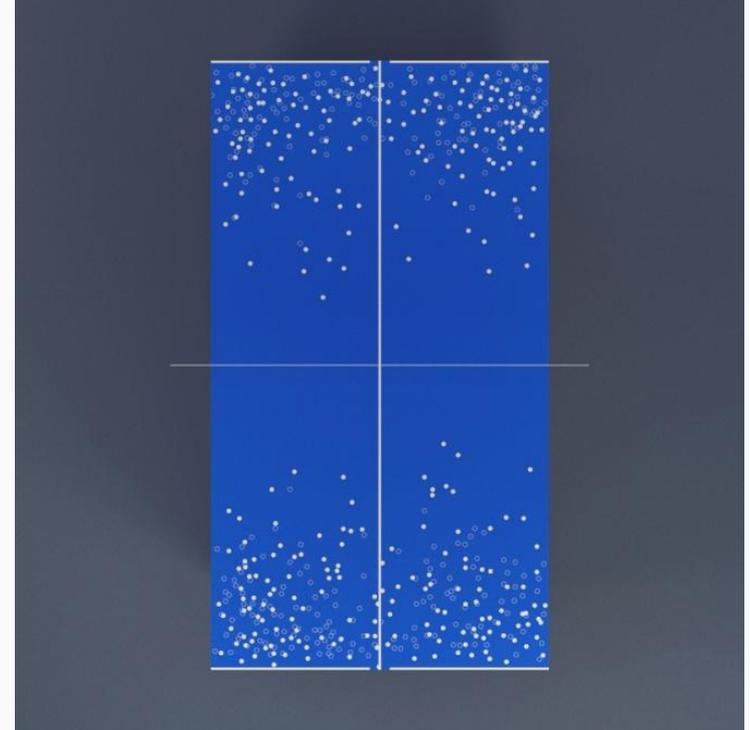
06/06/2018

v1.0

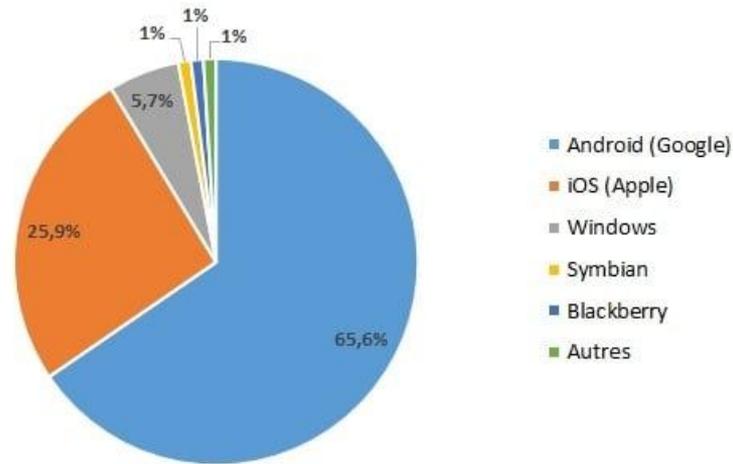




TABLE WALDNER



Répartition des OS smartphone en France au deuxième trimestre 2017



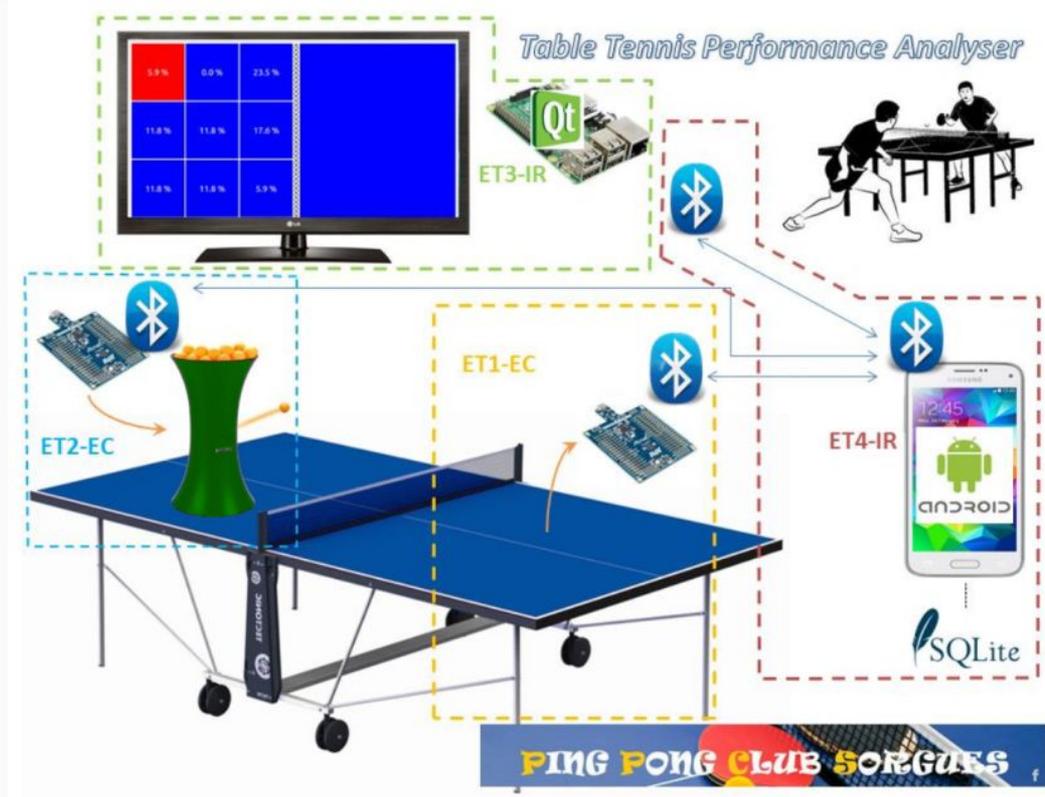
- Analyser les performances d'un joueur
- Proposer une phase d'entraînement
- Détecter l'impact des balles
- Lancer un exercice spécifique
- Visualiser l'évolution personnelle d'un joueur

