

Projet TTPA (Table Tennis Performance Analyser)

1.4

HAMMOUMA Youssef
BOTELLA-BROC Yohan

Table des matières

1	Le projet Projet TTPA (Table Tennis Performance Analyser)	2
1.1	Table des matières	2
2	Changelog	2
3	Installation	9
4	README	10
5	A propos	11
6	Licence GPL	11
7	Liste des choses à faire	11
8	Documentation des espaces de nommage	11
8.1	Référence de l'espace de nommage Ui	11
9	Documentation des classes	11
9.1	Référence de la classe BaseDeDonnees	11
9.1.1	Description détaillée	13
9.1.2	Documentation des constructeurs et destructeur	13
9.1.3	Documentation des fonctions membres	13
9.1.4	Documentation des données membres	21
9.2	Référence de la classe CommunicationBluetooth	22
9.2.1	Description détaillée	26
9.2.2	Documentation des énumérations membres	26
9.2.3	Documentation des constructeurs et destructeur	27
9.2.4	Documentation des fonctions membres	29
9.2.5	Documentation des données membres	49
9.2.6	Documentation des propriétés	51
9.3	Référence de la classe FenetreAccueil	54
9.3.1	Description détaillée	55
9.3.2	Documentation des données membres	55

9.3.3	Documentation des propriétés	59
9.4	Référence de la classe FenetreReglages	61
9.4.1	Description détaillée	62
9.4.2	Documentation des données membres	62
9.4.3	Documentation des propriétés	65
9.5	Référence de la classe FenetreStatistiques	66
9.5.1	Description détaillée	67
9.5.2	Documentation des données membres	67
9.6	Référence de la classe FenetreTTPA	68
9.6.1	Description détaillée	69
9.6.2	Documentation des fonctions membres	69
9.6.3	Documentation des données membres	69
9.7	Référence de la classe Joueur	71
9.7.1	Description détaillée	71
9.7.2	Documentation des constructeurs et destructeur	71
9.7.3	Documentation des fonctions membres	72
9.7.4	Documentation des propriétés	73
9.8	Référence de la classe Joueurs	73
9.8.1	Description détaillée	75
9.8.2	Documentation des constructeurs et destructeur	75
9.8.3	Documentation des fonctions membres	75
9.8.4	Documentation des données membres	76
9.8.5	Documentation des propriétés	77
9.9	Référence de la classe Parametre	77
9.9.1	Description détaillée	78
9.9.2	Documentation des constructeurs et destructeur	78
9.9.3	Documentation des fonctions membres	79
9.9.4	Documentation des propriétés	82
9.10	Référence de la classe Parametres	83
9.10.1	Description détaillée	85
9.10.2	Documentation des constructeurs et destructeur	85

9.10.3	Documentation des fonctions membres	85
9.10.4	Documentation des données membres	87
9.10.5	Documentation des propriétés	88
9.11	Référence de la classe PeripheriqueTTPA	88
9.11.1	Description détaillée	89
9.11.2	Documentation des constructeurs et destructeur	90
9.11.3	Documentation des fonctions membres	90
9.11.4	Documentation des données membres	94
9.11.5	Documentation des propriétés	94
9.12	Référence de la classe Profil	95
9.12.1	Description détaillée	96
9.12.2	Documentation des constructeurs et destructeur	96
9.12.3	Documentation des fonctions membres	96
9.12.4	Documentation des propriétés	98
9.13	Référence de la classe Profils	99
9.13.1	Description détaillée	100
9.13.2	Documentation des constructeurs et destructeur	100
9.13.3	Documentation des fonctions membres	100
9.13.4	Documentation des données membres	102
9.13.5	Documentation des propriétés	102
9.14	Référence de la classe StatistiquesSeance	103
9.14.1	Description détaillée	105
9.14.2	Documentation des énumérations membres	105
9.14.3	Documentation des constructeurs et destructeur	105
9.14.4	Documentation des fonctions membres	106
9.14.5	Documentation des données membres	114
9.15	Référence de la classe Ttpalhm	116
9.15.1	Description détaillée	118
9.15.2	Documentation des énumérations membres	118
9.15.3	Documentation des constructeurs et destructeur	118
9.15.4	Documentation des fonctions membres	120
9.15.5	Documentation des données membres	134

10 Documentation des fichiers	135
10.1 Référence du fichier BaseDeDonnees.cpp	135
10.1.1 Description détaillée	136
10.2 Référence du fichier BaseDeDonnees.h	136
10.2.1 Description détaillée	136
10.3 Référence du fichier Changelog.md	136
10.4 Référence du fichier communicationbluetooth.cpp	136
10.4.1 Description détaillée	136
10.5 Référence du fichier CommunicationBluetooth.cpp	137
10.5.1 Description détaillée	137
10.6 Référence du fichier communicationbluetooth.h	137
10.6.1 Description détaillée	137
10.6.2 Documentation des fonctions	138
10.7 Référence du fichier CommunicationBluetooth.h	138
10.7.1 Description détaillée	138
10.7.2 Documentation des macros	139
10.8 Référence du fichier FenetreAccueil.qml	139
10.8.1 Description détaillée	139
10.9 Référence du fichier FenetreReglages.qml	140
10.9.1 Description détaillée	140
10.10 Référence du fichier FenetreStatistiques.qml	140
10.10.1 Description détaillée	140
10.11 Référence du fichier FenetreTTPA.qml	140
10.11.1 Description détaillée	141
10.12 Référence du fichier INSTALL.md	141
10.13 Référence du fichier Joueur.cpp	141
10.13.1 Description détaillée	141
10.14 Référence du fichier Joueur.h	141
10.14.1 Description détaillée	141
10.15 Référence du fichier Joueurs.cpp	142
10.15.1 Description détaillée	142

10.16	Référence du fichier Joueurs.h	142
10.16.1	Description détaillée	142
10.17	Référence du fichier main.cpp	142
10.17.1	Description détaillée	143
10.17.2	Documentation des fonctions	143
10.18	Référence du fichier main.cpp	144
10.18.1	Description détaillée	144
10.18.2	Documentation des fonctions	144
10.19	Référence du fichier Parametre.cpp	145
10.19.1	Description détaillée	145
10.20	Référence du fichier Parametre.h	145
10.20.1	Description détaillée	145
10.21	Référence du fichier Parametres.cpp	145
10.21.1	Description détaillée	146
10.22	Référence du fichier Parametres.h	146
10.22.1	Description détaillée	146
10.23	Référence du fichier peripheriqueTTPA.cpp	146
10.23.1	Description détaillée	146
10.24	Référence du fichier peripheriqueTTPA.h	147
10.24.1	Description détaillée	147
10.25	Référence du fichier Profil.cpp	147
10.25.1	Description détaillée	147
10.26	Référence du fichier Profil.h	147
10.26.1	Description détaillée	148
10.27	Référence du fichier Profils.cpp	148
10.27.1	Description détaillée	148
10.28	Référence du fichier Profils.h	148
10.28.1	Description détaillée	148
10.29	Référence du fichier README.md	149
10.30	Référence du fichier statistiquesseance.cpp	149
10.30.1	Description détaillée	149
10.31	Référence du fichier statistiquesseance.h	149
10.31.1	Description détaillée	149
10.32	Référence du fichier ttpaihm.cpp	149
10.32.1	Description détaillée	150
10.33	Référence du fichier ttpaihm.h	150
10.33.1	Description détaillée	150
10.33.2	Documentation des macros	150

1 Le projet Projet TTPA (Table Tennis Performance Analyser)

Le système TTPA 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.

Auteur

Yohann BOTELLA-BROC botellabroc.yohann@gmail.com

Youssef HAMMOUMA yhammouma@gmail.com

1.1 Table des matières

- [README](#)
- [Changelog](#)
- [Installation](#)
- [Liste des choses à faire](#)
- [A propos](#)
- [Licence GPL](#)

2 Changelog

r107 | ybotella | 2019-06-04 15 :54 :56 +0200 (mar. 04 juin 2019) | 2 lignes

modification de la méthode commencerSeance

r106 | yhammouma | 2019-06-04 14 :31 :27 +0200 (mar. 04 juin 2019) | 3 lignes

Ajout d'un signal indiquant que la séance s'est terminée normalement + Reverse Engineering pour diagramme BOUML

r105 | ybotella | 2019-06-04 10 :48 :20 +0200 (mar. 04 juin 2019) | 2 lignes

Ajout de série dans l'objectif

r104 | ybotella | 2019-05-29 16 :56 :10 +0200 (mer. 29 mai 2019) | 3 lignes

AJout de ballesBonnes et le robot étant une zone hors table et amélioration de l'ihm écranResultats

r103 | yhammouma | 2019-05-29 15 :57 :08 +0200 (mer. 29 mai 2019) | 6 lignes

Ajout de la classe [Profil](#) + ajout de la base de donnée + ajout des commandes permettant de récupérer la bdd + modification du AndroidManifest.xml pour bloquer la rotation de l'ecran en mode portrait

- Mise au point de la sélection d'un joueur et affichage des profils en fonction du nom de joueur choisi + insertion du profil dans la bdd

r102 | yhammouma | 2019-05-28 19 :20 :28 +0200 (mar. 28 mai 2019) | 2 lignes

Ajout des classes [Parametre.h](#) et [Parametres.h](#)

r101 | yhammouma | 2019-05-28 18 :12 :09 +0200 (mar. 28 mai 2019) | 4 lignes

Ajout de la base de donnée + comboBox permettant de sélectionner un joueur de la bdd + enregistrement du nom du joueur dans la bdd lorsque qu'il appuye sur boutonJouer

r100 | ybotella | 2019-05-28 17 :00 :50 +0200 (mar. 28 mai 2019) | 2 lignes

AJout d'une méthodes pour afficherStatistiqueEcranResultats

r99 | tvaira | 2019-05-28 14 :55 :20 +0200 (mar. 28 mai 2019) | 1 ligne

Ajout fichier SQL

r98 | yhammouma | 2019-05-28 14 :44 :16 +0200 (mar. 28 mai 2019) | 1 ligne

suppression fichier 'bdd-ttpa'

r97 | ybotella | 2019-05-28 14 :25 :58 +0200 (mar. 28 mai 2019) | 4 lignes

Ajout de l'icon pause et amelioration des méthode commencerSeance reprendreSeance mettreEnPauseSeance et finirSeance puis amelioration du visuel de l'ihm

r96 | yhammouma | 2019-05-28 14 :18 :09 +0200 (mar. 28 mai 2019) | 1 ligne

Ajout de l'icone

r95 | yhammouma | 2019-05-28 14 :05 :56 +0200 (mar. 28 mai 2019) | 1 ligne

Ajout des icones en vue de la BDD

r94 | yhammouma | 2019-05-28 13 :49 :18 +0200 (mar. 28 mai 2019) | 3 lignes

Ajout de l'icon permettant d'ouvrir la boite de dialogue en vue de la bdd

r93 | tvaira | 2019-05-28 11 :07 :12 +0200 (mar. 28 mai 2019) | 1 ligne

Ajout classe [BaseDeDonnees](#) pour SQLite

r92 | yhammouma | 2019-05-26 13 :44 :25 +0200 (dim. 26 mai 2019) | 4 lignes

Ajout de la doc doxygen + Mise en place d'un message d'erreur sur le terminal mobile en cas d'erreur robot + envoie d'une trame à l'ecran lui indiquant l'erreur.

r91 | yhammouma | 2019-05-24 12 :45 :02 +0200 (ven. 24 mai 2019) | 4 lignes

Ajout du décodage des trames reçues + Gestion des états de la séance et envoie de trame en fonction de l'etat + envoie de la trame contenant la zone impactée à l'ecran ttpa

r90 | ybotella | 2019-05-23 18 :05 :21 +0200 (jeu. 23 mai 2019) | 6 lignes

Ajout du splash, Séquencement des trame Start, PAUSE, RESUME, END, RESET. Ajout du début séance commence à 0 puis on peut le faire changer d'etat grâce au différénte trame de type Enum EtatSeance dans communicationbluetooth.h . Amélioration de l'initialisation des stats avec commencerSeance

r89 | yhammouma | 2019-05-23 17 :08 :11 +0200 (jeu. 23 mai 2019) | 2 lignes

Ajout d'un todo

r88 | yhammouma | 2019-05-23 17 :06 :49 +0200 (jeu. 23 mai 2019) | 3 lignes

Mise en place de la selection de la zone du robot et de la zone objectif

— récupération des zones à l'aide de `getZoneObjectif` et `getZoneRobot`.

r87 | ybotella | 2019-05-23 15 :14 :35 +0200 (jeu. 23 mai 2019) | 3 lignes

Ajout du décodage de trame `Pause`, `rasume`, `end`, `reset`, `stat`, `start` et `signal/` slot adapter dans ihm et communication bluetooth

r86 | yhammouma | 2019-05-22 17 :19 :07 +0200 (mer. 22 mai 2019) | 1 ligne

Résolution du conflit

r85 | yhammouma | 2019-05-21 17 :29 :07 +0200 (mar. 21 mai 2019) | 1 ligne

Ajout du fichier sql de la base de donnée

r84 | yhammouma | 2019-05-21 16 :15 :18 +0200 (mar. 21 mai 2019) | 3 lignes

Résolution du problème qui empechait la connexion à plusieurs périphériques en même temps

r83 | yhammouma | 2019-05-21 10 :03 :52 +0200 (mar. 21 mai 2019) | 4 lignes

Ajout de l'envoi des trames `mettreEnPauseSeance` `reprenreLaSeance` et `arreterSeance` + Boite de dialogue qui permettra de choisir la position du robot et l'objectif.

r82 | tvaira | 2019-05-20 11 :33 :51 +0200 (lun. 20 mai 2019) | 1 ligne

Ajout documentation Doxygen pour tag 1.1

r81 | tvaira | 2019-05-20 11 :31 :23 +0200 (lun. 20 mai 2019) | 1 ligne

Passage des fichiers Doxygen au format Markdown

r80 | ybotella | 2019-05-17 11 :58 :39 +0200 (ven. 17 mai 2019) | 2 lignes

décoder trame

r79 | ybotella | 2019-05-10 12 :31 :19 +0200 (ven. 10 mai 2019) | 1 ligne

création du tag 1.1

r78 | ybotella | 2019-05-10 12 :30 :58 +0200 (ven. 10 mai 2019) | 1 ligne

supression erreur

r77 | ybotella | 2019-05-10 12 :24 :25 +0200 (ven. 10 mai 2019) | 1 ligne

création du tag 1.1

r76 | ybotella | 2019-05-10 12 :18 :15 +0200 (ven. 10 mai 2019) | 2 lignes

revue 3

r75 | yhammouma | 2019-05-10 12 :18 :03 +0200 (ven. 10 mai 2019) | 2 lignes

terminal-ttpa V1.1

r74 | ybotella | 2019-05-09 18 :49 :02 +0200 (jeu. 09 mai 2019) | 2 lignes

ajout de dioxygen

r73 | ybotella | 2019-05-09 16 :32 :38 +0200 (jeu. 09 mai 2019) | 2 lignes

Ajout de `decoderTrame`

r72 | yhammouma | 2019-05-09 16 :22 :38 +0200 (jeu. 09 mai 2019) | 3 lignes

Ajout d'un bouton qui permettra dans le futur de selectionner la zone du robot. Amélioration des méthodes d'envoi des trames.

