

TTPA (Mobile)

1.1

Généré par Doxygen 1.7.6.1

Jeudi Juin 7 2018 20 :53 :53

Table des matières

1	Page principale du projet TTPA (Table Tennis Performance Analyser)	1
1.1	Introduction	1
1.2	Table des matières	1
2	Changelog	1
3	Configuration	3
4	Manuel d'installation	4
5	Recette IR	4
6	Base de données	4
7	A propos	4
8	Licence GPL	5
9	Documentation des classes	5
9.1	Référence de la classe <code>com.ttpa.iris.ttpamobile.IHMEcranPrincipal</code> . . .	6
9.1.1	Description détaillée	9
9.1.2	Documentation des fonctions membres	10
9.1.3	Documentation des données membres	42
9.2	Référence de la classe <code>com.ttpa.iris.ttpamobile.IHMHistoriqueSeances</code> .	53
9.2.1	Description détaillée	55
9.2.2	Documentation des fonctions membres	55
9.2.3	Documentation des données membres	63
9.3	Référence de la classe <code>com.ttpa.iris.ttpamobile.Joueur</code>	64
9.3.1	Description détaillée	64
9.3.2	Documentation des constructeurs et destructeur	64
9.3.3	Documentation des fonctions membres	65
9.3.4	Documentation des données membres	66
9.4	Référence de la classe <code>com.ttpa.iris.ttpamobile.ParametreSeance</code>	66
9.4.1	Description détaillée	67
9.4.2	Documentation des constructeurs et destructeur	67
9.4.3	Documentation des fonctions membres	68

9.4.4	Documentation des données membres	76
9.5	Référence de la classe com.ttpa.iris.ttpamobile.PeripheriqueBluetooth	78
9.5.1	Description détaillée	79
9.5.2	Documentation des constructeurs et destructeur	79
9.5.3	Documentation des fonctions membres	80
9.5.4	Documentation des données membres	84
9.6	Référence de la classe com.ttpa.iris.ttpamobile.ReceveurBluetooth	86
9.6.1	Documentation des constructeurs et destructeur	87
9.6.2	Documentation des fonctions membres	87
9.7	Référence de la classe com.ttpa.iris.ttpamobile.Seance	88
9.7.1	Description détaillée	89
9.7.2	Documentation des constructeurs et destructeur	89
9.7.3	Documentation des fonctions membres	89
9.7.4	Documentation des données membres	96
9.8	Référence de la classe com.ttpa.iris.ttpamobile.ServeurBDD	99
9.8.1	Description détaillée	100
9.8.2	Documentation des constructeurs et destructeur	100
9.8.3	Documentation des fonctions membres	101
9.8.4	Documentation des données membres	111
9.9	Référence de la classe com.ttpa.iris.ttpamobile.ServeurSQLite	112
9.9.1	Description détaillée	113
9.9.2	Documentation des constructeurs et destructeur	113
9.9.3	Documentation des fonctions membres	113
9.9.4	Documentation des données membres	116
9.10	Référence de la classe com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.- TReception	122
9.10.1	Documentation des constructeurs et destructeur	123
9.10.2	Documentation des fonctions membres	123
9.10.3	Documentation des données membres	124
10	Documentation des fichiers	125
10.1	Référence du fichier AndroidManifest.xml	125
10.2	Référence du fichier Changelog.dox	125
10.3	Référence du fichier IHMEcranPrincipal.java	125

10.4	Référence du fichier IHMHistoriqueSeances.java	125
10.5	Référence du fichier Joueur.java	125
10.6	Référence du fichier ParametreSeance.java	125
10.7	Référence du fichier PeripheriqueBluetooth.java	125
10.8	Référence du fichier README.dox	125
10.9	Référence du fichier ReceveurBluetooth.java	125
10.10	Référence du fichier Seance.java	126
10.11	Référence du fichier ServeurBDD.java	126
10.12	Référence du fichier ServeurSQLite.java	126

1 Page principale du projet TTPA (Table Tennis Performance - Analyser)

1.1 Introduction

Le système doit permettre une analyse des performances du joueur (côté relanceur). Il doit proposer une phase d'entraînement adaptée au niveau du joueur, puis de détecter l'impact des balles afin d'afficher le rythme de jeu, la précision, le pourcentage de réussite. La zone d'impact (côté distributeur) est identifiée sur un écran de télévision en fin d'exercice. Le pourcentage de balles dans chacune des zones, le rythme de jeu et le pourcentage de réussite sont disponibles en fin d'exercice. Le joueur lance un exercice spécifique et pourra connaître son évolution individuelle.

Module : Application Mobile (Smaniotto Nathan)

1.2 Table des matières

- Configuration
- Manuel d'installation
- Changelog
- Recette IR
- Base de données
- A propos
- Licence GPL

Dépôt SVN : <https://svn.riouxsvn.com/ttpa>

2 Changelog

r114 | nsmaniotto | 2018-06-05 15 :49 :54 +0200 (mar., 05 juin 2018) | 1 line

[TTPA Mobile] Implémentation de voyants permettant de visualiser l'état individuel de chacun des appareils bluetooth (table, lanceur, écran) du système

r113 | nsmaniotto | 2018-06-05 12 :03 :45 +0200 (mar., 05 juin 2018) | 1 line

[TTPA Mobile] Lorsqu'une séance est en cours (démarrée ou en pause), il n'est désormais plus possible d'interagir avec les sliders

r112 | nsmaniotto | 2018-06-05 10 :23 :04 +0200 (mar., 05 juin 2018) | 1 line

[TTPA Mobile] Modification de la sélection des zones

r109 | nsmaniotto | 2018-06-04 12 :39 :38 +0200 (lun., 04 juin 2018) | 1 line

[TTPA Mobile] Modifications graphiques pour la sélection des zones

r108 | nsmaniotto | 2018-06-04 10 :02 :52 +0200 (lun., 04 juin 2018) | 1 line

[TTPA Mobile] Modification du manifest Android afin que les activités soient forcées en mode portrait

r107 | nsmaniotto | 2018-06-04 09 :54 :38 +0200 (lun., 04 juin 2018) | 1 line

[TTPA Mobile] Implémentation d'un bouton d'arrêt de séance, modifications graphiques dans la sélection de la zone

r106 | nsmaniotto | 2018-06-02 15 :33 :40 +0200 (sam., 02 juin 2018) | 1 line

[TTPA Mobile] Calcul de l'intensité de l'effet à afficher : intensiteEffet allant de 1 à 9, on affiche un effet de 10 à 90%, sauf quand il n'y a pas d'effet

r105 | nsmaniotto | 2018-06-02 15 :30 :34 +0200 (sam., 02 juin 2018) | 1 line

[TTPA Mobile] Implémentation du paramètre de séance intensiteEffet

r104 | nsmaniotto | 2018-05-31 16 :40 :50 +0200 (jeu., 31 mai 2018) | 1 line

[TTPA Mobile] Implémentation de la méthode incrementerBallesJouees() de la classe IHMEcranPrincipal permettant de déterminer la fin d'une séance, même lorsque la dernière balle jouée est hors table

r103 | nsmaniotto | 2018-05-30 16 :19 :33 +0200 (mer., 30 mai 2018) | 1 line

[TTPA Mobile] Ajout de fichiers layout/xml permettant la portabilité sur mobile de la sélection des zones robot et objectif

r102 | nsmaniotto | 2018-05-30 15 :19 :23 +0200 (mer., 30 mai 2018) | 1 line

[TTPA Mobile] Ajout de fichiers layout/xml permettant la portabilité sur mobile de l'activité IHMHistoriqueSeances

r101 | nsmaniotto | 2018-05-30 13 :55 :18 +0200 (mer., 30 mai 2018) | 1 line

[TTPA Mobile] Ajout de fichiers drawable/xml pour la dimension d'écran par défaut

r100 | nsmaniotto | 2018-05-30 13 :53 :37 +0200 (mer., 30 mai 2018) | 1 line

[TTPA Mobile] Rendre l'application utilisable sur mobile

r99 | nsmaniotto | 2018-05-28 15 :18 :08 +0200 (lun., 28 mai 2018) | 1 line

[TTPA Mobile] Correctif du crash généré à la sélection d'une zone

r93 | nsmaniotto | 2018-05-25 11 :06 :55 +0200 (ven., 25 mai 2018) | 1 line

[TTPA Mobile] Ajout d'une capture de l'écran principal de l'application, à l'accueil de la documentation html Doxygen

r92 | nsmaniotto | 2018-05-25 10 :26 :33 +0200 (ven., 25 mai 2018) | 1 line

[TTPA Mobile] Ajout d'images de la doc dans le svn

3 Configuration

Poste de développement :

- Distribution : Ubuntu 12.04.5 LTS
- OS : GNU/Linux
- Noyau : Linux
- Version : 3.8.0-44-generic
- Machine : x86_64
- Processeur : Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz
- Mémoire RAM : 8129984 kB

Informations de version sur les outils

- Android Studio 2.3
- java 1.8.0
- SDK Android API 25 : Android 7.1.1 (Nougat)
- svn, version 1.6.17 (r1128011)
- doxygen 1.7.6.1
- bouml Bouml 7.4

4 Manuel d'installation

Fabrication de l'exécutable :

- `qmake`
- `make`

5 Recette IR

Étudiant 4 : Smaniotto Nathan

- La base de données est fonctionnelle et complétée
- Le système est paramétrable
- La liaison Bluetooth est fonctionnelle
- Les informations de paramétrages sont transmises
- L'application mobile est déployée

6 Base de données

```
pragma foreign_keys = on ;
```

```
CREATE TABLE table_joueurs ( "ID_JOUEUR" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, "NOM" VARCHAR(255) NOT NULL ) ;
```

```
INSERT INTO table_joueurs(NOM) VALUES('LEGOUT Christophe') ; INSERT INTO table_joueurs(NOM) VALUES('MARTINEZ Michel') ; INSERT INTO table_joueurs(NOM) VALUES('LEBESSON Emmanuel') ; INSERT INTO table_joueurs(NOM) VALUES('ELOI Damien') ; INSERT INTO table_joueurs(NOM) VALUES('MATTENET Adrien') ; - INSERT INTO table_joueurs(NOM) VALUES('CHILA Patrick') ; INSERT INTO table_joueurs(NOM) VALUES('BEAUMONT Jérôme') ;
```

```
CREATE TABLE table_seances ( "ID_SEANCE" INTEGER PRIMARY KEY AUTOINCREMENT, "NOMBRE_BALLES" INTEGER NOT NULL, "FREQUENCE" INTEGER NOT NULL, "EFFET" VARCHAR(1) NOT NULL, "PUISSANCE" INTEGER NOT NULL, "ROTATION" INTEGER NOT NULL, "ZONE_OBJECTIF" INTEGER NOT NULL, "ZONE_ROBOT" INTEGER NOT NULL, "TAUX_REUSSITE" REAL NOT NULL, "DATE_DEBUT" DATETIME NOT NULL, "DATE_FIN" DATETIME NOT NULL, "ID_JOUEUR" INTEGER NOT NULL, CONSTRAINT fk_seances_1 FOREIGN KEY (ID_JOUEUR) REFERENCES table_joueurs (ID_JOUEUR) ON DELETE CASCADE ) ;
```

```
CREATE TABLE table_parametres ( "ID_PARAMETRE" INTEGER PRIMARY KEY - CHECK (ID_PARAMETRE = 1), "ID_JOUEUR" INTEGER NOT NULL, CONSTRAINT fk_parametres_1 FOREIGN KEY (ID_JOUEUR) REFERENCES table_joueurs (ID_JOUEUR) ) ;
```

```
INSERT INTO table_parametres(ID_PARAMETRE, ID_JOUEUR) VALUES(1, 1) ;
```

7 A propos

Auteur

Smaniotto Nathan <smaniotto.nathan@gmail.com>

Version

1.1

Date

2018

8 Licence GPL

This program is free software ; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation ; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY ; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program ; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

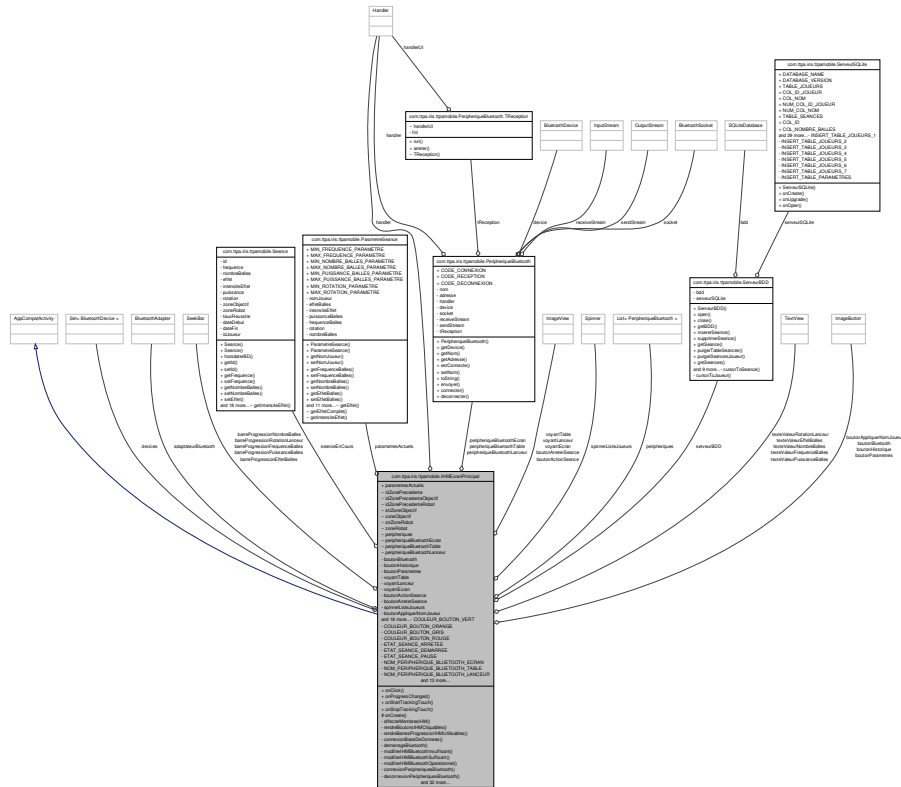
9 Documentation des classes

9.1 Référence de la classe com.ttpa.iris.ttpamobile.IHMEcranPrincipal

Graphe d'héritage de com.ttpa.iris.ttpamobile.IHMEcranPrincipal :



Graphe de collaboration de com.ttpa.iris.ttpamobile.IHMEcranPrincipal :



Fonctions membres publiques

- void `onClick` (View element)
- void `onProgressChanged` (SeekBar seekBar, int progress, boolean fromUser)
- void `onStartTrackingTouch` (SeekBar seekBar)
- void `onStopTrackingTouch` (SeekBar seekBar)

Attributs publics

- ```
– ParametreSeance parametresActuels = new ParametreSeance()
```

## Fonctions membres protégées

- void onCreate (Bundle savedInstanceState)

### Attributs de packaging

- int idZonePrecedente = -1
- int idZonePrecedenteObjectif = -1
- int idZonePrecedenteRobot = -1

- String `strZoneObjectif` = new String("ZONE 0")
- int `zoneObjectif` = 0
- String `strZoneRobot` = new String("ZONE 0")
- int `zoneRobot` = 0
- List< `PeripheriqueBluetooth` > `peripheriques`
- `PeripheriqueBluetooth` `peripheriqueBluetoothEcran` = null
- `PeripheriqueBluetooth` `peripheriqueBluetoothTable` = null
- `PeripheriqueBluetooth` `peripheriqueBluetoothLanceur` = null

#### Fonctions membres privées

- void `affecterMembresIHM` ()
- void `rendreBoutonsIHMClickables` ()
- void `rendreBarresProgressionIHMUtilisables` ()
- void `connexionBaseDeDonnees` ()
- void `demarrageBluetooth` ()
- void `modifierIHMBluetoothInsuffisant` ()
- void `modifierIHMBluetoothSuffisant` ()
- void `modifierIHMBluetoothOperationnel` ()
- void `connexionPeripheriquesBluetooth` ()
- void `deconnexionPeripheriquesBluetooth` ()
- void `connexionPeripheriqueBluetoothEcran` (BluetoothDevice appareilBluetooth)
- void `connexionPeripheriqueBluetoothTable` (BluetoothDevice appareilBluetooth)
- void `connexionPeripheriqueBluetoothLanceur` (BluetoothDevice appareilBluetooth)
- boolean `verifierConnexionAppareilsBluetoothRequis` ()
- void `actualiserIHMAppareilsBluetooth` (boolean tableEstConnectee, boolean lanceurEstConnecte, boolean ecranEstConnecte)
- void `actionnerSeance` ()
- void `demarrerSeance` ()
- void `pauserSeance` ()
- void `reprendreSeance` ()
- void `arreterSeance` (boolean seanceAEnregistrer)
- void `appliquerParametresSeance` ()
- void `modifierValeursParametresIHM` ()
- void `attendre` (int tempsMillisecondes)
- void `envoyerTramePeripheriqueBluetoothEcran` (String trame)
- void `envoyerTrameConnexionPeripheriqueBluetoothEcran` ()
- void `envoyerTrameParametrageSeancePeripheriqueBluetoothEcran` ()
- void `envoyerTrameDebutSeancePeripheriqueBluetoothEcran` ()
- void `envoyerTrameArretPeripheriqueBluetoothEcran` ()
- void `ajouterJoueur` ()
- void `creerListeJoueurs` ()
- void `traiterDonneesRecues` (String nomAppareilSource, String donnees)
- void `traiterDonneesRecuesLanceur` (String donnees)
- void `traiterErreurRecueLanceur` (String erreur)
- void `traiterDonneesRecuesTable` (String donnees)
- void `calculerReussiteSeance` (int zoneTouchee)
- void `incrementerBallesJouees` ()
- void `envoyerTrameArretPeripheriqueBluetoothLanceur` ()
- void `envoyerTrameArretPeripheriqueBluetoothTable` ()
- void `envoyerTrameDebutSeancePeripheriqueBluetoothLanceur` ()
- void `envoyerTrameRepriseSeancePeripheriqueBluetoothLanceur` ()
- void `envoyerTramePeripheriqueBluetoothLanceur` (String trame)
- void `selectionnerZone` (final int typeSelection, int choixObjectif, int choixRobot)
- void `redirectionActiviteHistorique` ()

#### Attributs privés

- ImageButton `boutonBluetooth`
- ImageButton `boutonHistorique`
- ImageButton `boutonParametres`
- ImageView `voyantTable`

- ImageView [voyantLanceur](#)
- ImageView [voyantEcran](#)
- ImageView [boutonActionSeance](#)
- ImageView [boutonArreterSeance](#)
- Spinner [spinnerListeJoueurs](#)
- ImageButton [boutonAppliquerNomJoueur](#)
- SeekBar [barreProgressionNombreBalles](#)
- TextView [texteValeurNombreBalles](#)
- SeekBar [barreProgressionFrequenceBalles](#)
- TextView [texteValeurFrequenceBalles](#)
- SeekBar [barreProgressionEffetBalles](#)
- TextView [texteValeurEffetBalles](#)
- SeekBar [barreProgressionPuissanceBalles](#)
- TextView [texteValeurPuissanceBalles](#)
- SeekBar [barreProgressionRotationLanceur](#)
- TextView [texteValeurRotationLanceur](#)
- [ServeurBDD](#) [serveurBDD](#)
- int [etatSeance](#) = [ETAT\\_SEANCE\\_ARRETEE](#)
- [Seance](#) [seanceEnCours](#)
- int [ballesJouees](#)
- int [ballesReussies](#)
- String [nomJoueur](#)
- BluetoothAdapter [adaptateurBluetooth](#) = null
- Set< BluetoothDevice > [devices](#)
- final Handler [handler](#)

#### Attributs privés statiques

- static final int [COULEUR\\_BOUTON\\_VERT](#) = Color.parseColor("#5eed7b")
- static final int [COULEUR\\_BOUTON\\_ORANGE](#) = Color.parseColor("#f7bb31")
- static final int [COULEUR\\_BOUTON\\_GRISE](#) = Color.parseColor("#c0c5c6")
- static final int [COULEUR\\_BOUTON\\_ROUGE](#) = Color.parseColor("#ee5e5e")
- static final int [ETAT\\_SEANCE\\_ARRETEE](#) = 0
- static final int [ETAT\\_SEANCE\\_DEMARREE](#) = 1
- static final int [ETAT\\_SEANCE\\_PAUSE](#) = 2
- static final String [NOM\\_PERIPHERIQUE\\_BLUETOOTH\\_ECRAN](#) = "TTPA-Ecran"
- static final String [NOM\\_PERIPHERIQUE\\_BLUETOOTH\\_TABLE](#) = "TTPA-Table"
- static final String [NOM\\_PERIPHERIQUE\\_BLUETOOTH\\_LANCEUR](#) = "TTPA--Lanceur"
- static final String [TRAME\\_ENTETE](#) = "\$TTPA"
- static final String [TRAME\\_FIN](#) = "\r\n"
- static final String [TRAME\\_ECRAN\\_DEBUT\\_SEANCE](#) = ":START"
- static final String [TRAME\\_ECRAN\\_PAUSE\\_SEANCE](#) = ":PAUSE"
- static final String [TRAME\\_ECRAN\\_REPRISE\\_SEANCE](#) = ":RESUME"
- static final String [TRAME\\_ECRAN\\_FIN\\_SEANCE](#) = ":FINSEANCE"
- static final String [TRAME\\_LANCEUR\\_PAUSE\\_SEANCE](#) = ":PAUSE : " + [TRAME\\_FIN](#)
- static final String [TRAME\\_LANCEUR\\_REPRISE\\_SEANCE](#) = ":RESUME : " + [TRAME\\_FIN](#)
- static final String [TRAME\\_LANCEUR\\_ARRET\\_SEANCE](#) = ":STOP : " + [TRAME\\_FIN](#)
- static final String [TRAME\\_LANCEUR\\_PING](#) = ":PING : " + [TRAME\\_FIN](#)
- static final String [TRAME\\_TABLE\\_ARRET\\_SEANCE](#) = ":RESET"
- static final int [REQUEST\\_CODE\\_ENABLE\\_BLUETOOTH](#) = 0
- static final int [SELECTION\\_ZONE\\_OBJECTIF](#) = 0
- static final int [SELECTION\\_ZONE\\_ROBOT](#) = 1

#### 9.1.1 Description détaillée

Created by smaniotto on 05/04/18. Classe [IHMEcranPrincipal](#) définissant le comportement du layout 'ecran\_principal'.

## 9.1.2 Documentation des fonctions membres

9.1.2.1 `void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actionnerSeance ( )`  
[private]

Méthode `actionnerSeance()` permettant de démarrer ou d'arrêter une séance en fonction de l'état actuel.

Références `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ETAT_SEANCE_ARRETEE`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ETAT_SEANCE_DEMARREE`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ETAT_SEANCE_PAUSE`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.etatSeance`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.idZonePrecedente`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.pauserSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.reprendreSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.SELECTION_ZONE_ROBOT`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.selectionnerZone()`.

Référéncé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick()`.

```
{
 Log.d("IHMEcranPrincipal", "actionnerSeance() état : " + etatSeance);

 // Démarrer ou arrêter la séance selon son état actuel
 switch (etatSeance)
 {
 case ETAT_SEANCE_ARRETEE:
 // La séance n'est pas commencée, il faut la démarrer
 if(idZonePrecedente == -1) // Si les zones n'ont pas été
 sélectionnées
 selectionnerZone(SELECTION_ZONE_ROBOT, 0, 0);
 else // Sinon, on peut démarrer la séance
 {
 demarrerSeance();
 etatSeance = ETAT_SEANCE_DEMARREE;
 }
 break;
 case ETAT_SEANCE_DEMARREE:
 // La séance est en cours, il faut la metre en pause
 pauserSeance();
 etatSeance = ETAT_SEANCE_PAUSE;
 break;
 case ETAT_SEANCE_PAUSE:
 // La séance est en pause, il faut la reprendre
 reprendreSeance();
 etatSeance = ETAT_SEANCE_DEMARREE;
 break;
 default:
 break;
 }
}
```

9.1.2.2 `void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actualiserIHMAppeareils-Bluetooth ( boolean tableEstConnectee, boolean lanceurEstConnecte, boolean ecranEstConnecte )` [private]

Méthode `actualiserIHMAppeareilsBluetooth()` permettant de modifier les voyants des appareils Bluetooth en fonction des appareils connectés

Références `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.voyantEcran`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.voyantLanceur`, et `com.ttpa.iris.ttpamobile.IHMEcran-`

`Principal.voyantTable`.

Référéncé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.verifierConnexionAppareilsBluetoothRequis()`.

```
{
 if (tableEstConnectee)
 voyantTable.setImageResource(R.drawable.table_connectee);
 else
 voyantTable.setImageResource(R.drawable.table_deconnectee);

 if (lanceurEstConnecte)
 voyantLanceur.setImageResource(R.drawable.lanceur_connecte);
 else
 voyantLanceur.setImageResource(R.drawable.lanceur_deconnecte);

 if (ecranEstConnecte)
 voyantEcran.setImageResource(R.drawable.ecran_connecte);
 else
 voyantEcran.setImageResource(R.drawable.ecran_deconnecte);
}
```

### 9.1.2.3 `void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affecterMembresIHM ( )` [private]

Méthode `affecterMembresIHM()` permettant l'affectation des membres de l'IHM (boutons, barres de progressions, textes, ...) aux attributs correspondants.

Références `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionEffetBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionFrequenceBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionNombreBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionPuissanceBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionRotationLanceur`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonActionSeance`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonAppliquerNomJoueur`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonArreterSeance`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonBluetooth`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonHistorique`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonParametres`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.spinnerListeJoueurs`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeurEffetBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeurFrequenceBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeurNombreBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeurPuissanceBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeurRotationLanceur`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.voyantEcran`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.voyantLanceur`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.voyantTable`.

Référéncé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onCreate()`.

```
{
 Log.d("IHMEcranPrincipal", "affecterMembresIHM()");

 //editTextNomJoueur = (EditText) findViewById(R.id.editTextNomJoueur);
 boutonAppliquerNomJoueur = (ImageButton) findViewById(R.id.boutonAppliquerNomJoueur);
 boutonBluetooth = (ImageButton) findViewById(R.id.boutonBluetooth);
 boutonHistorique = (ImageButton) findViewById(R.id.boutonHistorique);
 boutonParametres = (ImageButton) findViewById(R.id.boutonParametres);
 voyantTable = (ImageView) findViewById(R.id.voyantTable);
 voyantLanceur = (ImageView) findViewById(R.id.voyantLanceur);
 voyantEcran = (ImageView) findViewById(R.id.voyantEcran);
}
```

```

 boutonActionSeance = (ImageView) findViewById(R.id.boutonActionSeance);
 boutonArreterSeance = (ImageView) findViewById(R.id.boutonArreterSeance);
 };
 barreProgressionNombreBalles = (SeekBar) findViewById(R.id.
barreProgressionNombreBalles);
 texteValeurNombreBalles = (TextView) findViewById(R.id.
texteValeurNombreBalles);
 barreProgressionFrequenceBalles = (SeekBar) findViewById(R.id.
barreProgressionFrequenceBalles);
 texteValeurFrequenceBalles = (TextView) findViewById(R.id.
texteValeurFrequenceBalles);
 barreProgressionEffetBalles = (SeekBar) findViewById(R.id.
barreProgressionEffetBalles);
 texteValeurEffetBalles = (TextView) findViewById(R.id.
texteValeurEffetBalles);
 barreProgressionPuissanceBalles = (SeekBar) findViewById(R.id.
barreProgressionPuissanceBalles);
 texteValeurPuissanceBalles = (TextView) findViewById(R.id.
texteValeurPuissanceBalles);
 barreProgressionRotationLanceur = (SeekBar) findViewById(R.id.
barreProgressionRotationLanceur);
 texteValeurRotationLanceur = (TextView) findViewById(R.id.
texteValeurRotationLanceur);
 spinnerListeJoueurs = (Spinner) findViewById(R.id.spinnerListeJoueurs);
 spinnerListeJoueurs.setContentDescription("La liste des joueurs");
}

```

#### 9.1.2.4 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ajouterJoueur ( ) [private]

Méthode `ajouterJoueur()` ajoutant un joueur saisi à la base de données.

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.insererJoueur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.nomJoueur](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.serveurBDD](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick\(\)](#).

```

{
 AlertDialog.Builder ajoutJoueur = new AlertDialog.Builder(this);
 LayoutInflater factory = LayoutInflater.from(this);
 final View ajoutJoueurView = factory.inflate(R.layout.ajout_joueur,
null);
 ajoutJoueur.setView(ajoutJoueurView);

 ajoutJoueur.setTitle("Ajouter un nouveau joueur");

 ajoutJoueur.setPositiveButton("Valider", new DialogInterface.
OnClickListener()
 {
 public void onClick(DialogInterface dialog, int which)
 {
 //Lorsque l'on cliquera sur le bouton "OK", on récupère
l'EditText correspondant à notre vue personnalisée (cad à alertDialogView)
 EditText nomJoueur = (EditText)ajoutJoueurView.findViewById(R.
id.editTextNom);
 Joueur joueur = new Joueur(nomJoueur.getText().toString());
 long id = serveurBDD.insererJoueur(joueur);
 Log.d("IHMEcranPrincipal", "Nom joueur : " + nomJoueur.getText(
).toString() + " - id : " + id);
 Toast.makeText(getApplicationContext(), "Joueur " + nomJoueur.
getText() + " ajouté", Toast.LENGTH_SHORT).show();
 creerListeJoueurs();
 }
 });

 ajoutJoueur.setNegativeButton("Annuler", new DialogInterface.

```

```

OnClickListener()
{
 public void onClick(DialogInterface dialog, int which)
 {

 }
});

ajoutJoueur.show();
}

```

#### 9.1.2.5 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance ( ) [private]

Méthode [appliquerParametresSeance\(\)](#) appliquant les valeurs des paramètres à l'objet parametresActuels.

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionEffetBalles](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionFrequenceBalles](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionNombreBalles](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionPuissanceBalles](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionRotationLanceur](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ETAT\\_SEANCE\\_ARRETEE](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.etatSeance](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.nomJoueur](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.parametresActuels](#), [com.ttpa.iris.ttpamobile.ParametreSeance.setEffetBalles\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.setFrequenceBalles\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.setIntensiteEffet\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.setNombreBalles\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.setNomJoueur\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.setPuissanceBalles\(\)](#), et [com.ttpa.iris.ttpamobile.ParametreSeance.setRotation\(\)](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onCreate\(\)](#).

```

{
 Log.d("IHMEcranPrincipal", "appliquerParametresSeance()");

 if(etatSeance == ETAT_SEANCE_ARRETEE)
 {
 parametresActuels.setNomJoueur(nomJoueur);
 parametresActuels.setNombreBalles((barreProgressionNombreBalles.
getProgress() * 5) + 5); // 5 balles par palier, 5 balles minimum
 parametresActuels.setFrequenceBalles((
barreProgressionFrequenceBalles.getProgress() * 5) + 30); // 5 balles par
palier, 30 balles minimum

 String effet;
 int intensiteEffet = barreProgressionEffetBalles.getProgress() - 8;

 if (barreProgressionEffetBalles.getProgress() == 8)
 {
 effet = "Aucun";
 intensiteEffet = 1;
 }
 else if (barreProgressionEffetBalles.getProgress() < 8)
 {
 effet = "Coupé";
 intensiteEffet = 0 - intensiteEffet;
 }
 else
 effet = "Lifté";
 }
}

```



```

 parametresActuels.setEffetBalles(effet);
 parametresActuels.setIntensiteEffet(intensiteEffet);
 parametresActuels.setPuissanceBalles(barreProgressionPuissanceBalles
.getProgress() + 1); // 1 minimum (soit 10% minimum)
 parametresActuels.setRotation(barreProgressionRotationLanceur.
getProgress() * 5); // 5° par pallier
 }
}

```

#### 9.1.2.6 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance ( boolean seanceAEnregistrer ) [private]

Méthode `arreterSeance()` permettant d'envoyer les trames correspondantes aux appareils Bluetooth du projet.

Références `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionEffetBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionFrequenceBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionNombreBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionPuissanceBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionRotationLanceur`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonActionSeance`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonArreterSeance`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameArretPeripheriqueBluetoothEcran()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameArretPeripheriqueBluetoothLanceur()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameArretPeripheriqueBluetoothTable()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ETAT_SEANCE_ARRETEE`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.etatSeance`, `com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.seanceEnCours`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.serveurBDD`.

Référéncé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.calculerReussiteSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.incrementerBallesJouees()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothInsuffisant()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick()`.

```

{
 Log.d("IHMEcranPrincipal", "arreterSeance()");
 // Envoyer la trame d'arrêt à l'écran
 envoyerTrameArretPeripheriqueBluetoothEcran();

 // Envoyer la trame d'arrêt au lanceur
 envoyerTrameArretPeripheriqueBluetoothLanceur();

 // Envoyer la trame d'arrêt à la table
 envoyerTrameArretPeripheriqueBluetoothTable();

 if (seanceAEnregistrer)
 {
 // Enregistrer la séance dans la base de données
 serveurBDD.insererSeance(seanceEnCours);
 }

 // Changer l'icône du bouton d'action
 boutonActionSeance.setImageResource(R.drawable.bouton_demarrer);
 boutonActionSeance.setEnabled(true);

 // Changer l'état et la visibilité du bouton d'arrêt de séance
 boutonArreterSeance.setEnabled(false);
 boutonArreterSeance.setVisibility(View.INVISIBLE);
}

```

```

// Changer l'état de la séance
etatSeance = ETAT_SEANCE_ARRETEE;

// Changer les états des barres de paramétrage
barreProgressionNombreBalles.setEnabled(true);
barreProgressionFrequenceBalles.setEnabled(true);
barreProgressionEffetBalles.setEnabled(true);
barreProgressionPuissanceBalles.setEnabled(true);
barreProgressionRotationLanceur.setEnabled(true);
}

```

#### 9.1.2.7 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.attendre ( int tempsMillisecondes ) [private]

Méthode [attendre\(\)](#) permettant d'attendre un temps données.

##### Paramètres

|                                 |  |
|---------------------------------|--|
| <i>temps-<br/>Millisecondes</i> |  |
|---------------------------------|--|

```

{
 Log.d("IHMEcranPrincipal", "attendre()");

 try
 {
 Thread.sleep(tempsMillisecondes);
 }
 catch (InterruptedException e)
 {
 e.printStackTrace();
 }
}

```

#### 9.1.2.8 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.calculerReussiteSeance ( int zoneTouchee ) [private]

Méthode [calculerReussiteSeance\(\)](#) calculant le taux de réussite de la séance selon le nombre de balles ayant déjà touché l'objectif.