- Séances paramétrables
 - ◆ Nombre de balles
 - ◆ Effet
 - ◆ Vitesse

- Localisation des impacts des balles

- Retour visuel de la séance

1	2	3
4	5	6
7	8	9
	0	



→ Régler les paramètres d'une séance

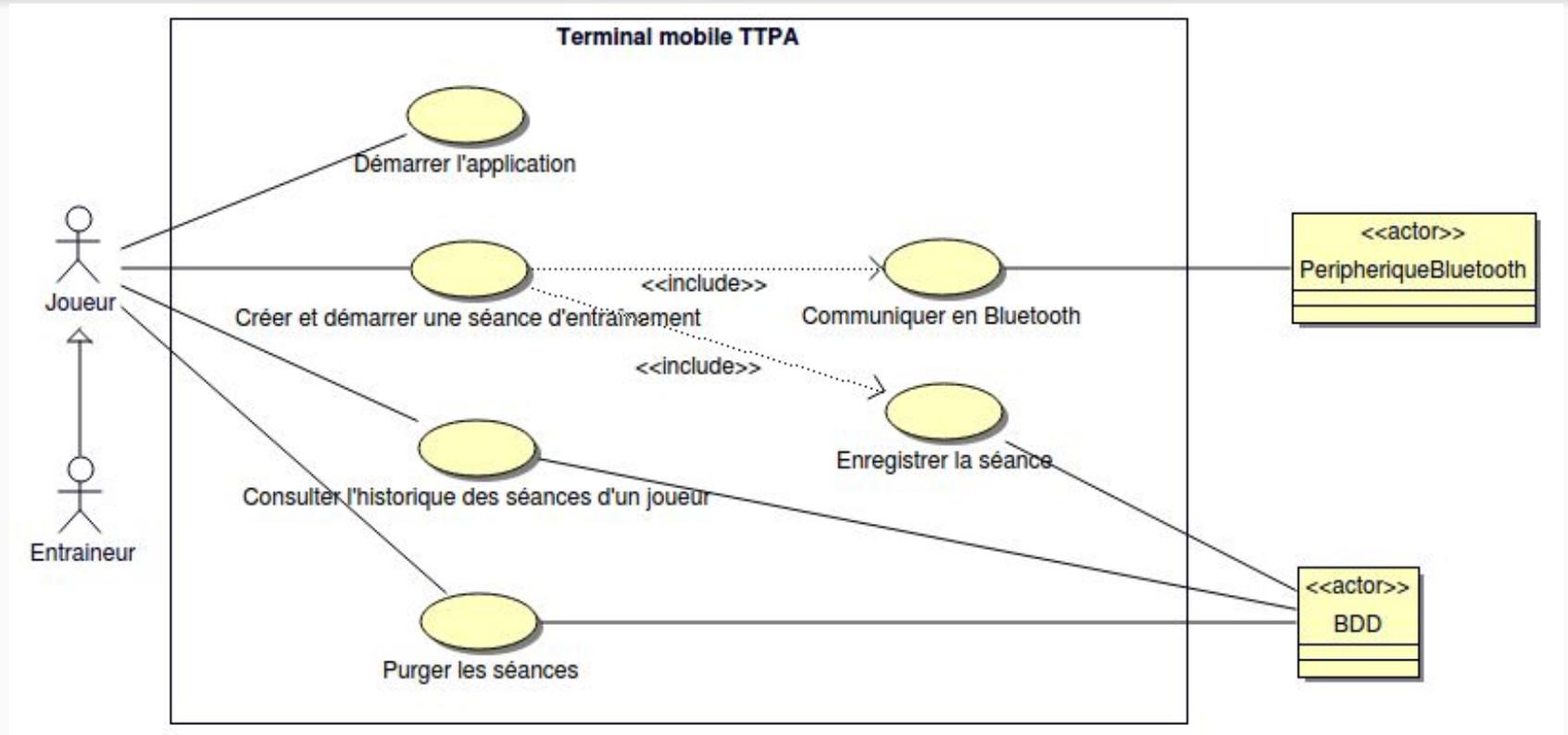
- ◆ Zone du robot
- ◆ Zone d'objectif
- ◆ Nombre de balles
- ◆ Fréquence d'envoi
- ◆ Effet des balles
 - coupé
 - lifté
 - sans effet
- ◆ Vitesse des balles
- ◆ Rotation du lanceur

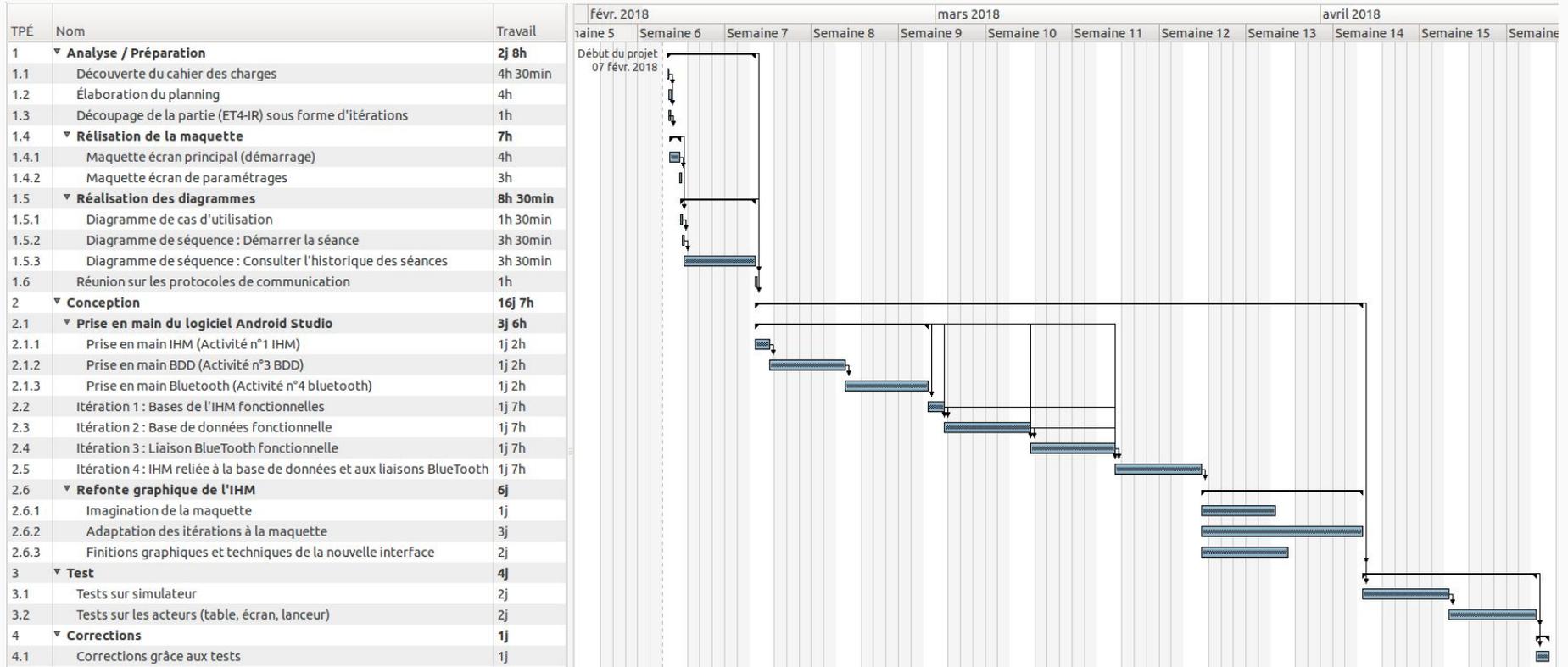
→ Localisation des impacts des balles sous forme de zones

→ Visualiser en temps réel les statistiques du joueur

Étudiant	Spécialité	Module
VIDAL Damien	EC	Robot Lanceur
GRENOD Pierre	EC	Détection impact
RACAMOND Adrien	IR	Écran TV
SMANIOTTO Nathan	IR	Terminal Mobile