r71 | yhammouma | 2019-05-09 10 :50 :12 +0200 (jeu. 09 mai 2019) | 7 lignes

Ajout d'une méthode finSeance qui envoie une trame de fin de séance aux appareils TTPA. Ajout de 3 méthodes demarrerSeance←Ecran, demarrerSeanceRobot et demarrerSeanceTable qui envoient une trame de démarrage de séance correspondante à chaque appareil TTPA. Ajout d'un bouton "Arreter" qui envoie la trame de finSeance et l'affichage du bouton "Pause" lorsque l'on démarre une séance.

r70 | tvaira | 2019-05-09 06 :08 :01 +0200 (jeu. 09 mai 2019) | 2 lignes

Correction Erreurs compilation

r69 | ybotella | 2019-05-08 21 :33 :58 +0200 (mer. 08 mai 2019) | 2 lignes

Ajout de Todo et simulation du Nom en attente de la connexion

r68 | tvaira | 2019-05-08 20 :02 :47 +0200 (mer. 08 mai 2019) | 2 lignes

nouveauClient() est un slot !

r67 | tvaira | 2019-05-08 19 :50 :39 +0200 (mer. 08 mai 2019) | 2 lignes

Verifications Bluetooth

r66 | ybotella | 2019-05-08 19 :28 :20 +0200 (mer. 08 mai 2019) | 3 lignes

Modification de l'ihm, et ajout du peripherique et les différents hostMode.

r65 | ybotella | 2019-05-08 18 :26 :09 +0200 (mer. 08 mai 2019) | 3 lignes

ajout du serveur dans communicationbluetooth.cpp et .h, et modification de la taille de l'image

r64 | ybotella | 2019-05-04 16 :17 :30 +0200 (sam. 04 mai 2019) | 5 lignes

ajout de dixxygen et de nombreuse méthodes dans communicationbluetooth.cpp et .h tel que connecter déconnecter, estconnecter les appareil et aussi la socketConnected, socketDisconnected, socketErreur et socketReadyRead

r63 | ybotella | 2019-05-04 14 :41 :56 +0200 (sam. 04 mai 2019) | 4 lignes

ajout de deux méthodes CommunicationBluetooth : :startDeviceDiscovery et deviceDiscovered pour la recherche de périphérique Bluetooth et ajout de dixxygen

r62 | ybotella | 2019-05-03 15 :45 :41 +0200 (ven. 03 mai 2019) | 2 lignes

ajout de la méthode information dans communicationbluetooth.cpp et .h

r61 | ybotella | 2019-05-03 15 :06 :32 +0200 (ven. 03 mai 2019) | 3 lignes

ajout du robot et de l'object sur l'ihm avec 4 méthodes deux setter et deux getter pour robot et la zone objective

r60 | ybotella | 2019-05-03 12 :00 :47 +0200 (ven. 03 mai 2019) | 3 lignes

commencerSeance et finirSeance en simulation plus début de la structure du bluetooth

r59 | yhammouma | 2019-05-03 12 :00 :05 +0200 (ven. 03 mai 2019) | 3 lignes

Modification de la valeur min et max du sliderFrequenceBalles à 20-70/ minutes.

r58 | ybotella | 2019-05-02 20 :42 :27 +0200 (jeu. 02 mai 2019) | 2 lignes

ajout du setZoneRobot

r57 | ybotella | 2019-05-02 19 :07 :21 +0200 (jeu. 02 mai 2019) | 3 lignes

Modificaiton de dioxygen pour les slots afficherBallesTotal, afficherBallesTotalHorsTable et la variable numeroZonelImpacte

r56 | ybotella | 2019-05-02 16 :53 :21 +0200 (jeu. 02 mai 2019) | 2 lignes

affichage de l'écranResultats du nombre ballesTotales

r55 | yhammouma | 2019-05-02 15 :56 :14 +0200 (jeu. 02 mai 2019) | 3 lignes

Ajout de la méthode demarrerSeance qui récupère bien les paramètres du robot entrés par l'utilisateur

r54 | ybotella | 2019-05-02 15 :25 :49 +0200 (jeu. 02 mai 2019) | 2 lignes

ajout de la couleur a chaque fois qu'on impact une zone

r53 | yhammouma | 2019-05-02 11 :45 :59 +0200 (jeu. 02 mai 2019) | 6 lignes

Réorganisation des fichiers QML à l'aide d'ajout de deux fichiers qml "FenetreReglages.qml" et "FenetreStatistiques.qml".

Ajout d'une classe demarrerSeance qui permettra entre autre de recevoir les parametres de la [FenetreReglages](#).

r52 | ybotella | 2019-04-25 16 :20 :19 +0200 (jeu. 25 avril 2019) | 2 lignes

ajout de afficherImpactCouleurs dans ttpa ihm

r51 | yhammouma | 2019-04-25 16 :17 :46 +0200 (jeu. 25 avril 2019) | 3 lignes

Ajout de la condition de la taille du nomJoueur pour activer le bouton "Jouer"

r50 | ybotella | 2019-04-25 15 :50 :23 +0200 (jeu. 25 avril 2019) | 4 lignes

Ajout d'un signal et un slot pour pouvoir afficher les balles total et balles balle ayant touché la Table(zone robot et zone hors table non comprise).

r49 | ybotella | 2019-04-25 13 :31 :47 +0200 (jeu. 25 avril 2019) | 1 ligne

création du tag 1.0

r48 | yhammouma | 2019-03-31 17 :58 :17 +0200 (dim. 31 mars 2019) | 3 lignes

Mise en place du squelette de la condition entre le nombre de balles entré et le bouton Regler

r47 | tvaira | 2019-03-31 07 :03 :50 +0200 (dim. 31 mars 2019) | 1 ligne

Controle Qualité Revue 2 (Terminal)

r46 | tvaira | 2019-03-30 11 :41 :19 +0100 (sam. 30 mars 2019) | 1 ligne

Controle Qualité Revue 2 (Ecran)

r45 | yhammouma | 2019-03-29 11 :53 :04 +0100 (ven. 29 mars 2019) | 2 lignes

Ajout de fonctionnalités graphiques pour le bouton Jouer

r44 | ybotella | 2019-03-29 11 :34 :17 +0100 (ven. 29 mars 2019) | 5 lignes

Modification de la méthode calculerPourcentages elle était en afficherpourcentages ajout d'un attribut nbBallesTotalTouche suppression d'un enum dans le [Ttpalhm.h](#), modification des nom d'attribut correction du francais, ajout de QDebug dans [statistiqueseance.cpp](#)

r43 | yhammouma | 2019-03-28 20 :47 :57 +0100 (jeu. 28 mars 2019) | 3 lignes

Détection des périphériques fonctionnelles ainsi que leurs voyants respectifs

r42 | yhammouma | 2019-03-28 18 :29 :40 +0100 (jeu. 28 mars 2019) | 3 lignes

Modification du thème de l'application ainsi que l'ajout du logo "PPC Sorgues"

r41 | yhammouma | 2019-03-28 17 :18 :41 +0100 (jeu. 28 mars 2019) | 1 ligne

Ajout de la modification du voyant lors de la détection d'un seul périphérique + ajout de l'icône écran

r40 | ybotella | 2019-03-28 16 :17 :00 +0100 (jeu. 28 mars 2019) | 3 lignes

Optimisation du code crée une méthode initialiserZone pour réduire le code dans le constructeur et ajout du filet dans l'ihm.

r39 | ybotella | 2019-03-28 14 :44 :51 +0100 (jeu. 28 mars 2019) | 3 lignes

changement du if de saisirNumeroZone en switch case, ajout de calculerPourcentage dans [statistiquesseance.cpp](#) et .h

r38 | yhammouma | 2019-03-28 14 :39 :36 +0100 (jeu. 28 mars 2019) | 1 ligne

Ajout des icônes

r37 | yhammouma | 2019-03-28 14 :36 :11 +0100 (jeu. 28 mars 2019) | 2 lignes

Modification graphique de la boîte "dialog" du bluetooth

r36 | ybotella | 2019-03-28 12 :21 :54 +0100 (jeu. 28 mars 2019) | 2 lignes

Ajout de simulation

r35 | ybotella | 2019-03-28 12 :09 :51 +0100 (jeu. 28 mars 2019) | 5 lignes

Ajout d'une simulation avec la saisie du clavier pour sélectionner les zones dans [tpaihm.h](#) et [tpaihm.cpp](#) il est invisible grâce à la commande `Maximize` à 0. ne pas oublier qui a le focus de la saisie seulement si c'est la dernière ligne du constructeur.

r34 | tvaira | 2019-03-28 04 :27 :38 +0100 (jeu. 28 mars 2019) | 3 lignes

Ajout de la classe JAVA `QtBluetoothBroadcastReceiver` au fichier `AndroidManifest.xml`

r33 | ybotella | 2019-03-27 12 :17 :06 +0100 (mer. 27 mars 2019) | 4 lignes

Ajout de deux attributs `balleToucheTable` et `ballePrecedentToucherTable` et modification de certains attributs comme `initialiserStatistiques`, `impacterZone` et amélioration de l'ihm

r32 | yhammouma | 2019-03-27 11 :55 :36 +0100 (mer. 27 mars 2019) | 1 ligne

Modification du design de la boîte Dialog du bluetooth

r31 | ybotella | 2019-03-25 12 :14 :21 +0100 (lun. 25 mars 2019) | 2 lignes

ajout de subversion sur qt

r30 | ybotella | 2019-03-25 12 :00 :58 +0100 (lun. 25 mars 2019) | 3 lignes

Ajout de Commentaires doxygen dans les fichiers [StatistiquesSeance.h](#) et .cpp

r29 | ybotella | 2019-03-22 12 :44 :02 +0100 (ven. 22 mars 2019) | 2 lignes

suppression des méthodes `reset` et ajout dans `initialiserStatistiques`

r28 | ybotella | 2019-03-22 12 :18 :25 +0100 (ven. 22 mars 2019) | 2 lignes

Ajout de la méthode `resetStatistiques` avec les attribue à 0

r27 | ybotella | 2019-03-22 12 :09 :13 +0100 (ven. 22 mars 2019) | 4 lignes

Ajout méthodes `impacterZone`, et `initialiserStatistiques` dans [statistiquesseance.cpp](#) et `.h`, ajout de commentaire doxygen sur `l'ihm.cpp` et `.h`

r26 | yhammouma | 2019-03-22 11 :26 :51 +0100 (ven. 22 mars 2019) | 2 lignes

Réalisation de la méthode `rechercherPeripheriquesTTPA`

r25 | ybotella | 2019-03-22 10 :01 :21 +0100 (ven. 22 mars 2019) | 3 lignes

Ajout de trois bouton "invisible" pour changer de page et renommer les layout et label de `l'ihm`

r24 | ybotella | 2019-03-21 16 :47 :06 +0100 (jeu. 21 mars 2019) | 3 lignes

Ajout d'un bouton en `Qdebug` pour aller de l'écran Attente à l'écran de seance

r23 | yhammouma | 2019-03-21 16 :22 :02 +0100 (jeu. 21 mars 2019) | 1 ligne

Ajout des permissions Bluetooth dans le `AndroidManifest` + Ajout du prototype d'etat des appareils dans la boite de dialogue

r22 | ybotella | 2019-03-21 16 :21 :07 +0100 (jeu. 21 mars 2019) | 3 lignes

Modification de `l'ihm`, Ajout de méthodes `reset seance` et ajout attribut, `numeroZoneObjectif`, `numeroZoneRobot`, `nbBallesTotal`.

r21 | ybotella | 2019-03-21 12 :36 :30 +0100 (jeu. 21 mars 2019) | 3 lignes

Nouveau nom classe [StatistiquesSeance](#) Integration des `QVector` pour gérer les zones

r20 | ybotella | 2019-03-21 10 :41 :54 +0100 (jeu. 21 mars 2019) | 1 ligne

ajout de todo et de nouvelle méthodes pour les futur bouton

r19 | tvaira | 2019-03-20 20 :43 :04 +0100 (mer. 20 mars 2019) | 2 lignes

Ajout de l'icône de l'application

r18 | yhammouma | 2019-03-20 12 :20 :12 +0100 (mer. 20 mars 2019) | 2 lignes

Ajout du fichier `main.cpp`

r17 | yhammouma | 2019-03-20 11 :59 :54 +0100 (mer. 20 mars 2019) | 3 lignes

Ajout des méthodes "sockets" qui étaient auparavant dans la classe [CommunicationBluetooth](#) dans la classe [PeripheriqueTTPA](#).

r16 | tvaira | 2019-03-16 18 :26 :10 +0100 (sam. 16 mars 2019) | 1 ligne

Notions de classes en QML (voir Doxygen)

r15 | tvaira | 2019-03-16 11 :47 :40 +0100 (sam. 16 mars 2019) | 2 lignes

Mode plein écran

r14 | tvaira | 2019-03-15 19 :32 :08 +0100 (ven. 15 mars 2019) | 1 ligne

Modification Doxyfile pour QML

r13 | tvaira | 2019-03-15 05 :48 :06 +0100 (ven. 15 mars 2019) | 1 ligne

Modifications documentation du code

r12 | tvaira | 2019-03-15 05 :36 :56 +0100 (ven. 15 mars 2019) | 1 ligne

Déplacement fichiers ecran-TTPA

r11 | ybotella | 2019-03-14 15 :34 :05 +0100 (jeu. 14 mars 2019) | 1 ligne

Ajout dans l'ihm.h un slot pour avoir un minuteur et un attribut compteur initialiser à 0, puis une méthode compteurTimerSequence avec pour fonction itération et d'afficher sur l'ihm le début séquence

r10 | yhammouma | 2019-03-14 10 :04 :17 +0100 (jeu. 14 mars 2019) | 1 ligne

Ajout du projet initial Terminal Android

r9 | tvaira | 2019-03-14 06 :32 :16 +0100 (jeu. 14 mars 2019) | 1 ligne

Correction et ajout des commentaires

r8 | ybotella | 2019-03-13 16 :27 :36 +0100 (mer. 13 mars 2019) | 1 ligne

Ajout de QTimer et d'un slots avec une fonction afficher heure avec une forme en (hh : mm : ss)

r7 | ybotella | 2019-03-13 09 :01 :12 +0100 (mer. 13 mars 2019) | 1 ligne

Modification fichier ressources.qrc et mise en amrche de la raspberry

r6 | tvaira | 2019-03-09 07 :51 :23 +0100 (sam. 09 mars 2019) | 1 ligne

Initialisation Doxygen

r5 | tvaira | 2019-03-08 18 :10 :30 +0100 (ven. 08 mars 2019) | 1 ligne

Initialisation de la documentation du code

r4 | ybotella | 2019-03-08 15 :28 :40 +0100 (ven. 08 mars 2019) | 1 ligne

Modification fichier ressources.qrc

r3 | ybotella | 2019-03-01 09 :17 :44 +0100 (ven. 01 mars 2019) | 1 ligne

Logo du projet TTPA

r2 | ybotella | 2019-03-01 08 :54 :24 +0100 (ven. 01 mars 2019) | 1 ligne

Création des dossier plus des Squelettes et début D'ihm de l'écran

r1 | www-data | 2019-02-06 20 :07 :59 +0100 (mer. 06 févr. 2019) | 1 ligne

Creating initial repository structure

3 Installation

Fabrication

```
$ qmake
$ make
$ sudo make install
```

4 README

Nom : Projet TTPA (Table Tennis Performance Analyser)