##### Paramètres

|                          |                            |
|--------------------------|----------------------------|
| <i>zone-<br/>Touchee</i> | étant la zone de l'impacte |
|--------------------------|----------------------------|

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ballesJouees](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ballesReussies](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ETAT\\_SEANCE\\_DEMARREE](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.etatSeance](#), [com.ttpa.iris.ttpamobile.Seance.getNombreBalles\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getZoneObjectif\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.seanceEnCours](#), et [com.ttpa.iris.ttpamobile.Seance.setTauxReussite\(\)](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonneesRecuesTable\(\)](#).

```

{
 Log.d("IHMEcranPrincipal", "calculerReussiteSeance() zone touchée : " +

```

```

zoneTouchee);

if(etatSeance == ETAT_SEANCE_DEMARREE)
{
 if (zoneTouchee != -1) // Si la balle a bien touché la table
 {
 if ((seanceEnCours.getZoneObjectif() != -1) && (seanceEnCours.
getZoneObjectif() != 0)) { // Si l'objectif a été défini
 if (zoneTouchee == seanceEnCours.getZoneObjectif()) // Si
la zone touchée est la même que l'objectif
 ballesReussies++;
 } else // Si aucun objectif n'a été défini: la table entière
est l'objectif
 ballesReussies++;
 }
 seanceEnCours.setTauxReussite((float) ballesReussies / (float)
seanceEnCours.getNombreBalles() * 100);

 if (ballesJouees == seanceEnCours.getNombreBalles())
 arreterSeance(true);
 }
}

```

#### 9.1.2.9 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionBaseDeDonnees ( ) [private]

Méthode [connexionBaseDeDonnees\(\)](#) permettant la création puis la connexion à la base de données.

Références [com.ttpa.iris.ttpamobile.ServeurBDD.open\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.serveurBDD](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onCreate\(\)](#).

```

{
 Log.d("IHMEcranPrincipal", "connexionBaseDeDonnees()");

 serveurBDD = new ServeurBDD(this);
 serveurBDD.open();
}

```

#### 9.1.2.10 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothEcran ( BluetoothDevice *appareilBluetooth* ) [private]

Méthode [connexionPeripheriqueBluetoothEcran\(\)](#) permettant la connexion Bluetooth à l'écran.

##### Paramètres

|                          |                               |
|--------------------------|-------------------------------|
| <i>appareilBluetooth</i> | étant le Bluetooth de l'écran |
|--------------------------|-------------------------------|

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.connecter\(\)](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.estConnecte\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.handler](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothEcran](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriquesBluetooth\(\)](#).

```
{
 Log.d("IHMEcranPrincipal", "connexionPeripheriqueBluetoothEcran()");

 peripheriqueBluetoothEcran = new PeripheriqueBluetooth(
 appareilBluetooth, handler);

 //Toast.makeText(getApplicationContext(), "Connexion à l'écran ...",
 Toast.LENGTH_SHORT).show();

 peripheriqueBluetoothEcran.connecter();

 //attendre(2000);

 if (peripheriqueBluetoothEcran.estConnecte())
 Toast.makeText(getApplicationContext(), "Ecran connecté !", Toast.
 LENGTH_SHORT).show();
}
```

#### 9.1.2.11 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexion- PeripheriqueBluetoothLanceur ( BluetoothDevice *appareilBluetooth* ) [private]

Méthode [connexionPeripheriqueBluetoothLanceur\(\)](#) permettant la connexion Bluetooth du lanceur.

##### Paramètres

|                           |                               |
|---------------------------|-------------------------------|
| <i>appareil-Bluetooth</i> | étant le Bluetooth du lanceur |
|---------------------------|-------------------------------|

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.connecter\(\)](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.estConnecte\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.handler](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothLanceur](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriquesBluetooth\(\)](#).

```
{
 Log.d("IHMEcranPrincipal", "connexionPeripheriqueBluetoothLanceur()");

 peripheriqueBluetoothLanceur = new PeripheriqueBluetooth(
 appareilBluetooth, handler);

 //Toast.makeText(getApplicationContext(), "Connexion au lanceur ...",
 Toast.LENGTH_SHORT).show();

 peripheriqueBluetoothLanceur.connecter();

 //attendre(2000);

 if (peripheriqueBluetoothLanceur.estConnecte())
 Toast.makeText(getApplicationContext(), "Lanceur connecté !", Toast.
 LENGTH_SHORT).show();
}
```

9.1.2.12 void **com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexion-  
PeripheriqueBluetoothTable** ( BluetoothDevice *appareilBluetooth* )  
[private]

Méthode **connexionPeripheriqueBluetoothTable()** permettant la connexion Bluetooth à la table.

Paramètres

|                                |                                |
|--------------------------------|--------------------------------|
| <i>appareil-<br/>Bluetooth</i> | étant le Bluetooth de la table |
|--------------------------------|--------------------------------|

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.connecter\(\)](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.estConnecte\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.handler](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothTable](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriquesBluetooth\(\)](#).

```
{
 Log.d("IHMEcranPrincipal", "connexionPeripheriqueBluetoothTable()");

 peripheriqueBluetoothTable = new PeripheriqueBluetooth(
 appareilBluetooth, handler);

 //Toast.makeText(getApplicationContext(), "Connexion à la table ...",
 Toast.LENGTH_SHORT).show();

 peripheriqueBluetoothTable.connecter();

 //attendre(2000);

 if (peripheriqueBluetoothTable.estConnecte())
 Toast.makeText(getApplicationContext(), "Table connectée !", Toast.
 LENGTH_SHORT).show();
}
```

9.1.2.13 void **com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexion-  
PeripheriquesBluetooth** ( ) [private]

Méthode **connexionPeripheriquesBluetooth()** permettant la connexion aux appareils - Bluetooth du projet détectés.

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothEcran\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothLanceur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothTable\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.deconnexionPeripheriquesBluetooth\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.devices](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.handler](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.NOM\\_PERIPHERIQUE\\_BLUETOOTH\\_ECRAN](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.NOM\\_PERIPHERIQUE\\_BLUETOOTH\\_LANCEUR](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.NOM\\_PERIPHERIQUE\\_BLUETOOTH\\_TABLE](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriques](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrageBluetooth\(\)](#).

```
{
```

```

Log.d("IHMEcranPrincipal", "connexionPeripheriquesBluetooth()");

// Déconnexion de tous les appareils avant une possible connexion
deconnexionPeripheriquesBluetooth();

for (BluetoothDevice appareilBluetooth : devices)
{
 //Toast.makeText(getApplicationContext(), "Périphérique = " +
 appareilBluetooth.getName(), Toast.LENGTH_SHORT).show();
 peripheriques.add(new PeripheriqueBluetooth(appareilBluetooth,
 handler));

 switch (appareilBluetooth.getName())
 {
 case NOM_PERIPHERIQUE_BLUETOOTH_ECRAN:
 connexionPeripheriqueBluetoothEcran(appareilBluetooth);
 break;
 case NOM_PERIPHERIQUE_BLUETOOTH_TABLE:
 connexionPeripheriqueBluetoothTable(appareilBluetooth);
 break;
 case NOM_PERIPHERIQUE_BLUETOOTH_LANCEUR:
 connexionPeripheriqueBluetoothLanceur(appareilBluetooth);
 break;
 default:
 break;
 }
}

if(peripheriques.size() == 0)
{
 Toast.makeText(getApplicationContext(), "Aucun périphérique détecté
 ! ", Toast.LENGTH_SHORT).show();
}
}

```

#### 9.1.2.14 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs ( ) [private]

Méthode `creerListeJoueurs()` créant la liste des joueurs présents dans la base de données

Références `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameConnexion-PeripheriqueBluetoothEcran()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ETA-T_SEANCE_ARRETEE`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.etatSeance`, `com.ttpa.iris.ttpamobile.Joueur.getId()`, `com.ttpa.iris.ttpamobile.ServeurBDD.getIdJoueurParametres()`, `com.ttpa.iris.ttpamobile.ServeurBDD.getJoueur()`, `com.ttpa.iris.ttpamobile.ServeurBDD.getJoueurs()`, `com.ttpa.iris.ttpamobile.Joueur.getNom()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.nomJoueur`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.parametresActuels`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.serveurBDD`, `com.ttpa.iris.ttpamobile.ServeurBDD.setIdJoueurParametres()`, `com.ttpa.iris.ttpamobile.ParametreSeance.setNomJoueur()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.spinnerListeJoueurs`.

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ajouterJoueur()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onCreate()`.

```

{
 final List<Joueur> listeJoueurs = serveurBDD.getJoueurs();
 final List<String> noms = new ArrayList<String>();

 // le dernier joueur a avoir utilisé l'application
 int idJoueur = serveurBDD.getIdJoueurParametres();
 for(int i = 0; i < listeJoueurs.size(); i++)
 {

```

```

 Joueur joueur = listeJoueurs.get(i);
 if(joueur.getId() == idJoueur)
 {
 noms.add(joueur.getNom());
 break;
 }
 }
 for(int i = 0; i < listeJoueurs.size(); i++)
 {
 Joueur joueur = listeJoueurs.get(i);
 if(joueur.getId() == idJoueur)
 continue;
 noms.add(joueur.getNom());
 }

 ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R
.layout.simple_spinner_item, noms);
 adapter.setDropDownViewResource(android.R.layout
.simple_spinner_dropdown_item);
 spinnerListeJoueurs.setAdapter(adapter);

 spinnerListeJoueurs.setOnItemClickListener(new AdapterView.
OnItemSelectedListener()
 {
 @Override
 public void onItemClick(AdapterView<?> arg0, View arg1, int
position, long id)
 {
 Joueur joueur = serveurBDD.getJoueur(noms.get(position));
 Log.d("IHMEcranPrincipal", "Nom joueur sélectionné : " + noms.
get(position));
 // On conserve son id pour la prochaine session
 serveurBDD.setIdJoueurParametres(joueur.getId());
 nomJoueur = joueur.getNom();
 if(etatSeance == ETAT_SEANCE_ARRETEE)
 parametresActuels.setNomJoueur(nomJoueur);

 // On envoie l'information à l'écran, si la séance n'a pas
encore été démarrée
 envoyerTrameConnexionPeripheriqueBluetoothEcran();
 }

 @Override
 public void onNothingSelected(AdapterView<?> arg0)
 {
 }
 });
}

```

### 9.1.2.15 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.deconnexion- PeripheriquesBluetooth ( ) [private]

Méthode `deconnexionPeripheriquesBluetooth()` permettant la déconnexion des appa-  
reils Bluetooth du projet connectés.

Références `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.deconnecter()`, `com.ttpa-  
iris.ttpamobile.IHMEcranPrincipal.modifierHMBluetoothInsuffisant()`, `com.ttpa.iris-  
ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothEcran`, `com.ttpa.iris.ttpamobile.IH-  
MEcranPrincipal.peripheriqueBluetoothLanceur`, et `com.ttpa.iris.ttpamobile.IHMEcran-  
Principal.peripheriqueBluetoothTable`.

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriques-  
Bluetooth()`.

```

{
 Log.d("IHMEcranPrincipal", "deconnexionPeripheriquesBluetooth()");
}

```

```

 if(peripheriqueBluetoothEcran != null)
 peripheriqueBluetoothEcran.deconnecter(true);

 if(peripheriqueBluetoothTable != null)
 peripheriqueBluetoothTable.deconnecter(true);

 if(peripheriqueBluetoothLanceur != null)
 peripheriqueBluetoothLanceur.deconnecter(true);

 modifierIHMBluetoothInsuffisant();
 }

```

#### 9.1.2.16 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrageBluetooth ( ) [private]

Méthode `demarrageBluetooth()` permettant le démarrage du Bluetooth puis la connexion automatique aux appareils du projet.

Références `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.adaptateurBluetooth`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriquesBluetooth()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.devices`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothInsuffisant()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriques`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.REQUEST_CODE_ENABLE_BLUETOOTH`.

Référéncé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onCreate()`.

```

{
 Log.d("IHMEcranPrincipal", "demarrageBluetooth()");

 adaptateurBluetooth = BluetoothAdapter.getDefaultAdapter();
 if (adaptateurBluetooth == null)
 {
 Toast.makeText(getApplicationContext(), "Bluetooth non activé !",
 Toast.LENGTH_SHORT).show();
 modifierIHMBluetoothInsuffisant();
 }
 else
 {
 if (!adaptateurBluetooth.isEnabled())
 {
 Toast.makeText(getApplicationContext(), "Bluetooth non activé !",
 Toast.LENGTH_SHORT).show();

 modifierIHMBluetoothInsuffisant();

 Intent activeBlueTooth = new Intent(BluetoothAdapter.
 ACTION_REQUEST_ENABLE);
 startActivityForResult(activeBlueTooth,
 REQUEST_CODE_ENABLE_BLUETOOTH);
 //bluetoothAdapter.enable();
 }
 else
 {
 Toast.makeText(getApplicationContext(), "Bluetooth activé,
 recherche en cours...", Toast.LENGTH_LONG).show();

 // Recherche des périphériques connus
 peripheriques = new ArrayList<PeripheriqueBluetooth>();
 devices = adaptateurBluetooth.getBondedDevices();

 // Connexion aux appareils Bluetooth du projet détectés
 connexionPeripheriquesBluetooth();
 }
 }
}

```



### 9.1.2.17 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance ( ) [private]

Méthode `demarrerSeance()` permettant d'envoyer les trames correspondantes aux appareils Bluetooth du projet.

Références `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ballesJouees`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ballesReussies`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionEffetBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionFrequenceBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionNombreBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionPuissanceBalles`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionRotationLanceur`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonActionSeance`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonArreterSeance`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeancePeripheriqueBluetoothEcran()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeancePeripheriqueBluetoothLanceur()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameParametrageSeancePeripheriqueBluetoothEcran()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getEffetCompleet()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getFrequenceBalles()`, `com.ttpa.iris.ttpamobile.ServeurBDD.getIdJoueurParametres()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getIntensiteEffet()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getNombreBalles()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getPuissanceBalles()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getRotation()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.parametresActuels`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.seanceEnCours`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.serveurBDD`, `com.ttpa.iris.ttpamobile.Seance.setIdJoueur()`, `com.ttpa.iris.ttpamobile.Seance.setZoneObjectif()`, `com.ttpa.iris.ttpamobile.Seance.setZoneRobot()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.zoneObjectif`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.zoneRobot`.

Référéncé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actionnerSeance()`.

```
{
 Log.d("IHMEcranPrincipal", "demarrerSeance()");

 // Lire les valeurs des paramètres actuels afin de les appliquer aux
 paramètres de la séance
 appliquerParametresSeance();

 // Envoyer la trame de paramétrage à l'écran, puis la trame de début de
 séance
 envoyerTrameDebutSeancePeripheriqueBluetoothEcran();

 envoyerTrameParametrageSeancePeripheriqueBluetoothEcran();

 // On applique les paramètres actuels à la séance en cours
 seanceEnCours = new Seance(parametresActuels.getFrequenceBalles(),
 parametresActuels.getNombreBalles(), parametresActuels.getEffetCompleet(),
 parametresActuels.getIntensiteEffet(), parametresActuels.getPuissanceBalles(),
 parametresActuels.getRotation());
 seanceEnCours.setIdJoueur(serveurBDD.getIdJoueurParametres());
 seanceEnCours.setZoneObjectif(zoneObjectif);
 seanceEnCours.setZoneRobot(zoneRobot);

 // Mettre à zéro les statistiques
 ballesJouees = 0;
 ballesReussies = 0;

 // Envoyer la trame de départ au lanceur
 envoyerTrameDebutSeancePeripheriqueBluetoothLanceur();
}
```

```

// Changer l'icône du bouton d'action
boutonActionSeance.setImageResource(R.drawable.bouton_pause);

// Changer l'état et la visibilité du bouton d'arrêt de séance
boutonArreterSeance.setEnabled(true);
boutonArreterSeance.setVisibility(View.VISIBLE);

// Changer les états des barres de paramétrage
barreProgressionNombreBalles.setEnabled(false);
barreProgressionFrequenceBalles.setEnabled(false);
barreProgressionEffetBalles.setEnabled(false);
barreProgressionPuissanceBalles.setEnabled(false);
barreProgressionRotationLanceur.setEnabled(false);
}

```

#### 9.1.2.18 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal. envoyerTrameArretPeripheriqueBluetoothEcran ( ) [private]

Méthode [envoyerTrameArretPeripheriqueBluetoothEcran\(\)](#) envoyant la trame de fin de séance à l'écran

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetoothEcran\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME\\_ECRAN\\_FIN\\_SEANCE](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance\(\)](#).

```

{
 Log.d("IHMEcranPrincipal", "
 envoyerTrameArretPeripheriqueBluetoothEcran()");

 envoyerTramePeripheriqueBluetoothEcran (TRAME_ECRAN_FIN_SEANCE);
}

```

#### 9.1.2.19 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal. envoyerTrameArretPeripheriqueBluetoothLanceur ( ) [private]

Méthode [envoyerTrameArretPeripheriqueBluetoothLanceur\(\)](#) envoyant la trame de fin de séance au lanceur.

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.envoyer\(\)](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.estConnecte\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothLanceur](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME\\_LANCEUR\\_ARRET\\_SEANCE](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance\(\)](#).

```

{
 Log.d("IHMEcranPrincipal", "
 envoyerTrameArretPeripheriqueBluetoothLanceur()");

 if (peripheriqueBluetoothLanceur != null)
 {
 if (peripheriqueBluetoothLanceur.estConnecte())
 {
 peripheriqueBluetoothLanceur.envoyer (TRAME_LANCEUR_ARRET_SEANCE
);
 Log.d("IHMEcranPrincipal", "Trame arrêt séance Lanceur : " +

```

```

 TRAME_LANCEUR_ARRET_SEANCE);
 }
}

```

#### 9.1.2.20 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal. envoyerTrameArretPeripheriqueBluetoothTable ( ) [private]

Méthode `envoyerTrameArretPeripheriqueBluetoothTable()` envoyant la trame de fin de séance au lanceur.

Références `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.envoyer()`, `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.estConnecte()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothTable`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.T-RAME_TABLE_ARRET_SEANCE`.

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance()`.

```

{
 Log.d("IHMEcranPrincipal", "
envoyerTrameArretPeripheriqueBluetoothTable()");

 if (peripheriqueBluetoothTable != null)
 {
 if (peripheriqueBluetoothTable.estConnecte())
 {
 peripheriqueBluetoothTable.envoyer(TRAME_TABLE_ARRET_SEANCE);
 Log.d("IHMEcranPrincipal", "Trame arrêt séance Table : " +
TRAME_TABLE_ARRET_SEANCE);
 }
 }
}

```

#### 9.1.2.21 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyer- TrameConnexionPeripheriqueBluetoothEcran ( ) [private]

Méthode `envoyerTrameConnexionPeripheriqueBluetoothEcran()` envoyant la trame de connexion à l'écran.

Références `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetoothEcran()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getNomJoueur()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.parametresActuels`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME_ENTETE`.

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick()`.

```

{
 Log.d("IHMEcranPrincipal", "
envoyerTrameConnexionPeripheriqueBluetoothEcran()");

 String trame = TRAME_ENTETE + ":CONNECT:" + parametresActuels.
getNomJoueur();

 envoyerTramePeripheriqueBluetoothEcran(trame);
}

```

**9.1.2.22 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyer-  
TrameDebutSeancePeripheriqueBluetoothEcran ( )**  
[private]

Méthode `envoyerTrameFinParametrageSeancePeripheriqueBluetoothEcran()` envoyant la trame de fin de paramétrage à l'écran.

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetoothEcran\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME\\_ECRAN\\_DEBUT\\_SEANCE](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#).

```
{
 Log.d("IHMEcranPrincipal", "
 envoyerTrameDebutSeancePeripheriqueBluetoothEcran()");

 envoyerTramePeripheriqueBluetoothEcran (TRAME_ECRAN_DEBUT_SEANCE);
}
```

**9.1.2.23 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyer-  
TrameDebutSeancePeripheriqueBluetoothLanceur ( )**  
[private]

Méthode `envoyerTrameDebutSeancePeripheriqueBluetoothLanceur()` envoyant la trame de début de séance au lanceur.

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.envoyer\(\)](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.estConnecte\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.getEffet\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.getFrequenceBalles\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.getIntensiteEffet\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.getNombreBalles\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.getPuissanceBalles\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.getRotation\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.parametresActuels](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothLanceur](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME\\_ENTETE](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME\\_FIN](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#).

```
{
 Log.d("IHMEcranPrincipal", "
 envoyerTrameDebutSeancePeripheriqueBluetoothLanceur()");

 if (peripheriqueBluetoothLanceur != null)
 {
 if (peripheriqueBluetoothLanceur.estConnecte())
 {
 String trame = TRAME_ENTETE + ":SET:" + parametresActuels.
 getEffet() + ":" + parametresActuels.getIntensiteEffet() + ":" +
 parametresActuels.getPuissanceBalles() + ":" + parametresActuels.
 getFrequenceBalles() + ":" + parametresActuels.getRotation() + ":" +
 parametresActuels.getNombreBalles() + TRAME_FIN;
 peripheriqueBluetoothLanceur.envoyer(trame);
 Log.d("IHMEcranPrincipal", "Trame départ séance Lanceur : " +
 trame);
 }
 }
}
```

### 9.1.2.24 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyer- TrameParametrageSeancePeripheriqueBluetoothEcran ( ) [private]

Méthode `envoyerTrameParametrageSeancePeripheriqueBluetoothEcran()` envoyant la trame de paramétrage à l'écran.

Références `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetoothEcran()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getNombreBalles()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.parametresActuels`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME_ENTETE`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.zoneObjectif`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.zoneRobot`.

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance()`.

```
{
 Log.d("IHMEcranPrincipal", "
envoyerTrameParametrageSeancePeripheriqueBluetoothEcran()");

 String trame = TRAME_ENTETE + ":SETSEANCE:" + zoneRobot + ":" +
zoneObjectif + ":" + parametresActuels.getNombreBalles();

 envoyerTramePeripheriqueBluetoothEcran(trame);
}
```

### 9.1.2.25 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyer- TramePeripheriqueBluetoothEcran ( String trame ) [private]

Méthode `envoyerTramePeripheriqueBluetoothEcran()` envoyant la trame à l'écran, si l'écran est connecté.

#### Paramètres

|              |                          |
|--------------|--------------------------|
| <i>trame</i> | étant la trame à envoyer |
|--------------|--------------------------|

Références `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.envoyer()`, `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.estConnecte()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothEcran`.

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameArretPeripheriqueBluetoothEcran()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameConnexionPeripheriqueBluetoothEcran()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeancePeripheriqueBluetoothEcran()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameParametrageSeancePeripheriqueBluetoothEcran()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.pauserSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.reprendreSeance()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonneesRecuesTable()`.

```
{
 Log.d("IHMEcranPrincipal", "envoyerTramePeripheriqueBluetoothEcran()");

 if(peripheriqueBluetoothEcran != null)
 {
 if (peripheriqueBluetoothEcran.estConnecte())
 peripheriqueBluetoothEcran.envoyer(trame);
 }
}
```

9.1.2.26 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyer-  
TramePeripheriqueBluetoothLanceur ( String trame )  
[private]

Méthode `envoyerTramePeripheriqueBluetoothLanceur()` envoyant la trame au lanceur, si le lanceur est connecté.

#### Paramètres

|              |                          |
|--------------|--------------------------|
| <i>trame</i> | étant la trame à envoyer |
|--------------|--------------------------|

Références `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.envoyer()`, `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.estConnecte()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothLanceur`.

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.pauserSeance()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.reprendreSeance()`.

```
{
 Log.d("IHMEcranPrincipal", "envoyerTramePeripheriqueBluetoothLanceur()");
}
if(peripheriqueBluetoothLanceur != null)
{
 if (peripheriqueBluetoothLanceur.estConnecte())
 peripheriqueBluetoothLanceur.envoyer(trame);
}
```

9.1.2.27 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyer-  
TrameRepriseSeancePeripheriqueBluetoothLanceur ( )  
[private]

Méthode `envoyerTrameRepriseSeancePeripheriqueBluetoothLanceur()` envoyant la trame de reprise de séance au lanceur.

Références `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ballesJouees`, `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.envoyer()`, `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.estConnecte()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getEffet()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getFrequenceBalles()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getIntensiteEffet()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getNombreBalles()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getPuissanceBalles()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getRotation()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.parametresActuels`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothLanceur`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME_ENTETE`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME_FIN`.

```
{
 Log.d("IHMEcranPrincipal", "envoyerTrameRepriseSeancePeripheriqueBluetoothLanceur()");
}
if (peripheriqueBluetoothLanceur != null)
{
 if (peripheriqueBluetoothLanceur.estConnecte())
 {
 String trame = TRAME_ENTETE + ":SET:" + parametresActuels.getEffet() + ":" + parametresActuels.getIntensiteEffet() + ":" + parametresActuels.getPuissanceBalles() + ":" + parametresActuels.getRotation();
 }
}
```

```

 getFrequenceBalles() + ":" + parametresActuels.getRotation() + ":" + (
 parametresActuels.getNombreBalles() - ballesJouees) + TRAME_FIN;
 peripheriqueBluetoothLanceur.envoyer(trame);
 Log.d("IHMEcranPrincipal", "Trame reprise séance Lanceur : " +
 trame);
 }
}

```

#### 9.1.2.28 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.incrementerBallesJouees( ) [private]

Méthode `incrementerBallesJouees()` permettant de traiter les nombre de balles jouées de la séance, et d'arrêter la séance si la dernière balle est en dehors.

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ballesJouees](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonActionSeance](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ETAT\\_SEANCE\\_DEMARREE](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.etatSeance](#), [com.ttpa.iris.ttpamobile.Seance.getNombreBalles\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.seanceEnCours](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonneesRecuesTable\(\)](#).

```

{
 Log.d("IHMEcranPrincipal", "incrementerBallesJouees()");

 if(etatSeance == ETAT_SEANCE_DEMARREE)
 {
 ++ballesJouees;

 if (ballesJouees == seanceEnCours.getNombreBalles())
 {
 boutonActionSeance.setEnabled(false);

 final Timer timerAsync = new Timer();
 final TimerTask timerTaskAsync = new TimerTask() {
 @Override
 public void run() {
 runOnUiThread(new Runnable() {
 @Override public void run() {
 if(etatSeance == ETAT_SEANCE_DEMARREE)
 arreterSeance(true);

 timerAsync.cancel();
 }
 });
 }
 };

 timerAsync.schedule(timerTaskAsync, 2000, 2000);
 }
 }
}

```

#### 9.1.2.29 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothInsuffisant( ) [private]

Méthode `modifierIHMBluetoothInsuffisant()` appelée lorsque le Bluetooth n'est pas activé ou que les appareils nécessaires ne sont pas présents/connectés

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonActionSeance](#), [com.ttpa.iris.ttpamobile.IHM-](#)

EcranPrincipal.boutonBluetooth, com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonParametres, et com.ttpa.iris.ttpamobile.IHMEcranPrincipal.COULEUR\_BOUTON\_ROUGE.

Référencé par com.ttpa.iris.ttpamobile.IHMEcranPrincipal.deconnexionPeripheriquesBluetooth(), com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrageBluetooth(), et com.ttpa.iris.ttpamobile.IHMEcranPrincipal.verifierConnexionAppareilsBluetoothRequis().

```
{
 Log.d("IHMEcranPrincipal", "modifierIHMBluetoothInsuffisant()");

 // Arrêter la séance en cours, sans enregistrer
 arreterSeance(false);

 // Modifier la couleur du bouton Bluetooth
 boutonBluetooth.setBackgroundColor(COULEUR_BOUTON_ROUGE);
 // Modifier la couleur du bouton réglages des zones
 boutonParametres.setBackgroundColor(COULEUR_BOUTON_ROUGE);
 // Autoriser le réglage des zones
 boutonParametres.setEnabled(false);
 // Changer l'icône du bouton d'action
 boutonActionSeance.setImageResource(R.drawable.
bouton_demarrer_desactive);
 // Empêcher une séance d'être jouée
 boutonActionSeance.setEnabled(false);
}
```

#### 9.1.2.30 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothOperationnel( ) [private]

Méthode modifierIHMBluetoothOperationnel() appelée lorsque tous les appareils - Bluetooth sont connectés.

Références com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonActionSeance, com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonBluetooth, com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonParametres, et com.ttpa.iris.ttpamobile.IHMEcranPrincipal.COULEUR\_BOUTON\_VERT.

Référencé par com.ttpa.iris.ttpamobile.IHMEcranPrincipal.verifierConnexionAppareilsBluetoothRequis().

```
{
 Log.d("IHMEcranPrincipal", "modifierIHMBluetoothOperationnel()");

 // Modifier la couleur du bouton Bluetooth
 boutonBluetooth.setBackgroundColor(COULEUR_BOUTON_VERT);
 // Modifier la couleur du bouton réglages des zones
 boutonParametres.setBackgroundColor(COULEUR_BOUTON_VERT);
 // Autoriser le réglage des zones
 boutonParametres.setEnabled(true);
 // Changer l'icône du bouton d'action
 boutonActionSeance.setImageResource(R.drawable.bouton_demarrer);
 // Autoriser le déroulement d'une séance
 boutonActionSeance.setEnabled(true);
}
```

#### 9.1.2.31 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothSuffisant( ) [private]

Méthode modifierIHMBluetoothSuffisant() appelée lorsque les appareils Bluetooth nécessaires sont connectés.



Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonActionSeance](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonBluetooth](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonParametres](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.COULEUR\\_BOUTON\\_ORANGE](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.COULEUR\\_BOUTON\\_VERT](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.verifierConnexionAppareilsBluetoothRequis\(\)](#).

```
{
 Log.d("IHMEcranPrincipal", "modifierIHMBluetoothSuffisant()");

 // Modifier la couleur du bouton Bluetooth
 boutonBluetooth.setBackgroundColor(COULEUR_BOUTON_ORANGE);
 // Modifier la couleur du bouton réglages des zones
 boutonParametres.setBackgroundColor(COULEUR_BOUTON_VERT);
 // Autoriser le réglage des zones
 boutonParametres.setEnabled(true);
 // Changer l'icône du bouton d'action
 boutonActionSeance.setImageResource(R.drawable.bouton_demarrer);
 // Autoriser le déroulement d'une séance
 boutonActionSeance.setEnabled(true);
}
```

#### 9.1.2.32 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierValeursParametresIHM( ) [private]

Méthode [modifierValeursParametresIHM\(\)](#) modifiant les valeurs des champs de paramètre en fonction de leur barre de progression correspondante.

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionEffetBalles](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionFrequenceBalles](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionNombreBalles](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionPuissanceBalles](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionRotationLanceur](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeurEffetBalles](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeurFrequenceBalles](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeurNombreBalles](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeurPuissanceBalles](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeurRotationLanceur](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onCreate\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onProgressChanged\(\)](#).

```
{
 Log.d("IHMEcranPrincipal", "modifierValeursParametresIHM()");

 texteValeurNombreBalles.setText(((barreProgressionNombreBalles.
 getProgress() * 5) + 5) + " balles"); // 5 balles par palier, 5 balles minimum
 texteValeurFrequenceBalles.setText(((barreProgressionFrequenceBalles.
 getProgress() * 5) + 30) + " balles/min"); // 5 balles par palier, 30 balles
 minimum

 String effet;
 int intensiteEffet = barreProgressionEffetBalles.getProgress() - 8;

 if (barreProgressionEffetBalles.getProgress() == 8)
 effet = "Aucun";
 else if (barreProgressionEffetBalles.getProgress() < 8)
 {
 effet = "Coupé";
 }
}
```

```

 intensiteEffet = 0 - intensiteEffet;
 }
 else
 effet = "Lifté";

 switch(effet)
 {
 case "Aucun":
 texteValeurEffetBalles.setText(effet);
 break;
 default:
 texteValeurEffetBalles.setText(effet + " " + intensiteEffet);
 }
 // 10% par palier

 texteValeurPuissanceBalles.setText(((barreProgressionPuissanceBalles.
 getProgress() * 10) + 10) + "%"); // 10% par palier, 10% minimum

 String stringRotation;
 int rotationActuelle = barreProgressionRotationLanceur.getProgress() *
 5; // 5° par palier

 if(rotationActuelle > 45)
 stringRotation = (rotationActuelle - 45) + "° à droite";
 else if(rotationActuelle < 45)
 stringRotation = (45 - rotationActuelle) + "° à gauche";
 else
 stringRotation = (rotationActuelle - 45) + "°";

 texteValeurRotationLanceur.setText(stringRotation);
}

```

### 9.1.2.33 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick ( View element )

Méthode onClick pour la gestion de l'évènement d'un click.

#### Paramètres

|                |                                                   |
|----------------|---------------------------------------------------|
| <i>element</i> | étant la vue sur laquelle l'utilisateur a cliqué. |
|----------------|---------------------------------------------------|

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actionnerSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ajouterJoueur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonActionSeance](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonAppliquerNomJoueur](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonArreterSeance](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonBluetooth](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonHistorique](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonParametres](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrageBluetooth\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameConnexionPeripheriqueBluetoothEcran\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.nomJoueur](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.parametresActuels](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.redirectionActiviteHistorique\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.SELECTION\\_ZONE\\_ROBOT](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.selectionnerZone\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.setNomJoueur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.verifierConnexionAppareilsBluetoothRequis\(\)](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ajouterJoueur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.selectionnerZone\(\)](#).

```
{
```

```

Log.d("IHMEcranPrincipal", "onClick()");

if(element == boutonAppliquerNomJoueur)
{
 ajouterJoueur();
}
else if(element == boutonBluetooth)
{
 if(!verifierConnexionAppareilsBluetoothRequis()) // Si tous les
appareils Bluetooth ne sont pas connectés
 {
 //Toast.makeText(getApplicationContext(), "Attendre..",
Toast.LENGTH_SHORT).show();
 demarrageBluetooth();
 }
}
else if(element == boutonHistorique)
{
 Log.d("IHMEcranPrincipal", "onClick() boutonHistorique");

 // Redirection vers l'activité de l'historique des séances
 redirectionActiviteHistorique();
}
else if(element == boutonParametres)
{
 // Gérer le click sur le bouton paramètres (engrenage)
 selectionnerZone(SELECTION_ZONE_ROBOT, 0, 0);
}
else if(element == boutonActionSeance)
{
 Log.d("IHMEcranPrincipal", "onClick() boutonActionSeance");
 parametresActuels.setNomJoueur(nomJoueur);
 envoyerTrameConnexionPeripheriqueBluetoothEcran();
 actionnerSeance();
}
else if(element == boutonArreterSeance)
{
 Log.d("IHMEcranPrincipal", "onClick() boutonArreterSeance");
 arreterSeance(false);
}
}

```

#### 9.1.2.34 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onCreate ( Bundle savedInstanceState ) [protected]

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affecterMembresIHM\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionBaseDeDonnees\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrageBluetooth\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierValeursParametresIHM\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBarresProgressionIHMUtilisables\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBoutonsIHMClickables\(\)](#).

```

{
 Log.d("IHMEcranPrincipal", "onCreate()");

 super.onCreate(savedInstanceState);
 setContentView(R.layout.ecran_principal);

 // Affectation des membres de l'IHM
 affecterMembresIHM();

 // Modifier les valeurs des champs de paramètre de l'IHM en fonction
des états des barres de progression
 modifierValeursParametresIHM();
}

```

```

// Rendre les boutons présents dans l'IHM cliquables
rendreBoutonsIHMClickables();

// Rendre les barres de progression présents dans l'IHM utilisables
rendreBarresProgressionIHMUtilisables();

// Connexion à la base de données
connexionBaseDeDonnees();

// Crée une liste des joueurs enregistrés
creerListeJoueurs();

// Appliquer les paramètres de séance actuels
appliquerParametresSeance();

// Démarrage du Bluetooth puis connexion aux appareils du projet
détectés
demarrageBluetooth();
}

```

#### 9.1.2.35 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onProgressChanged (SeekBar seekBar, int progress, boolean fromUser )

Méthode [onProgressChanged\(\)](#) appelée lorsqu'une barre de progression est modifiée.