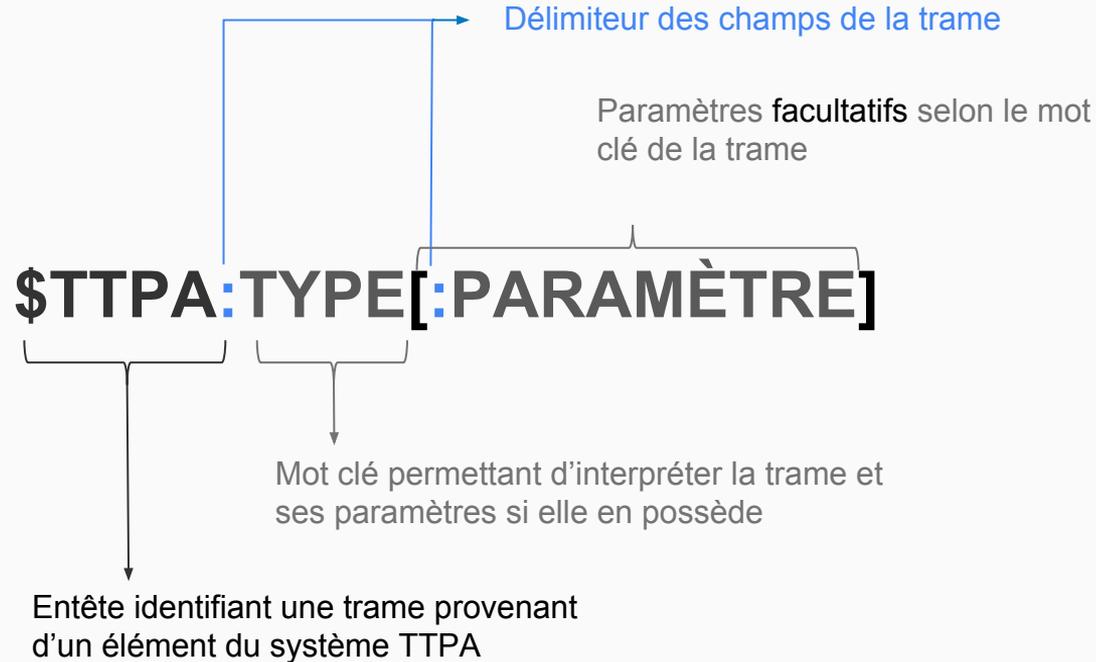
- Utilisation de [Subversion](#) pour les codes sources
- Utilisation d'un espace de stockage commun ([NAS](#) et [Google Drive](#)) pour tous les documents ressources
- Utilisation d'un [protocole de communication](#)

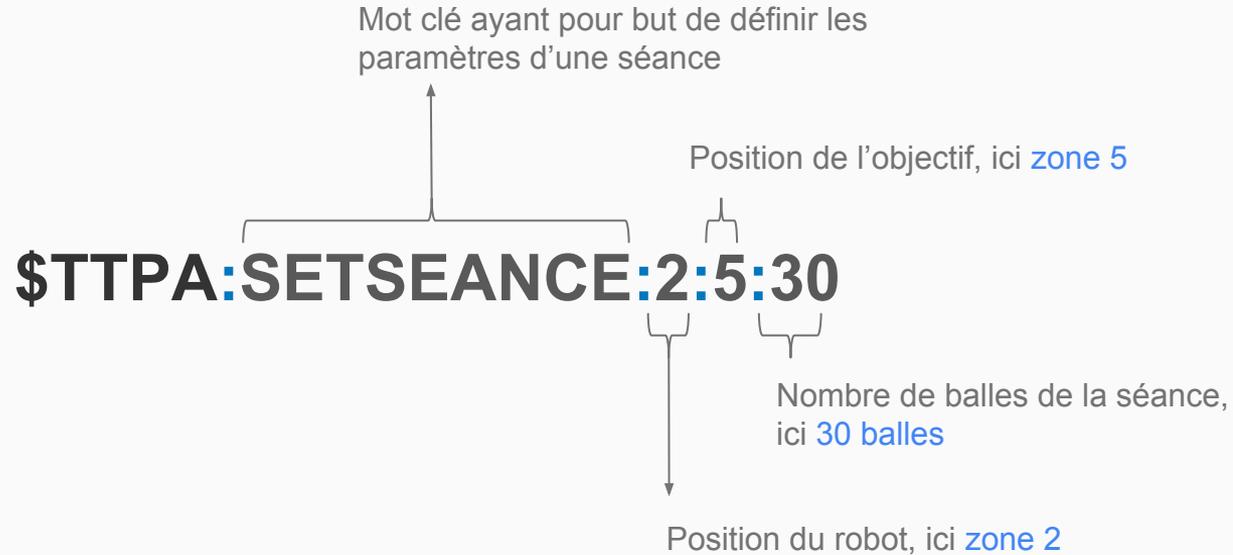




Fonctionnalité	Élément utilisé	Version
Système d'exploitation du poste de développement	GNU/LINUX	Ubuntu 12.04 LTS (Noyau 3.8.0)
Logiciel de planification	Planner	0.14.5
Génération des diagrammes	BOUML	7.4
Gestion de versions	Subversion	1.6.17
Environnement de développement	Android Studio	2.3
Génération de la documentation	Doxygen	1.7.6.1
Tablette utilisée pour les essais et démonstrations	Tablette Samsung	SM-T530

Composant	Description
Systeme d'exploitation	Android Lollipop 5.0.2
Processeur	Snapdragon 400 Quad-core 1.2GHz
Mémoire RAM	1.5 Go
Carte graphique	Adreno 305