Le système TTPA 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 Écran : Yohann BOTELLA-BROC
- Module Application Mobile : **Youssef HAMMOUMA**

Numéro de version : 1.3

Auteurs

Auteur

Yohann BOTELLA-BROC botellabroc.yohann@gmail.com
Youssef HAMMOUMA yhammouma@gmail.com

Dépôt SVN

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

Recette IR

- Étudiant : Yohann BOTELLA-BROC
 - Le système d'exploitation est installé et fonctionnel
 - L'écran est configuré en mode "kiosque"
 - La zone d'impact est identifiée et affichée en temps réel
 - Les données de la séance sont affichées en temps réel
 - Les liaisons sans fil sont opérationnelles
 - Les informations sont affichées en fin de séquence
- Étudiant : Youssef HAMMOUMA
 - 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

Base de données SQLite

```
CREATE TABLE joueurs ( "idJoueur" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, "nom" VARCHAR(255) NOT NULL
  UNIQUE )

CREATE TABLE parametres ( "idParametres" INTEGER PRIMARY KEY AUTOINCREMENT, "nbBalles" INTEGER NOT NULL,
  "nbBallesMinute" INTEGER NOT NULL, "effet" INTEGER NOT NULL, "puissance" INTEGER NOT NULL, "rotation" INTEGER
  NOT NULL, "zoneObjectif" INTEGER NOT NULL, "zoneRobot" INTEGER NOT NULL )

CREATE TABLE profils ( "idProfil" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, "idJoueur" INTEGER NOT NULL,
  "idParametres" INTEGER NOT NULL, "nom" VARCHAR(255) NOT NULL UNIQUE, CONSTRAINT fk_profils_1 FOREIGN KEY
  (idJoueur) REFERENCES joueurs (idJoueur) ON DELETE CASCADE, CONSTRAINT fk_profils_2 FOREIGN KEY (idParametres)
  REFERENCES parametres (idParametres) ON DELETE CASCADE )

CREATE TABLE seances ( "idSeance" INTEGER PRIMARY KEY AUTOINCREMENT, "idProfil" INTEGER NOT NULL,
  "reussite" REAL NOT NULL, "horodatage" DATETIME NOT NULL, CONSTRAINT fk_seances_1 FOREIGN KEY (idProfil) REFERENCES
  profils (idProfil) ON DELETE CASCADE )

CREATE TABLE ttpa ( "idTTPA" INTEGER PRIMARY KEY CHECK (idTTPA = 1), "idProfil" INTEGER NOT NULL,
  CONSTRAINT fk_ttpa_1 FOREIGN KEY (idProfil) REFERENCES profils (idProfil) ON DELETE CASCADE )
```

5 A propos

Auteur

Yohann BOTELLA-BROC botellabroc.yohann@gmail.com
Youssef HAMMOUMA yhammouma@gmail.com

Version

1.3

Date

2019

6 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

7 Liste des choses à faire

Membre [FenetreReglages](#) : [:positionRobot](#)

Faire en sorte que le bouton cancel reset le Dialog positionRobot et que le bouton Save valide les paramètres

8 Documentation des espaces de nommage

8.1 Référence de l'espace de nommage Ui

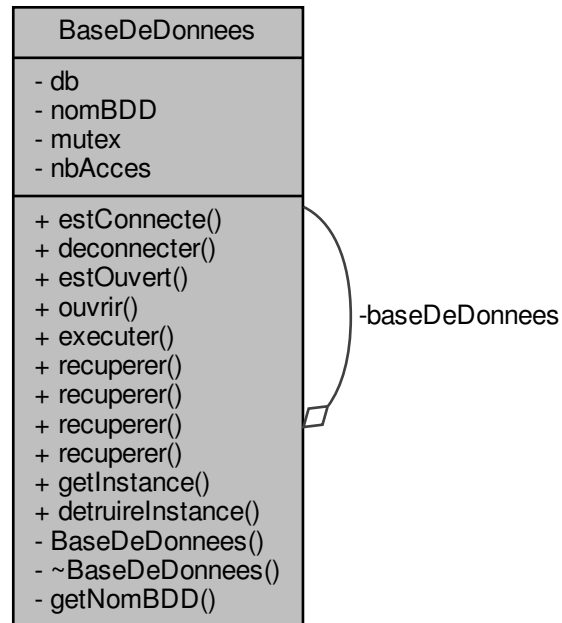
9 Documentation des classes

9.1 Référence de la classe [BaseDeDonnees](#)

Déclaration de la classe [BaseDeDonnees](#) pour Android.

```
#include <BaseDeDonnees.h>
```


Graphe de collaboration de BaseDeDonnees :



Fonctions membres publiques

- Q_INVOKABLE bool `estConnecte` ()
- Q_INVOKABLE void `deconnecter` ()
- Q_INVOKABLE bool `estOuvert` ()
- Q_INVOKABLE bool `ouvrir` (QString fichierBase)
- Q_INVOKABLE bool `executer` (QString requete)
- Q_INVOKABLE bool `recuperer` (QString requete, QString &donnees)
- Q_INVOKABLE bool `recuperer` (QString requete, QStringList &donnees)
- Q_INVOKABLE bool `recuperer` (QString requete, QVector< QString > &donnees)
- Q_INVOKABLE bool `recuperer` (QString requete, QVector< QStringList > &donnees)

Fonctions membres publiques statiques

- static `BaseDeDonnees` * `getInstance` ()
- static void `detruireInstance` ()

Fonctions membres privées

- `BaseDeDonnees` ()
- `~BaseDeDonnees` ()
- QSqlDatabase `getNomBDD` ()

Attributs privés

- QSqlDatabase `db`
- QString `nomBDD`
- QMutex `mutex`

Attributs privés statiques

- static [BaseDeDonnees](#) * [baseDeDonnees](#) = NULL
- static int [nbAcces](#) = 0

9.1.1 Description détaillée

Auteur

Thierry VAIRA

Version

1.1

9.1.2 Documentation des constructeurs et destructeur

9.1.2.1 BaseDeDonnees()

```
BaseDeDonnees::BaseDeDonnees ( ) [private]
```

Références [db](#), et [nomBDD](#).Référéncé par [getInstance\(\)](#).

```
00022 {
00023     #ifdef DEBUG_BASEDEDONNEES
00024     qDebug() << Q_FUNC_INFO;
00025     #endif
00026
00027     nomBDD = "sqlite";
00028     db = QSqlDatabase::addDatabase("SQLITE", nomBDD);
00029 }
```

9.1.2.2 ~BaseDeDonnees()

```
BaseDeDonnees::~~BaseDeDonnees ( ) [private]
```

```
00032 {
00033     #ifdef DEBUG_BASEDEDONNEES
00034     qDebug() << Q_FUNC_INFO;
00035     #endif
00036 }
```

9.1.3 Documentation des fonctions membres

9.1.3.1 deconnecter()

```
void BaseDeDonnees::deconnecter ( )
```

Références [getNomBDD\(\)](#), et [mutex](#).

Référencé par [destruireInstance\(\)](#).

```
00083 {
00084     QMutexLocker verrou(&mutex);
00085     if (getNomBDD().isOpen())
00086     {
00087         qDebug() << Q_FUNC_INFO << getNomBDD().driverName();
00088         getNomBDD().close();
00089     }
00090 }
```

9.1.3.2 destruireInstance()

```
void BaseDeDonnees::destruireInstance ( ) [static]
```

Références [baseDeDonnees](#), [deconnecter\(\)](#), et [nbAcces](#).

Référencé par [Joueurs : ~Joueurs\(\)](#), [Parametres : ~Parametres\(\)](#), et [Profils : ~Profils\(\)](#).

```
00052 {
00053     if (baseDeDonnees != NULL)
00054     {
00055         if (nbAcces > 0)
00056         {
00057             nbAcces--;
00058         }
00059     }
00060     #ifdef DEBUG_BASEDEDONNEES
00061     qDebug() << Q_FUNC_INFO << "nbAcces" << nbAcces;
00062     #endif
00063     if (nbAcces == 0)
00064     {
00065         baseDeDonnees->deconnecter();
00066         delete baseDeDonnees;
00067         baseDeDonnees = NULL;
00068     }
00069 }
```

9.1.3.3 estConnecte()

```
bool BaseDeDonnees::estConnecte ( )
```

Références [getNomBDD\(\)](#), et [mutex](#).

```
00077 {
00078     QMutexLocker verrou(&mutex);
00079     return getNomBDD().isOpen();
00080 }
```

9.1.3.4 estOuvert()

```
bool BaseDeDonnees::estOuvert ( )
```

Références [getNomBDD\(\)](#), et [mutex](#).

```
00093 {
00094     QMutexLocker verrou(&mutex);
00095     return getNomBDD().isOpen();
00096 }
```

9.1.3.5 executer()

```
bool BaseDeDonnees::executer (
    QString requete )
```

Références [db](#), [getNomBDD\(\)](#), et [mutex](#).

Référéncé par [Joueurs : :insérerJoueur\(\)](#), [Parametres : :insérerParametres\(\)](#), et [Profils : :insérerProfil\(\)](#).

```
00164 {
00165     QMutexLocker verrou(&mutex);
00166     QSqlQuery enregistrements(db);
00167     bool estCorrect;
00168
00169     if (getNomBDD().isOpen())
00170     {
00171         if (requete.contains("UPDATE") || requete.contains("INSERT") || requete.contains("DELETE"))
00172         {
00173             estCorrect = enregistrements.exec(requete);
00174
00175             #ifdef DEBUG_BASEDEDONNEES
00176             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Retour %1 pour la requete : %2").arg(
00177                 QString::number(estCorrect)).arg(requete);
00178             #endif
00179
00180             if (estCorrect)
00181             {
00182                 return true;
00183             }
00184             else
00185             {
00186                 qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : %1 pour la requête %2").arg(
00187                     enregistrements.lastError().text()).arg(requete);
00188                 return false;
00189             }
00190         }
00191         else
00192         {
00193             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : requête %1 non autorisée !").arg(requete);
00194             return false;
00195         }
00196     }
00197     else
00198     {
00199         return false;
00200     }
00201 }
00202 }
```

9.1.3.6 getInstance()

`BaseDeDonnees * BaseDeDonnees::getInstance () [static]`

Références [BaseDeDonnees\(\)](#), [baseDeDonnees](#), et [nbAcces](#).

Référencé par [Joueurs : :Joueurs\(\)](#), [Parametres : :Parametres\(\)](#), et [Profils : :Profils\(\)](#).

```
00039 {
00040     if (baseDeDonnees == NULL)
00041     {
00042         baseDeDonnees = new BaseDeDonnees();
00043     }
00044     nbAcces++;
00045     #ifdef DEBUG_BASEDEDONNEES
00046     qDebug() << Q_FUNC_INFO << "nbAcces" << nbAcces;
00047     #endif
00048     return baseDeDonnees;
00049 }
```

9.1.3.7 getNomBDD()

`QSqlDatabase BaseDeDonnees::getNomBDD () [private]`

Références [nomBDD](#).

Référencé par [deconnecter\(\)](#), [estConnecte\(\)](#), [estOuvert\(\)](#), [executer\(\)](#), [ouvrir\(\)](#), et [recuperer\(\)](#).

```
00072 {
00073     return QSqlDatabase::database(nomBDD, false);
00074 }
```

9.1.3.8 ouvrir()

`bool BaseDeDonnees::ouvrir (`
 `QString fichierBase)`

Références [getNomBDD\(\)](#), et [mutex](#).

Référencé par [Joueurs : :Joueurs\(\)](#), [Parametres : :Parametres\(\)](#), et [Profils : :Profils\(\)](#).

```
00099 {
00100     if (getNomBDD().driverName() != "SQLITE")
00101     {
00102         return false;
00103     }
00104     QMutexLocker verrou(&mutex);
00105     if (!getNomBDD().isOpen())
00106     {
00107         QFile sfile(QString("assets:/db") + QString("/") + fichierBase);
00108         QFile dfile(QString("./" + fichierBase));
00109         // supprime le fichier destination
00110         /*if (sfile.exists())
00111         {
00112             if (dfile.exists())
00113             {
00114                 dfile.remove();
00115             }
00116         }*/
00117         // copie
00118         if (sfile.exists())
```

```

00122     {
00123         sfile.copy(dfile.fileName());
00124         bool retour = dfile.setPermissions(QIODevice::ReadUser|QIODevice::WriteUser|
QIODevice::ReadOther);
00125         qDebug() << Q_FUNC_INFO << retour << dfile.permissions();
00126         //QFileInfo info(dfile);
00127         //qDebug() << Q_FUNC_INFO << "path" << info.absoluteFilePath();
00128         QFile dossierApplication("/data/data/org.qtproject.TTPA");
00129         retour = dossierApplication.setPermissions(QIODevice::ReadUser|QIODevice::WriteUser|
QIODevice::ExeUser|QIODevice::ReadOther|QIODevice::WriteOther|QIODevice::ExeOther);
00130         qDebug() << Q_FUNC_INFO << retour << dossierApplication.permissions();
00131         QFile dossierFichiers("/data/data/org.qtproject.TTPA/files");
00132         retour = dossierFichiers.setPermissions(QIODevice::ReadUser|QIODevice::WriteUser|
QIODevice::ExeUser|QIODevice::ReadOther|QIODevice::WriteOther|QIODevice::ExeOther);
00133         qDebug() << Q_FUNC_INFO << retour << dossierFichiers.permissions();
00134     }
00135
00136     getNomBDD().setDatabaseName(QString("./") + fichierBase);
00137
00138     #ifdef DEBUG_BASEDEDONNEES
00139     qDebug() << Q_FUNC_INFO << getNomBDD().databaseName();
00140     #endif
00141
00142     if(getNomBDD().open())
00143     {
00144         #ifdef DEBUG_BASEDEDONNEES
00145         qDebug() << Q_FUNC_INFO << QString::fromUtf8("Ouverture réussie de '%1'").arg(
getNomBDD().databaseName());
00146         #endif
00147         return true;
00148     }
00149     else
00150     {
00151         qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : impossible d'ouvrir la base de données !");
00152     }
00153     return false;
00154 }
00155 }
00156 else
00157 {
00158     return true;
00159 }
00160 }

```

9.1.3.9 recuperer() [1/4]

```

bool BaseDeDonnees::recuperer (
    QString requete,
    QString & donnees )

```

Références [db](#), [getNomBDD\(\)](#), et [mutex](#).