##### Paramètres

|                 |                                                             |
|-----------------|-------------------------------------------------------------|
| <i>seekBar</i>  | étant la barre de progression modifiée                      |
| <i>progress</i> | étant le progrès actuel de la barre de progression modifiée |
| <i>fromUser</i> |                                                             |

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierValeursParametresIHM\(\)](#).

```

{
 modifierValeursParametresIHM();
}

```

#### 9.1.2.36 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onStartTrackingTouch (SeekBar seekBar )

Méthode [onStartTrackingTouch\(\)](#) appelée lorsqu'une barre de progression commence à être modifiée.

##### Paramètres

|                |                                        |
|----------------|----------------------------------------|
| <i>seekBar</i> | étant la barre de progression modifiée |
|----------------|----------------------------------------|

```

{}

```

#### 9.1.2.37 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onStopTrackingTouch (SeekBar seekBar )

Méthode [onStopTrackingTouch\(\)](#) appelée lorsqu'une barre de progression a fini d'être modifiée.

## Paramètres

|                      |                                        |
|----------------------|----------------------------------------|
| <code>seekBar</code> | étant la barre de progression modifiée |
|----------------------|----------------------------------------|

```
{
 //appliquerParametresSeance();
}
```

### 9.1.2.38 `void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.pauserSeance ( )` [private]

Méthode `pauserSeance()` permettant de mettre en pause un séance en cours.

Références `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonActionSeance`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetoothEcran()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetoothLanceur()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ETAT_SEANCE_PAUSE`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.etatSeance`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME_ECRAN_PAUSE_SEANCE`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME_LANCEUR_PAUSE_SEANCE`.

Référéncé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actionnerSeance()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterErreurRecueLanceur()`.

```
{
 Log.d("IHMEcranPrincipal", "pauserSeance()");

 // Changer l'icône du bouton d'action
 boutonActionSeance.setImageResource(R.drawable.bouton_reprendre);

 envoyerTramePeripheriqueBluetoothEcran(TRAME_ECRAN_PAUSE_SEANCE);
 envoyerTramePeripheriqueBluetoothLanceur(TRAME_LANCEUR_PAUSE_SEANCE);

 etatSeance = ETAT_SEANCE_PAUSE;
}
```

### 9.1.2.39 `void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.redirectionActivite-Historique ( )` [private]

Méthode `redirectionActiviteHistorique()` démarrart l'activité `IHMHistoriqueSeances` permettant de visualiser l'historique des séances du joueur actuel.

Référéncé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick()`.

```
{
 Log.d("IHMEcranPrincipal", "redirectionActiviteHistorique()");

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

### 9.1.2.40 `void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBarres-ProgressionIHMUtilisables ( )` [private]

Méthode `rendreBarresProgressionIHMUtilisables()`

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionEffetBalles](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionFrequenceBalles](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionNombreBalles](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionPuissanceBalles](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgressionRotationLanceur](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onCreate\(\)](#).

```
{
 Log.d("IHMEcranPrincipal", "rendreBarresProgressionIHMUtilisables()");

 barreProgressionNombreBalles.setOnSeekBarChangeListener(this);
 barreProgressionFrequenceBalles.setOnSeekBarChangeListener(this);
 barreProgressionEffetBalles.setOnSeekBarChangeListener(this);
 barreProgressionPuissanceBalles.setOnSeekBarChangeListener(this);
 barreProgressionRotationLanceur.setOnSeekBarChangeListener(this);
}
```

#### 9.1.2.41 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBoutonsIHM- Cliquables ( ) [private]

Méthode [rendreBoutonsIHMCliquables\(\)](#)

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonActionSeance](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonAppliquerNomJoueur](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonArreterSeance](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonBluetooth](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonHistorique](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonParametres](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onCreate\(\)](#).

```
{
 Log.d("IHMEcranPrincipal", "rendreBoutonsIHMCliquables()");

 boutonAppliquerNomJoueur.setOnClickListener(this);
 boutonBluetooth.setOnClickListener(this);
 boutonHistorique.setOnClickListener(this);
 boutonParametres.setOnClickListener(this);
 boutonActionSeance.setOnClickListener(this);
 boutonArreterSeance.setOnClickListener(this);
}
```

#### 9.1.2.42 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.reprendreSeance ( ) [private]

Méthode [reprendreSeance\(\)](#) permettant de reprendre une séance actuellement en pause.

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonActionSeance](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetoothEcran\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetoothLanceur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ETAT\\_SEANCE\\_DEMARREE](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.etatSeance](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME\\_ECRAN\\_REPRISE\\_SEANCE](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME\\_LANCEUR\\_REPRISE\\_SEANCE](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actionnerSeance\(\)](#).

```

{
 Log.d("IHMEcranPrincipal", "reprendreSeance()");

 // Changer l'icône du bouton d'action
 boutonActionSeance.setImageResource(R.drawable.bouton_pause);

 envoyerTramePeripheriqueBluetoothEcran (TRAME_ECRAN_REPRISE_SEANCE);
 envoyerTramePeripheriqueBluetoothLanceur (TRAME_LANCEUR_REPRISE_SEANCE);

 etatSeance = ETAT_SEANCE_DEMARREE;
}

```

#### 9.1.2.43 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.selectionnerZone ( final int typeSelection, int choixObjectif, int choixRobot ) [private]

Méthode `selectionnerZone()` affichant une boîte de dialogue permettant de sélectionner la zone du robot ou la zone de l'objectif.

##### Paramètres

|                       |  |
|-----------------------|--|
| <i>type-Selection</i> |  |
| <i>choixObjectif</i>  |  |
| <i>choixRobot</i>     |  |

Références `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.idZonePrecedente`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.idZonePrecedenteObjectif`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.idZonePrecedenteRobot`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.SELECTION_ZONE_OBJECTIF`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.SELECTION_ZONE_ROBOT`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.strZoneObjectif`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.strZoneRobot`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.zoneObjectif`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.zoneRobot`.

Référéncé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actionnerSeance()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick()`.

```

{
 Log.d("IHMEcranPrincipal", "selectionnerZone()");

 final AlertDialog.Builder selectionZone = new AlertDialog.Builder(this);
;
 LayoutInflater factory = LayoutInflater.from(this);
 final View selectionZoneView = factory.inflate(R.layout.zones, null);
 selectionZone.setView(selectionZoneView);

 List<Button> boutonsZone = new ArrayList<Button>();

 Button btnZone1 = (Button)selectionZoneView.findViewById(R.id.case1);
 boutonsZone.add(btnZone1);
 Button btnZone2 = (Button)selectionZoneView.findViewById(R.id.case2);
 boutonsZone.add(btnZone2);
 Button btnZone3 = (Button)selectionZoneView.findViewById(R.id.case3);
 boutonsZone.add(btnZone3);
 Button btnZone4 = (Button)selectionZoneView.findViewById(R.id.case4);
 boutonsZone.add(btnZone4);
 Button btnZone5 = (Button)selectionZoneView.findViewById(R.id.case5);
 boutonsZone.add(btnZone5);
 Button btnZone6 = (Button)selectionZoneView.findViewById(R.id.case6);
 boutonsZone.add(btnZone6);
 Button btnZone7 = (Button)selectionZoneView.findViewById(R.id.case7);
 boutonsZone.add(btnZone7);
}

```

```

Button btnZone8 = (Button)selectionZoneView.findViewById(R.id.case8);
boutonsZone.add(btnZone8);
Button btnZone9 = (Button)selectionZoneView.findViewById(R.id.case9);
boutonsZone.add(btnZone9);

if(choixObjectif > 0)
{
 Button btnZone = boutonsZone.get(choixObjectif - 1);
 btnZone.setBackgroundResource(R.drawable.case_cible);
}
if(choixRobot > 0)
{
 Button btnZone = boutonsZone.get(choixRobot - 1);
 btnZone.setBackgroundResource(R.drawable.case_robot);
}

for(int i = 0; i < 9; i++)
{
 Button btnZone = boutonsZone.get(i);

 // Changer l'icône de fond du bouton si la zone correspondante est
 // actuellement occupée par le robot ou l'objectif
 if((i + 1) == zoneRobot) // i + 1 étant le numéro de zone actuel
 btnZone.setBackgroundResource(R.drawable.case_robot);

 /*if((i + 1) == zoneObjectif) // i + 1 étant le numéro de zone
 actuel
 btnZone.setBackgroundResource(R.drawable.case_cible);*/

 btnZone.setOnClickListener(new View.OnClickListener() {
 public void onClick(View v) {
 Button myButton = (Button)selectionZoneView.findViewById(v.
 getId());
 int zoneActuelle = Integer.parseInt(myButton.getText().
 toString().substring(myButton.getText().toString().length() - 1));

 switch (typeSelection)
 {
 case SELECTION_ZONE_OBJECTIF:
 idZonePrecedente = idZonePrecedenteObjectif;
 break;
 case SELECTION_ZONE_ROBOT:
 idZonePrecedente = idZonePrecedenteRobot;
 break;
 }

 if(idZonePrecedente != -1)
 {
 Button myButtonPrecedent = (Button)selectionZoneView.
 findViewById(idZonePrecedente);
 int numeroZonePrecedente = Integer.parseInt(
 myButtonPrecedent.getText().toString().substring(myButtonPrecedent.getText().toString().
 length() - 1));

 switch(typeSelection)
 {
 case SELECTION_ZONE_OBJECTIF:
 if((v.getId() != idZonePrecedenteRobot) && (
 myButtonPrecedent.getId() != idZonePrecedenteRobot))
 myButtonPrecedent.setBackgroundResource(R.
 drawable.case_libre);
 break;
 /*case SELECTION_ZONE_ROBOT:
 if(v.getId() != idZonePrecedenteObjectif)
 myButtonPrecedent.setBackgroundResource(R.
 drawable.case_libre);
 break;*/
 default:
 myButtonPrecedent.setBackgroundResource(R.
 drawable.case_libre);
 break;
 }
 }
 }
 });
}

```



```

 switch (typeSelection)
 {
 case SELECTION_ZONE_OBJECTIF:
 if (v.getId() != idZonePrecedenteRobot)
 idZonePrecedenteObjectif = v.getId();
 break;
 case SELECTION_ZONE_ROBOT:
 idZonePrecedenteRobot = v.getId();
 break;
 }

 idZonePrecedente = v.getId();

 if(typeSelection == SELECTION_ZONE_OBJECTIF)
 {
 // Vérifier si l'objectif ne tombe pas sur la zone du
 robot
 if(zoneActuelle != zoneRobot)
 {
 if (strZoneObjectif == myButton.getText().toString(
))
 {
 myButton.setBackgroundResource(R.drawable.
 case_libre);
 strZoneObjectif = "ZONE 0";
 }
 else
 {
 myButton.setBackgroundResource(R.drawable.
 case_cible);
 strZoneObjectif = myButton.getText().toString()
 ;
 }

 Log.d("IHMEcranPrincipal", "Objectif : " +
 strZoneObjectif);
 }
 }
 else if(typeSelection == SELECTION_ZONE_ROBOT)
 {
 if (strZoneRobot == myButton.getText().toString())
 {
 myButton.setBackgroundResource(R.drawable.
 case_libre);
 strZoneRobot = "ZONE 0";
 }
 else
 {
 myButton.setBackgroundResource(R.drawable.
 case_robot);
 strZoneRobot = myButton.getText().toString();
 }

 Log.d("IHMEcranPrincipal", "Robot : " + strZoneRobot);
 }
 }
}

if(typeSelection == SELECTION_ZONE_OBJECTIF)
 selectionZone.setTitle("Placer l'objectif");
if(typeSelection == SELECTION_ZONE_ROBOT)
 selectionZone.setTitle("Placer le robot");

selectionZone.setPositiveButton("Valider", new DialogInterface.
OnClickListener()
{
 public void onClick(DialogInterface dialog, int which)
 {
 if(typeSelection == SELECTION_ZONE_OBJECTIF)
 {
 Log.d("IHMEcranPrincipal", "Zone objectif validée : " +
 strZoneObjectif);
 zoneObjectif = Integer.parseInt(strZoneObjectif.substring(

```

```

 strZoneObjectif.length() - 1));
 }
 else if (typeSelection == SELECTION_ZONE_ROBOT)
 {
 Log.d("IHMEcranPrincipal", "Zone robot validée : " +
 strZoneRobot);
 zoneRobot = Integer.parseInt(strZoneRobot.substring(
 strZoneRobot.length() - 1));
 // Puis sélectionner l'objectif
 selectionnerZone(SELECTION_ZONE_OBJECTIF, 0, 0);
 }

 if (idZonePrecedente == -1)
 idZonePrecedente = 0;
 }
 });

 selectionZone.setNegativeButton("Annuler", new DialogInterface.
 OnClickListener()
 {
 public void onClick(DialogInterface dialog, int which)
 {
 if (typeSelection == SELECTION_ZONE_OBJECTIF)
 {
 strZoneObjectif = "";
 zoneObjectif = 0;
 }
 else if (typeSelection == SELECTION_ZONE_ROBOT)
 {
 strZoneRobot = "";
 zoneRobot = 0;
 // Puis sélectionner l'objectif
 selectionnerZone(SELECTION_ZONE_OBJECTIF, 0, 0);
 }
 }
 });

 selectionZone.show();
}

```

#### 9.1.2.44 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonneesRecues (String nomAppareilSource, String donnees) [private]

Méthode `traiterDonneesRecues()` permettant la traitement des données bluetooth reçues en fonction de l'appareil source de ces données.

##### Paramètres

|                            |                                               |
|----------------------------|-----------------------------------------------|
| <i>nom-Appareil-Source</i> | étant l'appareil duquel on reçoit les données |
| <i>donnees</i>             | étant les données reçues                      |

Références `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.NOM_PERIPHERIQUE_BLUETOOTH_LANCEUR`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.NOM_PERIPHERIQUE_BLUETOOTH_TABLE`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonneesRecuesLanceur()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonneesRecuesTable()`.

```

{
 Log.d("IHMEcranPrincipal", "traiterDonneesRecues()");

 switch (nomAppareilSource)
 {

```

```

 case NOM_PERIPHERIQUE_BLUETOOTH_LANCEUR:
 traiterDonneesRecuesLanceur (donnees);
 break;
 case NOM_PERIPHERIQUE_BLUETOOTH_TABLE:
 traiterDonneesRecuesTable (donnees);
 break;
 default:
 break;
 }
}

```

#### 9.1.2.45 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal. traiterDonneesRecuesLanceur ( String *donnees* ) [private]

Méthode [traiterDonneesRecuesLanceur\(\)](#) permettant le traitement des données reçues par le lanceur.

##### Paramètres

|                |                          |
|----------------|--------------------------|
| <i>donnees</i> | étant les données reçues |
|----------------|--------------------------|

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterErreurRecueLanceur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME\\_ENTETE](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonneesRecues\(\)](#).

```

{
 Log.d("IHMEcranPrincipal", "traiterDonneesRecuesLanceur()");

 List<String> donneeRecue = new ArrayList<String>(Arrays.asList(donnees.
split(":")));

 switch (donneeRecue.get(0))
 {
 case TRAME_ENTETE:
 if(donneeRecue.get(1).contains("ERREUR"))
 traiterErreurRecueLanceur(donneeRecue.get(1));
 break;
 default:
 Log.e("IHMEcranPrincipal", "traiterDonneesRecuesLanceur() :
trame non reconnue !");
 }
}

```

#### 9.1.2.46 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonneesRecues- Table ( String *donnees* ) [private]

Méthode [traiterDonneesRecuesTable\(\)](#) permettant le traitement des données reçues par la table.

##### Paramètres

|                |                          |
|----------------|--------------------------|
| <i>donnees</i> | étant les données reçues |
|----------------|--------------------------|

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.calculerReussiteSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetooth-Ecran\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.incrementerBallesJouees\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME\\_ENTETE](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonneesRecues\(\)](#).

```
{
 Log.d("IHMEcranPrincipal", "traiterDonneesRecuesTable()");

 List<String> donneeRecue = new ArrayList<String>(Arrays.asList(donnees.
split(":")));

 switch (donneeRecue.get(0))
 {
 case TRAME_ENTETE:
 // Transférer la trame reçue à l'écran
 envoyerTramePeripheriqueBluetoothEcran(donnees);

 switch(donneeRecue.get(1))
 {
 case "IMPACT":
 if(Integer.parseInt(donneeRecue.get(2)) != 0) // Si
l'impacte a eu lieu sur une des 9 zones et non sur le côté du lanceur
 calculerReussiteSeance(Integer.parseInt(donneeRecue
.get(2)));
 else
 incrementerBallesJouees();
 break;
 case "FAUTE":
 calculerReussiteSeance(-1);
 }
 break;
 default:
 Log.e("IHMEcranPrincipal", "traiterDonneesRecuesTable() : trame
non reconnue !");
 }
}
```

#### 9.1.2.47 void com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterErreurRecue- Lanceur ( String *erreur* ) [private]

Méthode [traiterErreurRecueLanceur\(\)](#) permettant la gestion des erreurs en provenance du lanceur.

##### Paramètres

|               |                             |
|---------------|-----------------------------|
| <i>erreur</i> | étant le code d'erreur reçu |
|---------------|-----------------------------|

Références [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.pauserSeance\(\)](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonneesRecues-Lanceur\(\)](#).

```
{
 Log.d("IHMEcranPrincipal", "traiterErreurRecueLanceur() code d'erreur :
" + erreur);

 pauserSeance();
}
```

#### 9.1.2.48 boolean com.ttpa.iris.ttpamobile.IHMEcranPrincipal.- verifierConnexionAppareilsBluetoothRequis ( ) [private]

Méthode [verifierConnexionAppareilsBluetoothRequis\(\)](#) vérifiant si les appareils - Bluetooth nécessaires au bon déroulement d'une séance sont détectés et connectés,

puis modifie l'IHM en conséquent.

#### Renvoie

booléen tous les appareils sont connectés ou non

Références [com.tpa.iris.ttpamobile.IHMEcranPrincipal.actualiserIHMAppareilsBluetooth\(\)](#), [com.tpa.iris.ttpamobile.PeripheriqueBluetooth.estConnecte\(\)](#), [com.tpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothInsuffisant\(\)](#), [com.tpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothOperationnel\(\)](#), [com.tpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothSuffisant\(\)](#), [com.tpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothEcran](#), [com.tpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothLanceur](#), et [com.tpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothTable](#).

Référéncé par [com.tpa.iris.ttpamobile.IHMEcranPrincipal.onClick\(\)](#).

```
{
 Log.d("IHMEcranPrincipal", "verifierConnexionAppareilsBluetoothRequis()");

 boolean ecranEstDetecte = (peripheriqueBluetoothEcran!=null);
 boolean lanceurEstDetecte = (peripheriqueBluetoothLanceur != null);
 boolean tableEstDetectee = (peripheriqueBluetoothTable != null);
 boolean ecranEstConnecte = false;
 boolean lanceurEstConnecte = false;
 boolean tableEstConnectee = false;

 if(ecranEstDetecte)
 ecranEstConnecte = peripheriqueBluetoothEcran.estConnecte();

 if(lanceurEstDetecte)
 lanceurEstConnecte = peripheriqueBluetoothLanceur.estConnecte();

 if(tableEstDetectee)
 tableEstConnectee = peripheriqueBluetoothTable.estConnecte();

 actualiserIHMAppareilsBluetooth(tableEstConnectee, lanceurEstConnecte,
 ecranEstConnecte);

 boolean appareilsTousConnectes = ecranEstConnecte && lanceurEstConnecte
 && tableEstConnectee;
 boolean aucunAppareilConnecte = !ecranEstConnecte && !
 lanceurEstConnecte && !tableEstConnectee;

 if(appareilsTousConnectes) // Si tous les appareils sont connectés
 {
 modifierIHMBluetoothOperationnel();
 return true;
 }
 else if(aucunAppareilConnecte) // Si aucun appareil n'est connecté
 {
 modifierIHMBluetoothInsuffisant();
 return false;
 }
 else
 {
 modifierIHMBluetoothSuffisant(); // Si au moins un appareil est
 connecté
 }

 return false;
}
```

#### 9.1.3 Documentation des données membres

**9.1.3.1 BluetoothAdapter** `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.adaptateur-Bluetooth = null` [private]

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrageBluetooth()`.

**9.1.3.2 int** `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ballesJouees` [private]

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.calculerReussiteSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameRepriseSeancePeripheriqueBluetoothLanceur()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.incrementerBallesJouees()`.

**9.1.3.3 int** `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ballesReussies` [private]

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.calculerReussiteSeance()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance()`.

**9.1.3.4 SeekBar** `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgression-EffetBalles` [private]

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affecterMembresIHM()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierValeursParametresIHM()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBarresProgressionIHMUtilisables()`.

**9.1.3.5 SeekBar** `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgression-FrequenceBalles` [private]

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affecterMembresIHM()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierValeursParametresIHM()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBarresProgressionIHMUtilisables()`.

**9.1.3.6 SeekBar** `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgression-NombreBalles` [private]

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affecterMembresIHM()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierValeursParametresIHM()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBarresProgressionIHMUtilisables()`.

**9.1.3.7 SeekBar `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgression-PuissanceBalles` [private]**

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affectedMembresIHM()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierValeursParametresIHM()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBarresProgressionIHMUtilisables()`.

**9.1.3.8 SeekBar `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.barreProgression-RotationLanceur` [private]**

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affectedMembresIHM()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierValeursParametresIHM()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBarresProgressionIHMUtilisables()`.

**9.1.3.9 ImageView `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonAction-Seance` [private]**

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affectedMembresIHM()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.incrementerBallesJouees()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothInsuffisant()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothOperationnel()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothSuffisant()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.pauserSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBoutonsIHMClickables()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.reprendreSeance()`.

**9.1.3.10 ImageButton `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonAppliquer-NomJoueur` [private]**

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affectedMembresIHM()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBoutonsIHMClickables()`.

**9.1.3.11 ImageView `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonArreter-Seance` [private]**

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affectedMembresIHM()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBoutonsIHMClickables()`.

### 9.1.3.12 ImageButton com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonBluetooth [private]

Ressources IHM

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affectedMembresIHM\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothInsuffisant\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothOperationnel\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothSuffisant\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBoutonsIHMClickables\(\)](#).

### 9.1.3.13 ImageButton com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonHistorique [private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affectedMembresIHM\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBoutonsIHMClickables\(\)](#).

### 9.1.3.14 ImageButton com.ttpa.iris.ttpamobile.IHMEcranPrincipal.boutonParametres [private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affectedMembresIHM\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothInsuffisant\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothOperationnel\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothSuffisant\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.rendreBoutonsIHMClickables\(\)](#).

### 9.1.3.15 final int com.ttpa.iris.ttpamobile.IHMEcranPrincipal.COULEUR\_BOUTON\_GRISE = Color.parseColor("#c0c5c6") [static, private]

### 9.1.3.16 final int com.ttpa.iris.ttpamobile.IHMEcranPrincipal.COULEUR\_BOUTON\_ORANGE = Color.parseColor("#f7bb31") [static, private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothSuffisant\(\)](#).

### 9.1.3.17 final int com.ttpa.iris.ttpamobile.IHMEcranPrincipal.COULEUR\_BOUTON\_ROUGE = Color.parseColor("#ee5e5e") [static, private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierIHMBluetoothInsuffisant\(\)](#).

### 9.1.3.18 final int com.ttpa.iris.ttpamobile.IHMEcranPrincipal.COULEUR\_BOUTON\_VERT = Color.parseColor("#5e8d7b") [static, private]

Définitions



Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierHMBluetoothOperationnel\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierHMBluetoothSuffisant\(\)](#).

**9.1.3.19** `Set<BluetoothDevice> com.ttpa.iris.ttpamobile.IHMEcranPrincipal.devices`  
[private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriquesBluetooth\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrageBluetooth\(\)](#).

**9.1.3.20** `final int com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ETAT_SEANCE_ARR-  
ETEE = 0` [static, private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actionnerSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs\(\)](#).

**9.1.3.21** `final int com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ETAT_SEANCE_DEM-  
ARREE = 1` [static, private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actionnerSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.calculerReussiteSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.incrementerBallesJouees\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.reprendreSeance\(\)](#).

**9.1.3.22** `final int com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ETAT_SEANCE_PAU-  
SE = 2` [static, private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actionnerSeance\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.pauserSeance\(\)](#).

**9.1.3.23** `int com.ttpa.iris.ttpamobile.IHMEcranPrincipal.etatSeance =  
ETAT_SEANCE_ARRETEE` [private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actionnerSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.calculerReussiteSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.incrementerBallesJouees\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.pauserSeance\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.reprendreSeance\(\)](#).

**9.1.3.24** `final Handler com.ttpa.iris.ttpamobile.IHMEcranPrincipal.handler`  
[private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothEcran\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothLanceur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothTable\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriquesBluetooth\(\)](#).

9.1.3.25 `int com.ttpa.iris.ttpamobile.IHMEcranPrincipal.idZonePrecedente = -1`  
[package]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actionnerSeance\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.selectionnerZone\(\)](#).

9.1.3.26 `int com.ttpa.iris.ttpamobile.IHMEcranPrincipal.idZonePrecedenteObjectif = -1` [package]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.selectionnerZone\(\)](#).

9.1.3.27 `int com.ttpa.iris.ttpamobile.IHMEcranPrincipal.idZonePrecedenteRobot = -1` [package]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.selectionnerZone\(\)](#).

9.1.3.28 `final String com.ttpa.iris.ttpamobile.IHMEcranPrincipal.NOM_PERIPHERIQUE_BLUETOOTH_ECRAN = "TPA-Ecran"` [static, private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriquesBluetooth\(\)](#).

9.1.3.29 `final String com.ttpa.iris.ttpamobile.IHMEcranPrincipal.NOM_PERIPHERIQUE_BLUETOOTH_LANCEUR = "TPA-Lanceur"` [static, private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriquesBluetooth\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonneesRecues\(\)](#).

9.1.3.30 `final String com.ttpa.iris.ttpamobile.IHMEcranPrincipal.NOM_PERIPHERIQUE_BLUETOOTH_TABLE = "TPA-Table"` [static, private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriquesBluetooth\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonneesRecues\(\)](#).

9.1.3.31 `String com.ttpa.iris.ttpamobile.IHMEcranPrincipal.nomJoueur` [private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ajouterJoueur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick\(\)](#).

9.1.3.32 `ParametreSeance com.ttpa.iris.ttpamobile.IHMEcranPrincipal.parametresActuels = new ParametreSeance()`

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameConnexionPeripheriqueBluetoothEcran\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.selectionnerZone\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonneesRecues\(\)](#).

[iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeancePeripheriqueBluetoothLanceur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameParametrageSeancePeripheriqueBluetoothEcran\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameRepriseSeancePeripheriqueBluetoothLanceur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick\(\)](#).

**9.1.3.33 PeripheriqueBluetooth com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothEcran = null**  
[package]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothEcran\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.deconnexionPeripheriquesBluetooth\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetoothEcran\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.verifierConnexionAppareilsBluetoothRequis\(\)](#).

**9.1.3.34 PeripheriqueBluetooth com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothLanceur = null**  
[package]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothLanceur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.deconnexionPeripheriquesBluetooth\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameArretPeripheriqueBluetoothLanceur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeancePeripheriqueBluetoothLanceur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetoothLanceur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameRepriseSeancePeripheriqueBluetoothLanceur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.verifierConnexionAppareilsBluetoothRequis\(\)](#).

**9.1.3.35 PeripheriqueBluetooth com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriqueBluetoothTable = null**  
[package]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothTable\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.deconnexionPeripheriquesBluetooth\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameArretPeripheriqueBluetoothTable\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.verifierConnexionAppareilsBluetoothRequis\(\)](#).

**9.1.3.36 List<PeripheriqueBluetooth> com.ttpa.iris.ttpamobile.IHMEcranPrincipal.peripheriques** [package]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriquesBluetooth\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrageBluetooth\(\)](#).

**9.1.3.37 final int com.ttpa.iris.ttpamobile.IHMEcranPrincipal.REQUEST\_CODE\_ENABLE\_BLUETOOTH = 0** [static, private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrageBluetooth\(\)](#).

**9.1.3.38** `Seance` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.seanceEnCours`  
[private]

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.calculerReussiteSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.incrementsBallesJouees()`.

**9.1.3.39** `final int` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.-SELECTION_ZONE_OBJECTIF = 0` [static, private]

Gestion des zones

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.selectionnerZone()`.

**9.1.3.40** `final int` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.SELECTION_ZONE_ROBOT = 1` [static, private]

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actionnerSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.selectionnerZone()`.

**9.1.3.41** `ServeurBDD` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.serveurBDD`  
[private]

Attributs de la classe

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ajouterJoueur()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionBaseDeDonnees()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance()`.

**9.1.3.42** `Spinner` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.spinnerListeJoueurs`  
[private]

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affecterMembresIHM()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs()`.

**9.1.3.43** `String` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.strZoneObjectif = new String("ZONE 0")` [package]

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.selectionnerZone()`.

**9.1.3.44** `String` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.strZoneRobot = new String("ZONE 0")` [package]

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.selectionnerZone()`.

9.1.3.45 `TextView` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeurEffet-Balles` [private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affecterMembresIHM\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierValeursParametresIHM\(\)](#).

9.1.3.46 `TextView` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeur-FrequenceBalles` [private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affecterMembresIHM\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierValeursParametresIHM\(\)](#).

9.1.3.47 `TextView` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeurNombre-Balles` [private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affecterMembresIHM\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierValeursParametresIHM\(\)](#).

9.1.3.48 `TextView` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeur-PuissanceBalles` [private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affecterMembresIHM\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierValeursParametresIHM\(\)](#).

9.1.3.49 `TextView` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.texteValeurRotation-Lanceur` [private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affecterMembresIHM\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.modifierValeursParametresIHM\(\)](#).

9.1.3.50 `final String` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TR-AME_ECRAN_DEBUT_SEANCE = ":START"` [static, private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeance-PeripheriqueBluetoothEcran\(\)](#).

9.1.3.51 `final String` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TR-AME_ECRAN_FIN_SEANCE = ":FINSEANCE"` [static, private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameArret-PeripheriqueBluetoothEcran\(\)](#).

9.1.3.52 `final String` `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TR-AME_ECRAN_PAUSE_SEANCE = ":PAUSE"` [static, private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.pauserSeance\(\)](#).

9.1.3.53 `final String com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRA-  
ME_ECRAN_REPRISE_SEANCE = ":RESUME" [static,  
private]`

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.reprendreSeance()`.

9.1.3.54 `final String com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME_ENTETE =  
"$TTPA" [static, private]`

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameConnexion-  
PeripheriqueBluetoothEcran()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyer-  
TrameDebutSeancePeripheriqueBluetoothLanceur()`, `com.ttpa.iris.ttpamobile.IHM-  
EcranPrincipal.envoyerTrameParametrageSeancePeripheriqueBluetoothEcran()`, `com.-  
ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameRepriseSeancePeripherique-  
BluetoothLanceur()`, `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonnees-  
RecuesLanceur()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.traiterDonnees-  
RecuesTable()`.

9.1.3.55 `final String com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME_FIN =  
"\r\n" [static, private]`

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeance-  
PeripheriqueBluetoothLanceur()`, et `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.-  
envoyerTrameRepriseSeancePeripheriqueBluetoothLanceur()`.

9.1.3.56 `final String com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME_L-  
ANCEUR_ARRET_SEANCE = ":STOP:" + TRAME_FIN [static,  
private]`

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameArret-  
PeripheriqueBluetoothLanceur()`.

9.1.3.57 `final String com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME_LA-  
NCEUR_PAUSE_SEANCE = ":PAUSE:" + TRAME_FIN [static,  
private]`

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.pauserSeance()`.

9.1.3.58 `final String com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRA-  
ME_LANCEUR_PING = ":PING:" + TRAME_FIN [static,  
private]`

9.1.3.59 `final String com.ttpa.iris.ttpamobile.IHMEcranPrincipal.TRAME_LAN-  
CEUR_REPRISE_SEANCE = ":RESUME:" + TRAME_FIN [static,  
private]`

Référencé par `com.ttpa.iris.ttpamobile.IHMEcranPrincipal.reprendreSeance()`.

9.1.3.60 final String com.ttpa.iris.ttpamobile.IHMEcranPrincipal.T-RAME\_TABLE\_ARRET\_SEANCE = ":RESET" [static, private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameArret-PeripheriqueBluetoothTable\(\)](#).

9.1.3.61 ImageView com.ttpa.iris.ttpamobile.IHMEcranPrincipal.voyantEcran [private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actualiserIHMAppeareils-Bluetooth\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affecterMembresIHM\(\)](#).

9.1.3.62 ImageView com.ttpa.iris.ttpamobile.IHMEcranPrincipal.voyantLanceur [private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actualiserIHMAppeareils-Bluetooth\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affecterMembresIHM\(\)](#).

9.1.3.63 ImageView com.ttpa.iris.ttpamobile.IHMEcranPrincipal.voyantTable [private]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.actualiserIHMAppeareils-Bluetooth\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.affecterMembresIHM\(\)](#).

9.1.3.64 int com.ttpa.iris.ttpamobile.IHMEcranPrincipal.zoneObjectif = 0 [package]

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameParametrageSeancePeripherique-BluetoothEcran\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.selectionnerZone\(\)](#).

9.1.3.65 int com.ttpa.iris.ttpamobile.IHMEcranPrincipal.zoneRobot = 0 [package]

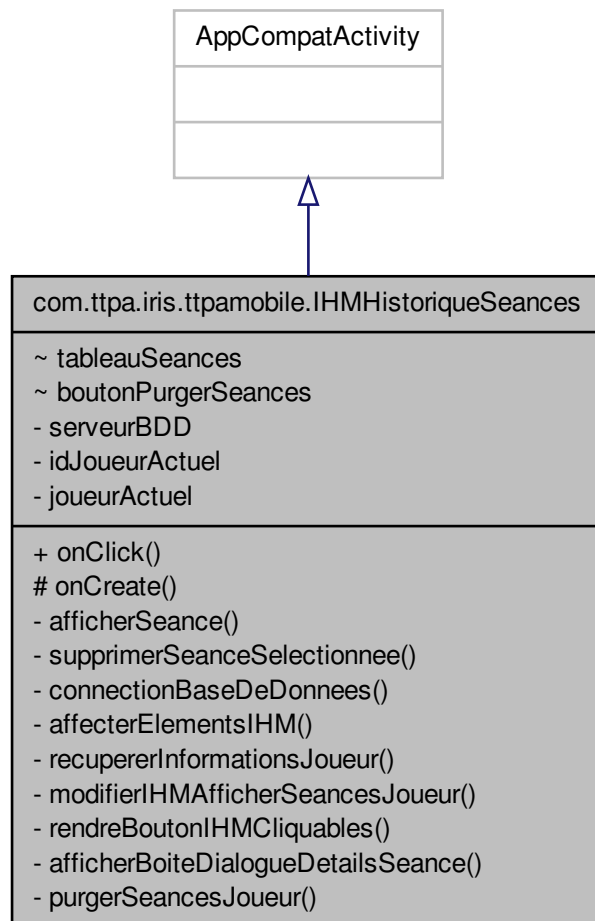
Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameParametrageSeancePeripherique-BluetoothEcran\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.selectionnerZone\(\)](#).

La documentation de cette classe a été générée à partir du fichier suivant :

– [IHMEcranPrincipal.java](#)

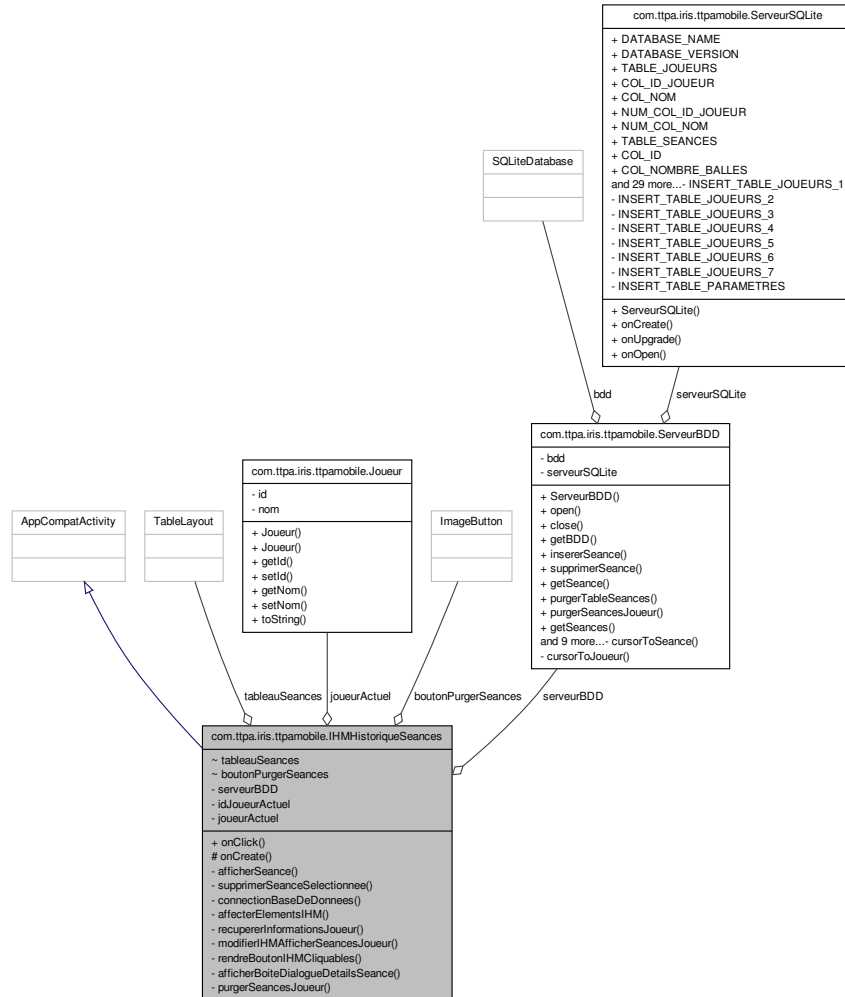
## 9.2 Référence de la classe com.ttpa.iris.ttpamobile.IHMHistoriqueSeances

Graphe d'héritage de com.ttpa.iris.ttpamobile.IHMHistoriqueSeances :





Grphe de collaboration de com.ttpa.iris.ttpamobile.IHMHistoriqueSeances :



### Fonctions membres publiques

- void `onClick` (View element)

### Fonctions membres protégées

- void `onCreate` (Bundle savedInstanceState)

### Attributs de paquetage

- TableLayout `tableauSeances`

- ImageButton [boutonPurgerSeances](#)

### Fonctions membres privées

- void [afficherSeance](#) ([Seance](#) seance)
- void [supprimerSeanceSelectionnee](#) (int idSeanceElementSelectionne, int index-LigneTableau)
- void [connectionBaseDeDonnees](#) ()
- void [affecterElementsIHM](#) ()
- void [recupererInformationsJoueur](#) ()
- void [modifierIHMAfficherSeancesJoueur](#) ()
- void [rendreBoutonIHMClickables](#) ()
- void [afficherBoiteDialogueDetailsSeance](#) (int idSeanceSelectionnee)
- void [purgerSeancesJoueur](#) ()

### Attributs privés

- [ServeurBDD](#) [serveurBDD](#)
- int [idJoueurActuel](#)
- [Joueur](#) [joueurActuel](#)

#### 9.2.1 Description détaillée

Created by smaniotto on 12/04/18. Classe [IHMHistoriqueSeances](#) définissant le comportement du layout 'ecran\_historique\_seances'.

#### 9.2.2 Documentation des fonctions membres

##### 9.2.2.1 void com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.affecterElementsIHM ( ) [private]

Méthode [affecterElementsIHM\(\)](#) permettant d'affecter les éléments graphiques actuels de l'IHM.

Références [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.boutonPurgerSeances](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.tableauSeances](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.onCreate\(\)](#).