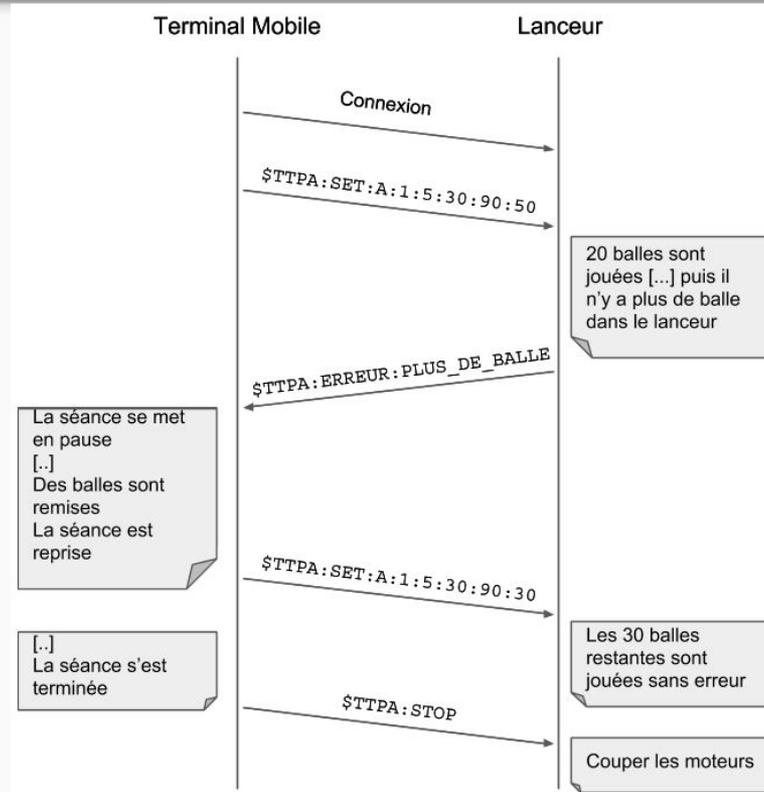
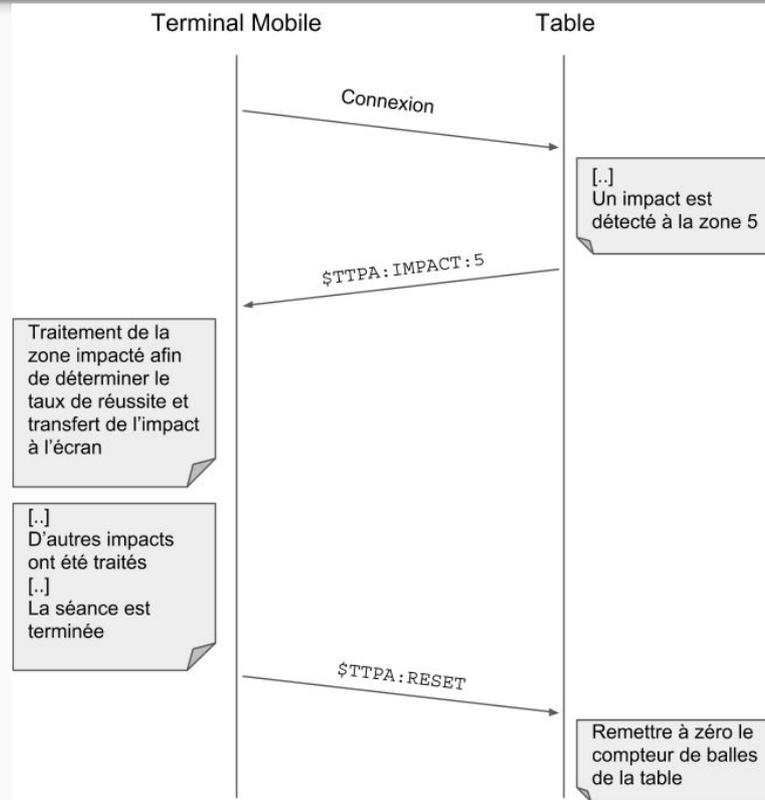
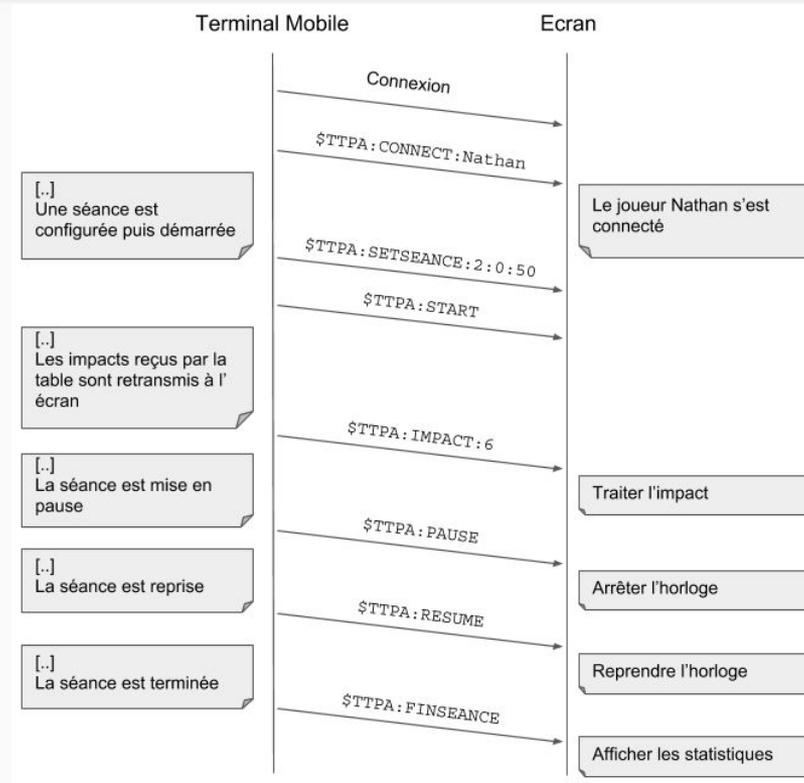
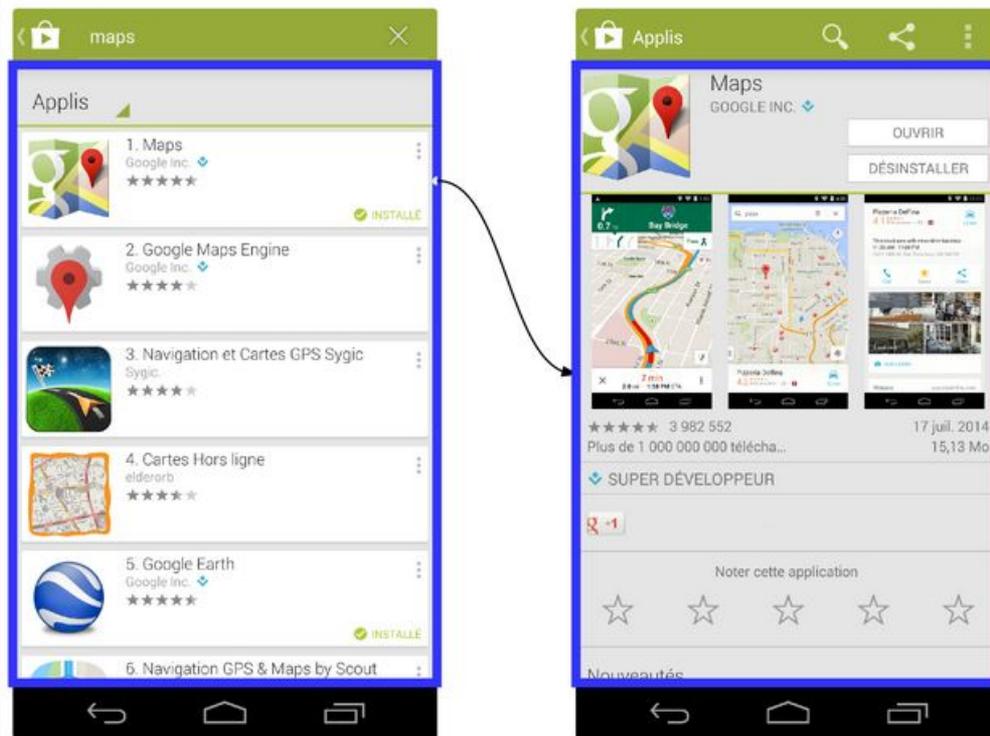
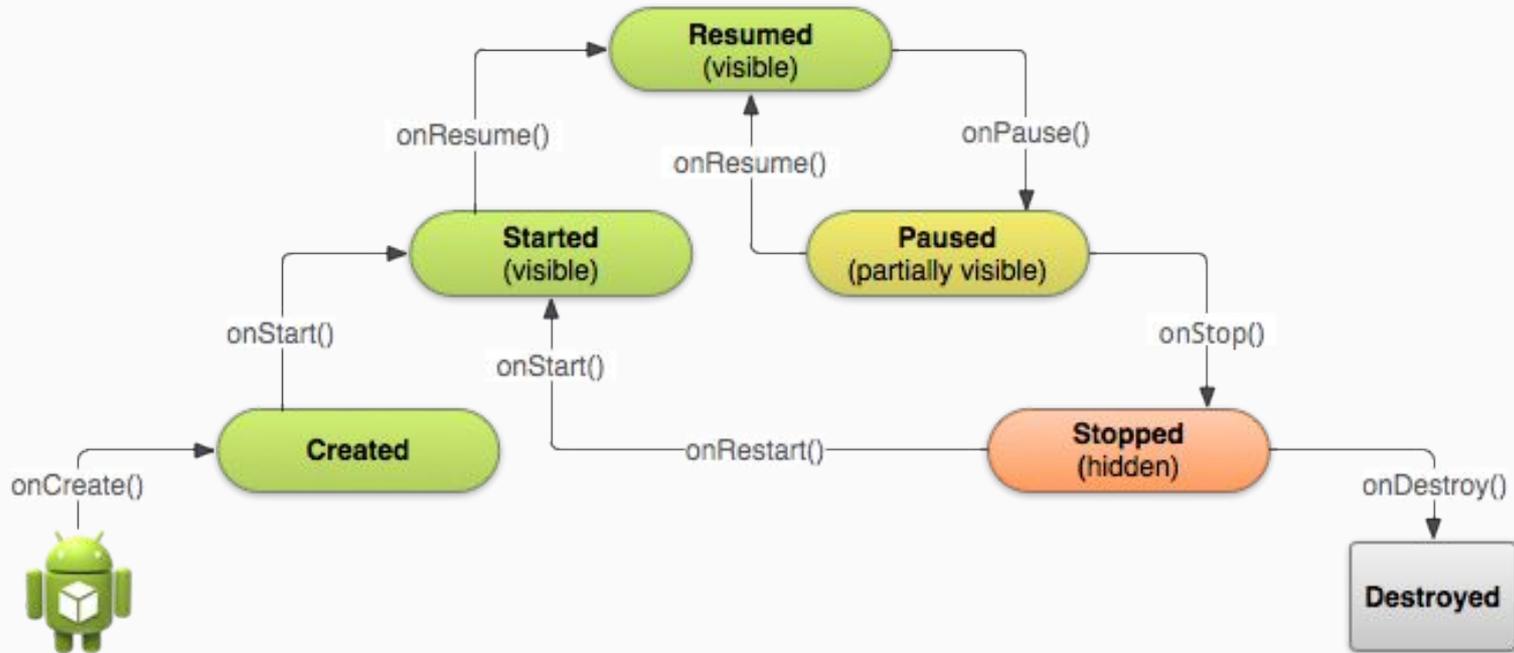