Référencé par [Joueurs : :insérerJoueur\(\)](#), [Parametres : :insérerParametres\(\)](#), [Joueurs : :recupererJoueurs\(\)](#), [Profils : :recupererNom←](#)
[Profils\(\)](#), [Parametres : :recupererParametres\(\)](#), et [Parametres : :recupererParametresProfil\(\)](#).

```

00208 {
00209     QMutexLocker verrou(&mutex);
00210     QSqlQuery enregistrements(db);
00211     bool estCorrect;
00212
00213     if (getNomBDD().isOpen())
00214     {
00215         if (requete.contains("SELECT"))
00216         {
00217             estCorrect = enregistrements.exec(requete);
00218             #ifdef DEBUG_BASEDEDONNEES
00219             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Retour %1 pour la requete : %2").arg(
QString::number(estCorrect)).arg(requete);
00220             #endif
00221
00222             if (estCorrect)
00223             {
00224                 // on se positionne sur l'enregistrement
00225                 enregistrements.first();
00226
00227                 // on vérifie l'état de l'enregistrement retourné
00228                 if(!enregistrements.isValid())
00229                 {

```

```

00230         #ifdef DEBUG_BASEDEDONNEES
00231         qDebug() << Q_FUNC_INFO << QString::fromUtf8("Résultat non valide !");
00232         #endif
00233
00234         return false;
00235     }
00236
00237     // on récupère sous forme de QString la valeur du champ
00238     if (enregistrements.isNull(0))
00239     {
00240         #ifdef DEBUG_BASEDEDONNEES
00241         qDebug() << Q_FUNC_INFO << QString::fromUtf8("Aucun résultat !");
00242         #endif
00243
00244         return false;
00245     }
00246     donnees = enregistrements.value(0).toString();
00247
00248     #ifdef DEBUG_BASEDEDONNEES
00249     qDebug() << Q_FUNC_INFO << "Enregistrement -> " << donnees;
00250     #endif
00251
00252     return true;
00253 }
00254 else
00255 {
00256     qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : %1 pour la requête %2").arg(
00257     enregistrements.lastError().text()).arg(requete);
00258
00259     return false;
00260 }
00261 else
00262 {
00263     qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : requête %1 non autorisée !").arg(requete);
00264
00265     return false;
00266 }
00267 else
00268 {
00269     return false;
00270 }
00271 }

```

9.1.3.10 recuperer() [2/4]

```

bool BaseDeDonnees::recuperer (
    QString requete,
    QStringList & donnees )

```

Références [db](#), [getNomBDD\(\)](#), et [mutex](#).

```

00277 {
00278     QMutexLocker verrou(&mutex);
00279     QSqlQuery enregistrements(db);
00280     bool estCorrect;
00281
00282     if (getNomBDD().isOpen())
00283     {
00284         if (requete.contains("SELECT"))
00285         {
00286             estCorrect = enregistrements.exec(requete);
00287
00288             #ifdef DEBUG_BASEDEDONNEES
00289             qDebug() << QString::fromUtf8("<BaseDeDonnees::recuperer(QString, QStringList)> retour %1 pour
la requete : %2").arg(QString::number(estCorrect)).arg(requete);
00290             #endif
00291
00292             if (estCorrect)
00293             {
00294                 // on se positionne sur l'enregistrement
00295                 enregistrements.first();
00296
00297                 // on vérifie l'état de l'enregistrement retourné
00298                 if (!enregistrements.isValid())
00299                 {
00300                     #ifdef DEBUG_BASEDEDONNEES
00301                     qDebug() << Q_FUNC_INFO << QString::fromUtf8("Résultat non valide !");
00302                     #endif

```

```

00303         return false;
00304     }
00305
00306     // on récupère sous forme de QString la valeur de tous les champs sélectionnés
00307     // et on les stocke dans une liste de QString
00308     for(int i = 0; i < enregistrements.record().count(); i++)
00309     {
00310         if (!enregistrements.isNull(i))
00311         {
00312             donnees << enregistrements.value(i).toString();
00313         }
00314     }
00315
00316     #ifdef DEBUG_BASEDEDONNEES
00317     qDebug() << Q_FUNC_INFO << "Enregistrement -> " << donnees;
00318     #endif
00319     return true;
00320
00321     }
00322     else
00323     {
00324         qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : %1 pour la requête %2").arg(
00325             enregistrements.lastError().text()).arg(requete);
00326
00327         return false;
00328     }
00329 }
00330 else
00331 {
00332     qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : requête %1 non autorisée !").arg(requete
00333 );
00334     return false;
00335 }
00336 }
00337 else
00338 {
00339     return false;
00340 }
00341 }

```

9.1.3.11 recuperer() [3/4]

```

bool BaseDeDonnees::recuperer (
    QString requete,
    QVector< QString > & donnees )

```

Références [db](#), [getNomBDD\(\)](#), et [mutex](#).

```

00347 {
00348     QMutexLocker verrou(&mutex);
00349     QSqlQuery enregistrements(db);
00350     bool estCorrect;
00351     QString valeur;
00352
00353     if (getNomBDD().isOpen())
00354     {
00355         if (requete.contains("SELECT"))
00356         {
00357             estCorrect = enregistrements.exec(requete);
00358
00359             #ifdef DEBUG_BASEDEDONNEES
00360             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Retour %1 pour la requete : %2").arg(
00361                 QString::number(estCorrect)).arg(requete);
00362             #endif
00363
00364             if(estCorrect)
00365             {
00366                 // pour chaque enregistrement
00367                 while (enregistrements.next() )
00368                 {
00369                     // on récupère sous forme de QString la valeur du champs sélectionné
00370                     valeur = enregistrements.value(0).toString();
00371
00372                     #ifdef DEBUG_BASEDEDONNEES
00373                     qDebug() << Q_FUNC_INFO << "Enregistrement -> " << data;
00374                     #endif
00375
00376                     // on stocke l'enregistrement dans le QVector

```



```

00376         donnees.push_back(valeur);
00377     }
00378
00379     #ifdef DEBUG_BASEDEDONNEES
00380     qDebug() << Q_FUNC_INFO << "Enregistrement -> " << donnees;
00381     #endif
00382
00383     return true;
00384 }
00385 else
00386 {
00387     qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : %1 pour la requête %2").arg(
enregistrements.lastError().text()).arg(requete);
00388     return false;
00389 }
00390 }
00391 else
00392 {
00393     qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : requête %1 non autorisée !").arg(requete
);
00394
00395     return false;
00396 }
00397 }
00398 else
00399 {
00400     return false;
00401 }
00402 }

```

9.1.3.12 recuperer() [4/4]

```

bool BaseDeDonnees::recuperer (
    QString requete,
    QVector< QStringList > & donnees )

```

Références [db](#), [getNomBDD\(\)](#), et [mutex](#).

```

00408 {
00409     QMutexLocker verrou(&mutex);
00410     QSqlQuery enregistrements(db);
00411     bool estCorrect;
00412     QStringList valeurs;
00413
00414     if (getNomBDD().isOpen())
00415     {
00416         if (requete.contains("SELECT"))
00417         {
00418             estCorrect = enregistrements.exec(requete);
00419             #ifdef DEBUG_BASEDEDONNEES
00420             qDebug() << Q_FUNC_INFO << QString::fromUtf8("Retour %1 pour la requete : %2").arg(
QString::number(estCorrect)).arg(requete);
00421             #endif
00422
00423             if (estCorrect)
00424             {
00425                 // pour chaque enregistrement
00426                 while (enregistrements.next() )
00427                 {
00428                     // on récupère sous forme de QString la valeur de tous les champs sélectionnés
00429                     // et on les stocke dans une liste de QString
00430                     for(int i=0; i < enregistrements.record().count(); i++)
00431                     {
00432                         valeurs << enregistrements.value(i).toString();
00433                     }
00434
00435                     #ifdef DEBUG_BASEDEDONNEES
00436                     //qDebug() << Q_FUNC_INFO << "Enregistrement -> " << data;
00437                     /*for(int i=0;i<r.record().count();i++)
00438                         qDebug() << r.value(i).toString();*/
00439                     #endif
00440
00441                     // on stocke l'enregistrement dans le QVector
00442                     donnees.push_back(valeurs);
00443
00444                     // on efface la liste de QString pour le prochain enregistrement
00445                     valeurs.clear();
00446                 }
00447
00448                 #ifdef DEBUG_BASEDEDONNEES

```

```

00449         qDebug() << Q_FUNC_INFO << "Enregistrement -> " << donnees;
00450         #endif
00451
00452         return true;
00453     }
00454     else
00455     {
00456         qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : %1 pour la requête %2").arg(
enregistrement.lastError().text()).arg(requete);
00457
00458         return false;
00459     }
00460 }
00461 else
00462 {
00463     qDebug() << Q_FUNC_INFO << QString::fromUtf8("Erreur : requête %1 non autorisée !").arg(requete
);
00464
00465     return false;
00466 }
00467 }
00468 else
00469 {
00470     return false;
00471 }
00472 }

```

9.1.4 Documentation des données membres

9.1.4.1 baseDeDonnees

`BaseDeDonnees * BaseDeDonnees::baseDeDonnees = NULL [static], [private]`

Référencé par [destruireInstance\(\)](#), et [getInstance\(\)](#).

9.1.4.2 db

`QSqlDatabase BaseDeDonnees::db [private]`

Référencé par [BaseDeDonnees\(\)](#), [executer\(\)](#), et [recuperer\(\)](#).

9.1.4.3 mutex

`QMutex BaseDeDonnees::mutex [private]`

Référencé par [deconnecter\(\)](#), [estConnecte\(\)](#), [estOuvert\(\)](#), [executer\(\)](#), [ouvrir\(\)](#), et [recuperer\(\)](#).

9.1.4.4 nbAcces

`int BaseDeDonnees::nbAcces = 0 [static], [private]`

Référencé par [destruireInstance\(\)](#), et [getInstance\(\)](#).

9.1.4.5 nomBDD

```
QString BaseDeDonnees::nomBDD [private]
```

Référencé par [BaseDeDonnees\(\)](#), et [getNomBDD\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

- [BaseDeDonnees.h](#)
- [BaseDeDonnees.cpp](#)

9.2 Référence de la classe CommunicationBluetooth

Déclaration de la classe [CommunicationBluetooth](#).

```
#include <communicationbluetooth.h>
```

Grphe de collaboration de CommunicationBluetooth :

CommunicationBluetooth
+ nomPeripheriqueBluetooth Local + adressePeripheriqueBluetooth Local + peripheriqueEcranTTPADetecte + peripheriqueEcranTTPACconnecte + peripheriqueEcranTTPADeconnecte + peripheriqueTableTTPADetecte + peripheriqueTableTTPACconnecte + peripheriqueTableTTPADeconnecte + peripheriqueRobotTTPADetecte + peripheriqueRobotTTPACconnecte et 6 de plus... - monBluetooth - serveur - socket - serviceInfo - nom - adresse - etatConnexion - connexionErreur - etatSeance - peripheriqueBluetoothLocal - peripheriquesTTPA - discoveryAgentDevice - etatSeance
+ CommunicationBluetooth() + ~CommunicationBluetooth() + getEtatSeance() + demarrer() + arreter() + CommunicationBluetooth() + ~CommunicationBluetooth() + getEtatSeance() + rechercherPeripheriquesTTPA() + arreterRecherche() et 20 de plus... + nouveauClient() + socketConnected() + socketDisconnected() + socketReadyRead() + socketErreur() + deviceConnected() + deviceDisconnected() + error() + ajouterPeripheriqueTTPA() + rechercheTerminee() + rechercheErreur() + recevoirTramePeripheriqueTTPA() + recevoirConnecte() + recevoirDeconnecte() - decoderTrame()

Types publics

- enum `EtatSeance` { `Initial` = 0, `EnCours` = 1, `EnPause` = 2, `Terminee` }
- Les différents état d'une séance d'entraînement.*
- enum `EtatDeLaSeance` { `Initial` = 0, `EnCours` = 1, `EnPause` = 2, `Arretee` }

Connecteurs publics

- void `nouveauClient` ()

- void `socketConnected` ()
la socket est connecté
- void `socketDisconnected` ()
la socket est déconnecté
- void `socketReadyRead` ()
la socket est prête à lire les informations
- void `socketErreur` (QBluetoothSocket : :SocketError `erreur`)
la socket indique une erreur de connexion
- void `deviceConnected` (const QBluetoothAddress &`adresse`)
- void `deviceDisconnected` (const QBluetoothAddress &`adresse`)
- void `error` (QBluetoothLocalDevice : :Error `erreur`)
- void `ajouterPeripheriqueTTPA` (const QBluetoothDeviceInfo &)
- void `rechercheTerminee` ()
- void `rechercheErreur` (QBluetoothDeviceDiscoveryAgent : :Error `erreur`)
- void `recevoirTramePeripheriqueTTPA` (QString `nomPeripherique`, QString `trame`)
- void `recevoirConnecte` ()
- void `recevoirDeconnecte` ()

Signaux

- void `connecte` ()
- void `erreur` ()
- void `impactZone` (QString `numeroZone`)
- void `departSeance` (QString `nomJoueur`, QString `zoneObjectif`, QString `zoneRobot`)
- void `mettreEnPauseSeance` ()
Met la séance en pause.
- void `repriseSeance` ()
- void `finSeance` ()
- void `resetSeance` ()
- void `erreurRobot` (QString `infoDuRobot`)
- void `recherche` ()
signal indiquant que l'état de la recherche de périphériques a changé
- void `detecte` ()
- void `connecte` ()
- void `deconnecte` ()
- void `robotDetecte` ()
- void `ecranDetecte` ()
- void `tableDetecte` ()
- void `peripheriquesTTPAUpdated` ()
- void `erreurRobot` ()
- void `seanceFinie` ()
- void `numeroTableUpdated` ()

Fonctions membres publiques

- `CommunicationBluetooth` (QObject *`parent`=nullptr)
- `~CommunicationBluetooth` ()
Définition du destructeur la classe `CommunicationBluetooth`.
- `EtatSeance` `getEtatSeance` ()
- void `demarrer` ()
demarre le serveur
- void `arreter` ()
arrete le serveur
- `CommunicationBluetooth` ()
Définition du constructeur la classe `CommunicationBluetooth`.
- `~CommunicationBluetooth` ()
- `EtatDeLaSeance` `getEtatSeance` ()
- Q_INVOKABLE void `rechercherPeripheriquesTTPA` ()
Assure la recherche d'appareils bluetooth "TTPA".
- Q_INVOKABLE void `arreterRecherche` ()
Permet d'arreter la recherche des appareils bluetooth "TTPA".
- Q_INVOKABLE bool `estValide` ()
Retourne vrai si l'appareil correspond à un appareil "TTPA".
- Q_INVOKABLE void `connecterPeripheriquesTTPA` ()
Connecte individuelle chaque appareil TTPA détecté
- Q_INVOKABLE void `deconnecterPeripheriquesTTPA` ()
Déconnecte individuelle chaque appareil TTPA détecté
- Q_INVOKABLE bool `envoyerPeripheriqueTTPA` (QString `nom`, QString `message`)
Permet d'envoyer une trame à un périphérique TTPA identifié par son nom.
- Q_INVOKABLE bool `demarrerSeance` (QString `nomJoueur`, QString `zoneObjectif`, QString `zoneRobot`, int `frequenceBalle`, int `nbBalles`, int `effet`, int `puissance`, bool `rotation`)