```
{
 Log.d("IHMHistoriqueSeances", "affecterElementsIHM()");

 // Tableau des séances
 tableauSeances = (TableLayout) findViewById(R.id.tableauSeances);
 // Bouton de purge des séances du joueur
 boutonPurgerSeances = (ImageButton) findViewById(R.id.
 boutonPurgerSeances);
}
```

### 9.2.2.2 void com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficher-BoiteDialogueDetailsSeance ( int idSeanceSelectionnee ) [private]

Méthode [afficherBoiteDialogueDetailsSeance\(\)](#) affichant une boîte de dialogue avec les informations détaillées de la séance sélectionnée.

#### Paramètres

|                              |                                    |
|------------------------------|------------------------------------|
| <i>idSeance-Selectionnee</i> | étant l'id de la séance à afficher |
|------------------------------|------------------------------------|

Références [com.ttpa.iris.ttpamobile.ServeurBDD.getSeance\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getZoneObjectif\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getZoneRobot\(\)](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.onClick\(\)](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.serveurBDD](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.onClick\(\)](#).

```
{
 Log.d("IHMHistoriqueSeances", "afficherBoiteDialogueDetailsSeance()");

 // Retrouver l'objet Seance correspondant grâce à la base de données
 Seance seanceSelectionnee = serveurBDD.getSeance(idSeanceSelectionnee);

 // Afficher la boîte de dialogue
 AlertDialog.Builder detailsSeance = new AlertDialog.Builder(this);
 LayoutInflater factory = LayoutInflater.from(this);
 final View detailsSeanceView = factory.inflate(R.layout.details_seance,
 null);
 detailsSeance.setView(detailsSeanceView);

 detailsSeance.setTitle("Informations complémentaires de la séance");

 // Affecter les informations de la séance aux champs correspondants
 TextView texteValeurZoneRobot = (TextView) detailsSeanceView.
 findViewById(R.id.texteValeurZoneRobot);
 TextView texteValeurZoneObjectif = (TextView) detailsSeanceView.
 findViewById(R.id.texteValeurZoneObjectif);

 switch(seanceSelectionnee.getZoneRobot())
 {
 case 0:
 case -1:
 texteValeurZoneRobot.setText("Aucune");
 break;
 default:
 texteValeurZoneRobot.setText("ZONE " + seanceSelectionnee.
 getZoneRobot());
 break;
 }

 switch(seanceSelectionnee.getZoneObjectif())
 {
 case 0:
 case -1:
 texteValeurZoneObjectif.setText("Aucune");
 break;
 default:
 texteValeurZoneObjectif.setText("ZONE " + seanceSelectionnee.
 getZoneObjectif());
 break;
 }

 detailsSeance.setNegativeButton("Retour", new DialogInterface.
 OnClickListener()
 {
```

```

 public void onClick(DialogInterface dialog, int which)
 {

 }
 });

 detailsSeance.show();
}

```

### 9.2.2.3 void com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherSeance ( Seance seance ) [private]

Méthode [afficherSeance\(\)](#) permettant d'afficher dans un tableau une séance.

#### Paramètres

|               |                            |
|---------------|----------------------------|
| <i>seance</i> | étant la séance à afficher |
|---------------|----------------------------|

Références [com.ttpa.iris.ttpamobile.Seance.getDateDebut\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getEffet\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getFrequence\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getId\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getNombreBalles\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getTauxReussite\(\)](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.tableauSeances](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.modifierIHMAfficherSeancesJoueur\(\)](#).

```

{
 Log.d("IHMHistoriqueSeances", "afficherSeance()");

 TableLayout tableauSeances = (TableLayout) findViewById(R.id.
tableauSeances);

 // Déclaration des paramètres visuels des champs
 TableRow.LayoutParams lpLigne = new TableRow.LayoutParams(TableRow.
LayoutParams.WRAP_CONTENT);
 TableRow.LayoutParams lpChamp = new TableRow.LayoutParams(TableRow.
LayoutParams.MATCH_PARENT, TableRow.LayoutParams.MATCH_PARENT, 1f);

 // Déclaration et paramétrage de la ligne accueillant la séance
 TableRow ligne= new TableRow(this);
 ligne.setLayoutParams(lpLigne);
 float taille_police_ecriture = (float) Integer.parseInt(getString(R.
string.taille_police_historique));

 // Déclaration et paramétrage du champ ID
 TextView texteSeanceId = new TextView(this);
 texteSeanceId.setLayoutParams(lpChamp);
 texteSeanceId.setGravity(Gravity.CENTER);
 texteSeanceId.setVisibility(View.GONE);
 texteSeanceId.setText(Integer.toString(seance.getId()));

 // Déclaration et paramétrage du champ Date
 TextView texteSeanceDate = new TextView(this);
 texteSeanceDate.setLayoutParams(lpChamp);
 texteSeanceDate.setGravity(Gravity.CENTER);
 texteSeanceDate.setTextSize(taille_police_ecriture);
 texteSeanceDate.setText(seance.getDateDebut());

 // Déclaration et paramétrage du champ Fréquence
 TextView texteSeanceFrequence = new TextView(this);
 texteSeanceFrequence.setLayoutParams(lpChamp);
 texteSeanceFrequence.setGravity(Gravity.CENTER);
 texteSeanceFrequence.setTextSize(taille_police_ecriture);
 texteSeanceFrequence.setText(Integer.toString(seance.getFrequence()) +

```

```

" balles/min");

// Déclaration et paramétrage du champ Nb Balles
TextView texteSeanceNombreBalles = new TextView(this);
texteSeanceNombreBalles.setLayoutParams(lpChamp);
texteSeanceNombreBalles.setGravity(Gravity.CENTER);
texteSeanceNombreBalles.setTextSize(taille_police_ecriture);
texteSeanceNombreBalles.setText(Integer.toString(seance.getNombreBalles()));

// Déclaration et paramétrage du champ Effet
TextView texteSeanceEffet = new TextView(this);
texteSeanceEffet.setLayoutParams(lpChamp);
texteSeanceEffet.setGravity(Gravity.CENTER);
texteSeanceEffet.setTextSize(taille_police_ecriture);
texteSeanceEffet.setText(seance.getEffet());

// Déclaration et paramétrage du champ Taux Réussite
TextView texteSeanceTauxReussite = new TextView(this);
texteSeanceTauxReussite.setLayoutParams(lpChamp);
texteSeanceTauxReussite.setGravity(Gravity.CENTER);
texteSeanceTauxReussite.setTextSize(taille_police_ecriture);
texteSeanceTauxReussite.setText(Float.toString(seance.getTauxReussite())
+ "%");

// Déclaration et paramétrage du champ Action
ImageButton boutonDetailsSeance = new ImageButton(this);
boutonDetailsSeance.setLayoutParams(lpChamp);

boutonDetailsSeance.setBackgroundColor(Color.argb(0, 0, 0, 0)); //
Rendre le fond transparent
boutonDetailsSeance.setImageResource(R.drawable.ic_loupe_afficher);
boutonDetailsSeance.setOnClickListener(this); // Rendre le bouton
clicquable

// Ajouter les champs à la ligne
ligne.addView(texteSeanceId);
ligne.addView(texteSeanceDate);
ligne.addView(texteSeanceFrequence);
ligne.addView(texteSeanceNombreBalles);
ligne.addView(texteSeanceEffet);
ligne.addView(texteSeanceTauxReussite);
ligne.addView(boutonDetailsSeance);

// Ajouter la ligne au tableau des séances à l'index 1 : après l'entête
du tableau
tableauSeances.addView(ligne, 1);
}

```

#### 9.2.2.4 void `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.connectionBaseDeDonnees()` [private]

Méthode `connectionBaseDeDonnees()` permettant de se connecter à la base de données.

Références `com.ttpa.iris.ttpamobile.ServeurBDD.open()`, et `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.serveurBDD`.

Référéncé par `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.onCreate()`.

```

{
 Log.d("IHMHistoriqueSeances", "connectionBaseDeDonnees()");

 serveurBDD = new ServeurBDD(this);
 serveurBDD.open();
}

```

### 9.2.2.5 void com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.modifierIHMAfficherSeancesJoueur( ) [private]

Méthode `modifierIHMAfficherSeancesJoueur()` affichant les séances du joueur actuel dans un tableau de l'IHM.

Références `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherSeance()`, `com.ttpa.iris.ttpamobile.Joueur.getId()`, `com.ttpa.iris.ttpamobile.Joueur.getNom()`, `com.ttpa.iris.ttpamobile.ServeurBDD.getSeances()`, `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.joueurActuel`, et `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.serveurBDD`.

Référéncé par `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.onCreate()`.

```
{
 Log.d("IHMHistoriqueSeances", "modifierIHMAfficherSeancesJoueur()");

 TextView texteHistoriqueSeances = (TextView) findViewById(R.id.
texteHistoriqueSeances);
 texteHistoriqueSeances.setText("Historique des séances de " +
joueurActuel.getNom() + " :");

 // Récupérer la liste des séances existantes du joueur
 List<Seance> seances = serveurBDD.getSeances(joueurActuel.getId());
 Log.d("IHMHistoriqueSeances", "modifierIHMAfficherSeancesJoueur()
Nombre de séances du joueur: " + seances.size());
 // Pour chaque séance existante
 for(int i = 0; i < seances.size(); i++)
 {
 Log.d("IHMHistoriqueSeances", "modifierIHMAfficherSeancesJoueur()
Séance n° " + seances.get(i).getId() + ": \n" + seances.get(i).toString());

 // Ajouter la séance dans le tableau de séances
 afficherSeance(seances.get(i));
 }
}
```

### 9.2.2.6 void com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.onClick ( View element )

Méthode `onClick()` permettant la gestion du click sur un bouton.

#### Paramètres

|                |                                                   |
|----------------|---------------------------------------------------|
| <i>element</i> | étant l'élément sur lequel l'utilisateur a cliqué |
|----------------|---------------------------------------------------|

Références `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherBoiteDialogueDetailsSeance()`, `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.boutonPurgerSeances`, `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.purgerSeancesJoueur()`, et `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.tableauSeances`.

Référéncé par `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherBoiteDialogueDetailsSeance()`, et `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.purgerSeancesJoueur()`.

```
{
 Log.d("IHMHistoriqueSeances", "onClick()");

 Seance seance = null;

 if(element == boutonPurgerSeances)
 {
```

```

 // Purger les séances déjà existantes du joueur
 purgerSeancesJoueur();
 }
 else // l'élément est un bouton du champ action
 {
 // Retrouver l'ID de la séance sélectionnée
 for (int i = 1; i < tableauSeances.getChildCount(); ++i) {

 TableRow ligne = (TableRow) tableauSeances.getChildAt(i);

 if (ligne == element.getParent()) {
 TextView texteIdSeance = (TextView) ligne.getChildAt(0); //
 0 : Position du champ ID de la ligne
 int idSeanceElementSelectionne = Integer.parseInt(
 texteIdSeance.getText().toString()); // ID de la séance à supprimer

 // Afficher la boîte de dialogue de la séance sélectionnée
 afficherBoiteDialogueDetailsSeance(
 idSeanceElementSelectionne);
 }
 }
 }
}

```

### 9.2.2.7 void com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.onCreate ( Bundle savedInstanceState ) [protected]

Méthode onCreate appelée au démarrage de l'activité permettant l'initialisation des composants.

#### Paramètres

|                                 |  |
|---------------------------------|--|
| <i>saved-<br/>InstanceState</i> |  |
|---------------------------------|--|

Références [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.affecterElementsIHM\(\)](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.connectionBaseDeDonnees\(\)](#), [com.-ttpa.iris.ttpamobile.IHMHistoriqueSeances.modifierIHMAfficherSeancesJoueur\(\)](#), [com.-ttpa.iris.ttpamobile.IHMHistoriqueSeances.recupererInformationsJoueur\(\)](#), et [com.-ttpa.iris.ttpamobile.IHMHistoriqueSeances.rendreBoutonIHMClickables\(\)](#).

```

{
 super.onCreate(savedInstanceState);
 setContentView(R.layout.ecran_historique_seances);

 // Se connecter à la base de données
 connectionBaseDeDonnees();

 // Affecter les éléments graphiques de l'IHM
 affecterElementsIHM();

 // Récupérer les informations du joueur actuel
 recupererInformationsJoueur();

 // Modifier l'IHM pour afficher le joueur
 modifierIHMAfficherSeancesJoueur();

 // Permettre à tous les boutons de suppression de séance d'être
 cliquables
 rendreBoutonIHMClickables();
}

```

### 9.2.2.8 void com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.purgerSeances-Joueur( ) [private]

Méthode `purgerSeancesJoueur()` affichant une boîte de dialogue qui va supprimer les séances du joueur actuel selon la décision prise.

Références `com.ttpa.iris.ttpamobile.Joueur.getNom()`, `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.idJoueurActuel`, `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.joueurActuel`, `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.onClick()`, `com.ttpa.iris.ttpamobile.ServeurBDD.purgerSeancesJoueur()`, `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.serveurBDD`, et `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.tableauSeances`.

Référéncé par `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.onClick()`.

```
{
 Log.d("IHMHistoriqueSeances", "purgerSeancesJoueur()");

 // Afficher la boîte de dialogue
 AlertDialog.Builder detailsSeance = new AlertDialog.Builder(this);

 detailsSeance.setMessage("Vous êtes sur le point de supprimer
 définitivement les séance du joueur " + joueurActuel.getNom() + ".");

 detailsSeance.setPositiveButton("Continuer", new DialogInterface.
 OnClickListener()
 {
 public void onClick(DialogInterface dialog, int which)
 {
 // Purger les séances déjà existantes du joueur
 Toast.makeText(getApplicationContext(), "Purge des séances...",
 Toast.LENGTH_SHORT).show();

 // Purger la table des séances de la base de données
 serveurBDD.purgerSeancesJoueur(idJoueurActuel);

 // Supprimer le contenu du tableau des séances
 int compteur = tableauSeances.getChildCount();
 for (int i = 1; i < compteur; i++) { // On ne supprime pas la
 1ere ligne qui est l'entête
 View child = tableauSeances.getChildAt(i);
 if (child instanceof TableRow) ((ViewGroup) child).
 removeAllViews();
 }
 }
 });

 detailsSeance.setNegativeButton("Annuler", new DialogInterface.
 OnClickListener()
 {
 public void onClick(DialogInterface dialog, int which)
 {
 }
 });

 detailsSeance.show();
}
```

### 9.2.2.9 void com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.recuperer- InformationsJoueur( ) [private]

Méthode `recupererInformationsJoueur()` affectant aux attributs les caractéristiques du joueur actuel, importées depuis la base de données.



Références [com.ttpa.iris.ttpamobile.ServeurBDD.getIdJoueurParametres\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getJoueur\(\)](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.idJoueurActuel](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.joueurActuel](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.serveurBDD](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.onCreate\(\)](#).

```
{
 Log.d("IHMHistoriqueSeances", "recupererInformationsJoueur()");

 idJoueurActuel = serveurBDD.getIdJoueurParametres();
 joueurActuel = serveurBDD.getJoueur(idJoueurActuel);
}
```

#### 9.2.2.10 void com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.rendreBoutonIHM-Cliquables ( ) [private]

Méthode [rendreBoutonIHM-Cliquables\(\)](#) permettant aux boutons de l'IHM d'être cliquables.

Références [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.boutonPurgerSeances](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.tableauSeances](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.onCreate\(\)](#).

```
{
 Log.d("IHMHistoriqueSeances", "rendreBoutonIHM-Cliquables()");

 // Boutons présents dans le tableau d'historique de séances du joueur
 int compteurLignes = tableauSeances.getChildCount();
 for (int i = 0; i < compteurLignes; ++i) {
 TableRow ligne = (TableRow) tableauSeances.getChildAt(i);

 if (ligne.getChildAt(ligne.getChildCount() - 1) instanceof
 ImageButton) {
 View element = ligne.getChildAt(ligne.getChildCount() - 1); //
 // Le bouton de suppression de séance étant le dernier élément de la ligne
 element.setOnClickListener(this);
 }
 }

 // Bouton de purge de séance du joueur
 boutonPurgerSeances.setOnClickListener(this);
}
```

#### 9.2.2.11 void com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.supprimerSeance-Selectionnee ( int idSeanceElementSelectionne, int indexLigneTableau ) [private]

Méthode [supprimerSeanceSelectionnee\(\)](#) permettant la suppression d'une séance précise.

##### Paramètres

|                                     |                                     |
|-------------------------------------|-------------------------------------|
| <i>idSeance-Element-Selectionne</i> | étant l'ID de la séance à supprimer |
| <i>indexLigne-Tableau</i>           | étant son index dans le tableau     |

Références [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.serveurBDD](#), [com.ttpa.iris.ttpamobile.ServeurBDD.supprimerSeance\(\)](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.tableauSeances](#).

```
{
 Log.d("IHMHistoriqueSeances", "supprimerSeanceSelectionnee()");

 // Supprimer la séance sélectionnée de la base de données
 serveurBDD.supprimerSeance(idSeanceElementSelectionnee);

 // Supprimer le contenu de la ligne correspondante du tableau des
 séances
 TableLayout tableauSeances = (TableLayout) findViewById(R.id.
 tableauSeances);
 TableRow ligne = (TableRow) tableauSeances.getChildAt(indexLigneTableau
);
 tableauSeances.removeView(ligne);

 Toast.makeText(getApplicationContext(), "Suppression de la séance " +
 idSeanceElementSelectionnee + "...", Toast.LENGTH_SHORT).show();
}
```

### 9.2.3 Documentation des données membres

#### 9.2.3.1 ImageButton [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.boutonPurgerSeances](#) [package]

Référéncé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.affecterElementsIHM\(\)](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.onClick\(\)](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.rendreBoutonIHMClickables\(\)](#).

#### 9.2.3.2 int [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.idJoueurActuel](#) [private]

Référéncé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.purgerSeancesJoueur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.recupererInformationsJoueur\(\)](#).

#### 9.2.3.3 Joueur [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.joueurActuel](#) [private]

Référéncé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.modifierIHMAfficherSeancesJoueur\(\)](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.purgerSeancesJoueur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.recupererInformationsJoueur\(\)](#).

#### 9.2.3.4 ServeurBDD [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.serveurBDD](#) [private]

Attributs de la classe

Référéncé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherBoiteDialogueDetailsSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.connectionBaseDeDonnees\(\)](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.modifierIHMAfficherSeancesJoueur\(\)](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.purgerSeancesJoueur\(\)](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.recupererInformationsJoueur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.supprimerSeance](#)

[Selectionnee\(\)](#).

#### 9.2.3.5 TableLayout `com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.tableau-Seances` [package]

Éléments graphiques de l'IHM

Référéncé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.affecterElementsIHM\(\)](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.onClick\(\)](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.purgerSeancesJoueur\(\)](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.rendreBoutonIHMClickables\(\)](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.supprimerSeanceSelectionnee\(\)](#).

La documentation de cette classe a été générée à partir du fichier suivant :

- [IHMHistoriqueSeances.java](#)

### 9.3 Référence de la classe com.ttpa.iris.ttpamobile.Joueur

Fonctions membres publiques

- [Joueur \(\)](#)
- [Joueur \(String nom\)](#)
- [int getId \(\)](#)
- [void setId \(int id\)](#)
- [String getNom \(\)](#)
- [void setNom \(String nom\)](#)
- [String toString \(\)](#)

Attributs privés

- [int id](#)
- [String nom](#)

#### 9.3.1 Description détaillée

Created by smaniotto on 10/04/18.

#### 9.3.2 Documentation des constructeurs et destructeur

##### 9.3.2.1 `com.ttpa.iris.ttpamobile.Joueur.Joueur ( )`

```
{
}
```

##### 9.3.2.2 `com.ttpa.iris.ttpamobile.Joueur.Joueur ( String nom )`

Références [com.ttpa.iris.ttpamobile.Joueur.nom](#).

```
{
 this.nom = nom;
}
```

### 9.3.3 Documentation des fonctions membres

#### 9.3.3.1 `int com.ttpa.iris.ttpamobile.Joueur.getId ( )`

Références [com.ttpa.iris.ttpamobile.Joueur.id](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs\(\)](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.modifierIHMAfficherSeancesJoueur\(\)](#).

```
{
 return id;
}
```

#### 9.3.3.2 `String com.ttpa.iris.ttpamobile.Joueur.getNom ( )`

Références [com.ttpa.iris.ttpamobile.Joueur.nom](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.insererJoueur\(\)](#), [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.modifierIHMAfficherSeancesJoueur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.purgerSeancesJoueur\(\)](#).

```
{
 return nom;
}
```

#### 9.3.3.3 `void com.ttpa.iris.ttpamobile.Joueur.setId ( int id )`

Références [com.ttpa.iris.ttpamobile.Joueur.id](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToJoueur\(\)](#).

```
{
 this.id = id;
}
```

#### 9.3.3.4 `void com.ttpa.iris.ttpamobile.Joueur.setNom ( String nom )`

Références [com.ttpa.iris.ttpamobile.Joueur.nom](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToJoueur\(\)](#).

```
{
 this.nom = nom;
}
```

#### 9.3.3.5 `String com.ttpa.iris.ttpamobile.Joueur.toString ( )`

Méthode `toString` permettant la visualisation des caractéristiques d'un joueur.

## Renvoie

String les caractéristiques d'un joueur.

Références [com.ttpa.iris.ttpamobile.Joueur.nom](#).

```
{
 return "Nom : " + nom;
}
```

## 9.3.4 Documentation des données membres

## 9.3.4.1 int com.ttpa.iris.ttpamobile.Joueur.id [private]

Référencé par [com.ttpa.iris.ttpamobile.Joueur.getId\(\)](#), et [com.ttpa.iris.ttpamobile.Joueur.setId\(\)](#).

## 9.3.4.2 String com.ttpa.iris.ttpamobile.Joueur.nom [private]

Référencé par [com.ttpa.iris.ttpamobile.Joueur.getNom\(\)](#), [com.ttpa.iris.ttpamobile.Joueur.Joueur\(\)](#), [com.ttpa.iris.ttpamobile.Joueur.setNom\(\)](#), et [com.ttpa.iris.ttpamobile.Joueur.toString\(\)](#).

La documentation de cette classe a été générée à partir du fichier suivant :

– [Joueur.java](#)

## 9.4 Référence de la classe com.ttpa.iris.ttpamobile.ParametreSeance

## Fonctions membres publiques

- [ParametreSeance \(\)](#)
- [ParametreSeance \(String nomJoueur, String effetBalles, int intensiteEffet, int puissanceBalles, int frequenceBalles, int rotation, int nombreBalles\)](#)
- [String getNomJoueur \(\)](#)
- [void setNomJoueur \(String nomJoueur\)](#)
- [int getFrequenceBalles \(\)](#)
- [void setFrequenceBalles \(int frequenceBalles\)](#)
- [int getNombreBalles \(\)](#)
- [void setNombreBalles \(int nombreBalles\)](#)
- [String getEffetBalles \(\)](#)
- [void setEffetBalles \(String effetBalles\)](#)
- [void setIntensiteEffet \(int intensiteEffet\)](#)
- [int getPuissanceBalles \(\)](#)
- [void setPuissanceBalles \(int puissanceBalles\)](#)
- [int getRotation \(\)](#)
- [void setRotation \(int rotation\)](#)
- [String toString \(\)](#)
- [boolean estValide \(\)](#)
- [final boolean frequenceEstValide \(int frequence\)](#)
- [final boolean nombreBallesEstValide \(int nombreBalles\)](#)
- [final boolean puissanceBallesEstValide \(int puissanceBalles\)](#)
- [final boolean rotationEstValide \(int rotation\)](#)
- [final boolean effetEstValide \(String effet\)](#)

## Attributs publics statiques

- static final int [MIN\\_FREQUENCE\\_PARAMETRE](#) = 1
- static final int [MAX\\_FREQUENCE\\_PARAMETRE](#) = 60
- static final int [MIN\\_NOMBRE\\_BALLES\\_PARAMETRE](#) = 1
- static final int [MAX\\_NOMBRE\\_BALLES\\_PARAMETRE](#) = 50
- static final int [MIN\\_PUISSANCE\\_BALLES\\_PARAMETRE](#) = 1
- static final int [MAX\\_PUISSANCE\\_BALLES\\_PARAMETRE](#) = 10
- static final int [MIN\\_ROTATION\\_PARAMETRE](#) = 0
- static final int [MAX\\_ROTATION\\_PARAMETRE](#) = 180

## Fonctions de paquetage

- final char [getEffet](#) ()
- final String [getEffetCompleet](#) ()
- final int [getIntensiteEffet](#) ()

## Attributs privés

- String [nomJoueur](#)
- String [effetBalles](#)
- int [intensiteEffet](#)
- int [puissanceBalles](#)
- int [frequenceBalles](#)
- int [rotation](#)
- int [nombreBalles](#)

## 9.4.1 Description détaillée

Created by smaniotto on 19/03/18. Classe [ParametreSeance](#) définissant les caractéristiques et le comportement d'un paramètre de séance.

## 9.4.2 Documentation des constructeurs et destructeur

9.4.2.1 [com.ttpa.iris.ttpamobile.ParametreSeance.ParametreSeance](#) ( )

Méthode [ParametreSeance](#) constructeur par défaut de la classe [ParametreSeance](#).

```
{ }
```

9.4.2.2 [com.ttpa.iris.ttpamobile.ParametreSeance.ParametreSeance](#) ( String *nomJoueur*, String *effetBalles*, int *intensiteEffet*, int *puissanceBalles*, int *frequenceBalles*, int *rotation*, int *nombreBalles* )

Méthode [ParametreSeance](#) constructeur de la classe [ParametreSeance](#).

## Paramètres

|                        |                                                                             |
|------------------------|-----------------------------------------------------------------------------|
| <i>nomJoueur</i>       | étant le nom du joueur pratiquant la séance.                                |
| <i>effetBalles</i>     | étant l'effet appliqué aux balles tout au long de la séance.                |
| <i>intensite-Effet</i> | étant l'intensité de l'effet appliqué aux balles tout au long de la séance. |

|                         |                                                                        |
|-------------------------|------------------------------------------------------------------------|
| <i>frequence-Balles</i> | étant la fréquence d'envoi des balles (en balles/minute) de la séance. |
| <i>puissance-Balles</i> | étant la puissance ou coefficient de vitesse balles à envoyer.         |
| <i>rotation</i>         | étant la rotation du lanceur en degrés.                                |
| <i>nombre-Balles</i>    | étant le nombre de balles à envoyer.                                   |

Références [com.ttpa.iris.ttpamobile.ParametreSeance.effetBalles](#), [com.ttpa.iris.ttpamobile.ParametreSeance.frequenceBalles](#), [com.ttpa.iris.ttpamobile.ParametreSeance.intensiteEffet](#), [com.ttpa.iris.ttpamobile.ParametreSeance.nombreBalles](#), [com.ttpa.iris.ttpamobile.ParametreSeance.nomJoueur](#), [com.ttpa.iris.ttpamobile.ParametreSeance.puissanceBalles](#), et [com.ttpa.iris.ttpamobile.ParametreSeance.rotation](#).

```
{
 this.nomJoueur = nomJoueur;
 this.effetBalles = effetBalles;
 this.intensiteEffet = intensiteEffet;
 this.puissanceBalles = puissanceBalles;
 this.frequenceBalles = frequenceBalles;
 this.rotation = rotation;
 this.nombreBalles = nombreBalles;
}
```

#### 9.4.3 Documentation des fonctions membres

##### 9.4.3.1 final boolean com.ttpa.iris.ttpamobile.ParametreSeance.effetEstValide (String effet )

Méthode effetEstValide vérifiant la validité de l'effet du paramétrage.

###### Paramètres

|              |                           |
|--------------|---------------------------|
| <i>effet</i> | étant l'effet à vérifier. |
|--------------|---------------------------|

###### Renvoie

Référéncé par [com.ttpa.iris.ttpamobile.ParametreSeance.estValide\(\)](#).

```
{
 switch (effet)
 {
 case "Lifté":
 return true;
 case "Aucun":
 return true;
 case "Coupé":
 return true;
 default:
 return false;
 }
}
```

#### 9.4.3.2 `boolean com.ttpa.iris.ttpamobile.ParametreSeance.estValide ( )`

Méthode `estValide` vérifiant la validité des caractéristiques du paramétrage.

Renvoie

Références [com.ttpa.iris.ttpamobile.ParametreSeance.effetBalles](#), [com.ttpa.iris.ttpamobile.ParametreSeance.effetEstValide\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.frequenceBalles](#), [com.ttpa.iris.ttpamobile.ParametreSeance.frequenceEstValide\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.nombreBalles](#), [com.ttpa.iris.ttpamobile.ParametreSeance.nombreBallesEstValide\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.puissanceBalles](#), [com.ttpa.iris.ttpamobile.ParametreSeance.puissanceBallesEstValide\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.rotation](#), et [com.ttpa.iris.ttpamobile.ParametreSeance.rotationEstValide\(\)](#).