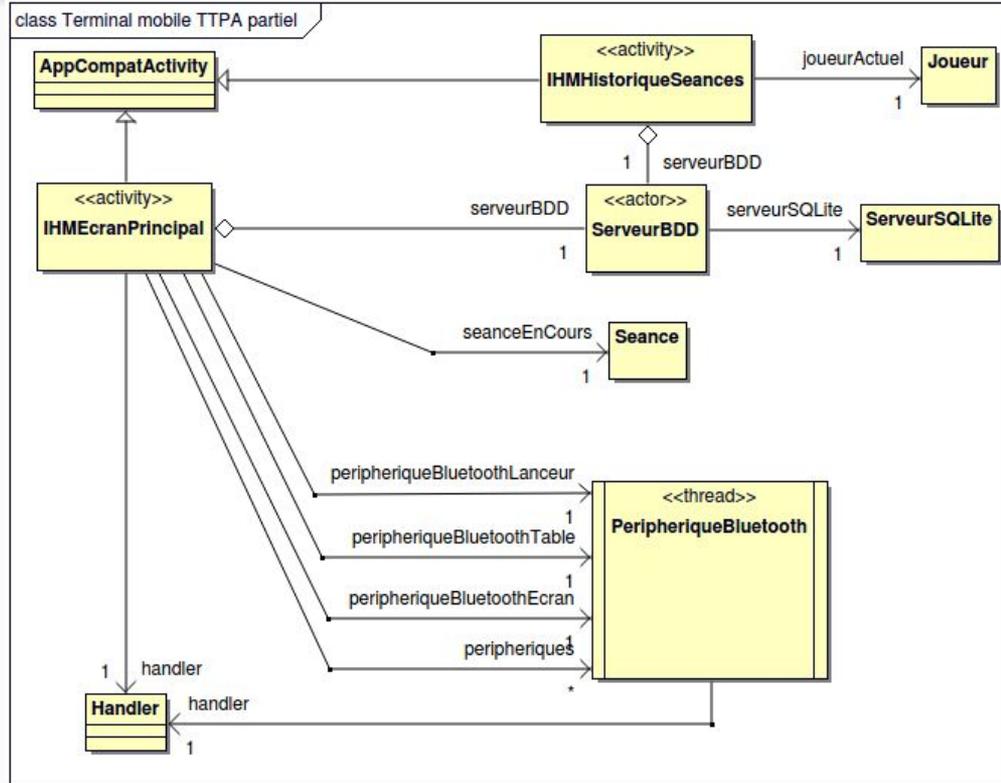
Diagramme d'échanges Terminal Mobile - Table