- Q_INVOKABLE bool [demarrerSeanceRobot](#) (int frequenceBalle, int nbBalles, int effet, int puissance, bool rotation)
Envoi une trame contenant les paramètres du robot au robot-tpa.
- Q_INVOKABLE bool [demarrerSeanceEcran](#) (QString nomJoueur, QString zoneObjectif, QString zoneRobot)
Envoi une trame de démarrage de séance à l'écran.
- Q_INVOKABLE bool [demarrerSeanceTable](#) (int nbBalles)
Envoi une trame de démarrage de séance à la table.
- Q_INVOKABLE bool [mettreEnPauseSeance](#) ()
- Q_INVOKABLE bool [reprendreLaSeance](#) ()
Reprend la séance.
- Q_INVOKABLE bool [arreterSeance](#) ()
Met subitement fin à la séance peut importe son état.
- bool [finirSeance](#) ()
Met fin à la séance lorsque le robot indique qu'il à tirer toutes ses balles.
- bool [envoyerZoneImpactee](#) (QString zone)
Dès que la table envoie une trame contenant la zone impactée, la méthode l'envoie à l'écran tpa.
- bool [envoyerErreurRobot](#) ()
Si le robot détecte un bourrage de balle ou une absence de balle, une trame est envoyée à l'écran tpa.
- Q_INVOKABLE QString [getZoneObjectif](#) (QString zoneObjectif)
Permet d'obtenir la zone objectif.
- Q_INVOKABLE QString [getZoneRobot](#) (QString zoneRobot)
Permet d'obtenir la zone robot.
- Q_INVOKABLE [PeripheriqueTTPA](#) * [getPeripheriqueTTPA](#) (QString nom)
- QString [getNomPeripheriqueBluetoothLocal](#) ()
- QString [getAdressePeripheriqueBluetoothLocal](#) ()
- QVariant [getPeripheriquesTTPA](#) ()

Propriétés

- QString [nomPeripheriqueBluetoothLocal](#)
le nom du périphérique Bluetooth de la tablette
- QString [adressePeripheriqueBluetoothLocal](#)
l'adresse MAC du périphérique Bluetooth de la tablette
- bool [peripheriqueEcranTTPADetecte](#)
- bool [peripheriqueEcranTTPAConnecte](#)
- bool [peripheriqueEcranTTPADeconnecte](#)
- bool [peripheriqueTableTTPADetecte](#)
- bool [peripheriqueTableTTPAConnecte](#)
- bool [peripheriqueTableTTPADeconnecte](#)
- bool [peripheriqueRobotTTPADetecte](#)
le périphérique Robot est détecté
- bool [peripheriqueRobotTTPAConnecte](#)
le périphérique Robot est connecté
- bool [peripheriqueRobotTTPADeconnecte](#)
le périphérique Robot est déconnecté
- bool [etatRecherche](#)
état de la recherche.
- QVariant [peripheriquesTTPA](#)
- bool [erreurRobotDetecte](#)
erreur sur le robot
- bool [seanceTerminee](#)
la séance est terminée
- int [numeroTable](#)
le numéro de la table où se situe les périphériques Robot, Table et Ecran

Fonctions membres privées

- void [decoderTrame](#) (QString trame)
permet de décoder la trame émise par le terminal mobile

Attributs privés

- QBluetoothLocalDevice [monBluetooth](#)
L'interface Bluetooth de la Raspberry Pi.
- QBluetoothServer * [serveur](#)
Le serveur Bluetooth.
- QBluetoothSocket * [socket](#)
la socket de communication Bluetooth

- `QBluetoothServiceInfo` [serviceInfo](#)
- `QString` [nom](#)
Le nom de l'interface Bluetooth.
- `QString` [adresse](#)
L'adresse MAC de l'interface Bluetooth.
- `bool` [etatConnexion](#)
L'état de connexion.
- `bool` [connexionErreur](#)
booléen indiquant une erreur de connexion
- [EtatSeance](#) [etatSeance](#)
L'état de la séance d'entraînement.
- `QBluetoothLocalDevice` [peripheriqueBluetoothLocal](#)
le périphérique Bluetooth de la tablette
- `QList< QObject * >` [peripheriquesTTPA](#)
liste des périphériques Bluetooth TTPA détectés au maximum 3 (Robot, Table et Ecran)
- `QBluetoothDeviceDiscoveryAgent *` [discoveryAgentDevice](#)
l'agent permettant de découvrir des périphériques Bluetooth
- [EtatDeLaSeance](#) [etatSeance](#)
état de la séance.

9.2.1 Description détaillée

Assure la détection des périphériques TTPA, leur connexion (et déconnexion) et une communication (envoyer et recevoir des trames)

Auteur

Botella-Broc Yohann

Version

1.4

Auteur

Hammouma Youssef

Version

1.4

9.2.2 Documentation des énumérations membres

9.2.2.1 EtatDeLaSeance

`enum` [CommunicationBluetooth::EtatDeLaSeance](#)

Valeurs énumérées

Initial	
EnCours	
EnPause	
Arretee	

```
00069      {
00070          Initial = 0,
```

```

00071         EnCours = 1,
00072         EnPause = 2,
00073         Arretee
00074     };

```

9.2.2.2 EtatSeance

```
enum CommunicationBluetooth::EtatSeance
```

Valeurs énumérées

Initial	
EnCours	
EnPause	
Terminee	

```

00053     {
00054         Initial = 0,
00055         EnCours = 1,
00056         EnPause,
00057         Terminee
00058     };

```

9.2.3 Documentation des constructeurs et destructeur

9.2.3.1 CommunicationBluetooth() [1/2]

```
CommunicationBluetooth::CommunicationBluetooth (
    QObject * parent = nullptr )
```

Références [adresse](#), [connexionErreur](#), [deviceConnected\(\)](#), [deviceDisconnected\(\)](#), [error\(\)](#), [etatConnexion](#), [etatSeance](#), [monBluetooth](#), et [nom](#).

```

00025                                     : QObject(parent),
00026     serveur(nullptr), socket(nullptr)
00027 {
00028     qDebug() << Q_FUNC_INFO;
00029     etatConnexion = false;
00030     connexionErreur = false;
00031     etatSeance = EtatSeance::Initial;
00032     if (!monBluetooth.isValid())
00033     {
00034         qDebug() << Q_FUNC_INFO << "Bluetooth valide" << monBluetooth.isValid();
00035         return;
00036     }
00037     qDebug() << Q_FUNC_INFO << monBluetooth.hostMode();
00038     monBluetooth.powerOn();
00039     nom = monBluetooth.name();
00040     adresse = monBluetooth.address().toString();
00041     qDebug() << Q_FUNC_INFO << "nom" << nom << "adresse" << adresse;
00042     monBluetooth.setHostMode(QBluetoothLocalDevice::HostDiscoverable);
00043     //monBluetooth.setHostMode(QBluetoothLocalDevice::HostConnectable);
00044
00045     connect(&monBluetooth, SIGNAL(deviceConnected(QBluetoothAddress)), this,
00046         SLOT(deviceConnected(QBluetoothAddress)));
00047     connect(&monBluetooth, SIGNAL(deviceDisconnected(QBluetoothAddress)),
00048         this, SLOT(deviceDisconnected(QBluetoothAddress)));
00049     connect(&monBluetooth, SIGNAL(error(QBluetoothLocalDevice::Error)), this, SLOT(
00050         error(QBluetoothLocalDevice::Error)));
00051 }

```


9.2.3.2 ~CommunicationBluetooth() [1/2]

CommunicationBluetooth::~~CommunicationBluetooth ()

Références [arreter\(\)](#), et [monBluetooth](#).

Référencé par [CommunicationBluetooth\(\)](#).

```
00059 {
00060     arreter();
00061     monBluetooth.setHostMode(QBluetoothLocalDevice::HostPoweredOff);
00062     qDebug() << Q_FUNC_INFO;
00063 }
```

9.2.3.3 CommunicationBluetooth() [2/2]

CommunicationBluetooth::CommunicationBluetooth ()

Paramètres

<i>parent</i>	QObject *
---------------	-----------

Références [adressePeripheriqueBluetoothLocal](#), [ajouterPeripheriqueTTPA\(\)](#), [deconnecterPeripheriquesTTPA\(\)](#), [discoveryAgentDevice](#), [error\(\)](#), [etatSeance](#), [nomPeripheriqueBluetoothLocal](#), [peripheriqueBluetoothLocal](#), [peripheriquesTTPA](#), [rechercheErreur\(\)](#), [rechercheTerminee\(\)](#), et [~CommunicationBluetooth\(\)](#).

```
00019     : nomPeripheriqueBluetoothLocal
    ("", adressePeripheriqueBluetoothLocal(""),
    discoveryAgentDevice(NULL), etatRecherche(false),
    peripheriqueRobotTTPADetecte(false),
    peripheriqueRobotTTPACconnecte(false),
    peripheriqueRobotTTPADeconnecte(false),
    peripheriqueEcranTTPADetecte(false),
    peripheriqueEcranTTPACconnecte(false),
    peripheriqueEcranTTPADeconnecte(false),
    peripheriqueTableTTPADetecte(false),
    peripheriqueTableTTPACconnecte(false),
    peripheriqueTableTTPADeconnecte(false),
    seanceTerminee(false), erreurRobotDetecte(false),
    numeroTable(2)
00020 {
00021     qDebug() << Q_FUNC_INFO;
00022     if (!peripheriqueBluetoothLocal.isValid())
00023     {
00024         qDebug() << Q_FUNC_INFO << "Bluetooth désactivé !";
00025         //qCritical("Bluetooth désactivé !");
00026         return;
00027     }
00028
00029     // Le Bluetooth de la tablette
00030     peripheriqueBluetoothLocal.powerOn();
00031     nomPeripheriqueBluetoothLocal =
    peripheriqueBluetoothLocal.name();
00032     adressePeripheriqueBluetoothLocal =
    peripheriqueBluetoothLocal.address().toString();
00033     qDebug() << Q_FUNC_INFO << nomPeripheriqueBluetoothLocal <<
    adressePeripheriqueBluetoothLocal;
00034
00035     // Recherche
00036     peripheriqueBluetoothLocal.setHostMode(
    QBluetoothLocalDevice::HostDiscoverable);
00037     discoveryAgentDevice = new QBluetoothDeviceDiscoveryAgent(this);
00038
00039     // Slot pour la recherche de périphériques Bluetooth TTPA
00040     connect(discoveryAgentDevice, SIGNAL(deviceDiscovered(QBluetoothDeviceInfo)), this,
00041     SLOT(ajouterPeripheriqueTTPA(QBluetoothDeviceInfo)));
00042     connect(discoveryAgentDevice, SIGNAL(canceled()), this, SLOT(
    rechercheTerminee()));
00043     connect(discoveryAgentDevice, SIGNAL(finished()), this, SLOT(
    rechercheTerminee()));
```

```

00044     connect(discoveryAgentDevice, SIGNAL(error(
00045         QBluetoothDeviceDiscoveryAgent::Error)), this, SLOT(rechercheErreur(QBluetoothDeviceDiscoveryAgent::Error)));
00046     etatSeance = EtatDeLaSeance::Initial;
00047 }

```

9.2.3.4 ~CommunicationBluetooth() [2/2]

CommunicationBluetooth::~CommunicationBluetooth ()

9.2.4 Documentation des fonctions membres

9.2.4.1 ajouterPeripheriqueTPPA

```

void CommunicationBluetooth::ajouterPeripheriqueTPPA (
    const QBluetoothDeviceInfo & info ) [slot]

```

Références [arreterRecherche\(\)](#), [detecte\(\)](#), [NOM_BLUETOOTH_ECRAN](#), [NOM_BLUETOOTH_ROBOT](#), [NOM_BLUETOOTH_TABLE](#), [numeroTable](#), [peripheriqueEcranTPPAConnecte](#), [peripheriqueRobotTPPAConnecte](#), [peripheriqueRobotTPPADetecte](#), [peripheriquesTPPA](#), [peripheriqueTableTPPAConnecte](#), et [peripheriqueTableTPPADetecte](#).

Référencé par [CommunicationBluetooth\(\)](#).

```

00328 {
00329     qDebug() << Q_FUNC_INFO << info.name() << info.address().toString();
00330
00331     if(info.name() == (NOM_BLUETOOTH_TABLE + QString::number(
00332         numeroTable)))
00333     {
00334         qDebug() << Q_FUNC_INFO << "périphérique Table TPPA découvert" << info.name();
00335         PeripheriqueTPPA *tableTPPA = new PeripheriqueTPPA(info.name(),
00336             info.address().toString(), this);
00337         peripheriquesTPPA.append(tableTPPA);
00338         peripheriqueTableTPPADetecte = true;
00339         emit detecte();
00340     }
00341     if(info.name() == (NOM_BLUETOOTH_ROBOT + QString::number(
00342         numeroTable)))
00343     {
00344         qDebug() << Q_FUNC_INFO << "périphérique Robot TPPA découvert" << info.name();
00345         PeripheriqueTPPA *robotTPPA = new PeripheriqueTPPA(info.name(),
00346             info.address().toString(), this);
00347         peripheriquesTPPA.append(robotTPPA);
00348         peripheriqueRobotTPPADetecte = true;
00349         emit detecte();
00350     }
00351     if(info.name() == (NOM_BLUETOOTH_ECRAN + QString::number(
00352         numeroTable)))
00353     {
00354         qDebug() << Q_FUNC_INFO << "périphérique Ecran TPPA découvert" << info.name();
00355         /*PeripheriqueTPPA *ecranTPPA = new PeripheriqueTPPA(info.name(), info.address().toString(), this);
00356         peripheriquesTPPA.append(ecranTPPA);
00357         peripheriqueEcranTPPADetecte = true;
00358         emit detecte();*/
00359     }
00360     // 3 sur 3
00361     if(peripheriqueEcranTPPAConnecte &&
00362         peripheriqueRobotTPPAConnecte &&
00363         peripheriqueTableTPPAConnecte)
00364     {
00365         arreterRecherche();
00366     }
00367     // 2 sur 3
00368     /*if(peripheriqueRobotTPPADetecte && peripheriqueTableTPPADetecte)
00369     {
00370         arreterRecherche();
00371     }
00372     // 1 sur 3
00373     /*if(peripheriqueRobotTPPADetecte)
00374     {
00375         arreterRecherche();
00376     }
00377 }

```

9.2.4.2 arreter()

```
void CommunicationBluetooth::arreter ( )
```

Références [serveur](#), [serviceInfo](#), et [socket](#).

Référencé par [~CommunicationBluetooth\(\)](#).

```
00088 {
00089     if (!serveur)
00090         return;
00091     serviceInfo.unregisterService();
00092
00093     if (socket)
00094     {
00095         if (socket->isOpen())
00096             socket->close();
00097         delete socket;
00098         socket = nullptr;
00099     }
00100
00101     delete serveur;
00102     serveur = nullptr;
00103 }
00104 }
```

9.2.4.3 arreterRecherche()

```
void CommunicationBluetooth::arreterRecherche ( )
```

Références [discoveryAgentDevice](#), [estValide\(\)](#), et [etatRecherche](#).