```
{
 return (frequenceEstValide(this.frequenceBalles) &&
 nombreBallesEstValide(this.nombreBalles) && effetEstValide(this.effetBalles) &&
 puissanceBallesEstValide(this.puissanceBalles) && rotationEstValide(this.
 rotation));
}
```

#### 9.4.3.3 `final boolean com.ttpa.iris.ttpamobile.ParametreSeance.frequenceEstValide ( int frequence )`

Méthode `frequenceEstValide` vérifiant la validité de la fréquence du paramétrage.

Renvoie

Références [com.ttpa.iris.ttpamobile.ParametreSeance.MAX\\_FREQUENCE\\_PARAMETRE](#), et [com.ttpa.iris.ttpamobile.ParametreSeance.MIN\\_FREQUENCE\\_PARAMETRE](#).

Référencé par [com.ttpa.iris.ttpamobile.ParametreSeance.estValide\(\)](#).

```
{
 return (frequence >= MIN_FREQUENCE_PARAMETRE && frequence <=
 MAX_FREQUENCE_PARAMETRE);
}
```

#### 9.4.3.4 `final char com.ttpa.iris.ttpamobile.ParametreSeance.getEffet ( )` `[package]`

Méthode `getEffet` vérifiant la validité de l'effet du paramétrage.

Renvoie

`char` l'effet pour la trame

Références [com.ttpa.iris.ttpamobile.ParametreSeance.effetBalles](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeancePeripheriqueBluetoothLanceur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameRepriseSeancePeripheriqueBluetoothLanceur\(\)](#).



```
{
 switch (this.effetBalles)
 {
 case "Lifté":
 return 'L';
 case "Aucun":
 return 'S';
 case "Coupé":
 return 'C';
 default:
 return 'S';
 }
}
```

#### 9.4.3.5 String com.ttpa.iris.ttpamobile.ParametreSeance.getEffetBalles ( )

Méthode [getEffetBalles\(\)](#) accesseur de l'attribut effetBalles.

Renvoie

effetBalles

Références [com.ttpa.iris.ttpamobile.ParametreSeance.effetBalles](#).

```
{ return effetBalles; }
```

#### 9.4.3.6 final String com.ttpa.iris.ttpamobile.ParametreSeance.getEffetComplet ( ) [package]

Méthode [getEffetComplet](#) accesseur de l'attribut effetBalles.

Renvoie

String l'effet pour la trame

Références [com.ttpa.iris.ttpamobile.ParametreSeance.effetBalles](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#).

```
{ return this.effetBalles; }
```

#### 9.4.3.7 int com.ttpa.iris.ttpamobile.ParametreSeance.getFrequenceBalles ( )

Méthode [getFrequenceBalles\(\)](#) accesseur de l'attribut frequenceBalles.

Renvoie

frequenceBalles

Références [com.ttpa.iris.ttpamobile.ParametreSeance.frequenceBalles](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeancePeripheriqueBluetoothLanceur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameRepriseSeancePeripheriqueBluetoothLanceur\(\)](#).

```
{
 return frequenceBalles;
}
```

#### 9.4.3.8 final int com.ttpa.iris.ttpamobile.ParametreSeance.getIntensiteEffet ( ) [package]

Méthode `getIntensiteEffet` accesseur de l'attribut `intensiteEffet`.

Renvoie

`intensiteEffet` l'effet pour la trame

Références [com.ttpa.iris.ttpamobile.ParametreSeance.intensiteEffet](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeancePeripheriqueBluetoothLanceur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameRepriseSeancePeripheriqueBluetoothLanceur\(\)](#).

```
{ return this.intensiteEffet; }
```

#### 9.4.3.9 int com.ttpa.iris.ttpamobile.ParametreSeance.getNombreBalles ( )

Méthode `getNombreBalles()` accesseur de l'attribut `nombreBalles`.

Renvoie

`nombreBalles`

Références [com.ttpa.iris.ttpamobile.ParametreSeance.nombreBalles](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeancePeripheriqueBluetoothLanceur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameParametrageSeancePeripheriqueBluetoothEcran\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameRepriseSeancePeripheriqueBluetoothLanceur\(\)](#).

```
{
 return nombreBalles;
}
```

#### 9.4.3.10 String com.ttpa.iris.ttpamobile.ParametreSeance.getNomJoueur ( )

Méthode `getNomJoueur()` accesseur de l'attribut `nomJoueur`.

Renvoie

`nomJoueur`

Références [com.ttpa.iris.ttpamobile.ParametreSeance.nomJoueur](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameConnexionPeripheriqueBluetoothEcran\(\)](#).

```
{ return nomJoueur; }
```

**9.4.3.11** `int com.ttpa.iris.ttpamobile.ParametreSeance.getPuissanceBalles ( )`

Références [com.ttpa.iris.ttpamobile.ParametreSeance.puissanceBalles](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeancePeripheriqueBluetoothLanceur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameRepriseSeancePeripheriqueBluetoothLanceur\(\)](#).

```
{
 return puissanceBalles;
}
```

**9.4.3.12** `int com.ttpa.iris.ttpamobile.ParametreSeance.getRotation ( )`

Références [com.ttpa.iris.ttpamobile.ParametreSeance.rotation](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeancePeripheriqueBluetoothLanceur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameRepriseSeancePeripheriqueBluetoothLanceur\(\)](#).

```
{
 return rotation;
}
```

**9.4.3.13** `final boolean com.ttpa.iris.ttpamobile.ParametreSeance.nombreBallesEstValide ( int nombreBalles )`

Méthode `nombreBallesEstValide` vérifiant la validité du nombre de balles du paramétrage.

**Paramètres**

|                      |                                       |
|----------------------|---------------------------------------|
| <i>nombre-Balles</i> | étant le nombre de balles à vérifier. |
|----------------------|---------------------------------------|

**Renvoie**

Références [com.ttpa.iris.ttpamobile.ParametreSeance.MAX\\_NOMBRE\\_BALLES\\_PARAMETRE](#), et [com.ttpa.iris.ttpamobile.ParametreSeance.MIN\\_NOMBRE\\_BALLES\\_PARAMETRE](#).

Référéncé par [com.ttpa.iris.ttpamobile.ParametreSeance.estValide\(\)](#).

```
{
 return (nombreBalles >= MIN_NOMBRE_BALLES_PARAMETRE && nombreBalles <=
 MAX_NOMBRE_BALLES_PARAMETRE);
}
```

9.4.3.14 final boolean com.ttpa.iris.ttpamobile.ParametreSeance.puissanceBallesEstValide ( int *puissanceBalles* )

Méthode puissanceBallesEstValide vérifiant la validité de la puissance des balles.

Paramètres

|                        |                    |
|------------------------|--------------------|
| <i>puissanceBalles</i> | étant la puissance |
|------------------------|--------------------|

Renvoie

Références [com.ttpa.iris.ttpamobile.ParametreSeance.MAX\\_PUISSANCE\\_BALLES\\_PARAMETER](#), et [com.ttpa.iris.ttpamobile.ParametreSeance.MIN\\_PUISSANCE\\_BALLES\\_PARAMETER](#).

Référéncé par [com.ttpa.iris.ttpamobile.ParametreSeance.estValide\(\)](#).

```
{
 return (puissanceBalles >= MIN_PUISSANCE_BALLES_PARAMETER &&
 puissanceBalles <= MAX_PUISSANCE_BALLES_PARAMETER);
}
```

9.4.3.15 final boolean com.ttpa.iris.ttpamobile.ParametreSeance.rotationEstValide ( int *rotation* )

Méthode rotationEstValide vérifiant la validité de la rotation.

Paramètres

|                 |                   |
|-----------------|-------------------|
| <i>rotation</i> | étant la rotation |
|-----------------|-------------------|

Renvoie

Références [com.ttpa.iris.ttpamobile.ParametreSeance.MAX\\_ROTATION\\_PARAMETER](#), et [com.ttpa.iris.ttpamobile.ParametreSeance.MIN\\_ROTATION\\_PARAMETER](#).

Référéncé par [com.ttpa.iris.ttpamobile.ParametreSeance.estValide\(\)](#).

```
{
 return (rotation >= MIN_ROTATION_PARAMETER && rotation <=
 MAX_ROTATION_PARAMETER);
}
```

9.4.3.16 void com.ttpa.iris.ttpamobile.ParametreSeance.setEffetBalles ( String *effetBalles* )

Méthode setEffet mutateur de l'attribut effetBalles.

## Paramètres

|                    |                           |
|--------------------|---------------------------|
| <i>effetBalles</i> | étant l'effet à affecter. |
|--------------------|---------------------------|

Références [com.ttpa.iris.ttpamobile.ParametreSeance.effetBalles](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance\(\)](#).

```
{
 this.effetBalles = effetBalles;
}
```

#### 9.4.3.17 void `com.ttpa.iris.ttpamobile.ParametreSeance.setFrequenceBalles` ( int *frequenceBalles* )

Méthode `setFrequenceBalles` mutateur de l'attribut `frequenceBalles`.

## Paramètres

|                         |                                |
|-------------------------|--------------------------------|
| <i>frequence-Balles</i> | étant la fréquence à affecter. |
|-------------------------|--------------------------------|

Références [com.ttpa.iris.ttpamobile.ParametreSeance.frequenceBalles](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance\(\)](#).

```
{
 this.frequenceBalles = frequenceBalles;
}
```

#### 9.4.3.18 void `com.ttpa.iris.ttpamobile.ParametreSeance.setIntensiteEffet` ( int *intensiteEffet* )

Méthode `setIntensiteEffet` mutateur de l'attribut `intensiteEffet`.

## Paramètres

|                        |                                          |
|------------------------|------------------------------------------|
| <i>intensite-Effet</i> | étant l'intensité de l'effet à affecter. |
|------------------------|------------------------------------------|

Références [com.ttpa.iris.ttpamobile.ParametreSeance.intensiteEffet](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance\(\)](#).

```
{
 this.intensiteEffet = intensiteEffet;
}
```

#### 9.4.3.19 void `com.ttpa.iris.ttpamobile.ParametreSeance.setNombreBalles` ( int *nombreBalles* )

Méthode `setNombreBalles()` mutateur de l'attribut `nombreBalles`.

## Paramètres

|                      |                                       |
|----------------------|---------------------------------------|
| <i>nombre-Balles</i> | étant le nombre de balles à affecter. |
|----------------------|---------------------------------------|

Références [com.ttpa.iris.ttpamobile.ParametreSeance.nombreBalles](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance\(\)](#).

```
{
 this.nombreBalles = nombreBalles;
}
```

#### 9.4.3.20 void `com.ttpa.iris.ttpamobile.ParametreSeance.setNomJoueur` ( String *nomJoueur* )

Méthode `setNomJoueur` mutateur de l'attribut `nomJoueur`.

## Paramètres

|                  |                         |
|------------------|-------------------------|
| <i>nomJoueur</i> | étant le nom à affecter |
|------------------|-------------------------|

Références [com.ttpa.iris.ttpamobile.ParametreSeance.nomJoueur](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.onClick\(\)](#).

```
{ this.nomJoueur = nomJoueur; }
```

#### 9.4.3.21 void `com.ttpa.iris.ttpamobile.ParametreSeance.setPuissanceBalles` ( int *puissanceBalles* )

Références [com.ttpa.iris.ttpamobile.ParametreSeance.puissanceBalles](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance\(\)](#).

```
{
 this.puissanceBalles = puissanceBalles;
}
```

#### 9.4.3.22 void `com.ttpa.iris.ttpamobile.ParametreSeance.setRotation` ( int *rotation* )

Références [com.ttpa.iris.ttpamobile.ParametreSeance.rotation](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.appliquerParametresSeance\(\)](#).

```
{
 this.rotation = rotation;
}
```

#### 9.4.3.23 `String com.ttpa.iris.ttpamobile.ParametreSeance.toString ( )`

Méthode `toString` permettant la visualisation des caractéristiques du paramétrage.

Renvoie

les caractéristiques du paramétrage.

Références [com.ttpa.iris.ttpamobile.ParametreSeance.effetBalles](#), [com.ttpa.iris.ttpamobile.ParametreSeance.frequenceBalles](#), [com.ttpa.iris.ttpamobile.ParametreSeance.intensiteEffet](#), [com.ttpa.iris.ttpamobile.ParametreSeance.nombreBalles](#), et [com.ttpa.iris.ttpamobile.ParametreSeance.puissanceBalles](#).

```
{
 return "Frequence : " + frequenceBalles + "\nNombre balles : " +
 nombreBalles + "\nEffet : " + effetBalles + "\nIntensité Effet : " +
 intensiteEffet + "\nPuissance balles : " + puissanceBalles;
}
```

#### 9.4.4 Documentation des données membres

##### 9.4.4.1 `String com.ttpa.iris.ttpamobile.ParametreSeance.effetBalles` [private]

Référencé par [com.ttpa.iris.ttpamobile.ParametreSeance.estValide\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.getEffet\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.getEffetBalles\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.getEffetCompleet\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.ParametreSeance\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.setEffetBalles\(\)](#), et [com.ttpa.iris.ttpamobile.ParametreSeance.toString\(\)](#).

##### 9.4.4.2 `int com.ttpa.iris.ttpamobile.ParametreSeance.frequenceBalles` [private]

Référencé par [com.ttpa.iris.ttpamobile.ParametreSeance.estValide\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.getFrequenceBalles\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.ParametreSeance\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.setFrequenceBalles\(\)](#), et [com.ttpa.iris.ttpamobile.ParametreSeance.toString\(\)](#).

##### 9.4.4.3 `int com.ttpa.iris.ttpamobile.ParametreSeance.intensiteEffet` [private]

Référencé par [com.ttpa.iris.ttpamobile.ParametreSeance.getIntensiteEffet\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.ParametreSeance\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.setIntensiteEffet\(\)](#), et [com.ttpa.iris.ttpamobile.ParametreSeance.toString\(\)](#).

##### 9.4.4.4 `final int com.ttpa.iris.ttpamobile.ParametreSeance.MAX_FREQUENCE_PARAMETRE = 60` [static]

Référencé par [com.ttpa.iris.ttpamobile.ParametreSeance.frequenceEstValide\(\)](#).

9.4.4.5 `final int com.ttpa.iris.ttpamobile.ParametreSeance.MAX_NOMBRE_BALLES_PARAMETRE = 50` `[static]`

Référencé par `com.ttpa.iris.ttpamobile.ParametreSeance.nombreBallesEstValide()`.

9.4.4.6 `final int com.ttpa.iris.ttpamobile.ParametreSeance.MAX_PUISSANCE_BALLES_PARAMETRE = 10` `[static]`

Référencé par `com.ttpa.iris.ttpamobile.ParametreSeance.puissanceBallesEstValide()`.

9.4.4.7 `final int com.ttpa.iris.ttpamobile.ParametreSeance.MAX_ROTATION_PARAMETRE = 180` `[static]`

Référencé par `com.ttpa.iris.ttpamobile.ParametreSeance.rotationEstValide()`.

9.4.4.8 `final int com.ttpa.iris.ttpamobile.ParametreSeance.MIN_FREQUENCE_PARAMETRE = 1` `[static]`

Référencé par `com.ttpa.iris.ttpamobile.ParametreSeance.frequenceEstValide()`.

9.4.4.9 `final int com.ttpa.iris.ttpamobile.ParametreSeance.MIN_NOMBRE_BALLES_PARAMETRE = 1` `[static]`

Référencé par `com.ttpa.iris.ttpamobile.ParametreSeance.nombreBallesEstValide()`.

9.4.4.10 `final int com.ttpa.iris.ttpamobile.ParametreSeance.MIN_PUISSANCE_BALLES_PARAMETRE = 1` `[static]`

Référencé par `com.ttpa.iris.ttpamobile.ParametreSeance.puissanceBallesEstValide()`.

9.4.4.11 `final int com.ttpa.iris.ttpamobile.ParametreSeance.MIN_ROTATION_PARAMETRE = 0` `[static]`

Référencé par `com.ttpa.iris.ttpamobile.ParametreSeance.rotationEstValide()`.

9.4.4.12 `int com.ttpa.iris.ttpamobile.ParametreSeance.nombreBalles` `[private]`

Référencé par `com.ttpa.iris.ttpamobile.ParametreSeance.estValide()`, `com.ttpa.iris.ttpamobile.ParametreSeance.getNombreBalles()`, `com.ttpa.iris.ttpamobile.ParametreSeance.ParametreSeance()`, `com.ttpa.iris.ttpamobile.ParametreSeance.setNombreBalles()`, et `com.ttpa.iris.ttpamobile.ParametreSeance.toString()`.

9.4.4.13 `String com.ttpa.iris.ttpamobile.ParametreSeance.nomJoueur` `[private]`

Référencé par `com.ttpa.iris.ttpamobile.ParametreSeance.getNomJoueur()`, `com.ttpa.iris.ttpamobile.ParametreSeance.ParametreSeance()`, et `com.ttpa.iris.ttpamobile.ParametreSeance.setNomJoueur()`.



9.4.4.14 int com.ttpa.iris.ttpamobile.ParametreSeance.puissanceBalles  
[private]

Référencé par [com.ttpa.iris.ttpamobile.ParametreSeance.estValide\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.getPuissanceBalles\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.ParametreSeance\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.setPuissanceBalles\(\)](#), et [com.ttpa.iris.ttpamobile.ParametreSeance.toString\(\)](#).

9.4.4.15 int com.ttpa.iris.ttpamobile.ParametreSeance.rotation [private]

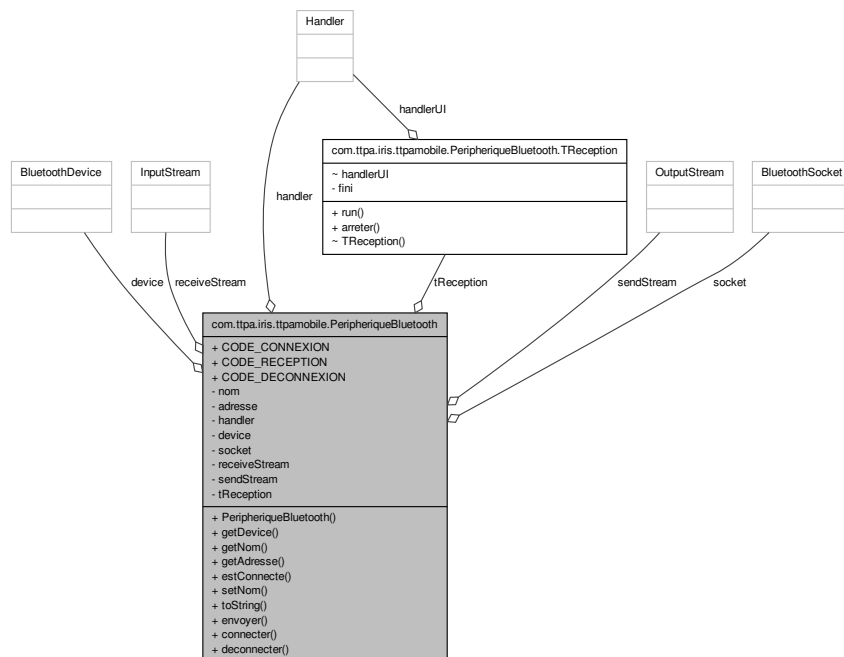
Référencé par [com.ttpa.iris.ttpamobile.ParametreSeance.estValide\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.getRotation\(\)](#), [com.ttpa.iris.ttpamobile.ParametreSeance.ParametreSeance\(\)](#), et [com.ttpa.iris.ttpamobile.ParametreSeance.setRotation\(\)](#).

La documentation de cette classe a été générée à partir du fichier suivant :

– [ParametreSeance.java](#)

9.5 Référence de la classe com.ttpa.iris.ttpamobile.PeripheriqueBluetooth

Graphe de collaboration de com.ttpa.iris.ttpamobile.PeripheriqueBluetooth :



### Classes

- class [TReception](#)

### Fonctions membres publiques

- [PeripheriqueBluetooth](#) (BluetoothDevice [device](#), Handler [handler](#))
- BluetoothDevice [getDevice](#) ()
- String [getNom](#) ()
- String [getAdresse](#) ()
- boolean [estConnecte](#) ()
- void [setNom](#) (String [nom](#))
- String [toString](#) ()
- void [envoyer](#) (String data)
- void [connecter](#) ()
- boolean [deconnecter](#) (boolean fermeture)

### Attributs publics statiques

- static final int [CODE\\_CONNEXION](#) = 0
- static final int [CODE\\_RECEPTION](#) = 1
- static final int [CODE\\_DECONNEXION](#) = 2

### Attributs privés

- String [nom](#)
- String [adresse](#)
- Handler [handler](#) = null
- BluetoothDevice [device](#) = null
- BluetoothSocket [socket](#) = null
- InputStream [receiveStream](#) = null
- OutputStream [sendStream](#) = null
- [TReception](#) [tReception](#)

#### 9.5.1 Description détaillée

Created by smaniotto on 04/04/18.

#### 9.5.2 Documentation des constructeurs et destructeur

##### 9.5.2.1 [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.PeripheriqueBluetooth](#) (BluetoothDevice [device](#), Handler [handler](#) )

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.adresse](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.device](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.handler](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.nom](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.receiveStream](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.sendStream](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.socket](#), et [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.tReception](#).

```
{
 if(device != null)
 {
```

```

 this.device = device;
 this.nom = device.getName();
 this.adresse = device.getAddress();
 this.handler = handler;
 }
 else
 {
 this.device = device;
 this.nom = "Aucun";
 this.adresse = "";
 this.handler = handler;
 }

 try
 {
 System.out.println("<Bluetooth> nom " + device.getName());
 ParcelUuid[] uuids = device.getUuids();
 if(uuids != null)
 for(int i = 0; i < uuids.length; i++)
 System.out.println("<Bluetooth> uuid " + uuids[i].getUuid()
.toString());
 else
 System.out.println("<Bluetooth> uuid null !");

 socket = device.createRfcommSocketToServiceRecord(UUID.fromString("
00001101-0000-1000-8000-00805F9B34FB"));
 //socket =
 device.createRfcommSocketToServiceRecord(uuids[0].getUuid());
 System.out.println("<Bluetooth> new socket");

 // API 23
 //System.out.println("<Bluetooth> max receive packet size : " +
socket.getMaxReceivePacketSize());
 //System.out.println("<Bluetooth> max transmit packet size : " +
socket.getMaxTransmitPacketSize());
 //System.out.println("<Bluetooth> connection type : " +
socket.getConnectionType());

 receiveStream = socket.getInputStream();
 sendStream = socket.getOutputStream();
 }
 catch (IOException e)
 {
 e.printStackTrace();
 System.out.println("<Bluetooth> Erreur socket !");
 socket = null;
 }

 if(socket != null)
 TReception = new TReception(handler);
}

```

### 9.5.3 Documentation des fonctions membres

#### 9.5.3.1 void com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.connecter ( )

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.CODE\\_CONNEXION](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.getAdresse\(\)](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.getNom\(\)](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.socket](#), et [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothEcran\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothLanceur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothTable\(\)](#).

```

{
 new Thread()
 {
 @Override public void run()
 {
 try
 {
 System.out.println("<Bluetooth> socket connect ...");
 socket.connect();
 System.out.println("<Bluetooth> socket connect ok");
 Message msg = Message.obtain();
 //msg.arg1 = CODE_CONNEXION;
 Bundle b = new Bundle();
 b.putString("nom", getNom());
 b.putString("adresse", getAdresse());
 b.putInt("etat", CODE_CONNEXION);
 b.putString("donnees", "");
 msg.setData(b);
 handler.sendMessage(msg);

 tReception.start();
 }
 catch (IOException e)
 {
 System.out.println("<Bluetooth> Erreur connect !");
 e.printStackTrace();
 }
 }
 }.start();
}

```

#### 9.5.3.2 boolean `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.deconnecter (boolean fermeture)`

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.arreter\(\)](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.socket](#), et [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.tReception](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.deconnexionPeripheriquesBluetooth\(\)](#).

```

{
 try
 {
 //if(estConnecte())
 {
 tReception.arreter();

 if(fermeture)
 {
 socket.close();
 System.out.println("<Bluetooth> socket close");
 }
 return true;
 }
 }
 catch (IOException e)
 {
 System.out.println("<Bluetooth> Erreur close !");
 e.printStackTrace();
 return false;
 }
}

```

## 9.5.3.3 void com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.envoyer ( String data )

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.sendStream](#), et [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.socket](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameArretPeripheriqueBluetoothLanceur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameArretPeripheriqueBluetoothTable\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeancePeripheriqueBluetoothLanceur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetoothEcran\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetoothLanceur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameRepriseSeancePeripheriqueBluetoothLanceur\(\)](#).

```
{
 if(socket == null)
 return;

 try
 {
 //if(estConnecte())
 if(socket.isConnected())
 {
 System.out.println("<Bluetooth> Envoyer " + data);
 sendStream.write(data.getBytes());
 sendStream.flush();
 }
 else
 {
 System.out.println("<Bluetooth> Envoyer (non connecté) " + data
);
 }
 }
 catch (IOException e)
 {
 System.out.println("<Bluetooth> Erreur socket write !");
 e.printStackTrace();
 }
}
```

## 9.5.3.4 boolean com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.estConnecte ( )

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.socket](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothEcran\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothLanceur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionPeripheriqueBluetoothTable\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameArretPeripheriqueBluetoothLanceur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameArretPeripheriqueBluetoothTable\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTrameDebutSeancePeripheriqueBluetoothLanceur\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetoothEcran\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.envoyerTramePeripheriqueBluetoothLanceur\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.verifierConnexionAppareilsBluetoothRequis\(\)](#).

```
{
 if(socket != null)
```

```
 return socket.isConnected();
 return false;
}
```

**9.5.3.5 String com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.getAdresse ( )**

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.adresse](#).

Référencé par [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.connecter\(\)](#), et [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.run\(\)](#).

```
{
 return adresse;
}
```

**9.5.3.6 BluetoothDevice com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.getDevice ( )**

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.device](#).

```
{
 return device;
}
```

**9.5.3.7 String com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.getNom ( )**

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.nom](#).

Référencé par [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.connecter\(\)](#), et [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.run\(\)](#).

```
{
 return nom;
}
```

**9.5.3.8 void com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.setNom ( String nom )**

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.nom](#).

```
{
 this.nom = nom;
}
```

**9.5.3.9 String com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.toString ( )**

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.adresse](#), et [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.nom](#).

```
{
 return "\nNom : " + nom + "\nAdresse : " + adresse;
}
```

### 9.5.4 Documentation des données membres

**9.5.4.1 String `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.adresse`**  
[private]

Référencé par `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.getAdresse()`, `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.PeripheriqueBluetooth()`, et `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.toString()`.

**9.5.4.2 final int `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.CODE_CONNEXION` = 0** [static]

Référencé par `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.connecter()`.

**9.5.4.3 final int `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.CODE_DECONNEXION` = 2** [static]

Référencé par `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.run()`.

**9.5.4.4 final int `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.CODE_RECEPTION` = 1** [static]

Référencé par `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.run()`.

**9.5.4.5 BluetoothDevice `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.device` = null**  
[private]

Référencé par `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.getDevice()`, et `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.PeripheriqueBluetooth()`.

**9.5.4.6 Handler `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.handler` = null**  
[private]

Référencé par `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.connecter()`, `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.PeripheriqueBluetooth()`, et `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.run()`.

**9.5.4.7 String `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.nom`** [private]

Référencé par `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.getNom()`, `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.PeripheriqueBluetooth()`, `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.setNom()`, et `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.toString()`.

**9.5.4.8 InputStream `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.receiveStream` = null** [private]

Référencé par `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.PeripheriqueBluetooth()`, et `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.run()`.

**9.5.4.9 OutputStream** `com.tpa.iris.ttpamobile.PeripheriqueBluetooth.sendStream = null [private]`

Référencé par [com.tpa.iris.ttpamobile.PeripheriqueBluetooth.envoyer\(\)](#), et [com.tpa.iris.ttpamobile.PeripheriqueBluetooth.PeripheriqueBluetooth\(\)](#).

**9.5.4.10 BluetoothSocket** `com.tpa.iris.ttpamobile.PeripheriqueBluetooth.socket = null [private]`

Référencé par [com.tpa.iris.ttpamobile.PeripheriqueBluetooth.connecter\(\)](#), [com.tpa.iris.ttpamobile.PeripheriqueBluetooth.deconnecter\(\)](#), [com.tpa.iris.ttpamobile.PeripheriqueBluetooth.envoyer\(\)](#), [com.tpa.iris.ttpamobile.PeripheriqueBluetooth.estConnecte\(\)](#), et [com.tpa.iris.ttpamobile.PeripheriqueBluetooth.PeripheriqueBluetooth\(\)](#).

**9.5.4.11 TReception** `com.tpa.iris.ttpamobile.PeripheriqueBluetooth.tReception [private]`

Référencé par [com.tpa.iris.ttpamobile.PeripheriqueBluetooth.connecter\(\)](#), [com.tpa.iris.ttpamobile.PeripheriqueBluetooth.deconnecter\(\)](#), et [com.tpa.iris.ttpamobile.PeripheriqueBluetooth.PeripheriqueBluetooth\(\)](#).

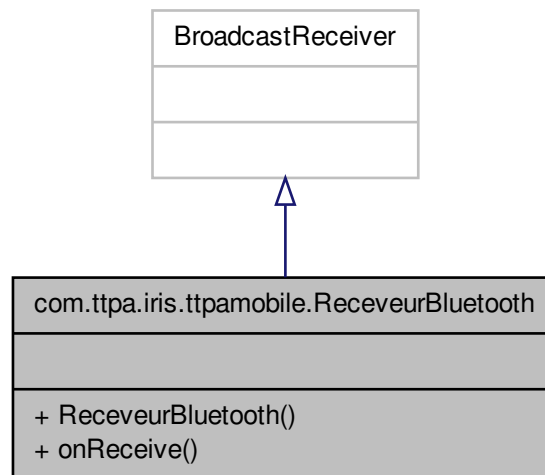
La documentation de cette classe a été générée à partir du fichier suivant :

– [PeripheriqueBluetooth.java](#)

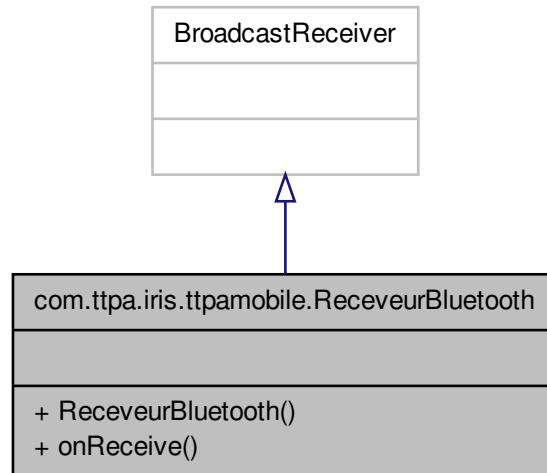


## 9.6 Référence de la classe com.ttpa.iris.ttpamobile.ReceveurBluetooth

Graphe d'héritage de com.ttpa.iris.ttpamobile.ReceveurBluetooth :



Graphe de collaboration de com.ttpa.iris.ttpamobile.ReceveurBluetooth :



#### Fonctions membres publiques

- [ReceveurBluetooth](#) ()
- void [onReceive](#) (Context context, Intent intent)

#### 9.6.1 Documentation des constructeurs et destructeur

##### 9.6.1.1 com.ttpa.iris.ttpamobile.ReceveurBluetooth.ReceveurBluetooth ( )

```

{
}

```

#### 9.6.2 Documentation des fonctions membres

##### 9.6.2.1 void com.ttpa.iris.ttpamobile.ReceveurBluetooth.onReceive ( Context context, Intent intent )

```

{
 String action = intent.getAction();
 if (BluetoothDevice.ACTION_FOUND.equals(action))
 {
 BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.
 EXTRA_DEVICE);
 Toast.makeText(context, "Nouveau périphérique : " + device.getName(
), Toast.LENGTH_SHORT).show();
 }
}

```

La documentation de cette classe a été générée à partir du fichier suivant :

- [ReceveurBluetooth.java](#)

## 9.7 Référence de la classe com.ttpa.iris.ttpamobile.Seance

### Fonctions membres publiques

- [Seance](#) ()
- [Seance](#) (int [frequence](#), int [nombreBalles](#), String [effet](#), int [intensiteEffet](#), int [puissance](#), int [rotation](#))
- String [horodaterBD](#) ()
- int [getId](#) ()
- void [setId](#) (int [id](#))
- int [getFrequence](#) ()
- void [setFrequence](#) (int [frequence](#))
- int [getNombreBalles](#) ()
- void [setNombreBalles](#) (int [nombreBalles](#))
- void [setEffet](#) (String [effet](#))
- String [getEffet](#) ()
- void [setIntensiteEffet](#) (int [intensiteEffet](#))
- float [getTauxReussite](#) ()
- void [setTauxReussite](#) (float [tauxReussite](#))
- int [getPuissance](#) ()
- void [setPuissance](#) (int [puissance](#))
- int [getRotation](#) ()
- void [setRotation](#) (int [rotation](#))
- int [getZoneObjectif](#) ()
- void [setZoneObjectif](#) (int [zoneObjectif](#))
- int [getZoneRobot](#) ()
- void [setZoneRobot](#) (int [zoneRobot](#))
- String [getDateDebut](#) ()
- void [setDateDebut](#) (String [dateDebut](#))
- String [getDateFin](#) ()
- void [setDateFin](#) (String [dateFin](#))
- int [getIdJoueur](#) ()
- void [setIdJoueur](#) (int [idJoueur](#))
- String [toString](#) ()

### Fonctions de paquetage

- final int [getIntensiteEffet](#) ()

### Attributs privés

- int [id](#)
- int [frequence](#)
- int [nombreBalles](#)
- String [effet](#)
- int [intensiteEffet](#)
- int [puissance](#)
- int [rotation](#)
- int [zoneObjectif](#)
- int [zoneRobot](#)
- float [tauxReussite](#)
- String [dateDebut](#)
- String [dateFin](#)
- int [idJoueur](#)

## 9.7.1 Description détaillée

Created by smaniotto on 16/03/18. Classe [Seance](#) définissant les caractéristiques et le comportement d'une séance.

## 9.7.2 Documentation des constructeurs et destructeur

## 9.7.2.1 com.ttpa.iris.ttpamobile.Seance.Seance ( )

Méthode [Seance](#) constructeur par défaut de la classe [Seance](#).