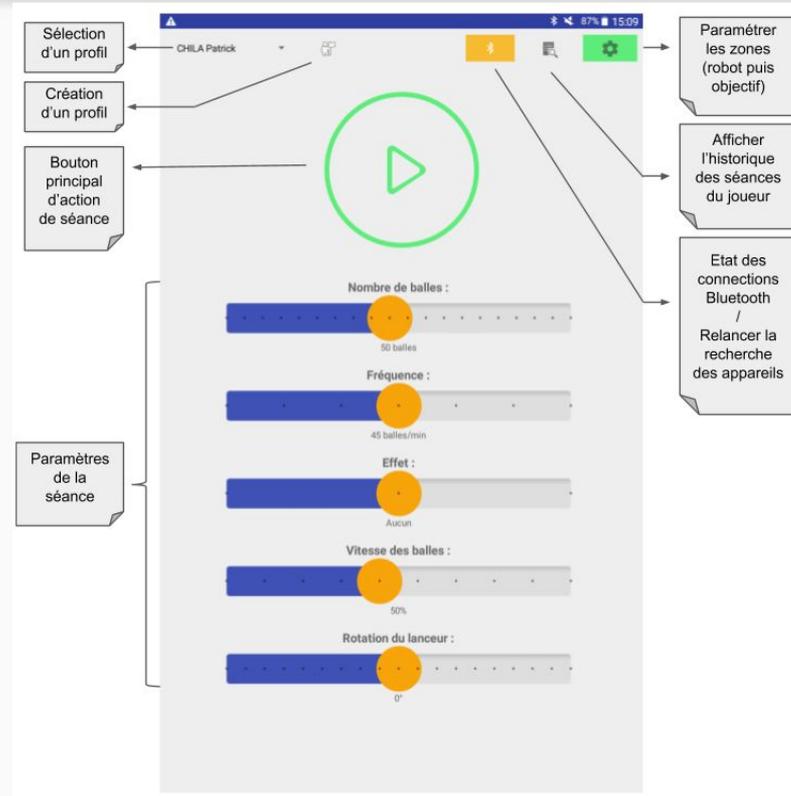












Historique des séances de **CHILA Patrick :**

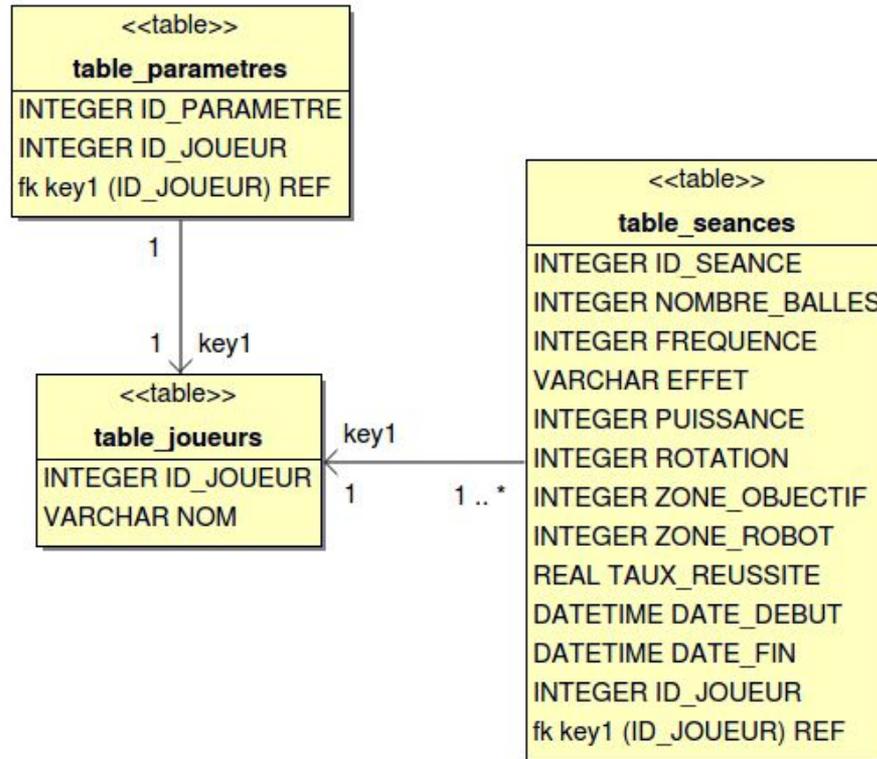
Date	Fréquence	Nb Balles	Effet	Taux Réussite	Action
2018-05-24 10:52:37	50 balles/min	20	Lité	40.0%	
2018-05-24 10:51:38	60 balles/min	15	Aucun	70.0%	
2018-05-24 10:51:00	50 balles/min	35	Aucun	90.0%	
2018-05-24 10:50:09	35 balles/min	75	Couplé	30.0%	
2018-05-24 10:49:18	35 balles/min	80	Couplé	47.0%	
2018-05-24 10:47:14	30 balles/min	30	Lité	86.0%	
2018-05-24 10:43:50	50 balles/min	50	Couplé	40.0%	
2018-05-24 10:40:49	60 balles/min	20	Aucun	50.0%	
.
.
.
.

Joueur actuellement sélectionné

Afficher les détails de la séance

Séances jouées par le joueur actuellement sélectionné

Purger toutes les séances du joueur actuellement sélectionné

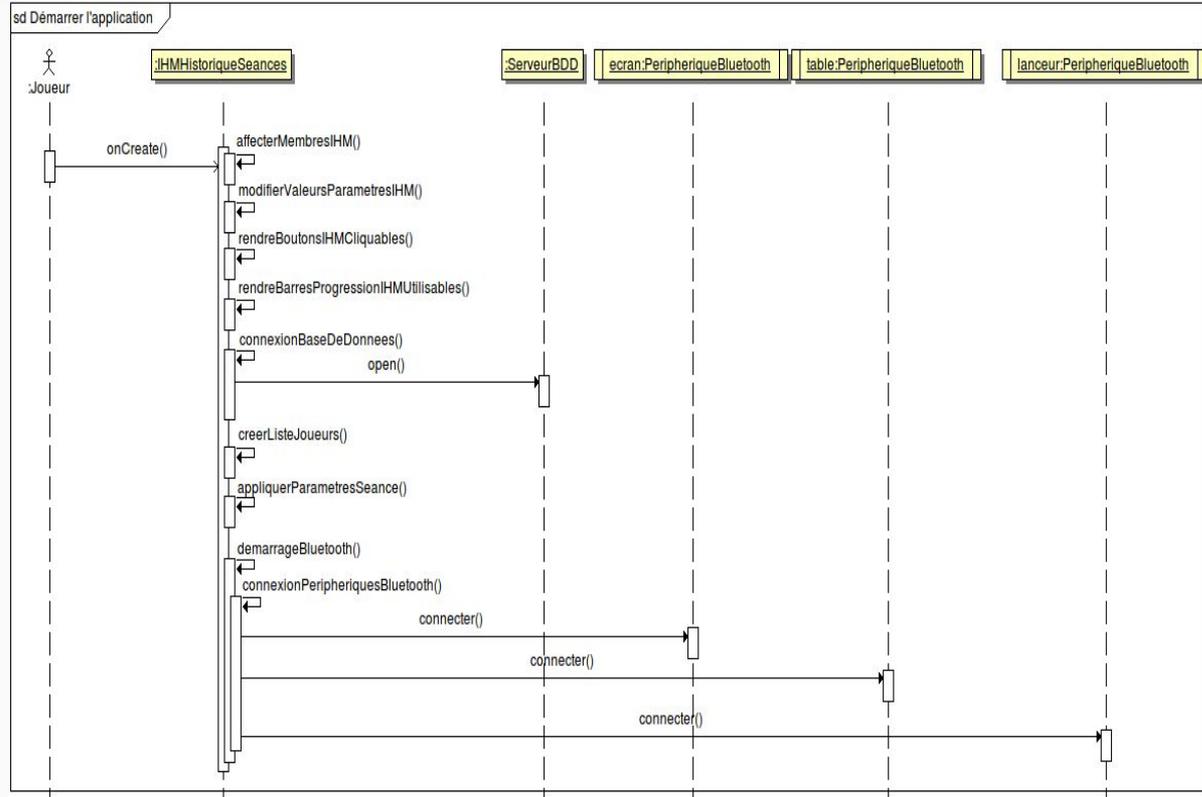


Exemple de requête SQL - Insérer une séance

Champ	Valeur
NOMBRE_BALLES	15 (balles)
FREQUENCE	45 (balles/min)
EFFET	"Aucun"
PUISSANCE	5 (coefficient)
ROTATION	45 (°)
ZONE_OBJECTIF	1
ZONE_ROBOT	3
TAUX_REUSSITE	40.0 (%)
DATE_FIN	2018-05-25 15:45:13
ID_JOUEUR	8