Référencé par [ajouterPeripheriqueTTPA\(\)](#).

```
00088 {
00089     if (!estValide())
00090         return;
00091     qDebug() << Q_FUNC_INFO;
00092     if (!etatRecherche)
00093         return;
00094     if (discoveryAgentDevice != NULL && discoveryAgentDevice->
00095         isActive())
00096     {
00097         discoveryAgentDevice->stop();
00098     }
00099 }
```

9.2.4.4 arreterSeance()

```
bool CommunicationBluetooth::arreterSeance ( )
```

Renvoie

bool vrai si la trame a pu être envoyé au périphérique TTPA

Références [envoyerPeripheriqueTTPA\(\)](#), [etatSeance](#), [NOM_BLUETOOTH_ECRAN](#), [NOM_BLUETOOTH_ROBOT](#), [NOM_BLUETOOTH_TABLE](#), et [numeroTable](#).

```
00253 {
00254     qDebug() << Q_FUNC_INFO;
00255     QString trameArret = ":RESET*00\r\n";
00256
00257     envoyerPeripheriqueTTPA(NOM_BLUETOOTH_ROBOT + QString::number
00258         (numeroTable), trameArret);
00259     envoyerPeripheriqueTTPA(NOM_BLUETOOTH_ECRAN + QString::number
00260         (numeroTable), trameArret);
00261     envoyerPeripheriqueTTPA(NOM_BLUETOOTH_TABLE + QString::number
00262         (numeroTable), trameArret);
00263     etatSeance = EtatDeLaSeance::Initial;
00264     return true;
00265 }
```

9.2.4.5 connecte [1/2]

```
void CommunicationBluetooth::connecte ( ) [signal]
```

Référencé par [connecterPeripheriquesTTPA\(\)](#), [deconnecterPeripheriquesTTPA\(\)](#), [nouveauClient\(\)](#), et [recevoirConnecte\(\)](#).

9.2.4.6 connecte [2/2]

```
void CommunicationBluetooth::connecte ( ) [signal]
```

9.2.4.7 connecterPeripheriquesTTPA()

```
void CommunicationBluetooth::connecterPeripheriquesTTPA ( )
```

Références [connecte\(\)](#), [PeripheriqueTTPA : :connecter\(\)](#), [deconnecte\(\)](#), [PeripheriqueTTPA : :estConnecte\(\)](#), [PeripheriqueTTPA : :getNom\(\)](#), [peripheriquesTTPA](#), [recevoirConnecte\(\)](#), [recevoirDeconnecte\(\)](#), et [recevoirTramePeripheriqueTTPA\(\)](#).

```
00448 {
00449     qDebug() << Q_FUNC_INFO << peripheriquesTTPA.size();
00450
00451     // liste les appareils TTPA détectés
00452     for(int i=0; i < peripheriquesTTPA.size(); i++)
00453     {
00454         PeripheriqueTTPA *peripheriqueTTPA = (PeripheriqueTTPA *)
peripheriquesTTPA[i];
00455         if(!peripheriqueTTPA->estConnecte())
00456         {
00457             qDebug() << Q_FUNC_INFO << peripheriqueTTPA << peripheriqueTTPA->
getNom();
00458             peripheriqueTTPA->connecter();
00459
00460             connect(peripheriqueTTPA, SIGNAL(trameRecue(QString,QString)), this, SLOT(
recevoirTramePeripheriqueTTPA(QString,QString)));
00461             connect(peripheriqueTTPA, SIGNAL(connecte()), this, SLOT(
recevoirConnecte()));
00462             connect(peripheriqueTTPA, SIGNAL(deconnecte()), this, SLOT(
recevoirDeconnecte()));
00463         }
00464     }
00465 }
```

9.2.4.8 decoderTrame()

```
void CommunicationBluetooth::decoderTrame (
    QString trame ) [private]
```

Paramètres

<i>trame</i>	QString la trame ecran-tpa-1
--------------	------------------------------

Démarrage \$ecran-tpa-1 :START;JULIEN;2;1*XX

	+ - Zone Robot

| +— Zone Objectif +----- Nom **Joueur**

\$ecran-ttpa-1 :STAT ;2*XX || +----- Numéro Zone Impact

Références **departSeance()**, **erreurRobot()**, **etatSeance**, **finSeance()**, **impactZone()**, **mettreEnPauseSeance()**, **repriseSeance()**, et **resetSeance()**.

Référéncé par **socketConnected()**, **socketDisconnected()**, et **socketReadyRead()**.

```

00162 {
00163     QStringList trames;
00164     trame.remove("\r\n");
00165     qDebug() << Q_FUNC_INFO << "trame" << trame;
00166
00167     const QString typeTrame = "ecran-ttpa";
00168     QString donnees;
00169     QString nomJoueur;
00170     QString zoneRobot;
00171     QString zoneObjectif;
00172     QString numeroZone;
00173     QString infoDuRobot;
00174
00175     if(trame.startsWith("$" + typeTrame))
00176     {
00191         donnees = trame.section(':', 1, 1); // START;JULIEN;2;1*XX
00192         if(trame.contains("START") && (etatSeance == EtatSeance::Initial ||
etatSeance == EtatSeance::Terminee))
00193         {
00194             etatSeance = EtatSeance::EnCours;
00195             // START;JULIEN;2;1*XX
00196             nomJoueur = donnees.section(';', 1, 1);
00197             zoneObjectif = donnees.section(';', 2, 2);
00198             zoneRobot = donnees.section(';', 3, 3).at(0);
00199             qDebug() << Q_FUNC_INFO << "nomJoueur" << nomJoueur << "zoneObjectif" << zoneObjectif << "
zoneRobot" << zoneRobot;
00200             emit departSeance(nomJoueur, zoneObjectif, zoneRobot);
00201         }
00202         else if(trame.contains("STAT") && etatSeance == EtatSeance::EnCours)
00203         {
00204             // STAT;2*XX
00205             numeroZone = donnees.section(';', 1, 1).at(0);
00206             qDebug() << Q_FUNC_INFO << "numeroZone" << numeroZone;
00207             emit impactZone(numeroZone);
00208         }
00209         else if(trame.contains("PAUSE") && etatSeance == EtatSeance::EnCours)
00210         {
00211             etatSeance = EtatSeance::EnPause;
00212             //PAUSE*XX
00213             emit mettreEnPauseSeance();
00214         }
00215         else if(trame.contains("RESUME") && etatSeance == EtatSeance::EnPause)
00216         {
00217             etatSeance = EtatSeance::EnCours;
00218             //RESUME*XX
00219             emit repriseSeance();
00220         }
00221         else if(trame.contains("END") && etatSeance == EtatSeance::EnCours)
00222         {
00223             etatSeance = EtatSeance::Terminee;
00224             //END*XX
00225             emit finSeance();
00226         }
00227         else if(trame.contains("RESET") && (etatSeance == EtatSeance::EnCours ||
etatSeance == EtatSeance::EnPause || etatSeance == EtatSeance::Terminee))
00228         {
00229             etatSeance = EtatSeance::Initial;
00230             //RESET*XX
00231             emit resetSeance();
00232         }
00233         else if(trame.contains("INFO") && etatSeance == EtatSeance::EnCours)
00234         {
00235             etatSeance = EtatSeance::EnPause;
00236             //INFO*XX
00237             infoDuRobot = donnees.section(';', 1, 1);
00238             QString remove(20,33);
00239             //QString s = "Bourrage du robot ou absence de balles*00";
00240             infoDuRobot.remove(38, 41);
00241             emit erreurRobot(infoDuRobot);
00242             qDebug() << Q_FUNC_INFO << "infoDuRobot" << infoDuRobot;
00243         }
00244     }
00245 }

```

9.2.4.9 deconnecte

```
void CommunicationBluetooth::deconnecte ( ) [signal]
```

Référencé par [connecterPeripheriquesTTPA\(\)](#), [deconnecterPeripheriquesTTPA\(\)](#), et [recevoirDeconnecte\(\)](#).

9.2.4.10 deconnecterPeripheriquesTTPA()

```
void CommunicationBluetooth::deconnecterPeripheriquesTTPA ( )
```

Références [connecte\(\)](#), [deconnecte\(\)](#), [PeripheriqueTTPA : :deconnecter\(\)](#), [PeripheriqueTTPA : :getNom\(\)](#), [peripheriquesTTPA](#), [recevoirConnecte\(\)](#), [recevoirDeconnecte\(\)](#), et [recevoirTramePeripheriqueTTPA\(\)](#).

Référencé par [CommunicationBluetooth\(\)](#).

```
00491 {
00492     qDebug() << Q_FUNC_INFO;
00493
00494     // liste les appareils TTPA détectés
00495     for(int i=0; i < peripheriquesTTPA.size(); i++)
00496     {
00497         PeripheriqueTTPA *peripheriqueTTPA = (PeripheriqueTTPA *)
peripheriquesTTPA[i];
00498         qDebug() << Q_FUNC_INFO << "déconnecter" << peripheriqueTTPA->getNom();
00499         disconnect(peripheriqueTTPA, SIGNAL(trameRecue(QString,QString)), this, SLOT(
recevoirTramePeripheriqueTTPA(QString,QString)));
00500         peripheriqueTTPA->deconnecter();
00501         disconnect(peripheriqueTTPA, SIGNAL(connecte()), this, SLOT(
recevoirConnecte()));
00502         disconnect(peripheriqueTTPA, SIGNAL(deconnecte()), this, SLOT(
recevoirDeconnecte()));
00503     }
00504 }
```

9.2.4.11 demarrer()

```
void CommunicationBluetooth::demarrer ( )
```

Références [nouveauClient\(\)](#), [serveur](#), [serviceInfo](#), [serviceNom\(\)](#), et [serviceUuid\(\)](#).

Référencé par [Ttpalhm : :Ttpalhm\(\)](#).

```
00071 {
00072     if (!serveur)
00073     {
00074         serveur = new QBluetoothServer(QBluetoothServiceInfo::RfcommProtocol, this);
00075         connect(serveur, SIGNAL(newConnection()), this, SLOT(
nouveauClient()));
00076
00077         QBluetoothUuid uuid = QBluetoothUuid(serviceUuid);
00078         serviceInfo = serveur->listen(uuid, serviceNom);
00079         qDebug() << Q_FUNC_INFO << "serviceInfo" << serviceInfo;
00080     }
00081 }
```

9.2.4.12 demarrerSeance()

```
bool CommunicationBluetooth::demarrerSeance (
    QString nomJoueur,
    QString zoneObjectif,
    QString zoneRobot,
    int frequenceBalle,
    int nbBalles,
    int effet,
    int puissance,
    bool rotation )
```

Références [demarrerSeanceEcran\(\)](#), [demarrerSeanceRobot\(\)](#), [demarrerSeanceTable\(\)](#), [etatSeance](#), et [Initial](#).

```
00156 {
00157     if(etatSeance == Initial)
00158     {
00159         demarrerSeanceRobot(frequenceBalle, nbBalles, effet, puissance, rotation);
00160         demarrerSeanceEcran(nomJoueur, zoneObjectif, zoneRobot);
00161         demarrerSeanceTable(nbBalles);
00162         etatSeance = EtatDeLaSeance::EnCours;
00163         return true;
00164     }
00165     return false;
00166 }
```

9.2.4.13 demarrerSeanceEcran()

```
bool CommunicationBluetooth::demarrerSeanceEcran (
    QString nomJoueur,
    QString zoneObjectif,
    QString zoneRobot )
```

Paramètres

<i>nomJoueur</i>	QString nom du joueur
<i>zoneObjectif</i>	QString trame respectant le protocole TTPA
<i>zoneRobot</i>	QString trame respectant le protocole TTPA

Renvoie

bool vrai si la trame a pu être envoyé au périphérique TTPA

Références [envoyerPeripheriqueTTPA\(\)](#), [NOM_BLUETOOTH_ECRAN](#), et [numeroTable](#).

Référencé par [demarrerSeance\(\)](#).

```
00189 {
00190     qDebug() << Q_FUNC_INFO << NOM_BLUETOOTH_ECRAN + QString::number(
00191         numeroTable) << nomJoueur << zoneObjectif << zoneRobot;
00192     //QString trameReset = ":RESET*00\r\n";
00193     //envoyerPeripheriqueTTPA(NOM_BLUETOOTH_ECRAN + QString::number(numeroTable), trameReset);
00194
00195     QString trameDemarrageEcran = ":START;" + nomJoueur + ";" + zoneObjectif + ";" + zoneRobot + "*00\r\n";
00196     envoyerPeripheriqueTTPA(NOM_BLUETOOTH_ECRAN + QString::number
00197         (numeroTable), trameDemarrageEcran);
00198
00199     return true;
00200 }
```

9.2.4.14 demarrerSeanceRobot()

```
bool CommunicationBluetooth::demarrerSeanceRobot (
    int  frequenceBalle,
    int  nbBalles,
    int  effet,
    int  puissance,
    bool rotation )
```

Paramètres

<i>frequenceBalle</i>	int nom du périphérique Bluetooth
<i>nbBalles</i>	int trame respectant le protocole TTPA
<i>effet</i>	int trame respectant le protocole TTPA
<i>puissance</i>	int trame respectant le protocole TTPA
<i>rotation</i>	bool trame respectant le protocole TTPA

Renvoie

bool vrai si la trame a pu être envoyé au périphérique TTPA

Références [envoyerPeripheriqueTTPA\(\)](#), [NOM_BLUETOOTH_ROBOT](#), et [numeroTable](#).

Référencé par [demarrerSeance\(\)](#).

```
00169 {
00170     qDebug() << Q_FUNC_INFO << NOM_BLUETOOTH_ROBOT + QString::number(
        numeroTable) << frequenceBalle << nbBalles << effet << puissance << rotation;
00171
00172     QString QfrequenceBalle = QString::number(frequenceBalle);
00173     QString QnbBalles = QString::number(nbBalles);
00174     QString Qeffet = QString::number(effet);
00175     QString Qpuissance = QString::number(puissance);
00176     QString Qrotation = QString::number(rotation);
00177
00178     QString trameReset = ":RESET*00\r\n";
00179     //envoyerPeripheriqueTTPA(NOM_BLUETOOTH_ROBOT + QString::number(numeroTable), trameReset);
00180
00181     QString trameDemarrageRobot = ":START;" + QfrequenceBalle + ";" + QnbBalles + ";" + Qeffet + ";" +
        Qpuissance + ";" + Qrotation + "*00\r\n";
00182
00183     envoyerPeripheriqueTTPA(NOM_BLUETOOTH_ROBOT + QString::number
        (numeroTable), trameDemarrageRobot);
00184
00185     return true;
00186 }
```

9.2.4.15 demarrerSeanceTable()

```
bool CommunicationBluetooth::demarrerSeanceTable (
    int  nbBalles )
```

Paramètres

<i>nbBalles</i>	int Nombre de balles
-----------------	----------------------

Renvoie

bool vrai si la trame a pu être envoyé au périphérique TTPA

Références [envoyerPeripheriqueTTPA\(\)](#), [NOM_BLUETOOTH_TABLE](#), et [numeroTable](#).