```
{ }
```

9.7.2.2 com.ttpa.iris.ttpamobile.Seance.Seance ( int *frequence*, int *nombreBalles*, String *effet*, int *intensiteEffet*, int *puissance*, int *rotation* )

Méthode [Seance](#) constructeur de la classe [Seance](#).

## Paramètres

|                      |                                                                        |
|----------------------|------------------------------------------------------------------------|
| <i>frequence</i>     | étant la fréquence d'envoi des balles (en balles/minute) de la séance. |
| <i>nombre-Balles</i> | étant le nombre de balles à envoyer.                                   |
| <i>effet</i>         | étant l'effet appliqué aux balles tout au long de la séance.           |
| <i>puissance</i>     | étant la puissance de la balle envoyée.                                |
| <i>rotation</i>      | étant la rotation du lanceur.                                          |

Références [com.ttpa.iris.ttpamobile.Seance.dateDebut](#), [com.ttpa.iris.ttpamobile.Seance.dateFin](#), [com.ttpa.iris.ttpamobile.Seance.effet](#), [com.ttpa.iris.ttpamobile.Seance.frequence](#), [com.ttpa.iris.ttpamobile.Seance.horodaterBD\(\)](#), [com.ttpa.iris.ttpamobile.Seance.idJoueur](#), [com.ttpa.iris.ttpamobile.Seance.intensiteEffet](#), [com.ttpa.iris.ttpamobile.Seance.nombreBalles](#), [com.ttpa.iris.ttpamobile.Seance.puissance](#), [com.ttpa.iris.ttpamobile.Seance.rotation](#), [com.ttpa.iris.ttpamobile.Seance.tauxReussite](#), [com.ttpa.iris.ttpamobile.Seance.zoneObjectif](#), et [com.ttpa.iris.ttpamobile.Seance.zone-Robot](#).

```
{
 this.frequence = frequence;
 this.nombreBalles = nombreBalles;
 this.effet = effet;
 this.intensiteEffet = intensiteEffet;
 this.puissance = puissance;
 this.rotation = rotation;
 this.zoneObjectif = -1;
 this.zoneRobot = -1;
 this.tauxReussite = 0;
 this.dateDebut = horodaterBD();
 this.dateFin = horodaterBD();
 this.idJoueur = -1;
}
```

## 9.7.3 Documentation des fonctions membres

#### 9.7.3.1 `String com.ttpa.iris.ttpamobile.Seance.getDateDebut ( )`

Références [com.ttpa.iris.ttpamobile.Seance.dateDebut](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherSeance\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

```
{
 return dateDebut;
}
```

#### 9.7.3.2 `String com.ttpa.iris.ttpamobile.Seance.getDateFin ( )`

Références [com.ttpa.iris.ttpamobile.Seance.dateFin](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

```
{
 return dateFin;
}
```

#### 9.7.3.3 `String com.ttpa.iris.ttpamobile.Seance.getEffet ( )`

Méthode `getEffet` accesseur de l'attribut `effet`.

Renvoie

`effet`

Références [com.ttpa.iris.ttpamobile.Seance.effet](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherSeance\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

```
{ return effet; }
```

#### 9.7.3.4 `int com.ttpa.iris.ttpamobile.Seance.getFrequence ( )`

Méthode `getFrequence` accesseur de l'attribut `frequence`.

Renvoie

`frequence`

Références [com.ttpa.iris.ttpamobile.Seance.frequence](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherSeance\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

```
{
 return frequence;
}
```

**9.7.3.5** `int com.ttpa.iris.ttpamobile.Seance.getId ( )`

Méthode `getId` accesseur de l'attribut `id`.

Renvoie

`id`

Références [com.ttpa.iris.ttpamobile.Seance.id](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherSeance\(\)](#).

```
{ return id; }
```

**9.7.3.6** `int com.ttpa.iris.ttpamobile.Seance.getIdJoueur ( )`

Références [com.ttpa.iris.ttpamobile.Seance.idJoueur](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

```
{
 return idJoueur;
}
```

**9.7.3.7** `final int com.ttpa.iris.ttpamobile.Seance.getIntensiteEffet ( )` [package]

Méthode `getIntensiteEffet` accesseur de l'attribut `intensiteEffet`.

Renvoie

`intensiteEffet`

Références [com.ttpa.iris.ttpamobile.Seance.intensiteEffet](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

```
{ return intensiteEffet; }
```

**9.7.3.8** `int com.ttpa.iris.ttpamobile.Seance.getNombreBalles ( )`

Méthode `getNombreBalles` accesseur de l'attribut `nombreBalles`.

Renvoie

`nombreBalles`

Références [com.ttpa.iris.ttpamobile.Seance.nombreBalles](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.calculerReussiteSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.incrementerBallesJouees\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

```
{
 return nombreBalles;
}
```

**9.7.3.9** `int com.ttpa.iris.ttpamobile.Seance.getPuissance ( )`

Références [com.ttpa.iris.ttpamobile.Seance.puissance](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

```
{
 return puissance;
}
```

**9.7.3.10** `int com.ttpa.iris.ttpamobile.Seance.getRotation ( )`

Références [com.ttpa.iris.ttpamobile.Seance.rotation](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

```
{
 return rotation;
}
```

**9.7.3.11** `float com.ttpa.iris.ttpamobile.Seance.getTauxReussite ( )`

Méthode `getTauxReussite` accesseur de l'attribut `tauxReussite`.

Renvoie

Références [com.ttpa.iris.ttpamobile.Seance.tauxReussite](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherSeance\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

```
{
 return tauxReussite;
}
```

**9.7.3.12** `int com.ttpa.iris.ttpamobile.Seance.getZoneObjectif ( )`

Références [com.ttpa.iris.ttpamobile.Seance.zoneObjectif](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherBoiteDialogue-DetailsSeance\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.calculerReussiteSeance\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

```
{
 return zoneObjectif;
}
```

**9.7.3.13** `int com.ttpa.iris.ttpamobile.Seance.getZoneRobot ( )`

Références [com.ttpa.iris.ttpamobile.Seance.zoneRobot](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherBoiteDialogue-DetailsSeance\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

```

{
 return zoneRobot;
}

```

#### 9.7.3.14 String com.ttpa.iris.ttpamobile.Seance.horodaterBD ( )

Référencé par [com.ttpa.iris.ttpamobile.Seance.Seance\(\)](#).

```

{
 Calendar calendar = Calendar.getInstance();
 // Format SQLite : "2018-04-28 14:11:52"
 SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
 final String strDate = simpleDateFormat.format(calendar.getTime());
 return strDate;
}

```

#### 9.7.3.15 void com.ttpa.iris.ttpamobile.Seance.setDateDebut ( String dateDebut )

Références [com.ttpa.iris.ttpamobile.Seance.dateDebut](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance\(\)](#).

```

{
 this.dateDebut = dateDebut;
}

```

#### 9.7.3.16 void com.ttpa.iris.ttpamobile.Seance.setDateFin ( String dateFin )

Références [com.ttpa.iris.ttpamobile.Seance.dateFin](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance\(\)](#).

```

{
 this.dateFin = dateFin;
}

```

#### 9.7.3.17 void com.ttpa.iris.ttpamobile.Seance.setEffet ( String effet )

Méthode setEffet mutateur de l'attribut effet.

##### Paramètres

|              |                           |
|--------------|---------------------------|
| <i>effet</i> | étant l'effet à affecter. |
|--------------|---------------------------|

Références [com.ttpa.iris.ttpamobile.Seance.effet](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance\(\)](#).

```

{
 this.effet = effet;
}

```



**9.7.3.18 void com.ttpa.iris.ttpamobile.Seance.setFrequence ( int *frequence* )**

Méthode setFrequence mutateur de l'attribut frequence.

**Paramètres**

|                  |                                |
|------------------|--------------------------------|
| <i>frequence</i> | étant la fréquence à affecter. |
|------------------|--------------------------------|

Références [com.ttpa.iris.ttpamobile.Seance.frequence](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance\(\)](#).

```
{
 this.frequence = frequence;
}
```

**9.7.3.19 void com.ttpa.iris.ttpamobile.Seance.setId ( int *id* )**

Méthode setId mutateur de l'attribut id.

**Paramètres**

|           |                        |
|-----------|------------------------|
| <i>id</i> | étant l'id à affecter. |
|-----------|------------------------|

Références [com.ttpa.iris.ttpamobile.Seance.id](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance\(\)](#).

```
{ this.id = id; }
```

**9.7.3.20 void com.ttpa.iris.ttpamobile.Seance.setIdJoueur ( int *idJoueur* )**

Références [com.ttpa.iris.ttpamobile.Seance.idJoueur](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#).

```
{
 this.idJoueur = idJoueur;
}
```

**9.7.3.21 void com.ttpa.iris.ttpamobile.Seance.setIntensiteEffet ( int *intensiteEffet* )**

Méthode setIntensiteEffet mutateur de l'attribut intensiteEffet.

**Paramètres**

|                        |                                          |
|------------------------|------------------------------------------|
| <i>intensite-Effet</i> | étant l'intensité de l'effet à affecter. |
|------------------------|------------------------------------------|

Références [com.ttpa.iris.ttpamobile.Seance.intensiteEffet](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance\(\)](#).

```
{
```

```
 this.intensiteEffet = intensiteEffet;
}
```

#### 9.7.3.22 `void com.ttpa.iris.ttpamobile.Seance.setNombreBalles ( int nombreBalles )`

Méthode `setNombreBalles` mutateur de l'attribut `nombreBalles`.

##### Paramètres

|                      |                                       |
|----------------------|---------------------------------------|
| <i>nombre-Balles</i> | étant le nombre de balles à affecter. |
|----------------------|---------------------------------------|

Références [com.ttpa.iris.ttpamobile.Seance.nombreBalles](#).

Référéncé par [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance\(\)](#).

```
{
 this.nombreBalles = nombreBalles;
}
```

#### 9.7.3.23 `void com.ttpa.iris.ttpamobile.Seance.setPuissance ( int puissance )`

Références [com.ttpa.iris.ttpamobile.Seance.puissance](#).

Référéncé par [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance\(\)](#).

```
{
 this.puissance = puissance;
}
```

#### 9.7.3.24 `void com.ttpa.iris.ttpamobile.Seance.setRotation ( int rotation )`

Références [com.ttpa.iris.ttpamobile.Seance.rotation](#).

Référéncé par [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance\(\)](#).

```
{
 this.rotation = rotation;
}
```

#### 9.7.3.25 `void com.ttpa.iris.ttpamobile.Seance.setTauxReussite ( float tauxReussite )`

Méthode `setTauxReussite` mutateur de l'attribut `tauxReussite`.

##### Paramètres

|                      |                                       |
|----------------------|---------------------------------------|
| <i>taux-Reussite</i> | étant le taux de réussite à affecter. |
|----------------------|---------------------------------------|

Références [com.ttpa.iris.ttpamobile.Seance.tauxReussite](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.calculerReussiteSeance\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance\(\)](#).

```
{ this.tauxReussite = tauxReussite; }
```

#### 9.7.3.26 void com.ttpa.iris.ttpamobile.Seance.setZoneObjectif ( int zoneObjectif )

Références [com.ttpa.iris.ttpamobile.Seance.zoneObjectif](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#).

```
{
 this.zoneObjectif = zoneObjectif;
}
```

#### 9.7.3.27 void com.ttpa.iris.ttpamobile.Seance.setZoneRobot ( int zoneRobot )

Références [com.ttpa.iris.ttpamobile.Seance.zoneRobot](#).

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#).

```
{
 this.zoneRobot = zoneRobot;
}
```

#### 9.7.3.28 String com.ttpa.iris.ttpamobile.Seance.toString ( )

Méthode toString permettant la visualisation des caractéristiques de la séance.

Renvoie

les caractéristiques de la séance.

Références [com.ttpa.iris.ttpamobile.Seance.dateDebut](#), [com.ttpa.iris.ttpamobile.Seance.dateFin](#), [com.ttpa.iris.ttpamobile.Seance.effet](#), [com.ttpa.iris.ttpamobile.Seance.frequence](#), [com.ttpa.iris.ttpamobile.Seance.idJoueur](#), [com.ttpa.iris.ttpamobile.Seance.intensiteEffet](#), [com.ttpa.iris.ttpamobile.Seance.nombreBalles](#), [com.ttpa.iris.ttpamobile.Seance.puissance](#), [com.ttpa.iris.ttpamobile.Seance.rotation](#), [com.ttpa.iris.ttpamobile.Seance.tauxReussite](#), [com.ttpa.iris.ttpamobile.Seance.zoneObjectif](#), et [com.ttpa.iris.ttpamobile.Seance.zoneRobot](#).

```
{
 return "Frequence : " + frequence + "\nNombre balles : " + nombreBalles
 + "\nEffet : " + effet + "\nIntensité effet : " + intensiteEffet + "\nTaux
 réussite : " + tauxReussite + "\nPuissance : " + puissance + "\nRotation robot : " +
 rotation + "\nZone objectif : " + zoneObjectif + "\nZone robot : " + zoneRobot
 + "\nDate début : " + dateDebut + "\nDate fin : " + dateFin + "\nID joueur : " +
 idJoueur;
}
```

### 9.7.4 Documentation des données membres

#### 9.7.4.1 String com.ttpa.iris.ttpamobile.Seance.dateDebut [private]

Référencé par [com.ttpa.iris.ttpamobile.Seance.getDateDebut\(\)](#), [com.ttpa.iris.ttpamobile.Seance.Seance\(\)](#), [com.ttpa.iris.ttpamobile.Seance.setDateDebut\(\)](#), et [com.ttpa.iris.ttpamobile.Seance.toString\(\)](#).

**9.7.4.2 String `com.ttpa.iris.ttpamobile.Seance.dateFin` [private]**

Référencé par `com.ttpa.iris.ttpamobile.Seance.getDateFin()`, `com.ttpa.iris.ttpamobile.Seance.Seance()`, `com.ttpa.iris.ttpamobile.Seance.setDateFin()`, et `com.ttpa.iris.ttpamobile.Seance.toString()`.

**9.7.4.3 String `com.ttpa.iris.ttpamobile.Seance.effet` [private]**

Référencé par `com.ttpa.iris.ttpamobile.Seance.getEffet()`, `com.ttpa.iris.ttpamobile.Seance.Seance()`, `com.ttpa.iris.ttpamobile.Seance.setEffet()`, et `com.ttpa.iris.ttpamobile.Seance.toString()`.

**9.7.4.4 int `com.ttpa.iris.ttpamobile.Seance.frequence` [private]**

Référencé par `com.ttpa.iris.ttpamobile.Seance.getFrequence()`, `com.ttpa.iris.ttpamobile.Seance.Seance()`, `com.ttpa.iris.ttpamobile.Seance.setFrequence()`, et `com.ttpa.iris.ttpamobile.Seance.toString()`.

**9.7.4.5 int `com.ttpa.iris.ttpamobile.Seance.id` [private]**

Référencé par `com.ttpa.iris.ttpamobile.Seance.getId()`, et `com.ttpa.iris.ttpamobile.Seance.setId()`.

**9.7.4.6 int `com.ttpa.iris.ttpamobile.Seance.idJoueur` [private]**

Référencé par `com.ttpa.iris.ttpamobile.Seance.getIdJoueur()`, `com.ttpa.iris.ttpamobile.Seance.Seance()`, `com.ttpa.iris.ttpamobile.Seance.setIdJoueur()`, et `com.ttpa.iris.ttpamobile.Seance.toString()`.

**9.7.4.7 int `com.ttpa.iris.ttpamobile.Seance.intensiteEffet` [private]**

Référencé par `com.ttpa.iris.ttpamobile.Seance.getIntensiteEffet()`, `com.ttpa.iris.ttpamobile.Seance.Seance()`, `com.ttpa.iris.ttpamobile.Seance.setIntensiteEffet()`, et `com.ttpa.iris.ttpamobile.Seance.toString()`.

**9.7.4.8 int `com.ttpa.iris.ttpamobile.Seance.nombreBalles` [private]**

Référencé par `com.ttpa.iris.ttpamobile.Seance.getNombreBalles()`, `com.ttpa.iris.ttpamobile.Seance.Seance()`, `com.ttpa.iris.ttpamobile.Seance.setNombreBalles()`, et `com.ttpa.iris.ttpamobile.Seance.toString()`.

**9.7.4.9 int `com.ttpa.iris.ttpamobile.Seance.puissance` [private]**

Référencé par `com.ttpa.iris.ttpamobile.Seance.getPuissance()`, `com.ttpa.iris.ttpamobile.Seance.Seance()`, `com.ttpa.iris.ttpamobile.Seance.setPuissance()`, et `com.ttpa.iris.ttpamobile.Seance.toString()`.

**9.7.4.10 int `com.ttpa.iris.ttpamobile.Seance.rotation` [private]**

Référencé par `com.ttpa.iris.ttpamobile.Seance.getRotation()`, `com.ttpa.iris.ttpamobile.Seance.Seance()`, `com.ttpa.iris.ttpamobile.Seance.setRotation()`, et `com.ttpa.iris.ttpamobile.Seance.toString()`.

**9.7.4.11** `float com.ttpa.iris.ttpamobile.Seance.tauxReussite` [private]

Référencé par `com.ttpa.iris.ttpamobile.Seance.getTauxReussite()`, `com.ttpa.iris.ttpamobile.Seance.Seance()`, `com.ttpa.iris.ttpamobile.Seance.setTauxReussite()`, et `com.ttpa.iris.ttpamobile.Seance.toString()`.

**9.7.4.12** `int com.ttpa.iris.ttpamobile.Seance.zoneObjectif` [private]

Référencé par `com.ttpa.iris.ttpamobile.Seance.getZoneObjectif()`, `com.ttpa.iris.ttpamobile.Seance.Seance()`, `com.ttpa.iris.ttpamobile.Seance.setZoneObjectif()`, et `com.ttpa.iris.ttpamobile.Seance.toString()`.

**9.7.4.13** `int com.ttpa.iris.ttpamobile.Seance.zoneRobot` [private]

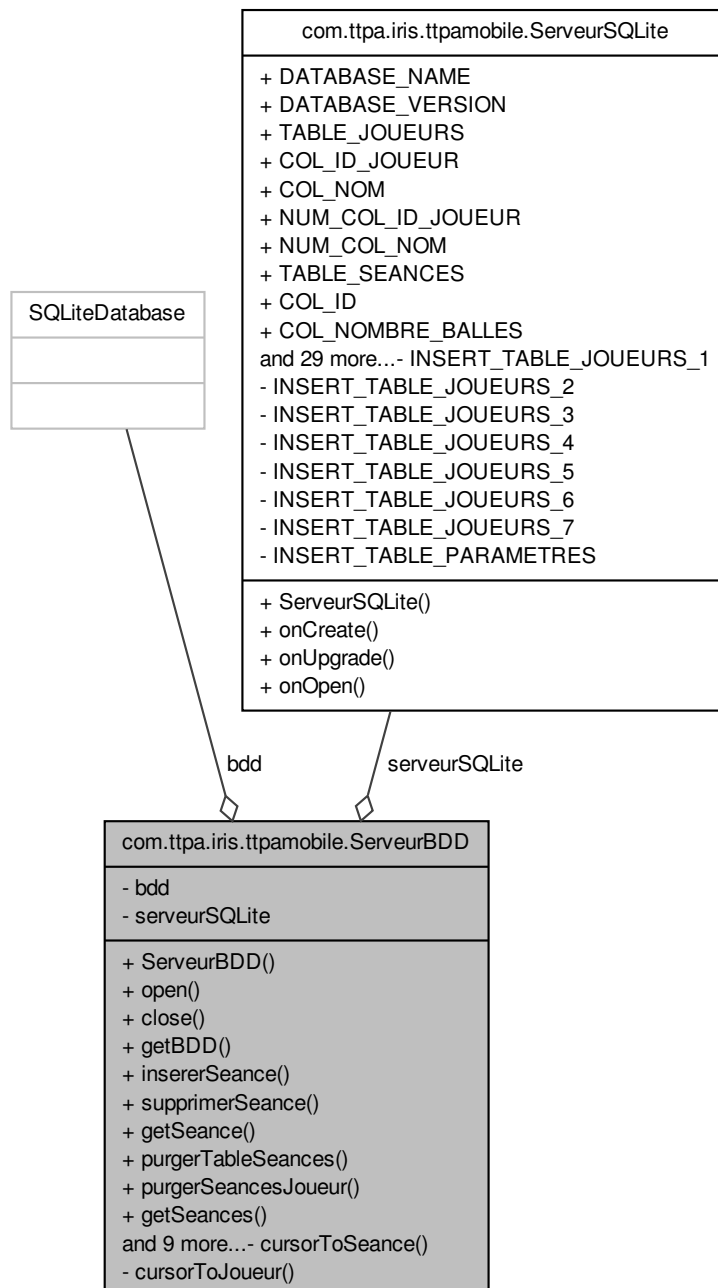
Référencé par `com.ttpa.iris.ttpamobile.Seance.getZoneRobot()`, `com.ttpa.iris.ttpamobile.Seance.Seance()`, `com.ttpa.iris.ttpamobile.Seance.setZoneRobot()`, et `com.ttpa.iris.ttpamobile.Seance.toString()`.

La documentation de cette classe a été générée à partir du fichier suivant :

- [Seance.java](#)

## 9.8 Référence de la classe com.ttpa.iris.ttpamobile.ServeurBDD

Graphe de collaboration de com.ttpa.iris.ttpamobile.ServeurBDD :



## Fonctions membres publiques

- [ServeurBDD](#) (Context context)
  - void [open](#) ()
  - void [close](#) ()
  - SQLiteDatabase [getBDD](#) ()
  - long [insérerSeance](#) ([Seance](#) seance)
  - int [supprimerSeance](#) (int id)
  - [Seance](#) [getSeance](#) (int id)
  - void [purgerTableSeances](#) ()
  - int [purgerSeancesJoueur](#) (int idJoueur)
  - List< [Seance](#) > [getSeances](#) ()
  - List< [Seance](#) > [getSeances](#) (int idJoueur)
  - long [insérerJoueur](#) ([Joueur](#) joueur)
  - int [supprimerJoueur](#) (int id)
  - int [supprimerJoueur](#) (String nom)
  - [Joueur](#) [getJoueur](#) (int id)
  - [Joueur](#) [getJoueur](#) (String nom)
  - void [purgerTableJoueurs](#) ()
  - List< [Joueur](#) > [getJoueurs](#) ()
  - int [getIdJoueurParametres](#) ()
  - int [setIdJoueurParametres](#) (int idJoueur)
- Permet de mettre à jour un enregistrement de la table.*

## Fonctions membres privées

- [Seance](#) [cursorToSeance](#) (Cursor c, boolean one)
- [Joueur](#) [cursorToJoueur](#) (Cursor c, boolean one)

## Attributs privés

- SQLiteDatabase [bdd](#) = null
- [ServeurSQLite](#) [serveurSQLite](#) = null

## 9.8.1 Description détaillée

Classe [ServeurBDD](#) définissant les caractéristiques et le comportement d'un serveur de base de données.

## 9.8.2 Documentation des constructeurs et destructeur

## 9.8.2.1 com.ttpa.iris.ttpamobile.ServeurBDD.ServeurBDD ( Context context )

Méthode [ServeurBDD](#) constructeur de la classe [ServeurBDD](#).

## Paramètres

|                |  |
|----------------|--|
| <i>context</i> |  |
|----------------|--|

Références [com.ttpa.iris.ttpamobile.ServeurBDD.serveurSQLite](#).

```

{
 // cn crée la BDD et ses tables
 serveurSQLite = new ServeurSQLite(context);
}

```

## 9.8.3 Documentation des fonctions membres

## 9.8.3.1 void com.ttpa.iris.ttpamobile.ServeurBDD.close ( )

Méthode close fermant la base de données.

Références [com.ttpa.iris.ttpamobile.ServeurBDD.bdd](#).

```
{
 if (bdd != null)
 if (bdd.isOpen())
 bdd.close();
}
```

9.8.3.2 Joueur com.ttpa.iris.ttpamobile.ServeurBDD.cursorToJoueur ( Cursor *c*, boolean *one* ) [private]

Méthode cursorToJoueur permettant de convertir un curseur en un objet de type [Joueur](#).

## Paramètres

|            |  |
|------------|--|
| <i>c</i>   |  |
| <i>one</i> |  |

## Renvoie

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.NUM\\_COL\\_ID\\_JOUEUR](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.NUM\\_COL\\_NOM](#), [com.ttpa.iris.ttpamobile.Joueur.setId\(\)](#), et [com.ttpa.iris.ttpamobile.Joueur.setNom\(\)](#).

```
{
 if (c.getCount() == 0)
 return null;

 if (one == true)
 c.moveToFirst();

 Joueur joueur = new Joueur();

 joueur.setId(c.getInt(ServeurSQLite.NUM_COL_ID_JOUEUR));
 joueur.setNom(c.getString(ServeurSQLite.NUM_COL_NOM));

 if (one == true)
 c.close();

 return joueur;
}
```

9.8.3.3 Seance com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance ( Cursor *c*, boolean *one* ) [private]

Méthode cursorToSeance permettant de convertir un curseur en un objet de type [Seance](#).



## Paramètres

|            |  |
|------------|--|
| <i>c</i>   |  |
| <i>one</i> |  |

## Renvoi

Références `com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_DATE_DEBUT`, `com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_DATE_FIN`, `com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_EFFET`, `com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_FREQUENCE`, `com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_ID`, `com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_ID_JOUEUR_SEANCE`, `com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_INTENSITE_EFFET`, `com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_NOMBRE_BALLES`, `com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_PUISSANCE`, `com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_ROTATION`, `com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_TAUX_REUSSITE`, `com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_ZONE_OBJECTIF`, `com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_ZONE_ROBOT`, `com.ttpa.iris.ttpamobile.Seance.setDateDebut()`, `com.ttpa.iris.ttpamobile.Seance.setDateFin()`, `com.ttpa.iris.ttpamobile.Seance.setEffet()`, `com.ttpa.iris.ttpamobile.Seance.setFrequence()`, `com.ttpa.iris.ttpamobile.Seance.setId()`, `com.ttpa.iris.ttpamobile.Seance.setIdJoueur()`, `com.ttpa.iris.ttpamobile.Seance.setIntensiteEffet()`, `com.ttpa.iris.ttpamobile.Seance.setNombreBalles()`, `com.ttpa.iris.ttpamobile.Seance.setPuissance()`, `com.ttpa.iris.ttpamobile.Seance.setRotation()`, `com.ttpa.iris.ttpamobile.Seance.setTauxReussite()`, `com.ttpa.iris.ttpamobile.Seance.setZoneObjectif()`, et `com.ttpa.iris.ttpamobile.Seance.setZoneRobot()`.

```
{
 if (c.getCount() == 0)
 return null;

 if (one == true)
 c.moveToFirst();

 Seance seance = new Seance();

 seance.setId(c.getInt(ServeurSQLite.NUM_COL_ID));
 seance.setNombreBalles(c.getInt(ServeurSQLite.NUM_COL_NOMBRE_BALLES));
 seance.setFrequence(c.getInt(ServeurSQLite.NUM_COL_FREQUENCE));
 seance.setEffet(c.getString(ServeurSQLite.NUM_COL_EFFET));
 seance.setIntensiteEffet(c.getInt(ServeurSQLite.NUM_COL_INTENSITE_EFFET));
 seance.setPuissance(c.getInt(ServeurSQLite.NUM_COL_PUISSANCE));
 seance.setRotation(c.getInt(ServeurSQLite.NUM_COL_ROTATION));
 seance.setZoneObjectif(c.getInt(ServeurSQLite.NUM_COL_ZONE_OBJECTIF));
 seance.setZoneRobot(c.getInt(ServeurSQLite.NUM_COL_ZONE_ROBOT));
 seance.setTauxReussite(c.getInt(ServeurSQLite.NUM_COL_TAUX_REUSSITE));
 seance.setDateDebut(c.getString(ServeurSQLite.NUM_COL_DATE_DEBUT));
 seance.setDateFin(c.getString(ServeurSQLite.NUM_COL_DATE_FIN));
 seance.setIdJoueur(c.getInt(ServeurSQLite.NUM_COL_ID_JOUEUR_SEANCE));

 if (one == true)
 c.close();

 return seance;
}
```

## 9.8.3.4 SQLiteDatabase com.ttpa.iris.ttpamobile.ServeurBDD.getBDD ( )

Méthode getBDD accesseur de l'attribut bdd.

Renvoie

Références [com.ttpa.iris.ttpamobile.ServeurBDD.bdd](#).

```
{
 return bdd;
}
```

## 9.8.3.5 int com.ttpa.iris.ttpamobile.ServeurBDD.getIdJoueurParametres ( )

Renvoie

int l'id du joueur

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID\\_JOUEUR](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_PARAMETRES](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs\(\)](#), [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.demarrerSeance\(\)](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.recupererInformationsJoueur\(\)](#).

```
{
 Cursor c = bdd.query(ServeurSQLite.TABLE_PARAMETRES, new String[] {
 ServeurSQLite.COL_ID_JOUEUR}, ServeurSQLite.COL_ID_PARAMETRE + " = '1'", null, null,
 null, null);

 if (c.getCount() == 0)
 return 0;

 c.moveToFirst();

 int idJoueur = c.getInt(0);

 c.close();

 return idJoueur;
}
```

## 9.8.3.6 Joueur com.ttpa.iris.ttpamobile.ServeurBDD.getJoueur ( int id )

Méthode getJoueur permettant l'accès à un joueur grâce à son attribut id.

Paramètres

|           |                                   |
|-----------|-----------------------------------|
| <i>id</i> | étant l'id du joueur à retourner. |
|-----------|-----------------------------------|

Renvoie

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID\\_JOUEUR](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_NOM](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_JOUEURS](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs\(\)](#), et [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.recupererInformationsJoueur\(\)](#).

```
{
 Cursor c = bdd.query(ServeurSQLite.TABLE_JOUEURS, new String[] {
 ServeurSQLite.COL_ID_JOUEUR, ServeurSQLite.COL_NOM}, ServeurSQLite.COL_ID_JOUEUR + " = "
 + id, null, null, null, null);

 return cursorToJoueur(c, true);
}
```

#### 9.8.3.7 Joueur [com.ttpa.iris.ttpamobile.ServeurBDD.getJoueur \( String nom \)](#)

Méthode getJoueur permettant l'accès à un joueur grâce à son nom.

Paramètres

|            |                                      |
|------------|--------------------------------------|
| <i>nom</i> | String le nom du joueur à retourner. |
|------------|--------------------------------------|

Renvoie

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID\\_JOUEUR](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_NOM](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_JOUEURS](#).

```
{
 Cursor c = bdd.query(ServeurSQLite.TABLE_JOUEURS, new String[] {
 ServeurSQLite.COL_ID_JOUEUR, ServeurSQLite.COL_NOM}, ServeurSQLite.COL_NOM + " = '" +
 nom + "'", null, null, null, null);

 return cursorToJoueur(c, true);
}
```

#### 9.8.3.8 List<Joueur> [com.ttpa.iris.ttpamobile.ServeurBDD.getJoueurs \( \)](#)

Méthode getJoueurs retournant tous les joueurs présents dans la table des joueurs.

Renvoie

liste d'objets [Joueur](#).

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID\\_JOUEUR](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_NOM](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_JOUEURS](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs\(\)](#).

```

{
 List<Joueur> joueurs = new ArrayList<Joueur>();

 Cursor cursor = bdd.query(ServeurSQLite.TABLE_JOUEURS, new String[] {
 ServeurSQLite.COL_ID_JOUEUR, ServeurSQLite.COL_NOM}, null, null, null, null, null)
 ;

 cursor.moveToFirst();
 while (!cursor.isAfterLast())
 {
 Joueur joueur = cursorToJoueur(cursor, false);
 joueurs.add(joueur);
 cursor.moveToNext();
 }

 cursor.close();

 return joueurs;
}

```

#### 9.8.3.9 Seance com.ttpa.iris.ttpamobile.ServeurBDD.getSeance ( int id )

Méthode getSeance permettant l'accès à une séance grâce à son attribut id.

##### Paramètres

|           |                                      |
|-----------|--------------------------------------|
| <i>id</i> | étant l'id de la séance à retourner. |
|-----------|--------------------------------------|

##### Renvoie

Références [com.ttpa.iris.ttpamobile.ServeurBDD.bdd](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_DATE\\_DEBUT](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_DATE\\_FIN](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_EFFET](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_FREQUENCE](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID\\_JOUEUR](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_INTENSITE\\_EFFET](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_NOMBRE\\_BALLES](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_PUISSANCE](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ROTATION](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_TAUX\\_REUSSITE](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ZONE\\_OBJECTIF](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ZONE\\_ROBOT](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_SEANCES](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.afficherBoiteDialogueDetailsSeance\(\)](#).

```

{
 Cursor c = bdd.query(ServeurSQLite.TABLE_SEANCES, new String[] {
 ServeurSQLite.COL_ID, ServeurSQLite.COL_NOMBRE_BALLES, ServeurSQLite.COL_FREQUENCE,
 ServeurSQLite.COL_EFFET, ServeurSQLite.COL_INTENSITE_EFFET, ServeurSQLite.
 COL_PUISSANCE, ServeurSQLite.COL_ROTATION, ServeurSQLite.COL_ZONE_OBJECTIF, ServeurSQLite.
 COL_ZONE_ROBOT, ServeurSQLite.COL_TAUX_REUSSITE, ServeurSQLite.COL_DATE_DEBUT,
 ServeurSQLite.COL_DATE_FIN, ServeurSQLite.COL_ID_JOUEUR}, ServeurSQLite.COL_ID +
 " = " + id, null, null, null, null);

 return cursorToSeance(c, true);
}

```

### 9.8.3.10 List<Seance> com.ttpa.iris.ttpamobile.ServeurBDD.getSeances ( )

Méthode getSeances retournant toutes les séances présentes dans la table des séances.

#### Renvoie

liste d'objets [Seance](#).

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_DATE\\_DEBUT](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_DATE\\_FIN](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_EFFET](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_FREQUENCE](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID\\_JOUEUR](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_INTENSITE\\_EFFET](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_NOMBRE\\_BALLES](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_PUISSANCE](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ROTATION](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_TAUX\\_REUSSITE](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ZONE\\_OBJECTIF](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ZONE\\_ROBOT](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_SEANCES](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.modifierIHMAfficherSeancesJoueur\(\)](#).