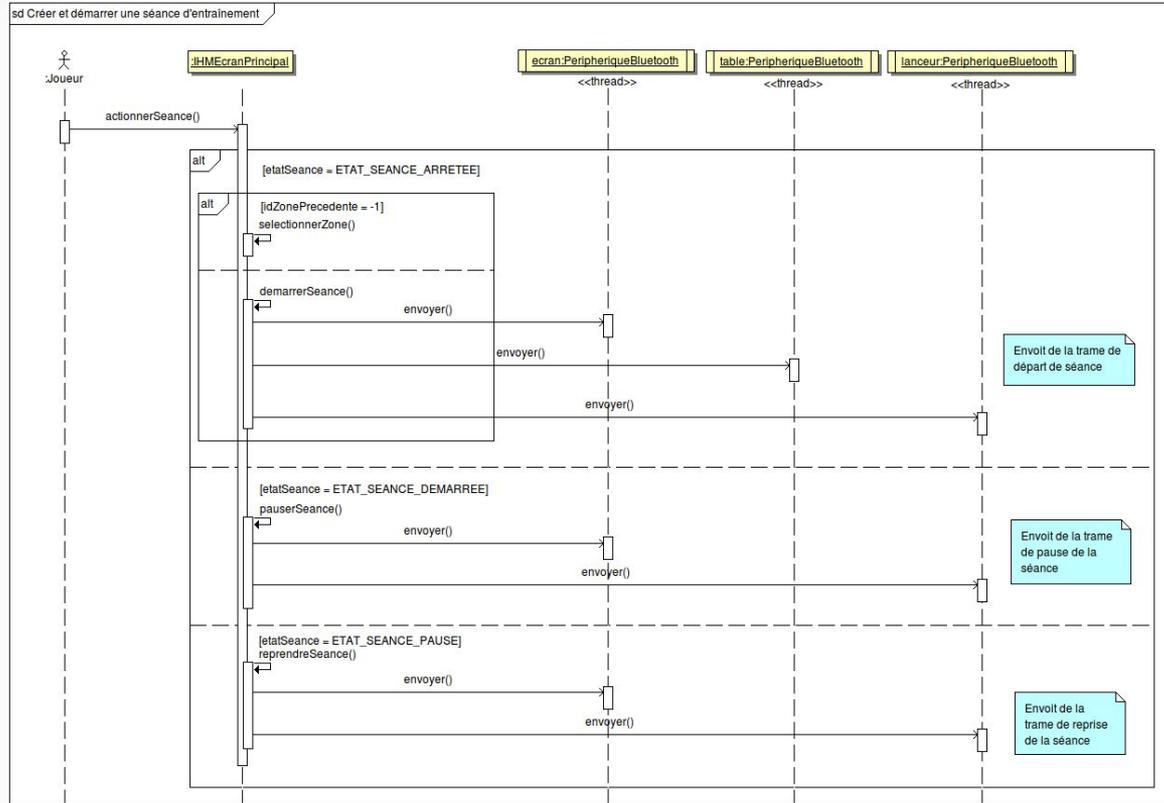
```
INSERT INTO table_seances (NOMBRE_BALLES, FREQUENCE, EFFET, PUISSANCE, ROTATION,
                           ZONE_OBJECTIF, ZONE_ROBOT, TAUX_REUSSITE, DATE_FIN, ID_JOUEUR)
VALUES (15, 45, "Aucun", 5, 45, 1 , 3, 40.0, 2018-05-25 15:45:13, 8);
```