Référencé par [demarrerSeance\(\)](#).

```
00203 {
00204     qDebug() << Q_FUNC_INFO << NOM\_BLUETOOTH\_TABLE + QString::number(
        numeroTable) << nbBalles;
00205
00206     QString trameReset = ":RESET*00\r\n";
00207     envoyerPeripheriqueTTPA(NOM\_BLUETOOTH\_TABLE + QString::number
        (numeroTable), trameReset);
00208
00209     QString QnbBalles = QString::number(nbBalles);
00210     QString trameDemarrageTable = ":START;" + QnbBalles + "*00\r\n";
00211
00212     envoyerPeripheriqueTTPA(NOM\_BLUETOOTH\_TABLE + QString::number
        (numeroTable), trameDemarrageTable);
00213
00214     return true;
00215 }
```

9.2.4.16 departSeance

```
void CommunicationBluetooth::departSeance (
    QString nomJoueur,
    QString zoneObjectif,
    QString zoneRobot ) [signal]
```

Référencé par [decoderTrame\(\)](#).

9.2.4.17 detecte

```
void CommunicationBluetooth::detecte ( ) [signal]
```

Référencé par [ajouterPeripheriqueTTPA\(\)](#), [rechercherPeripheriquesTTPA\(\)](#), et [rechercheTerminee\(\)](#).

9.2.4.18 deviceConnected

```
void CommunicationBluetooth::deviceConnected (
    const QBluetoothAddress & adresse ) [slot]
```

Références [monBluetooth](#).

Référencé par [CommunicationBluetooth\(\)](#).

```
00282 {
00283     qDebug() << Q_FUNC_INFO << adresse << adresse.toString() <<
        monBluetooth.pairingStatus(adresse);
00284 }
```

9.2.4.19 deviceDisconnected

```
void CommunicationBluetooth::deviceDisconnected (
    const QBluetoothAddress & adresse ) [slot]
```

Références [adresse](#).

Référencé par [CommunicationBluetooth\(\)](#).

```
00287 {
00288     qDebug() << Q_FUNC_INFO << adresse;
00289 }
```

9.2.4.20 ecranDetecte

```
void CommunicationBluetooth::ecranDetecte ( ) [signal]
```

9.2.4.21 envoyerErreurRobot()

```
bool CommunicationBluetooth::envoyerErreurRobot ( )
```

Renvoie

bool vrai si la trame a pu être envoyée au périphérique TTPA

Références [envoyerPeripheriqueTTPA\(\)](#), [NOM_BLUETOOTH_ECRAN](#), et [numeroTable](#).

Référencé par [recevoirTramePeripheriqueTTPA\(\)](#).

```
00297 {
00298     qDebug() << Q_FUNC_INFO;
00299
00300     QString trameErreur = ":INFO;Bourrage du robot ou Absence de balles*00\r\n";
00301
00302     envoyerPeripheriqueTTPA(NOM_BLUETOOTH_ECRAN + QString::number
00303                             (numeroTable), trameErreur);
00304
00305     return true;
00306 }
```

9.2.4.22 envoyerPeripheriqueTTPA()

```
bool CommunicationBluetooth::envoyerPeripheriqueTTPA (
    QString nom,
    QString message )
```

Paramètres

<i>nom</i>	QString nom du périphérique Bluetooth
<i>message</i>	QString trame respectant le protocole TTPA

Renvoie

bool vrai si la trame a pu être envoyé au périphérique TTPA

Références [PeripheriqueTTPA :envoyer\(\)](#), [PeripheriqueTTPA :getNom\(\)](#), [nom](#), et [peripheriquesTTPA](#).

Référencé par [arreterSeance\(\)](#), [demarrerSeanceEcran\(\)](#), [demarrerSeanceRobot\(\)](#), [demarrerSeanceTable\(\)](#), [envoyerErreurRobot\(\)](#), [envoyerZoneImpactee\(\)](#), [finirSeance\(\)](#), et [reprendreLaSeance\(\)](#).

```
00140 {
00141     qDebug() << Q_FUNC_INFO << "nom" << nom << "message" << message;
00142     // liste les appareils TTPA détectés
00143     for(int i=0; i < peripheriquesTTPA.size(); i++)
00144     {
00145         if(((PeripheriqueTTPA *)peripheriquesTTPA[i])->getNom() ==
nom)
00146         {
00147             PeripheriqueTTPA *peripheriqueTTPA = (
PeripheriqueTTPA *)peripheriquesTTPA[i];
00148             qDebug() << Q_FUNC_INFO << "envoyer" << peripheriqueTTPA->getNom();
00149             return peripheriqueTTPA->envoyer(nom + message);
00150         }
00151     }
00152     return false;
00153 }
```

9.2.4.23 envoyerZoneImpactee()

```
bool CommunicationBluetooth::envoyerZoneImpactee (
    QString zone )
```

Paramètres

<i>zone</i>	QString zone impactée
-------------	-----------------------

Renvoie

bool vrai si la trame a pu être envoyée au périphérique TTPA

Références [envoyerPeripheriqueTTPA\(\)](#), [NOM_BLUETOOTH_ECRAN](#), et [numeroTable](#).

Référencé par [recevoirTramePeripheriqueTTPA\(\)](#).

```
00286 {
00287     qDebug() << Q_FUNC_INFO;
00288
00289     QString trameStatistique = ":STAT;" + zone + "*00\r\n";
00290
00291     envoyerPeripheriqueTTPA(NOM_BLUETOOTH_ECRAN + QString::number
(numeroTable), trameStatistique);
00292
00293     return true;
00294 }
```

9.2.4.24 erreur

```
void CommunicationBluetooth::erreur ( ) [signal]
```

Référencé par [error\(\)](#), et [rechercheErreur\(\)](#).

9.2.4.25 erreurRobot [1/2]

```
void CommunicationBluetooth::erreurRobot (
    QString infoDuRobot ) [signal]
```

9.2.4.26 erreurRobot [2/2]

```
void CommunicationBluetooth::erreurRobot ( ) [signal]
```

Référéncé par [decoderTrame\(\)](#), et [recevoirTramePeripheriqueTTPA\(\)](#).

9.2.4.27 error

```
void CommunicationBluetooth::error (
    QBluetoothLocalDevice::Error erreur ) [slot]
```

Références [erreur\(\)](#).

Référéncé par [CommunicationBluetooth\(\)](#), et [socketErreur\(\)](#).

```
00292 {
00293     qDebug() << Q_FUNC_INFO << erreur;
00294 }
```

9.2.4.28 estValide()

```
bool CommunicationBluetooth::estValide ( )
```

Renvoie

Vrai si l'appareil est conforme aux normes "TTPA"

Références [peripheriqueBluetoothLocal](#).

Référéncé par [arreterRecherche\(\)](#), et [rechercherPeripheriquesTTPA\(\)](#).

```
00101 {
00102     return peripheriqueBluetoothLocal.isValid();
00103 }
```

9.2.4.29 finirSeance()

```
bool CommunicationBluetooth::finirSeance ( )
```

Renvoie

bool vrai si la trame a pu être envoyée au périphérique TTPA

Références [EnCours](#), [envoyerPeripheriqueTTPA\(\)](#), [etatSeance](#), [NOM_BLUETOOTH_ECRAN](#), [NOM_BLUETOOTH_ROBOT](#), [NOM_←_BLUETOOTH_TABLE](#), [numeroTable](#), [seanceFinie\(\)](#), et [seanceTerminee](#).

Référencé par [recevoirTramePeripheriqueTTPA\(\)](#).

```
00267 {
00268     qDebug() << Q_FUNC_INFO;
00269
00270     QString trameFin = ":END*00\r\n";
00271
00272     if(etatSeance == EnCours)
00273     {
00274         envoyerPeripheriqueTTPA(NOM_BLUETOOTH_ROBOT +
00275     QString::number(numeroTable), trameFin);
00276         envoyerPeripheriqueTTPA(NOM_BLUETOOTH_ECRAN +
00277     QString::number(numeroTable), trameFin);
00278         envoyerPeripheriqueTTPA(NOM_BLUETOOTH_TABLE +
00279     QString::number(numeroTable), trameFin);
00280         seanceTerminee = true;
00281         emit seanceFinie();
00282         etatSeance = EtatDeLaSeance::Initial;
00283         return true;
00284     }
00285     return false;
00286 }
```

9.2.4.30 finSeance

```
void CommunicationBluetooth::finSeance ( ) [signal]
```

Référencé par [decoderTrame\(\)](#).

9.2.4.31 getAdressePeripheriqueBluetoothLocal()

```
QString CommunicationBluetooth::getAdressePeripheriqueBluetoothLocal ( )
```

Références [adressePeripheriqueBluetoothLocal](#).

```
00111 {
00112     return adressePeripheriqueBluetoothLocal;
00113 }
```

9.2.4.32 getEtatSeance() [1/2]

```
CommunicationBluetooth::EtatSeance CommunicationBluetooth::getEtatSeance ( )
```

Références [etatSeance](#).

```
00260 {
00261     return etatSeance;
00262 }
```

9.2.4.33 getEtatSeance() [2/2]

EtatDeLaSeance CommunicationBluetooth::getEtatSeance ()

9.2.4.34 getNomPeripheriqueBluetoothLocal()

QString CommunicationBluetooth::getNomPeripheriqueBluetoothLocal ()

Références [nomPeripheriqueBluetoothLocal](#).

```
00106 {
00107     return nomPeripheriqueBluetoothLocal;
00108 }
```

9.2.4.35 getPeripheriquesTPPA()

QVariant CommunicationBluetooth::getPeripheriquesTPPA ()

Références [peripheriquesTPPA](#).

```
00116 {
00117     return QVariant::fromValue(peripheriquesTPPA);
00118 }
```

9.2.4.36 getPeripheriqueTPPA()

PeripheriqueTPPA * CommunicationBluetooth::getPeripheriqueTPPA (
 QString nom)

Références [PeripheriqueTPPA : :getNom\(\)](#), [nom](#), et [peripheriquesTPPA](#).

```
00121 {
00122     bool trouve = false;
00123     PeripheriqueTPPA *peripheriqueTPPA = nullptr;
00124
00125     // liste les appareils TPA détectés
00126     for(int i=0; i < peripheriquesTPPA.size() && !trouve; i++)
00127     {
00128         if(((PeripheriqueTPPA *)peripheriquesTPPA[i])->getNom() ==
nom)
00129         {
00130             peripheriqueTPPA = (PeripheriqueTPPA *)
peripheriquesTPPA[i];
00131             qDebug() << Q_FUNC_INFO << "return" << peripheriqueTPPA->getNom();
00132             trouve = true;
00133         }
00134     }
00135     return peripheriqueTPPA;
00136 }
```

9.2.4.37 getZoneObjectif()

QString CommunicationBluetooth::getZoneObjectif (
 QString zoneObjectif)

Paramètres

<i>zoneObjectif</i>	QString numéro de la zone objectif choisie par le joueur
---------------------	--

Renvoie

QString numéro de la zone objectif

```

00308 {
00309     QString numeroZoneObjectif = zoneObjectif;
00310     QStringRef b(&numeroZoneObjectif, 5, 1);
00311     numeroZoneObjectif = b.toString();
00312     qDebug() << Q_FUNC_INFO << "Numero zone objectif : " << numeroZoneObjectif;
00313
00314     return numeroZoneObjectif;
00315 }
```

9.2.4.38 getZoneRobot()

```

QString CommunicationBluetooth::getZoneRobot (
    QString zoneRobot )
```

Paramètres

<i>zoneRobot</i>	QString numéro de la zone robot choisie par le joueur
------------------	---

Renvoie

QString numéro de la zone robot

```

00318 {
00319     QString numeroZoneRobot = zoneRobot;
00320     QStringRef b(&numeroZoneRobot, 5, 1);
00321     numeroZoneRobot = b.toString();
00322     qDebug() << Q_FUNC_INFO << "Numero zone robot : " << numeroZoneRobot;
00323
00324     return numeroZoneRobot;
00325 }
```

9.2.4.39 impactZone

```

void CommunicationBluetooth::impactZone (
    QString numeroZone ) [signal]
```

Référencé par [decoderTrame\(\)](#).

9.2.4.40 mettreEnPauseSeance [1/2]

```
bool CommunicationBluetooth::mettreEnPauseSeance ( ) [signal]
```

Renvoi

bool vrai si la trame a pu être envoyé au périphérique TTPA

Référencé par [decoderTrame\(\)](#), et [recevoirTramePeripheriqueTTPA\(\)](#).

```
00218 {
00219     qDebug() << Q_FUNC_INFO;
00220
00221     QString tramePause = ":PAUSE*00\r\n";
00222
00223     if(etatSeance == EtatDeLaSeance::EnCours)
00224     {
00225         envoyerPeripheriqueTTPA(NOM_BLUETOOTH_ROBOT +
00226             QString::number(numeroTable), tramePause);
00227         envoyerPeripheriqueTTPA(NOM_BLUETOOTH_ECRAN +
00228             QString::number(numeroTable), tramePause);
00229         envoyerPeripheriqueTTPA(NOM_BLUETOOTH_TABLE +
00230             QString::number(numeroTable), tramePause);
00231         etatSeance = EtatDeLaSeance::EnPause;
00232         return true;
00233     }
00234     return false;
00235 }
```

9.2.4.41 mettreEnPauseSeance() [2/2]

```
Q_INVOKABLE bool CommunicationBluetooth::mettreEnPauseSeance ( )
```

9.2.4.42 nouveauClient

```
void CommunicationBluetooth::nouveauClient ( ) [slot]
```

Références [connecte\(\)](#), [etatConnexion](#), [serveur](#), [socket](#), [socketDisconnected\(\)](#), et [socketReadyRead\(\)](#).

Référencé par [demarrer\(\)](#).

```
00265 {
00266     // on récupère la socket
00267     socket = serveur->nextPendingConnection();
00268     if (!socket)
00269         return;
00270
00271     connect(socket, SIGNAL(disconnected()), this, SLOT(socketDisconnected()));
00272     connect(socket, SIGNAL(readyRead()), this, SLOT(socketReadyRead()));
00273     //connect(socket, SIGNAL(error(QBluetoothSocket::SocketError)), this,
00274     //connect(socket, SIGNAL(stateChanged(QBluetoothSocket::SocketState)), this,
00275     //connect(socket, SIGNAL(stateChanged(QBluetoothSocket::SocketState)));
00276     etatConnexion = true;
00277     qDebug() << Q_FUNC_INFO;
00278     emit connecte();
00279 }
```


9.2.4.43 numeroTableUpdated

```
void CommunicationBluetooth::numeroTableUpdated ( ) [signal]
```

9.2.4.44 peripheriquesTTPAUpdated

```
void CommunicationBluetooth::peripheriquesTTPAUpdated ( ) [signal]
```

Référencé par [rechercherPeripheriquesTTPA\(\)](#), et [rechercheTerminee\(\)](#).

9.2.4.45 recevoirConnecte

```
void CommunicationBluetooth::recevoirConnecte ( ) [slot]
```

Références [connecte\(\)](#), [PeripheriqueTTPA : :getNom\(\)](#), [peripheriqueEcranTTPAConnecte](#), [peripheriqueRobotTTPAConnecte](#), et [peripheriqueTableTTPAConnecte](#).