```
{
 List<Seance> seances = new ArrayList<Seance>();

 Cursor cursor = bdd.query(ServeurSQLite.TABLE_SEANCES, new String[] {
 ServeurSQLite.COL_ID, ServeurSQLite.COL_FREQUENCE, ServeurSQLite.COL_NOMBRE_BALLES
 , ServeurSQLite.COL_EFFET, ServeurSQLite.COL_INTENSITE_EFFET, ServeurSQLite.
 COL_PUISSANCE, ServeurSQLite.COL_ROTATION, ServeurSQLite.COL_ZONE_OBJECTIF,
 ServeurSQLite.COL_ZONE_ROBOT, ServeurSQLite.COL_TAUX_REUSSITE, ServeurSQLite.
 COL_DATE_DEBUT, ServeurSQLite.COL_DATE_FIN, ServeurSQLite.COL_ID_JOUEUR}, null, null, null,
 null, null);

 cursor.moveToFirst();
 while (!cursor.isAfterLast())
 {
 Seance seance = cursorToSeance(cursor, false);
 seances.add(seance);
 cursor.moveToNext();
 }

 cursor.close();

 return seances;
}
```

### 9.8.3.11 List<Seance> com.ttpa.iris.ttpamobile.ServeurBDD.getSeances ( int idJoueur )

Méthode getSeances retournant toutes les séances présentes dans la table des séances ayant pour id idJoueur.

#### Paramètres

|                 |  |
|-----------------|--|
| <i>idJoueur</i> |  |
|-----------------|--|

**Renvoie**

liste d'objets [Seance](#).

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_DATE\\_DEBUT](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_DATE\\_FIN](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_EFFET](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_FREQUENCE](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID\\_JOUEUR](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_INTENSITE\\_EFFET](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_NOMBRE\\_BALLES](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_PUISSANCE](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ROTATION](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_TAUX\\_REUSSITE](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ZONE\\_OBJECTIF](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ZONE\\_ROBOT](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_SEANCES](#).

```
{
 List<Seance> seances = new ArrayList<Seance>();

 Cursor cursor = bdd.query(ServeurSQLite.TABLE_SEANCES, new String[] {
 ServeurSQLite.COL_ID, ServeurSQLite.COL_FREQUENCE, ServeurSQLite.COL_NOMBRE_BALLES
 , ServeurSQLite.COL_EFFET, ServeurSQLite.COL_INTENSITE_EFFET, ServeurSQLite.
 COL_PUISSANCE, ServeurSQLite.COL_ROTATION, ServeurSQLite.COL_ZONE_OBJECTIF,
 ServeurSQLite.COL_ZONE_ROBOT, ServeurSQLite.COL_TAUX_REUSSITE, ServeurSQLite.
 COL_DATE_DEBUT, ServeurSQLite.COL_DATE_FIN, ServeurSQLite.COL_ID_JOUEUR}, ServeurSQLite.
 COL_ID_JOUEUR + " = " + idJoueur, null, null, null, null);

 cursor.moveToFirst();
 while (!cursor.isAfterLast())
 {
 Seance seance = cursorToSeance(cursor, false);
 seances.add(seance);
 cursor.moveToNext();
 }

 cursor.close();

 return seances;
}
```

**9.8.3.12 long com.ttpa.iris.ttpamobile.ServeurBDD.insererJoueur ( Joueur joueur )**

Méthode insererJoueur permettant l'insertion d'un joueur.

**Paramètres**

|               |                                                    |
|---------------|----------------------------------------------------|
| <i>joueur</i> | étant le joueur à insérer dans la base de données. |
|---------------|----------------------------------------------------|

**Renvoie**

l'id dans la base de données.

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_NOM](#), [com.ttpa.iris.ttpamobile.Joueur.getNom\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_JOUEURS](#).

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.ajouterJoueur\(\)](#).

```
{
 ContentValues values = new ContentValues();
```

```
//values.put(ServeurSQLite.COL_ID_JOUEUR, joueur.getId());
values.put(ServeurSQLite.COL_NOM, joueur.getNom());

return bdd.insert(ServeurSQLite.TABLE_JOUEURS, null, values);
}
```

### 9.8.3.13 long com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance ( Seance seance )

Méthode insererSeance permettant l'insertion d'une séance dans la base de données.

#### Paramètres

|               |                                                    |
|---------------|----------------------------------------------------|
| <i>seance</i> | étant la séance à insérer dans la base de données. |
|---------------|----------------------------------------------------|

#### Renvoie

l'id de la séance dans la base de données.

Références [com.ttpa.iris.ttpamobile.ServeurBDD.bdd](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_DATE\\_DEBUT](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_DATE\\_FIN](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_EFFECT](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_FREQUENCE](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID\\_JOUEUR](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_INTENSITE\\_EFFECT](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_NOMBRE\\_BALLES](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_PUISSANCE](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ROTATION](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_TAUX\\_REUSSITE](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ZONE\\_OBJECTIF](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ZONE\\_ROBOT](#), [com.ttpa.iris.ttpamobile.Seance.getDateDebut\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getDateFin\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getEffet\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getFrequence\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getIdJoueur\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getIntensiteEffet\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getNombreBalles\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getPuissance\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getRotation\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getTauxReussite\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getZoneObjectif\(\)](#), [com.ttpa.iris.ttpamobile.Seance.getZoneRobot\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_SEANCES](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.arreterSeance\(\)](#).

```
{
 ContentValues values = new ContentValues();

 values.put(ServeurSQLite.COL_FREQUENCE, seance.getFrequence());
 values.put(ServeurSQLite.COL_NOMBRE_BALLES, seance.getNombreBalles());
 values.put(ServeurSQLite.COL_EFFECT, seance.getEffet());
 values.put(ServeurSQLite.COL_INTENSITE_EFFECT, seance.getIntensiteEffet());
 values.put(ServeurSQLite.COL_PUISSANCE, seance.getPuissance());
 values.put(ServeurSQLite.COL_ROTATION, seance.getRotation());
 values.put(ServeurSQLite.COL_ZONE_OBJECTIF, seance.getZoneObjectif());
 values.put(ServeurSQLite.COL_ZONE_ROBOT, seance.getZoneRobot());
 values.put(ServeurSQLite.COL_TAUX_REUSSITE, seance.getTauxReussite());
 values.put(ServeurSQLite.COL_DATE_DEBUT, seance.getDateDebut());
 values.put(ServeurSQLite.COL_DATE_FIN, seance.getDateFin());
 values.put(ServeurSQLite.COL_ID_JOUEUR, seance.getIdJoueur());

 return bdd.insert(ServeurSQLite.TABLE_SEANCES, null, values);
}
```

**9.8.3.14 void com.ttpa.iris.ttpamobile.ServeurBDD.open ( )**

Méthode open ouvrant la base de données en écriture.

Références [com.ttpa.iris.ttpamobile.ServeurBDD.bdd](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.serveurSQLite](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.connectionBaseDeDonnees\(\)](#), et [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.connexionBaseDeDonnees\(\)](#).

```
{
 // on ouvre la BDD en écriture
 if (bdd == null)
 bdd = serveurSQLite.getWritableDatabase();
}
```

**9.8.3.15 int com.ttpa.iris.ttpamobile.ServeurBDD.purgerSeancesJoueur ( int idJoueur )**

Méthode purgerSeancesJoueur permettant la suppression des séances d'un joueur.

Paramètres

|                 |                                                         |
|-----------------|---------------------------------------------------------|
| <i>idJoueur</i> | étant l'id du joueur auquel on doit purger les séances. |
|-----------------|---------------------------------------------------------|

Renvoie

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID\\_JOUEUR](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_SEANCES](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.purgerSeancesJoueur\(\)](#).

```
{
 return bdd.delete(ServeurSQLite.TABLE_SEANCES, ServeurSQLite.
 COL_ID_JOUEUR + " = " + idJoueur, null);
}
```

**9.8.3.16 void com.ttpa.iris.ttpamobile.ServeurBDD.purgerTableJoueurs ( )**

Méthode purgerTableJoueurs permettant la purge (suppression totale) de la table des joueurs.

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.CREATE\\_BDD\\_JOUEURS](#).

```
{
 bdd.execSQL("DROP TABLE IF EXISTS " + serveurSQLite.TABLE_JOUEURS); //
 Supprimer la table
 bdd.execSQL(ServeurSQLite.CREATE_BDD_JOUEURS); // Recréer la table
}
```

**9.8.3.17 void com.ttpa.iris.ttpamobile.ServeurBDD.purgerTableSeances ( )**

Méthode purgerTableSeances permettant la purge (suppression totale) de la table des séances.



Références [com.ttpa.iris.ttpamobile.ServeurSQLite.CREATE\\_BDD\\_SEANCES](#).

```
{
 bdd.execSQL("DROP TABLE IF EXISTS " + serveurSQLite.TABLE_SEANCES); //
 Supprimer la table
 bdd.execSQL(ServeurSQLite.CREATE_BDD_SEANCES); // Recréer la table
}
```

**9.8.3.18** `int com.ttpa.iris.ttpamobile.ServeurBDD.setIdJoueurParametres ( int idJoueur )`

#### Paramètres

|                 |                                        |
|-----------------|----------------------------------------|
| <i>idJoueur</i> | int représente l'identifiant du joueur |
|-----------------|----------------------------------------|

#### Renvoie

un int qui permet de savoir si la mise à jour de l'enregistrement a réussi

Référencé par [com.ttpa.iris.ttpamobile.IHMEcranPrincipal.creerListeJoueurs\(\)](#).

```
{
 ContentValues values = new ContentValues();
 values.put("ID_JOUEUR", idJoueur);

 return bdd.update("table_parametres", values, "ID_PARAMETRE = " + 1,
 null);
}
```

**9.8.3.19** `int com.ttpa.iris.ttpamobile.ServeurBDD.supprimerJoueur ( int id )`

Méthode `supprimerJoueur` permettant la suppression d'un joueur.

#### Paramètres

|           |                                   |
|-----------|-----------------------------------|
| <i>id</i> | étant l'id du joueur à supprimer. |
|-----------|-----------------------------------|

#### Renvoie

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID\\_JOUEUR](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_JOUEURS](#).

```
{
 return bdd.delete(ServeurSQLite.TABLE_JOUEURS, ServeurSQLite.
 COL_ID_JOUEUR + " = " + id, null);
}
```

**9.8.3.20** `int com.ttpa.iris.ttpamobile.ServeurBDD.supprimerJoueur ( String nom )`

Méthode `supprimerJoueur` permettant la suppression d'un joueur.

#### Paramètres

|            |                                      |
|------------|--------------------------------------|
| <i>nom</i> | String le nom du joueur à supprimer. |
|------------|--------------------------------------|

Renvoie

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_NOM](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_JOUEURS](#).

```
{
 return bdd.delete(ServeurSQLite.TABLE_JOUEURS, ServeurSQLite.COL_NOM +
 " = '" + nom + "'", null);
}
```

#### 9.8.3.21 `int com.ttpa.iris.ttpamobile.ServeurBDD.supprimerSeance ( int id )`

Méthode `supprimerSeance` permettant la suppression d'une séance dans la base de données.

Paramètres

|                 |                                      |
|-----------------|--------------------------------------|
| <code>id</code> | étant l'id de la séance à supprimer. |
|-----------------|--------------------------------------|

Renvoie

Références [com.ttpa.iris.ttpamobile.ServeurBDD.bdd](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_SEANCES](#).

Référéncé par [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances.supprimerSeanceSelectionnee\(\)](#).

```
{
 return bdd.delete(ServeurSQLite.TABLE_SEANCES, ServeurSQLite.COL_ID + "
 = " + id, null);
}
```

### 9.8.4 Documentation des données membres

#### 9.8.4.1 `SQLiteDatabase com.ttpa.iris.ttpamobile.ServeurBDD.bdd = null`

`[private]`

Référéncé par [com.ttpa.iris.ttpamobile.ServeurBDD.close\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getBDD\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getSeance\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.open\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.supprimerSeance\(\)](#).

#### 9.8.4.2 `ServeurSQLite com.ttpa.iris.ttpamobile.ServeurBDD.serveurSQLite = null`

`[private]`

Référéncé par [com.ttpa.iris.ttpamobile.ServeurBDD.open\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.ServeurBDD\(\)](#).

La documentation de cette classe a été générée à partir du fichier suivant :

– [ServeurBDD.java](#)

## 9.9 Référence de la classe com.ttpa.iris.ttpamobile.ServeurSQLite

### Fonctions membres publiques

- [ServeurSQLite](#) (Context context)
  - void [onCreate](#) (SQLiteDatabase db)
  - void [onUpgrade](#) (SQLiteDatabase db, int oldVersion, int newVersion)
  - void [onOpen](#) (SQLiteDatabase db)
- Ajoute les droits en lecture et en écriture à la base de données lors de son ouverture.*

### Attributs publics statiques

- static final String [DATABASE\\_NAME](#) = "ttpa\_mobile.db"
- static final int [DATABASE\\_VERSION](#) = 1
- static final String [TABLE\\_JOUEURS](#) = "table\_joueurs"
- static final String [COL\\_ID\\_JOUEUR](#) = "ID\_JOUEUR"
- static final String [COL\\_NOM](#) = "NOM"
- static final int [NUM\\_COL\\_ID\\_JOUEUR](#) = 0
- static final int [NUM\\_COL\\_NOM](#) = 1
- static final String [TABLE\\_SEANCES](#) = "table\_seances"
- static final String [COL\\_ID](#) = "ID\_SEANCE"
- static final String [COL\\_NOMBRE\\_BALLES](#) = "NOMBRE\_BALLES"
- static final String [COL\\_FREQUENCE](#) = "FREQUENCE"
- static final String [COL\\_EFFET](#) = "EFFET"
- static final String [COL\\_INTENSITE\\_EFFET](#) = "INTENSITE\_EFFET"
- static final String [COL\\_PUISSANCE](#) = "PUISSANCE"
- static final String [COL\\_ROTATION](#) = "ROTATION"
- static final String [COL\\_ZONE\\_OBJECTIF](#) = "ZONE\_OBJECTIF"
- static final String [COL\\_ZONE\\_ROBOT](#) = "ZONE\_ROBOT"
- static final String [COL\\_TAUX\\_REUSSITE](#) = "TAUX\_REUSSITE"
- static final String [COL\\_DATE\\_DEBUT](#) = "DATE\_DEBUT"
- static final String [COL\\_DATE\\_FIN](#) = "DATE\_FIN"
- static final int [NUM\\_COL\\_ID](#) = 0
- static final int [NUM\\_COL\\_FREQUENCE](#) = 1
- static final int [NUM\\_COL\\_NOMBRE\\_BALLES](#) = 2
- static final int [NUM\\_COL\\_EFFET](#) = 3
- static final int [NUM\\_COL\\_INTENSITE\\_EFFET](#) = 4
- static final int [NUM\\_COL\\_PUISSANCE](#) = 5
- static final int [NUM\\_COL\\_ROTATION](#) = 6
- static final int [NUM\\_COL\\_ZONE\\_OBJECTIF](#) = 7
- static final int [NUM\\_COL\\_ZONE\\_ROBOT](#) = 8
- static final int [NUM\\_COL\\_TAUX\\_REUSSITE](#) = 9
- static final int [NUM\\_COL\\_DATE\\_DEBUT](#) = 10
- static final int [NUM\\_COL\\_DATE\\_FIN](#) = 11
- static final int [NUM\\_COL\\_ID\\_JOUEUR\\_SEANCE](#) = 12
- static final String [TABLE\\_PARAMETRES](#) = "table\_parametres"
- static final String [COL\\_ID\\_PARAMETRE](#) = "ID\_PARAMETRE"
- static final int [NUM\\_COL\\_ID\\_PARAMETRE](#) = 0
- static final int [NUM\\_COL\\_ID\\_JOUEUR\\_PARAMETRE](#) = 1
- static final String [CREATE\\_BDD\\_JOUEURS](#) = " VARCHAR(255) NOT NULL);"
- static final String [CREATE\\_BDD\\_SEANCES](#) = "CONSTRAINT fk\_seances\_1 FOREIGN KEY (ID\_JOUEUR) REFERENCES table\_joueurs (ID\_JOUEUR) ON DELETE CASCADE);"
- static final String [CREATE\\_BDD\\_PARAMETRES](#) = "CONSTRAINT fk\_parametres\_1 FOREIGN KEY (ID\_JOUEUR) REFERENCES table\_joueurs (ID\_JOUEUR));"

### Attributs privés statiques

- static final String [INSERT\\_TABLE\\_JOUEURS\\_1](#) = "INSERT INTO table\_joueurs(NOM) VALUES('LEGOUT Christophe');"

```

– static final String INSERT_TABLE_JOUEURS_2 = "INSERT INTO table_joueurs(N-
OM) VALUES('MARTINEZ Michel');"
– static final String INSERT_TABLE_JOUEURS_3 = "INSERT INTO table_joueurs(N-
OM) VALUES('LEBESSON Emmanuel');"
– static final String INSERT_TABLE_JOUEURS_4 = "INSERT INTO table_joueurs(N-
OM) VALUES('ELOI Damien');"
– static final String INSERT_TABLE_JOUEURS_5 = "INSERT INTO table_joueurs(N-
OM) VALUES('MATTENET Adrien');"
– static final String INSERT_TABLE_JOUEURS_6 = "INSERT INTO table_joueurs(N-
OM) VALUES('CHILA Patrick');"
– static final String INSERT_TABLE_JOUEURS_7 = "INSERT INTO table_joueurs(N-
OM) VALUES('BEAUMONT Jérôme');"
– static final String INSERT_TABLE_PARAMETRES = "INSERT INTO table_
parametres(ID_PARAMETRE, ID_JOUEUR) VALUES(1, 1);"

```

### 9.9.1 Description détaillée

Classe [ServeurSQLite](#) définissant les caractéristiques de la base de données.

### 9.9.2 Documentation des constructeurs et destructeur

#### 9.9.2.1 com.ttpa.iris.ttpamobile.ServeurSQLite.ServeurSQLite ( Context context )

Méthode [ServeurSQLite](#) constructeur de la classe [ServeurSQLite](#).

##### Paramètres

|                |  |
|----------------|--|
| <i>context</i> |  |
|----------------|--|

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.DATABASE\\_NAME](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.DATABASE\\_VERSION](#).

```

{
 super(context, DATABASE_NAME, null, DATABASE_VERSION);
}

```

### 9.9.3 Documentation des fonctions membres

#### 9.9.3.1 void com.ttpa.iris.ttpamobile.ServeurSQLite.onCreate ( SQLiteDatabase db )

Méthode onCreate appelée à la création de l'objet et permettant l'exécution des requêtes créant les tables de la base de données.

##### Paramètres

|           |  |
|-----------|--|
| <i>db</i> |  |
|-----------|--|

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.CREATE\\_BDD\\_JOUEURS](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.CREATE\\_BDD\\_PARAMETRES](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.CREATE\\_BDD\\_SEANCES](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT\\_TABLE\\_JOUEURS\\_1](#), [com.ttpa.iris.ttpamobile.ServeurSQ-](#)

Lite.INSERT\_TABLE\_JOUEURS\_2, com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT\_TABLE\_JOUEURS\_3, com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT\_TABLE\_JOUEURS\_4, com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT\_TABLE\_JOUEURS\_5, com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT\_TABLE\_JOUEURS\_6, com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT\_TABLE\_JOUEURS\_7, et com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT\_TABLE\_PARAMETRES.

Référencé par com.ttpa.iris.ttpamobile.ServeurSQLite.onUpgrade().

```
{
 // On crée la table des séances
 db.execSQL("pragma foreign_keys = on;");
 db.execSQL(CREATE_BDD_JOUEURS);
 db.execSQL(CREATE_BDD_SEANCES);
 db.execSQL(CREATE_BDD_PARAMETRES);
 db.execSQL(INSERT_TABLE_JOUEURS_1);
 db.execSQL(INSERT_TABLE_JOUEURS_2);
 db.execSQL(INSERT_TABLE_JOUEURS_3);
 db.execSQL(INSERT_TABLE_JOUEURS_4);
 db.execSQL(INSERT_TABLE_JOUEURS_5);
 db.execSQL(INSERT_TABLE_JOUEURS_6);
 db.execSQL(INSERT_TABLE_JOUEURS_7);
 db.execSQL(INSERT_TABLE_PARAMETRES);

 String path = db.getPath();
 File f = new File(path);
 boolean r = f.setReadable(true, false);
 if(r)
 {
 Log.d("TTPA", "onCreate : Ajout droit lecture " + path); // d =
 debug
 }
 else
 {
 Log.e("TTPA", "onCreate : Erreur ajout droit lecture " + path); //
 e = erreur
 }
 r = f.setWritable(true, false);
 if(r)
 {
 Log.d("TTPA", "onCreate : Ajout droit écriture " + path); // d =
 debug
 }
 else
 {
 Log.e("TTPA", "onCreate : Erreur ajout droit écriture " + path); //
 e = erreur
 }
 File parentDir = f.getAbsoluteFile().getParentFile();
 r = parentDir.setReadable(true, false);
 if(r)
 {
 Log.d("TTPA", "onCreate : Ajout droit lecture " + parentDir.getPath
 ()); // d = debug
 }
 else
 {
 Log.e("TTPA", "onCreate : Erreur ajout droit lecture " + parentDir.
 getPath()); // e = erreur
 }
 r = parentDir.setWritable(true, false);
 if(r)
 {
 Log.d("TTPA", "onCreate : Ajout droit écriture " + parentDir.
 getPath()); // d = debug
 }
 else
 {
 Log.e("TTPA", "onCreate : Erreur ajout droit écriture " + parentDir.
 getPath()); // e = erreur
 }
}
```

```

 }

```

### 9.9.3.2 void com.ttpa.iris.ttpamobile.ServeurSQLite.onOpen ( SQLiteDatabase db )

#### Paramètres

|           |                                                          |
|-----------|----------------------------------------------------------|
| <i>db</i> | un type SQLiteDatabase qui représente la base de données |
|-----------|----------------------------------------------------------|

```

{
 //onUpgrade(db, 1, 2);

 String path = db.getPath();
 File f = new File(path);
 boolean r = f.setReadable(true, false);
 if(r)
 {
 Log.d("TTPA", "onOpen : Ajout droit lecture " + path); // d = debug
 }
 else
 {
 Log.e("TTPA", "onOpen : Erreur ajout droit lecture " + path); // e
 = erreur
 }
 r = f.setWritable(true, false);
 if(r)
 {
 Log.d("TTPA", "onOpen : Ajout droit écriture " + path); // d =
 debug
 }
 else
 {
 Log.e("TTPA", "onOpen : Erreur ajout droit écriture " + path); // e
 = erreur
 }
 File parentDir = f.getAbsolutePath().getParentFile();
 r = parentDir.setReadable(true, false);
 if(r)
 {
 Log.d("TTPA", "onOpen : Ajout droit lecture " + parentDir.getPath()
); // d = debug
 }
 else
 {
 Log.e("TTPA", "onOpen : Erreur ajout droit lecture " + parentDir.
 getPath()); // e = erreur
 }
 r = parentDir.setWritable(true, false);
 if(r)
 {
 Log.d("TTPA", "onOpen : Ajout droit écriture " + parentDir.getPat
 h()
); // d = debug
 }
 else
 {
 Log.e("TTPA", "onOpen : Erreur ajout droit écriture " + parentDir.
 getPath()); // e = erreur
 }
}
}

```

### 9.9.3.3 void com.ttpa.iris.ttpamobile.ServeurSQLite.onUpgrade ( SQLiteDatabase db, int oldVersion, int newVersion )

Méthode onUpgrade supprimant et recréant toutes les tables de la base de données.

## Paramètres

|                   |  |
|-------------------|--|
| <i>db</i>         |  |
| <i>oldVersion</i> |  |
| <i>newVersion</i> |  |

Références [com.ttpa.iris.ttpamobile.ServeurSQLite.onCreate\(\)](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_JOUEURS](#), [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_PARAMETRES](#), et [com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE\\_SEANCES](#).

```
{
 // On supprime la table puis on la recrée
 db.execSQL("DROP TABLE IF EXISTS " + TABLE_JOUEURS + ";");
 db.execSQL("DROP TABLE IF EXISTS " + TABLE_SEANCES + ";");
 db.execSQL("DROP TABLE IF EXISTS " + TABLE_PARAMETRES + ";");
 onCreate(db);
}
```

## 9.9.4 Documentation des données membres

**9.9.4.1** final String [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_DATE\\_DEBUT](#) = "DATE\_DEBUT" [static]

Référéncé par [com.ttpa.iris.ttpamobile.ServeurBDD.getSeance\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getSeances\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

**9.9.4.2** final String [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_DATE\\_FIN](#) = "DATE\_FIN" [static]

Référéncé par [com.ttpa.iris.ttpamobile.ServeurBDD.getSeance\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getSeances\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

**9.9.4.3** final String [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_EFFET](#) = "EFFET" [static]

Référéncé par [com.ttpa.iris.ttpamobile.ServeurBDD.getSeance\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getSeances\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

**9.9.4.4** final String [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_FREQUENCE](#) = "FREQUENCE" [static]

Référéncé par [com.ttpa.iris.ttpamobile.ServeurBDD.getSeance\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getSeances\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

**9.9.4.5** final String [com.ttpa.iris.ttpamobile.ServeurSQLite.COL\\_ID](#) = "ID\_SEANCE" [static]

Référéncé par [com.ttpa.iris.ttpamobile.ServeurBDD.getSeance\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getSeances\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

[supprimerSeance\(\)](#).

**9.9.4.6** final String com.ttpa.iris.ttpamobile.ServeurSQLite.COL\_ID\_JOUEUR =  
"ID\_JOUEUR" [static]

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.getIdJoueurParametres\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getJoueur\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getJoueurs\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getSeance\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getSeances\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.purgerSeancesJoueur\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.supprimerJoueur\(\)](#).

**9.9.4.7** final String com.ttpa.iris.ttpamobile.ServeurSQLite.COL\_ID\_PARAMETRE =  
"ID\_PARAMETRE" [static]

**9.9.4.8** final String com.ttpa.iris.ttpamobile.ServeurSQLite.COL\_INTENSITE\_EFFECT =  
"INTENSITE\_EFFECT" [static]

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.getSeance\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getSeances\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

**9.9.4.9** final String com.ttpa.iris.ttpamobile.ServeurSQLite.COL\_NOM = "NOM"  
[static]

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.getJoueur\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getJoueurs\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.insererJoueur\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.supprimerJoueur\(\)](#).

**9.9.4.10** final String com.ttpa.iris.ttpamobile.ServeurSQLite.COL\_NOMBRE\_BALLES =  
"NOMBRE\_BALLES" [static]

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.getSeance\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getSeances\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

**9.9.4.11** final String com.ttpa.iris.ttpamobile.ServeurSQLite.COL\_PUISSANCE =  
"PUISSANCE" [static]

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.getSeance\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getSeances\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).

**9.9.4.12** final String com.ttpa.iris.ttpamobile.ServeurSQLite.COL\_ROTATION =  
"ROTATION" [static]

Référencé par [com.ttpa.iris.ttpamobile.ServeurBDD.getSeance\(\)](#), [com.ttpa.iris.ttpamobile.ServeurBDD.getSeances\(\)](#), et [com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance\(\)](#).



9.9.4.13 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.COL_TAUX_REUSSITE = "TAUX_REUSSITE" [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.getSeance()`, `com.ttpa.iris.ttpamobile.ServeurBDD.getSeances()`, et `com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance()`.

9.9.4.14 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.COL_ZONE_OBJECTIF = "ZONE_OBJECTIF" [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.getSeance()`, `com.ttpa.iris.ttpamobile.ServeurBDD.getSeances()`, et `com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance()`.

9.9.4.15 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.COL_ZONE_ROBOT = "ZONE_ROBOT" [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.getSeance()`, `com.ttpa.iris.ttpamobile.ServeurBDD.getSeances()`, et `com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance()`.

9.9.4.16 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.CREATE_BDD_JOUEURS = " VARCHAR(255) NOT NULL);" [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurSQLite.onCreate()`, et `com.ttpa.iris.ttpamobile.ServeurBDD.purgerTableJoueurs()`.

9.9.4.17 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.CREATE_BDD_PARAMETERS = "CONSTRAINT fk_parametres_1 FOREIGN KEY (ID_JOUEUR) REFERENCES table_joueurs (ID_JOUEUR));" [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurSQLite.onCreate()`.

9.9.4.18 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.CREATE_BDD_SEANCES = "CONSTRAINT fk_seances_1 FOREIGN KEY (ID_JOUEUR) REFERENCES table_joueurs (ID_JOUEUR) ON DELETE CASCADE);" [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurSQLite.onCreate()`, et `com.ttpa.iris.ttpamobile.ServeurBDD.purgerTableSeances()`.

9.9.4.19 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.DATABASE_NAME = "ttpa_mobile.db" [static]`

Attributs de la classe `ServeurSQLite`.

Référencé par `com.ttpa.iris.ttpamobile.ServeurSQLite.ServeurSQLite()`.

9.9.4.20 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.DATABASE_VERSION = 1 [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurSQLite.ServeurSQLite()`.

9.9.4.21 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT_TABLE_JOUEURS_1 = "INSERT INTO table_joueurs(NOM) VALUES('LEGOUT Christophe');"`  
[static, private]

Référencé par `com.ttpa.iris.ttpamobile.ServeurSQLite.onCreate()`.

9.9.4.22 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT_TABLE_JOUEURS_2 = "INSERT INTO table_joueurs(NOM) VALUES('MARTINEZ Michel');"`  
[static, private]

Référencé par `com.ttpa.iris.ttpamobile.ServeurSQLite.onCreate()`.

9.9.4.23 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT_TABLE_JOUEURS_3 = "INSERT INTO table_joueurs(NOM) VALUES('LEBESSON Emmanuel');"`  
[static, private]

Référencé par `com.ttpa.iris.ttpamobile.ServeurSQLite.onCreate()`.

9.9.4.24 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT_TABLE_JOUEURS_4 = "INSERT INTO table_joueurs(NOM) VALUES('ELOI Damien');"`  
[static, private]

Référencé par `com.ttpa.iris.ttpamobile.ServeurSQLite.onCreate()`.

9.9.4.25 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT_TABLE_JOUEURS_5 = "INSERT INTO table_joueurs(NOM) VALUES('MATTENET Adrien');"`  
[static, private]

Référencé par `com.ttpa.iris.ttpamobile.ServeurSQLite.onCreate()`.

9.9.4.26 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT_TABLE_JOUEURS_6 = "INSERT INTO table_joueurs(NOM) VALUES('CHILA Patrick');"`  
[static, private]

Référencé par `com.ttpa.iris.ttpamobile.ServeurSQLite.onCreate()`.

9.9.4.27 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT_TABLE_JOUEURS_7 = "INSERT INTO table_joueurs(NOM) VALUES('BEAUMONT Jérôme');"`  
[static, private]

Référencé par `com.ttpa.iris.ttpamobile.ServeurSQLite.onCreate()`.

9.9.4.28 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.INSERT_TABLE_PARAMETERS = "INSERT INTO table_parametres(ID_PARAMETRE, ID_JOUEUR) VALUES(1,1);"` [static, private]

Référencé par `com.ttpa.iris.ttpamobile.ServeurSQLite.onCreate()`.

9.9.4.29 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_DATE_DEBUT = 10` [static]

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance()`.

9.9.4.30 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_DATE_FIN = 11`  
`[static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance()`.

9.9.4.31 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_EFFECT = 3`  
`[static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance()`.

9.9.4.32 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_FREQUENCE`  
`= 1 [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance()`.

9.9.4.33 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_ID = 0`  
`[static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance()`.

9.9.4.34 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_ID_JOUEUR =`  
`0 [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToJoueur()`.

9.9.4.35 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_ID_JOUEUR -`  
`PARAMETRE = 1 [static]`

9.9.4.36 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_ID_JOUEUR -`  
`SEANCE = 12 [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance()`.

9.9.4.37 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_ID_PARAMET-`  
`RE = 0 [static]`

9.9.4.38 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_INTENSITE_E-`  
`FFET = 4 [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance()`.

9.9.4.39 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_NOM = 1`  
`[static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToJoueur()`.

9.9.4.40 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_NOMBRE_BA-`  
`LLES = 2 [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance()`.

9.9.4.41 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_PUISSANCE = 5 [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance()`.

9.9.4.42 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_ROTATION = 6 [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance()`.

9.9.4.43 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_TAUX_REUSITE = 9 [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance()`.

9.9.4.44 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_ZONE_OBJECTIF = 7 [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance()`.

9.9.4.45 `final int com.ttpa.iris.ttpamobile.ServeurSQLite.NUM_COL_ZONE_ROBOT = 8 [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.cursorToSeance()`.

9.9.4.46 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE_JOUEURS = "table_joueurs" [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.getJoueur()`, `com.ttpa.iris.ttpamobile.ServeurBDD.getJoueurs()`, `com.ttpa.iris.ttpamobile.ServeurBDD.insererJoueur()`, `com.ttpa.iris.ttpamobile.ServeurSQLite.onUpgrade()`, et `com.ttpa.iris.ttpamobile.ServeurBDD.supprimerJoueur()`.

9.9.4.47 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE_PARAMETRES = "table_parametres" [static]`

Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.getIdJoueurParametres()`, et `com.ttpa.iris.ttpamobile.ServeurSQLite.onUpgrade()`.

9.9.4.48 `final String com.ttpa.iris.ttpamobile.ServeurSQLite.TABLE_SEANCES = "table_seances" [static]`

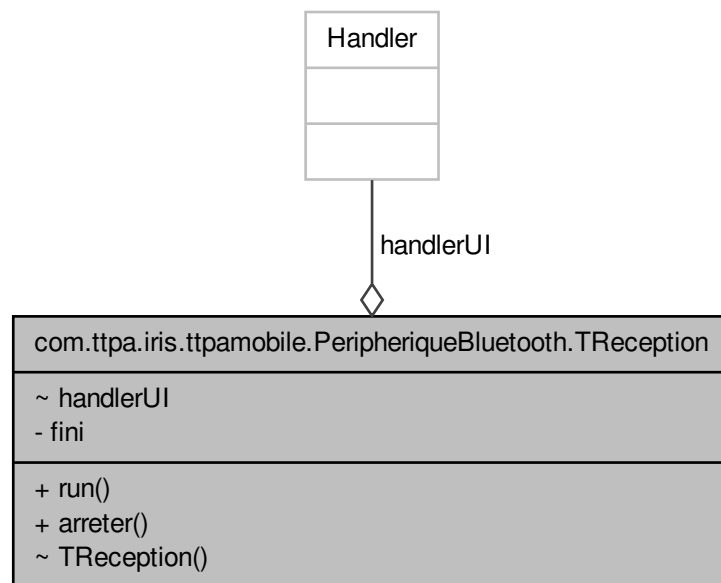
Référencé par `com.ttpa.iris.ttpamobile.ServeurBDD.getSeance()`, `com.ttpa.iris.ttpamobile.ServeurBDD.getSeances()`, `com.ttpa.iris.ttpamobile.ServeurBDD.insererSeance()`, `com.ttpa.iris.ttpamobile.ServeurSQLite.onUpgrade()`, `com.ttpa.iris.ttpamobile.ServeurBDD.purgerSeancesJoueur()`, et `com.ttpa.iris.ttpamobile.ServeurBDD.supprimerSeance()`.

La documentation de cette classe a été générée à partir du fichier suivant :

– [ServeurSQLite.java](#)

### 9.10 Référence de la classe com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception

Graphe de collaboration de com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception-  
:



#### Fonctions membres publiques

- void `run()`
- void `arreter()`

#### Fonctions de paquetage

- `TReception` (Handler h)

#### Attributs de paquetage

- Handler `handlerUI`

#### Attributs privés

- boolean `fini`

## 9.10 Référence de la classe

**com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception**

123

### 9.10.1 Documentation des constructeurs et destructeur

**9.10.1.1** `com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.TReception (Handler h)` [*package*]

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.fini](#), et [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.handlerUI](#).