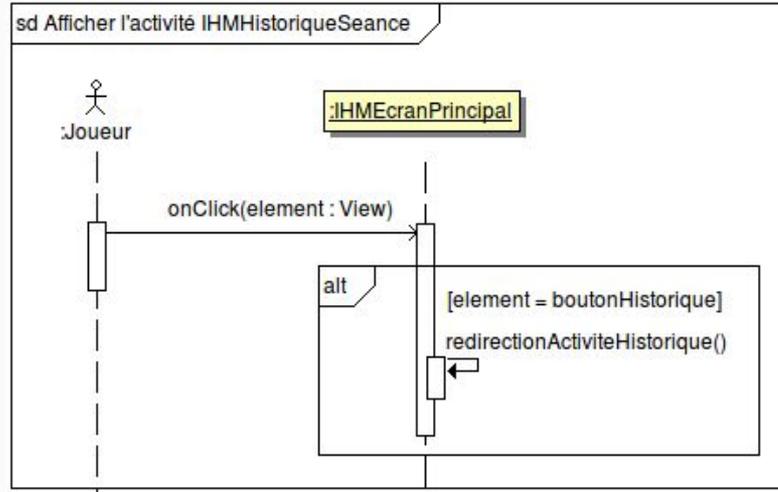
Scénario - Démarrer une application



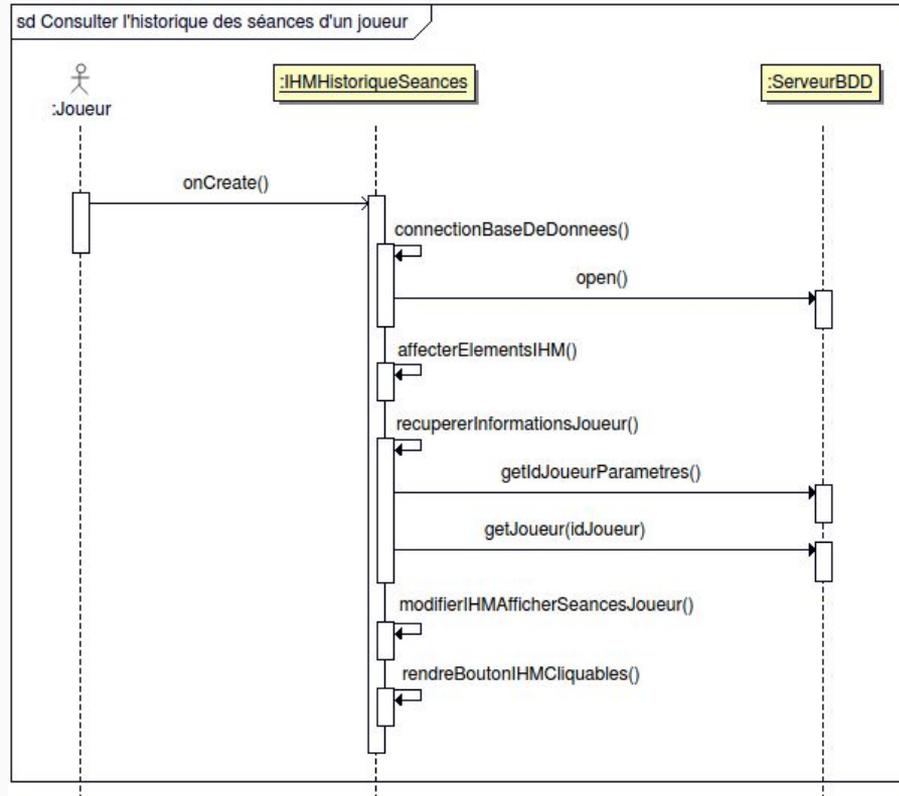
Initialisation
des éléments
graphiques

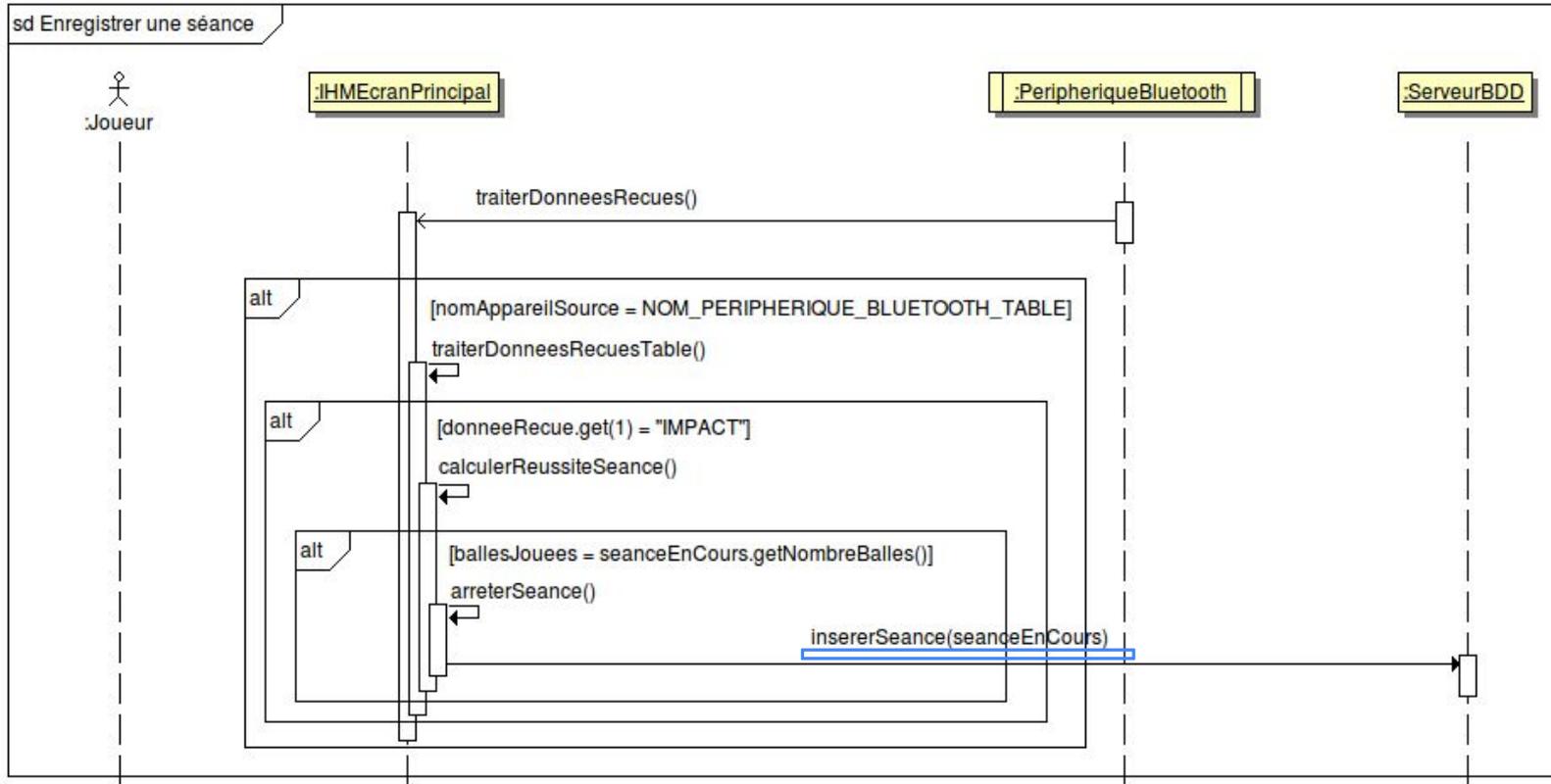
Connexion
aux appareils
Bluetooth





```
private void redirectionActiviteHistorique()  
{  
    Intent intent = new Intent(IHMEcranPrincipal.this, IHMHistoriqueSeances.class);  
    startActivity(intent);  
}
```





- **Classe `SQLiteOpenHelper`** : pour gérer la création d'une base de données (BDD)
 - ◆ Création d'une sous-classe `ServeurSQLite` qui hérite de `SQLiteOpenHelper`
 - ◆ Implémentation des méthodes `onCreate()`, `onUpgrade()` et `onOpen()`

- **Classe `SQLiteDatabase`** : pour effectuer des requêtes sur les tables de la BDD
 - ◆ Création d'une classe `ServeurBDD` qui est composé d'un objet `SQLiteDatabase`
`private SQLiteDatabase bdd = null;`
`private ServeurSQLite serveurSQLite = new ServeurSQLite(context);`
 - ◆ Création et/ou ouverture de la base de données qui sera utilisée pour la lecture et l'écriture avec `getWritableDatabase()`
`bdd = serveurSQLite.getWritableDatabase();`
 - ◆ Insérer, modifier et supprimer un enregistrement avec la méthode `insert()`, `update()` et `delete()`
 - ◆ Récupérer des données avec `query()`

```
/**  
 * Déclaration de la méthode insérerSeance permettant l'insertion d'une séance dans  
 * la base de données.  
 * @param seance séance à insérer dans la base de données  
 * @return l'id de la séance dans la base de données  
 */  
public long insérerSeance(Seance seance)  
{  
    ContentValues values = new ContentValues();  
  
    values.put(ServeurSQLite.COL_FREQUENCE, seance.getFrequence());  
    values.put(ServeurSQLite.COL_NOMBRE_BALLES, seance.getNombreBalles());  
    values.put(ServeurSQLite.COL_EFFET, seance.getEffet());  
    ...  
  
    return bdd.insert(ServeurSQLite.TABLE_SEANCES, null, values);  
}
```

Description	OUI	NON
La base de données est fonctionnelle et complétée	✓	
Le système est paramétrable	✓	
La liaison Bluetooth est fonctionnelle	✓	
Les informations de paramétrage sont transmises	✓	
L'application mobile est déployée	✓	

- 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
 - Arrêter une séance en cours
 - Actualisation des appareils Bluetooth
 - Supprimer individuellement une séance
 - Calcul du temps d'une séance, avec les pauses
- Bilan global
 - ◆ Enjeu réel : partenariat avec le club PPC Sorgues