```
{
 handlerUI = h;
 fini = false;
}
```

### 9.10.2 Documentation des fonctions membres

**9.10.2.1** `void com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.arreter ()`

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.fini](#).

Référencé par [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.deconnecter\(\)](#).

```
{
 if(fini == false)
 {
 fini = true;
 }
 try
 {
 Thread.sleep(500);
 }
 catch (InterruptedException e)
 {
 e.printStackTrace();
 }
}
```

**9.10.2.2** `void com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.run ()`

Références [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.CODE\\_DECONNEXION](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.CODE\\_RECEPTION](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.fini](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.getAdresse\(\)](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.getNom\(\)](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.handler](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.handlerUI](#), et [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.receiveStream](#).

```
{
 System.out.println("<Bluetooth> Attente reception");
 while(!fini)
 {
 try
 {
 if(receiveStream.available() > 0)
 {
 byte buffer[] = new byte[100];
 int k = receiveStream.read(buffer, 0, 100);
 }
 }
 }
}
```

```

 if (k > 0)
 {
 byte rawdata[] = new byte[k];
 for (int i=0; i<k; i++)
 rawdata[i] = buffer[i];

 String data = new String(rawdata);
 System.out.println("<Bluetooth> Reception " + data)
 }

 Message msg = Message.obtain();
 /*msg.what = Peripherique.CODE_RECEPTION;
 msg.obj = data;
 handlerUI.sendMessage(msg); */

 //Message msg = handlerUI.obtainMessage();
 Bundle b = new Bundle();
 b.putString("nom", getNom());
 b.putString("adresse", getAdresse());
 b.putInt("etat", CODE_RECEPTION);
 b.putString("donnees", data);
 msg.setData(b);
 handlerUI.sendMessage(msg);
 }
}

try
{
 Thread.sleep(250);
}
catch (InterruptedException e)
{
 e.printStackTrace();
}
}
catch (IOException e)
{
 //System.out.println("<Socket> error read");
 e.printStackTrace();
}
}

Message msg = Message.obtain();
//msg.arg1 = CODE_DECONNEXION;
Bundle b = new Bundle();
b.putString("nom", getNom());
b.putString("adresse", getAdresse());
b.putInt("etat", CODE_DECONNEXION);
b.putString("donnees", "");
msg.setData(b);
handler.sendMessage(msg);
System.out.println("<Bluetooth> Fin reception");
}
}

```

### 9.10.3 Documentation des données membres

#### 9.10.3.1 boolean com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.fini [private]

Référéncé par [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.arreter\(\)](#), [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.run\(\)](#), et [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.TReception\(\)](#).

#### 9.10.3.2 Handler com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.-handlerUI [package]

Référéncé par [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.run\(\)](#), et [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception.TReception\(\)](#).

La documentation de cette classe a été générée à partir du fichier suivant :

- [PeripheriqueBluetooth.java](#)

## 10 Documentation des fichiers

### 10.1 Référence du fichier AndroidManifest.xml

### 10.2 Référence du fichier Changelog.dox

### 10.3 Référence du fichier IHMEcranPrincipal.java

#### Classes

- class [com.ttpa.iris.ttpamobile.IHMEcranPrincipal](#)

### 10.4 Référence du fichier IHMHistoriqueSeances.java

#### Classes

- class [com.ttpa.iris.ttpamobile.IHMHistoriqueSeances](#)

### 10.5 Référence du fichier Joueur.java

#### Classes

- class [com.ttpa.iris.ttpamobile.Joueur](#)

### 10.6 Référence du fichier ParametreSeance.java

#### Classes

- class [com.ttpa.iris.ttpamobile.ParametreSeance](#)

### 10.7 Référence du fichier PeripheriqueBluetooth.java

#### Classes

- class [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth](#)
- class [com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception](#)

### 10.8 Référence du fichier README.dox

### 10.9 Référence du fichier ReceveurBluetooth.java



**Classes**

- class [com.ttpa.iris.ttpamobile.ReceveurBluetooth](#)

**10.10 Référence du fichier Seance.java****Classes**

- class [com.ttpa.iris.ttpamobile.Seance](#)

**10.11 Référence du fichier ServeurBDD.java****Classes**

- class [com.ttpa.iris.ttpamobile.ServeurBDD](#)

**10.12 Référence du fichier ServeurSQLite.java****Classes**

- class [com.ttpa.iris.ttpamobile.ServeurSQLite](#)

## Index

AndroidManifest.xml, [125](#)  
CODE\_CONNEXION  
    com : :tpa : :iris : :ttpamobile : :-  
        PeripheriqueBluetooth, [84](#)  
CODE\_RECEPTION  
    com : :tpa : :iris : :ttpamobile : :-  
        PeripheriqueBluetooth, [84](#)  
COL\_DATE\_DEBUT  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [116](#)  
COL\_DATE\_FIN  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [116](#)  
COL\_EFFECT  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [116](#)  
COL\_FREQUENCE  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [116](#)  
COL\_ID  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [116](#)  
COL\_ID\_JOUEUR  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [117](#)  
COL\_NOM  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [117](#)  
COL\_PUISSANCE  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [117](#)  
COL\_ROTATION  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [117](#)  
COL\_ZONE\_ROBOT  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [118](#)  
Changelog.dox, [125](#)  
DATABASE\_NAME  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [118](#)  
IHMEcranPrincipal.java, [125](#)  
IHMHistoriqueSeances.java, [125](#)  
Joueur  
    com : :tpa : :iris : :ttpamobile : :-  
        Joueur, [64](#)  
Joueur.java, [125](#)  
NUM\_COL\_EFFECT  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [120](#)  
NUM\_COL\_ID  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [120](#)  
NUM\_COL\_NOM  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [120](#)  
ParametreSeance  
    com : :tpa : :iris : :ttpamobile : :-  
        ParametreSeance, [67](#)  
ParametreSeance.java, [125](#)  
PeripheriqueBluetooth  
    com : :tpa : :iris : :ttpamobile : :-  
        PeripheriqueBluetooth, [79](#)  
PeripheriqueBluetooth.java, [125](#)  
README.dox, [125](#)  
ReceveurBluetooth  
    com : :tpa : :iris : :ttpamobile : :-  
        ReceveurBluetooth, [87](#)  
ReceveurBluetooth.java, [125](#)  
Seance  
    com : :tpa : :iris : :ttpamobile : :-  
        Seance, [89](#)  
Seance.java, [126](#)  
ServeurBDD  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurBDD, [100](#)  
ServeurBDD.java, [126](#)  
ServeurSQLite  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [113](#)  
ServeurSQLite.java, [126](#)  
TABLE\_JOUEURS  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [121](#)  
TABLE\_SEANCES  
    com : :tpa : :iris : :ttpamobile : :-  
        ServeurSQLite, [121](#)  
TRAME\_ENTETE  
    com : :tpa : :iris : :ttpamobile : :IHM-  
        EcranPrincipal, [51](#)  
TRAME\_FIN  
    com : :tpa : :iris : :ttpamobile : :IHM-

- EcranPrincipal, [51](#)
- TReception
  - com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth : :T-  
Reception, [123](#)
- actionnerSeance
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [10](#)
- actualiserIHMAppareilsBluetooth
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [10](#)
- adaptateurBluetooth
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [42](#)
- adresse
  - com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth, [84](#)
- affecterElementsIHM
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
HistoriqueSeances, [55](#)
- affecterMembresIHM
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [11](#)
- afficherBoiteDialogueDetailsSeance
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
HistoriqueSeances, [55](#)
- afficherSeance
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
HistoriqueSeances, [57](#)
- ajouterJoueur
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [12](#)
- appliquerParametresSeance
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [13](#)
- arreter
  - com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth : :T-  
Reception, [123](#)
- arreterSeance
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [14](#)
- attendre
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [15](#)
- ballesJouees
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [43](#)
- ballesReussies
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [43](#)
- barreProgressionEffetBalles
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [43](#)
- barreProgressionFrequenceBalles
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [43](#)
- barreProgressionNombreBalles
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [43](#)
- barreProgressionPuissanceBalles
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [43](#)
- barreProgressionRotationLanceur
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [44](#)
- bdd
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurBDD, [111](#)
- boutonActionSeance
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [44](#)
- boutonAppliquerNomJoueur
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [44](#)
- boutonArreterSeance
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [44](#)
- boutonBluetooth
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [44](#)
- boutonHistorique
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [45](#)
- boutonParametres
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [45](#)
- boutonPurgerSeances
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
HistoriqueSeances, [63](#)
- calculerReussiteSeance
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [15](#)
- close
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurBDD, [101](#)

- com.ttpa.iris.ttpamobile.IHMEcranPrincipal, 6
- com.ttpa.iris.ttpamobile.IHMHistorique-Seances, 53
- com.ttpa.iris.ttpamobile.Joueur, 64
- com.ttpa.iris.ttpamobile.Parametre-Seance, 66
- com.ttpa.iris.ttpamobile.PeripheriqueBluetooth, 78
- com.ttpa.iris.ttpamobile.PeripheriqueBluetooth.TReception, 122
- com.ttpa.iris.ttpamobile.ReceveurBluetooth, 86
- com.ttpa.iris.ttpamobile.Seance, 88
- com.ttpa.iris.ttpamobile.ServeurBDD, 99
- com.ttpa.iris.ttpamobile.ServeurSQLite, 112
- com : : ttpa : : iris : : ttpamobile : : IHMEcran-Principal
  - TRAME\_FIN, 51
  - actionnerSeance, 10
  - actualiserIHMAppareilsBluetooth, 10
  - adaptateurBluetooth, 42
  - affecterMembresIHM, 11
  - ajouterJoueur, 12
  - appliquerParametresSeance, 13
  - arreterSeance, 14
  - attendre, 15
  - ballesJouees, 43
  - ballesReussies, 43
  - barreProgressionEffetBalles, 43
  - barreProgressionFrequenceBalles, 43
  - barreProgressionNombreBalles, 43
  - barreProgressionPuissanceBalles, 43
  - barreProgressionRotationLanceur, 44
  - boutonActionSeance, 44
  - boutonAppliquerNomJoueur, 44
  - boutonArreterSeance, 44
  - boutonBluetooth, 44
  - boutonHistorique, 45
  - boutonParametres, 45
  - calculerReussiteSeance, 15
  - connexionBaseDeDonnees, 16
  - connexionPeripheriqueBluetooth-Ecran, 16
  - connexionPeripheriqueBluetooth-Lanceur, 17
  - connexionPeripheriqueBluetooth-Table, 17
  - connexionPeripheriquesBluetooth, 18
  - creerListeJoueurs, 19
  - deconnexionPeripheriquesBluetooth, 20
  - demarrageBluetooth, 21
  - demarrerSeance, 22
  - devices, 46
  - envoyerTrameArretPeripheriqueBluetoothEcran, 23
  - envoyerTrameArretPeripheriqueBluetoothLanceur, 23
  - envoyerTrameArretPeripheriqueBluetoothTable, 24
  - envoyerTrameConnexionPeripheriqueBluetoothEcran, 24
  - envoyerTrameDebutSeancePeripheriqueBluetoothEcran, 24
  - envoyerTrameDebutSeancePeripheriqueBluetoothLanceur, 25
  - envoyerTrameParametrageSeance-PeripheriqueBluetoothEcran, 25
  - envoyerTramePeripheriqueBluetooth-Ecran, 26
  - envoyerTramePeripheriqueBluetooth-Lanceur, 27
  - envoyerTrameRepriseSeance-PeripheriqueBluetoothLanceur, 27
  - etatSeance, 46
  - handler, 46
  - idZonePrecedente, 46
  - idZonePrecedenteObjectif, 47
  - idZonePrecedenteRobot, 47
  - incrementerBallesJouees, 28
  - modifierIHMBluetoothInsuffisant, 28
  - modifierIHMBluetoothOperationnel, 29
  - modifierIHMBluetoothSuffisant, 29
  - modifierValeursParametresIHM, 30
  - nomJoueur, 47
  - onClick, 31
  - onCreate, 32
  - onProgressChanged, 33
  - onStartTrackingTouch, 33
  - onStopTrackingTouch, 33
  - parametresActuels, 47

- pauserSeance, [34](#)
- peripheriqueBluetoothEcran, [48](#)
- peripheriqueBluetoothLanceur, [48](#)
- peripheriqueBluetoothTable, [48](#)
- peripheriques, [48](#)
- redirectionActiviteHistorique, [34](#)
- rendreBarresProgressionIHM-Utilisables, [34](#)
- rendreBoutonsIHMClickables, [35](#)
- reprendreSeance, [35](#)
- seanceEnCours, [48](#)
- selectionnerZone, [36](#)
- serveurBDD, [49](#)
- spinnerListeJoueurs, [49](#)
- strZoneObjectif, [49](#)
- strZoneRobot, [49](#)
- texteValeurEffetBalles, [49](#)
- texteValeurFrequenceBalles, [50](#)
- texteValeurNombreBalles, [50](#)
- texteValeurPuissanceBalles, [50](#)
- texteValeurRotationLanceur, [50](#)
- traiterDonneesRecues, [39](#)
- traiterDonneesRecuesLanceur, [40](#)
- traiterDonneesRecuesTable, [40](#)
- traiterErreurRecueLanceur, [41](#)
- verifierConnexionAppareilsBluetooth-Requis, [41](#)
- voyantEcran, [52](#)
- voyantLanceur, [52](#)
- voyantTable, [52](#)
- zoneObjectif, [52](#)
- zoneRobot, [52](#)
- com : :tpa : :iris : :ttpamobile : :IHM-HistoriqueSeances
  - affecterElementsIHM, [55](#)
  - afficherBoiteDialogueDetailsSeance, [55](#)
  - afficherSeance, [57](#)
  - boutonPurgerSeances, [63](#)
  - connectionBaseDeDonnees, [58](#)
  - idJoueurActuel, [63](#)
  - joueurActuel, [63](#)
  - modifierIHMAfficherSeancesJoueur, [58](#)
  - onClick, [59](#)
  - onCreate, [60](#)
  - purgerSeancesJoueur, [60](#)
  - recupererInformationsJoueur, [61](#)
  - rendreBoutonIHMClickables, [62](#)
  - serveurBDD, [63](#)
  - supprimerSeanceSelectionnee, [62](#)
  - tableauSeances, [64](#)
- com : :tpa : :iris : :ttpamobile : :Joueur
  - Joueur, [64](#)
  - getId, [65](#)
  - getNom, [65](#)
  - id, [66](#)
  - nom, [66](#)
  - setId, [65](#)
  - setNom, [65](#)
  - toString, [65](#)
- com : :tpa : :iris : :ttpamobile : :Parametre-Seance
  - ParametreSeance, [67](#)
  - effetBalles, [76](#)
  - effetEstValide, [68](#)
  - estValide, [68](#)
  - frequenceBalles, [76](#)
  - frequenceEstValide, [69](#)
  - getEffet, [69](#)
  - getEffetBalles, [70](#)
  - getEffetComplet, [70](#)
  - getFrequenceBalles, [70](#)
  - getIntensiteEffet, [70](#)
  - getNomJoueur, [71](#)
  - getNombreBalles, [71](#)
  - getPuissanceBalles, [71](#)
  - getRotation, [72](#)
  - intensiteEffet, [76](#)
  - nomJoueur, [77](#)
  - nombreBalles, [77](#)
  - nombreBallesEstValide, [72](#)
  - puissanceBalles, [77](#)
  - puissanceBallesEstValide, [72](#)
  - rotation, [78](#)
  - rotationEstValide, [73](#)
  - setEffetBalles, [73](#)
  - setFrequenceBalles, [74](#)
  - setIntensiteEffet, [74](#)
  - setNomJoueur, [75](#)
  - setNombreBalles, [74](#)
  - setPuissanceBalles, [75](#)
  - setRotation, [75](#)
  - toString, [75](#)
- com : :tpa : :iris : :ttpamobile : :-PeripheriqueBluetooth
  - PeripheriqueBluetooth, [79](#)
  - adresse, [84](#)
  - connecter, [80](#)
  - deconnecter, [81](#)

- device, [84](#)
- envoyer, [81](#)
- estConnecte, [82](#)
- getAdresse, [83](#)
- getDevice, [83](#)
- getNom, [83](#)
- handler, [84](#)
- nom, [84](#)
- receiveStream, [84](#)
- sendStream, [84](#)
- setNom, [83](#)
- socket, [85](#)
- tReception, [85](#)
- toString, [83](#)
- com : :tpa : :iris : :tpamobile : :-  
  - PeripheriqueBluetooth : :T-  
    - Reception
  - TReception, [123](#)
  - arreter, [123](#)
  - fini, [124](#)
  - handlerUI, [124](#)
  - run, [123](#)
- com : :tpa : :iris : :tpamobile : :Receveur-  
  - Bluetooth
  - ReceveurBluetooth, [87](#)
  - onReceive, [87](#)
- com : :tpa : :iris : :tpamobile : :Seance  
  - Seance, [89](#)
  - dateDebut, [96](#)
  - dateFin, [96](#)
  - effet, [97](#)
  - frequence, [97](#)
  - getDateDebut, [89](#)
  - getDateFin, [90](#)
  - getEffet, [90](#)
  - getFrequence, [90](#)
  - getId, [90](#)
  - getIdJoueur, [91](#)
  - getIntensiteEffet, [91](#)
  - getNombreBalles, [91](#)
  - getPuissance, [91](#)
  - getRotation, [92](#)
  - getTauxReussite, [92](#)
  - getZoneObjectif, [92](#)
  - getZoneRobot, [92](#)
  - horodaterBD, [93](#)
  - id, [97](#)
  - idJoueur, [97](#)
  - intensiteEffet, [97](#)
  - nombreBalles, [97](#)
  - puissance, [97](#)
  - rotation, [97](#)
  - setDateDebut, [93](#)
  - setDateFin, [93](#)
  - setEffet, [93](#)
  - setFrequence, [93](#)
  - setId, [94](#)
  - setIdJoueur, [94](#)
  - setIntensiteEffet, [94](#)
  - setNombreBalles, [95](#)
  - setPuissance, [95](#)
  - setRotation, [95](#)
  - setTauxReussite, [95](#)
  - setZoneObjectif, [96](#)
  - setZoneRobot, [96](#)
  - tauxReussite, [97](#)
  - toString, [96](#)
  - zoneObjectif, [98](#)
  - zoneRobot, [98](#)
- com : :tpa : :iris : :tpamobile : :ServeurB-  
  - DD
  - ServeurBDD, [100](#)
  - bdd, [111](#)
  - close, [101](#)
  - cursorToJoueur, [101](#)
  - cursorToSeance, [101](#)
  - getBDD, [102](#)
  - getIdJoueurParametres, [103](#)
  - getJoueur, [103](#), [104](#)
  - getJoueurs, [104](#)
  - getSeance, [105](#)
  - getSeances, [105](#), [106](#)
  - insérerJoueur, [107](#)
  - insérerSeance, [108](#)
  - open, [108](#)
  - purgerSeancesJoueur, [109](#)
  - purgerTableJoueurs, [109](#)
  - purgerTableSeances, [109](#)
  - serveurSQLite, [111](#)
  - setIdJoueurParametres, [110](#)
  - supprimerJoueur, [110](#)
  - supprimerSeance, [111](#)
- com : :tpa : :iris : :tpamobile : :ServeurS-  
  - QLite
  - COL\_EFFECT, [116](#)
  - COL\_ID, [116](#)
  - COL\_NOM, [117](#)
  - NUM\_COL\_ID, [120](#)
  - ServeurSQLite, [113](#)
  - onCreate, [113](#)

- onOpen, [115](#)
- onUpgrade, [115](#)
- connecter
  - com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth, [80](#)
- connectionBaseDeDonnees
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
HistoriqueSeances, [58](#)
- connexionBaseDeDonnees
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [16](#)
- connexionPeripheriqueBluetoothEcran
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [16](#)
- connexionPeripheriqueBluetoothLanceur
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [17](#)
- connexionPeripheriqueBluetoothTable
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [17](#)
- connexionPeripheriquesBluetooth
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [18](#)
- creerListeJoueurs
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [19](#)
- cursorToJoueur
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurBDD, [101](#)
- cursorToSeance
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurBDD, [101](#)
- dateDebut
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [96](#)
- dateFin
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [96](#)
- deconnecter
  - com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth, [81](#)
- deconnexionPeripheriquesBluetooth
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [20](#)
- demarrageBluetooth
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [21](#)
- demarrerSeance
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [22](#)
- device
  - com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth, [84](#)
- devices
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [46](#)
- effet
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [97](#)
- effetBalles
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [76](#)
- effetEstValide
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [68](#)
- envoyer
  - com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth, [81](#)
- envoyerTrameArretPeripheriqueBluetooth-  
Ecran
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [23](#)
- envoyerTrameArretPeripheriqueBluetooth-  
Lanceur
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [23](#)
- envoyerTrameArretPeripheriqueBluetooth-  
Table
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [24](#)
- envoyerTrameConnexionPeripherique-  
BluetoothEcran
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [24](#)
- envoyerTrameDebutSeancePeripherique-  
BluetoothEcran
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [24](#)
- envoyerTrameDebutSeancePeripherique-  
BluetoothLanceur
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [25](#)
- envoyerTrameParametrageSeance-  
PeripheriqueBluetoothEcran
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [25](#)

envoyerTramePeripheriqueBluetooth-  
   Ecran  
     com : :ttpa : :iris : :ttpamobile : :IHM-  
       EcranPrincipal, [26](#)  
 envoyerTramePeripheriqueBluetooth-  
   Lanceur  
     com : :ttpa : :iris : :ttpamobile : :IHM-  
       EcranPrincipal, [27](#)  
 envoyerTrameRepriseSeancePeripherique-  
   BluetoothLanceur  
     com : :ttpa : :iris : :ttpamobile : :IHM-  
       EcranPrincipal, [27](#)  
 estConnecte  
   com : :ttpa : :iris : :ttpamobile : :-  
     PeripheriqueBluetooth, [82](#)  
 estValide  
   com : :ttpa : :iris : :ttpamobile : :-  
     ParametreSeance, [68](#)  
 etatSeance  
   com : :ttpa : :iris : :ttpamobile : :IHM-  
     EcranPrincipal, [46](#)  
 fini  
   com : :ttpa : :iris : :ttpamobile : :-  
     PeripheriqueBluetooth : :T-  
       Reception, [124](#)  
 frequence  
   com : :ttpa : :iris : :ttpamobile : :-  
     Seance, [97](#)  
 frequenceBalles  
   com : :ttpa : :iris : :ttpamobile : :-  
     ParametreSeance, [76](#)  
 frequenceEstValide  
   com : :ttpa : :iris : :ttpamobile : :-  
     ParametreSeance, [69](#)  
 getAdresse  
   com : :ttpa : :iris : :ttpamobile : :-  
     PeripheriqueBluetooth, [83](#)  
 getBDD  
   com : :ttpa : :iris : :ttpamobile : :-  
     ServeurBDD, [102](#)  
 getDateDebut  
   com : :ttpa : :iris : :ttpamobile : :-  
     Seance, [89](#)  
 getDateFin  
   com : :ttpa : :iris : :ttpamobile : :-  
     Seance, [90](#)  
 getDevice  
   com : :ttpa : :iris : :ttpamobile : :-  
     PeripheriqueBluetooth, [83](#)  
 getEffet  
   com : :ttpa : :iris : :ttpamobile : :-  
     ParametreSeance, [69](#)  
 getEffetBalles  
   com : :ttpa : :iris : :ttpamobile : :-  
     ParametreSeance, [70](#)  
 getEffetCompleet  
   com : :ttpa : :iris : :ttpamobile : :-  
     ParametreSeance, [70](#)  
 getFrequence  
   com : :ttpa : :iris : :ttpamobile : :-  
     Seance, [90](#)  
 getFrequenceBalles  
   com : :ttpa : :iris : :ttpamobile : :-  
     ParametreSeance, [70](#)  
 getId  
   com : :ttpa : :iris : :ttpamobile : :-  
     Joueur, [65](#)  
   com : :ttpa : :iris : :ttpamobile : :-  
     Seance, [90](#)  
 getIdJoueur  
   com : :ttpa : :iris : :ttpamobile : :-  
     Seance, [91](#)  
 getIdJoueurParametres  
   com : :ttpa : :iris : :ttpamobile : :-  
     ServeurBDD, [103](#)  
 getIntensiteEffet  
   com : :ttpa : :iris : :ttpamobile : :-  
     ParametreSeance, [70](#)  
   com : :ttpa : :iris : :ttpamobile : :-  
     Seance, [91](#)  
 getJoueur  
   com : :ttpa : :iris : :ttpamobile : :-  
     ServeurBDD, [103](#), [104](#)  
 getJoueurs  
   com : :ttpa : :iris : :ttpamobile : :-  
     ServeurBDD, [104](#)  
 getNom  
   com : :ttpa : :iris : :ttpamobile : :-  
     Joueur, [65](#)  
   com : :ttpa : :iris : :ttpamobile : :-  
     PeripheriqueBluetooth, [83](#)  
 getNomJoueur  
   com : :ttpa : :iris : :ttpamobile : :-  
     ParametreSeance, [71](#)  
 getNombreBalles



- com : :ttpa : :iris : :ttpamobile : :- ParametreSeance, [71](#)
- com : :ttpa : :iris : :ttpamobile : :- Seance, [91](#)
- getPuissance
  - com : :ttpa : :iris : :ttpamobile : :- Seance, [91](#)
- getPuissanceBalles
  - com : :ttpa : :iris : :ttpamobile : :- ParametreSeance, [71](#)
- getRotation
  - com : :ttpa : :iris : :ttpamobile : :- ParametreSeance, [72](#)
  - com : :ttpa : :iris : :ttpamobile : :- Seance, [92](#)
- getSeance
  - com : :ttpa : :iris : :ttpamobile : :- ServeurBDD, [105](#)
- getSeances
  - com : :ttpa : :iris : :ttpamobile : :- ServeurBDD, [105](#), [106](#)
- getTauxReussite
  - com : :ttpa : :iris : :ttpamobile : :- Seance, [92](#)
- getZoneObjectif
  - com : :ttpa : :iris : :ttpamobile : :- Seance, [92](#)
- getZoneRobot
  - com : :ttpa : :iris : :ttpamobile : :- Seance, [92](#)
- handler
  - com : :ttpa : :iris : :ttpamobile : :IHM-EcranPrincipal, [46](#)
  - com : :ttpa : :iris : :ttpamobile : :- PeripheriqueBluetooth, [84](#)
- handlerUI
  - com : :ttpa : :iris : :ttpamobile : :- PeripheriqueBluetooth : :T-Reception, [124](#)
- horodaterBD
  - com : :ttpa : :iris : :ttpamobile : :- Seance, [93](#)
- id
  - com : :ttpa : :iris : :ttpamobile : :- Joueur, [66](#)
  - com : :ttpa : :iris : :ttpamobile : :- Seance, [97](#)
- idJoueur
  - com : :ttpa : :iris : :ttpamobile : :- Seance, [97](#)
- idJoueurActuel
  - com : :ttpa : :iris : :ttpamobile : :IHM-HistoriqueSeances, [63](#)
- idZonePrecedente
  - com : :ttpa : :iris : :ttpamobile : :IHM-EcranPrincipal, [46](#)
- idZonePrecedenteObjectif
  - com : :ttpa : :iris : :ttpamobile : :IHM-EcranPrincipal, [47](#)
- idZonePrecedenteRobot
  - com : :ttpa : :iris : :ttpamobile : :IHM-EcranPrincipal, [47](#)
- incrementerBallesJouees
  - com : :ttpa : :iris : :ttpamobile : :IHM-EcranPrincipal, [28](#)
- insererJoueur
  - com : :ttpa : :iris : :ttpamobile : :- ServeurBDD, [107](#)
- insererSeance
  - com : :ttpa : :iris : :ttpamobile : :- ServeurBDD, [108](#)
- intensiteEffet
  - com : :ttpa : :iris : :ttpamobile : :- ParametreSeance, [76](#)
  - com : :ttpa : :iris : :ttpamobile : :- Seance, [97](#)
- joueurActuel
  - com : :ttpa : :iris : :ttpamobile : :IHM-HistoriqueSeances, [63](#)
- modifierIHMAfficherSeancesJoueur
  - com : :ttpa : :iris : :ttpamobile : :IHM-HistoriqueSeances, [58](#)
- modifierIHMBluetoothInsuffisant
  - com : :ttpa : :iris : :ttpamobile : :IHM-EcranPrincipal, [28](#)
- modifierIHMBluetoothOperationnel
  - com : :ttpa : :iris : :ttpamobile : :IHM-EcranPrincipal, [29](#)
- modifierIHMBluetoothSuffisant
  - com : :ttpa : :iris : :ttpamobile : :IHM-EcranPrincipal, [29](#)
- modifierValeursParametresIHM
  - com : :ttpa : :iris : :ttpamobile : :IHM-EcranPrincipal, [30](#)
- nom

- com : :ttpa : :iris : :ttpamobile : :-  
Joueur, [66](#)
- com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth, [84](#)
- nomJoueur
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [47](#)
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [77](#)
- nombreBalles
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [77](#)
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [97](#)
- nombreBallesEstValide
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [72](#)
- onClick
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [31](#)
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
HistoriqueSeances, [59](#)
- onCreate
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [32](#)
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
HistoriqueSeances, [60](#)
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurSQLite, [113](#)
- onOpen
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurSQLite, [115](#)
- onProgressChanged
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [33](#)
- onReceive
  - com : :ttpa : :iris : :ttpamobile : :-  
ReceveurBluetooth, [87](#)
- onStartTrackingTouch
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [33](#)
- onStopTrackingTouch
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [33](#)
- onUpgrade
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurSQLite, [115](#)
- open
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurBDD, [108](#)
- parametresActuels
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [47](#)
- pauserSeance
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [34](#)
- peripheriqueBluetoothEcran
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [48](#)
- peripheriqueBluetoothLanceur
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [48](#)
- peripheriqueBluetoothTable
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [48](#)
- peripheriques
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [48](#)
- puissance
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [97](#)
- puissanceBalles
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [77](#)
- puissanceBallesEstValide
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [72](#)
- purgerSeancesJoueur
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
HistoriqueSeances, [60](#)
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurBDD, [109](#)
- purgerTableJoueurs
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurBDD, [109](#)
- purgerTableSeances
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurBDD, [109](#)
- receiveStream
  - com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth, [84](#)
- recupererInformationsJoueur
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
HistoriqueSeances, [61](#)
- redirectionActiviteHistorique

- com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [34](#)
- rendreBarresProgressionIHMUtilisables
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [34](#)
- rendreBoutonIHMClickables
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
HistoriqueSeances, [62](#)
- rendreBoutonsIHMClickables
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [35](#)
- reprendreSeance
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [35](#)
- rotation
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [78](#)
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [97](#)
- rotationEstValide
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [73](#)
- run
  - com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth : :T-  
Reception, [123](#)
- seanceEnCours
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [48](#)
- selectionnerZone
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [36](#)
- sendStream
  - com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth, [84](#)
- serveurBDD
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [49](#)
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
HistoriqueSeances, [63](#)
- serveurSQLite
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurBDD, [111](#)
- setDateDebut
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [93](#)
- setDateFin
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [93](#)
- setEffet
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [93](#)
- setEffetBalles
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [73](#)
- setFrequence
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [93](#)
- setFrequenceBalles
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [74](#)
- setId
  - com : :ttpa : :iris : :ttpamobile : :-  
Joueur, [65](#)
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [94](#)
- setIdJoueur
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [94](#)
- setIdJoueurParametres
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurBDD, [110](#)
- setIntensiteEffet
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [74](#)
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [94](#)
- setNom
  - com : :ttpa : :iris : :ttpamobile : :-  
Joueur, [65](#)
  - com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth, [83](#)
- setNomJoueur
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [75](#)
- setNombreBalles
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [74](#)
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [95](#)
- setPuissance
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [95](#)
- setPuissanceBalles
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [75](#)
- setRotation
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [75](#)

- com : :ttpa : :iris : :ttpamobile : :-  
Seance, [95](#)
- setTauxReussite
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [95](#)
- setZoneObjectif
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [96](#)
- setZoneRobot
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [96](#)
- socket
  - com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth, [85](#)
- spinnerListeJoueurs
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [49](#)
- strZoneObjectif
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [49](#)
- strZoneRobot
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [49](#)
- supprimerJoueur
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurBDD, [110](#)
- supprimerSeance
  - com : :ttpa : :iris : :ttpamobile : :-  
ServeurBDD, [111](#)
- supprimerSeanceSelectionnee
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
HistoriqueSeances, [62](#)
- tReception
  - com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth, [85](#)
- tableauSeances
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
HistoriqueSeances, [64](#)
- tauxReussite
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [97](#)
- texteValeurEffetBalles
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [49](#)
- texteValeurFrequenceBalles
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [50](#)
- texteValeurNombreBalles
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [50](#)
- texteValeurPuissanceBalles
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [50](#)
- texteValeurRotationLanceur
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [50](#)
- toString
  - com : :ttpa : :iris : :ttpamobile : :-  
Joueur, [65](#)
  - com : :ttpa : :iris : :ttpamobile : :-  
ParametreSeance, [75](#)
  - com : :ttpa : :iris : :ttpamobile : :-  
PeripheriqueBluetooth, [83](#)
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [96](#)
- traiterDonneesRecues
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [39](#)
- traiterDonneesRecuesLanceur
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [40](#)
- traiterDonneesRecuesTable
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [40](#)
- traiterErreurRecueLanceur
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [41](#)
- verifierConnexionAppareilsBluetooth-  
Requis
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [41](#)
- voyantEcran
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [52](#)
- voyantLanceur
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [52](#)
- voyantTable
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [52](#)
- zoneObjectif
  - com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [52](#)
  - com : :ttpa : :iris : :ttpamobile : :-  
Seance, [98](#)
- zoneRobot

com : :ttpa : :iris : :ttpamobile : :IHM-  
EcranPrincipal, [52](#)  
com : :ttpa : :iris : :ttpamobile : :-  
Seance, [98](#)