Référencé par [connecterPeripheriquesTTPA\(\)](#), et [deconnecterPeripheriquesTTPA\(\)](#).

```
00426 {
00427     PeripheriqueTTPA *peripheriqueTTPA = qobject_cast<
    PeripheriqueTTPA*>(sender());
00428     qDebug() << Q_FUNC_INFO << peripheriqueTTPA->getNom();
00429
00430     if(peripheriqueTTPA->getNom().contains("table-ttpa"))
00431     {
00432         peripheriqueTableTTPAConnecte = true;
00433         emit connecte();
00434     }
00435     else if(peripheriqueTTPA->getNom().contains("robot-ttpa"))
00436     {
00437         peripheriqueRobotTTPAConnecte = true;
00438         emit connecte();
00439     }
00440     else if(peripheriqueTTPA->getNom().contains("ecran-ttpa"))
00441     {
00442         peripheriqueEcranTTPAConnecte = true;
00443         emit connecte();
00444     }
00445 }
```

9.2.4.46 recevoirDeconnecte

```
void CommunicationBluetooth::recevoirDeconnecte ( ) [slot]
```

Références [deconnecte\(\)](#), [PeripheriqueTTPA : :getNom\(\)](#), [peripheriqueEcranTTPADeconnecte](#), [peripheriqueRobotTTPADeconnecte](#), et [peripheriqueTableTTPADeconnecte](#).

Référencé par [connecterPeripheriquesTTPA\(\)](#), et [deconnecterPeripheriquesTTPA\(\)](#).

```
00468 {
00469     PeripheriqueTTPA *peripheriqueTTPA = qobject_cast<
    PeripheriqueTTPA*>(sender());
00470     qDebug() << Q_FUNC_INFO << peripheriqueTTPA->getNom();
00471
00472     if(peripheriqueTTPA->getNom().contains("table-ttpa"))
00473     {
00474         peripheriqueTableTTPADeconnecte = true;
00475         emit deconnecte();
00476     }
00477     else if(peripheriqueTTPA->getNom().contains("robot-ttpa"))
00478     {
00479         peripheriqueRobotTTPADeconnecte = true;
00480         emit deconnecte();
00481     }
00482     else if(peripheriqueTTPA->getNom().contains("ecran-ttpa"))
00483     {
00484         peripheriqueEcranTTPADeconnecte = true;
00485         emit deconnecte();
00486     }
00487
00488 }
```

9.2.4.47 recevoirTramePeripheriqueTTPA

```
void CommunicationBluetooth::recevoirTramePeripheriqueTTPA (
    QString nomPeripherique,
    QString trame ) [slot]
```

Références [envoyerErreurRobot\(\)](#), [envoyerZoneImpactee\(\)](#), [erreurRobot\(\)](#), [erreurRobotDetecte](#), [etatSeance](#), [finirSeance\(\)](#), et [mettreEnPauseSeance\(\)](#).

Référencé par [connecterPeripheriquesTTPA\(\)](#), et [deconnecterPeripheriquesTTPA\(\)](#).

```
00387 {
00388     qDebug() << Q_FUNC_INFO << "nomPeripherique" << nomPeripherique << "trame" << trame;
00389
00390     if(etatSeance != EtatDeLaSeance::EnCours)
00391         return;
00392
00393     if(trame.contains("IMPACT"))
00394     {
00395         QStringRef traitementImpactTable(&trame,21, 1);
00396         QString donneesImpact = traitementImpactTable.toString();
00397         int zoneImpact = donneesImpact.toInt();
00398         donneesImpact = QString::number(zoneImpact-1);
00399         qDebug() << Q_FUNC_INFO << "donneesImpact" << donneesImpact;
00400         envoyerZoneImpactee(donneesImpact);
00401     }
00402     else if(trame.contains("ERROR")) // ROBOT
00403     {
00404         QStringRef traitementBourrageRobot(&trame, 21, 1);
00405         QStringRef traitementAbsenceDeBalleRobot(&trame, 23, 1);
00406         QStringRef traitementErreurParametresRobot(&trame, 25, 1);
00407         QString bourrageDeBalle = traitementBourrageRobot.toString();
00408         QString absenceDeBalle = traitementAbsenceDeBalleRobot.toString();
00409         QString erreurParametres = traitementErreurParametresRobot.toString();
00410         qDebug() << Q_FUNC_INFO << "bourrageBalle" << bourrageDeBalle << "absenceBalle" << absenceDeBalle <
00411         < "erreurParametre" << erreurParametres;
00412         // 1. signaler à l'IHM
00413         erreurRobotDetecte = true;
00414         emit erreurRobot();
00415         // 2. envoyer trame à l'écran
00416         envoyerErreurRobot();
00417         // A voir : passage à l'état ERREUR ou PAUSE ?
00418         mettreEnPauseSeance();
00419     }
00420     else if(trame.contains("FINISH"))
00421     {
00422         finirSeance();
00423     }
00424 }
```

9.2.4.48 recherche

```
void CommunicationBluetooth::recherche ( ) [signal]
```

Référencé par [rechercherPeripheriquesTTPA\(\)](#), et [rechercheTerminee\(\)](#).

9.2.4.49 rechercheErreur

```
void CommunicationBluetooth::rechercheErreur (
    QBluetoothDeviceDiscoveryAgent::Error erreur ) [slot]
```

Références [erreur\(\)](#).

Référencé par [CommunicationBluetooth\(\)](#).

```
00382 {
00383     qDebug() << Q_FUNC_INFO << erreur;
00384 }
```

9.2.4.50 rechercherPeripheriquesTTPA()

CommunicationBluetooth::rechercherPeripheriquesTTPA ()

Références [detecte\(\)](#), [discoveryAgentDevice](#), [estValide\(\)](#), [etatRecherche](#), [NOM_BLUETOOTH_ECRAN](#), [numeroTable](#), [peripheriqueEcranTTPADetecte](#), [peripheriqueRobotTTPADetecte](#), [peripheriquesTTPA](#), [peripheriquesTTPAUpdated\(\)](#), [peripheriqueTableTTPADetecte](#), et [recherche\(\)](#).

```
00057 {
00058     if(!estValide())
00059         return;
00060     qDebug() << Q_FUNC_INFO;
00061
00062     qDeleteAll(peripheriquesTTPA);
00063     peripheriquesTTPA.clear();
00064     peripheriqueTableTTPADetecte = false;
00065     peripheriqueRobotTTPADetecte = false;
00066     peripheriqueEcranTTPADetecte = false;
00067     emit detecte();
00068     emit peripheriquesTTPAUpdated();
00069
00070     PeripheriqueTTPA *ecranTTPA = new PeripheriqueTTPA(
00071         NOM_BLUETOOTH_ECRAN + QString::number(numeroTable), "B8:27:EB:13:DA:A5", this
00072     );
00073     peripheriquesTTPA.append(ecranTTPA);
00074     peripheriqueEcranTTPADetecte = true;
00075     emit detecte();
00076
00077     if(discoveryAgentDevice != NULL)
00078     {
00079         qDebug() << Q_FUNC_INFO << discoveryAgentDevice->isActive();
00080         discoveryAgentDevice->start();
00081         if (discoveryAgentDevice->isActive())
00082         {
00083             etatRecherche = true;
00084             emit recherche();
00085         }
00086     }
00087 }
```

9.2.4.51 rechercheTerminee

void CommunicationBluetooth::rechercheTerminee () [slot]

Références [detecte\(\)](#), [etatRecherche](#), [peripheriquesTTPAUpdated\(\)](#), et [recherche\(\)](#).

Référencé par [CommunicationBluetooth\(\)](#).

```
00374 {
00375     etatRecherche = false;
00376     emit recherche();
00377     emit detecte();
00378     emit peripheriquesTTPAUpdated();
00379 }
```

9.2.4.52 reprendreLaSeance()

```
bool CommunicationBluetooth::reprendreLaSeance ( )
```

Renvoie

bool vrai si la trame a pu être envoyé au périphérique TTPA

Références [envoyerPeripheriqueTTPA\(\)](#), [erreurRobotDetecte](#), [etatSeance](#), [NOM_BLUETOOTH_ECRAN](#), [NOM_BLUETOOTH_ROBOT](#), [NOM_BLUETOOTH_TABLE](#), et [numeroTable](#).

```
00235 {
00236     qDebug() << Q_FUNC_INFO;
00237
00238     QString trameReprise = ":RESUME*00\r\n";
00239
00240     if(etatSeance == EtatDeLaSeance::EnPause)
00241     {
00242         envoyerPeripheriqueTTPA(NOM_BLUETOOTH_ROBOT +
00243             QString::number(numeroTable), trameReprise);
00244         envoyerPeripheriqueTTPA(NOM_BLUETOOTH_ECRAN +
00245             QString::number(numeroTable), trameReprise);
00246         envoyerPeripheriqueTTPA(NOM_BLUETOOTH_TABLE +
00247             QString::number(numeroTable), trameReprise);
00248         etatSeance = EtatDeLaSeance::EnCours;
00249         erreurRobotDetecte = false;
00250         return true;
00251     }
00252     return false;
00253 }
```

9.2.4.53 repriseSeance

```
void CommunicationBluetooth::repriseSeance ( ) [signal]
```

Référencé par [decoderTrame\(\)](#).

9.2.4.54 resetSeance

```
void CommunicationBluetooth::resetSeance ( ) [signal]
```

Référencé par [decoderTrame\(\)](#).

9.2.4.55 robotDetecte

```
void CommunicationBluetooth::robotDetecte ( ) [signal]
```

9.2.4.56 seanceFinie

```
void CommunicationBluetooth::seanceFinie ( ) [signal]
```

Référencé par [finirSeance\(\)](#).

9.2.4.57 socketConnected

```
void CommunicationBluetooth::socketConnected ( ) [slot]
```

Références [connexionErreur](#), [decoderTrame\(\)](#), et [etatConnexion](#).

```
00112 {
00113     qDebug() << Q_FUNC_INFO;
00114     connect(this, SIGNAL(donneesRecues(QString)),this, SLOT(decoderTrame(QString)));
00115     etatConnexion = true;
00116     connexionErreur = false;
00117 }
```

9.2.4.58 socketDisconnected

```
void CommunicationBluetooth::socketDisconnected ( ) [slot]
```

Références [connexionErreur](#), [decoderTrame\(\)](#), et [etatConnexion](#).

Référencé par [nouveauClient\(\)](#).

```
00125 {
00126     qDebug() << Q_FUNC_INFO;
00127     disconnect(this, SIGNAL(donneesRecues(QString)),this, SLOT(decoderTrame(QString)));
00128     etatConnexion = false;
00129     connexionErreur = false;
00130 }
```

9.2.4.59 socketErreur

```
void CommunicationBluetooth::socketErreur (
    QBluetoothSocket::SocketError error ) [slot]
```

Paramètres

<i>error</i>	QBluetoothSocket::SocketError code d'erreur
--------------	---

Références [connexionErreur](#), [error\(\)](#), et [etatConnexion](#).

```
00253 {
00254     qDebug() << Q_FUNC_INFO << "erreur" << error;
00255     etatConnexion = false;
00256     connexionErreur = true;
00257 }
```

9.2.4.60 socketReadyRead

```
void CommunicationBluetooth::socketReadyRead ( ) [slot]
```

Références [decoderTrame\(\)](#), et [socket](#).

Référencé par [nouveauClient\(\)](#).

```
00138 {
00139     QByteArray donneesRecues;
00140
00141     while(socket->canReadLine())
00142     {
00143         donneesRecues += socket->readLine();
00144     }
00145
00146     if(donneesRecues.startsWith("$"))
00147     {
00148         qDebug() << Q_FUNC_INFO << "donneesRecues" << donneesRecues;
00149         decoderTrame(QString(donneesRecues.data()));
00150         donneesRecues.clear();
00151     }
00152 }
```

9.2.4.61 tableDetecte

```
void CommunicationBluetooth::tableDetecte ( ) [signal]
```

9.2.5 Documentation des données membres

9.2.5.1 adresse

```
QString CommunicationBluetooth::adresse [private]
```

Référéncé par [CommunicationBluetooth\(\)](#), et [deviceDisconnected\(\)](#).

9.2.5.2 connexionErreur

```
bool CommunicationBluetooth::connexionErreur [private]
```

Référéncé par [CommunicationBluetooth\(\)](#), [socketConnected\(\)](#), [socketDisconnected\(\)](#), et [socketErreur\(\)](#).

9.2.5.3 discoveryAgentDevice

```
QBluetoothDeviceDiscoveryAgent* CommunicationBluetooth::discoveryAgentDevice [private]
```

Référéncé par [arreterRecherche\(\)](#), [CommunicationBluetooth\(\)](#), et [rechercherPeripheriquesTTPA\(\)](#).

9.2.5.4 etatConnexion

```
bool CommunicationBluetooth::etatConnexion [private]
```

Référéncé par [CommunicationBluetooth\(\)](#), [nouveauClient\(\)](#), [socketConnected\(\)](#), [socketDisconnected\(\)](#), et [socketErreur\(\)](#).

9.2.5.5 `etatSeance` [1/2]

`EtatSeance` `CommunicationBluetooth::etatSeance` [private]

Référencé par `arreterSeance()`, `CommunicationBluetooth()`, `decoderTrame()`, `demarrerSeance()`, `finirSeance()`, `getEtatSeance()`, `recevoirTramePeripheriqueTTPA()`, et `reprendreLaSeance()`.

9.2.5.6 `etatSeance` [2/2]

`EtatDeLaSeance` `CommunicationBluetooth::etatSeance` [private]

9.2.5.7 `monBluetooth`

`QBluetoothLocalDevice` `CommunicationBluetooth::monBluetooth` [private]

Référencé par `CommunicationBluetooth()`, `deviceConnected()`, et `~CommunicationBluetooth()`.

9.2.5.8 `nom`

`QString` `CommunicationBluetooth::nom` [private]

Référencé par `CommunicationBluetooth()`, `envoyerPeripheriqueTTPA()`, et `getPeripheriqueTTPA()`.

9.2.5.9 `peripheriqueBluetoothLocal`

`QBluetoothLocalDevice` `CommunicationBluetooth::peripheriqueBluetoothLocal` [private]

Référencé par `CommunicationBluetooth()`, et `estValide()`.

9.2.5.10 `peripheriquesTTPA`

`QList<QObject*>` `CommunicationBluetooth::peripheriquesTTPA` [private]

9.2.5.11 `serveur`

`QBluetoothServer*` `CommunicationBluetooth::serveur` [private]

Référencé par `arreter()`, `demarrer()`, et `nouveauClient()`.

9.2.5.12 serviceInfo

```
QBluetoothServiceInfo CommunicationBluetooth::serviceInfo [private]
```

Référencé par [arreter\(\)](#), et [demarrer\(\)](#).

9.2.5.13 socket

```
QBluetoothSocket* CommunicationBluetooth::socket [private]
```

Référencé par [arreter\(\)](#), [nouveauClient\(\)](#), et [socketReadyRead\(\)](#).

9.2.6 Documentation des propriétés

9.2.6.1 adressePeripheriqueBluetoothLocal

```
QString CommunicationBluetooth::adressePeripheriqueBluetoothLocal
```

Référencé par [CommunicationBluetooth\(\)](#), et [getAdressePeripheriqueBluetoothLocal\(\)](#).

9.2.6.2 erreurRobotDetecte

```
bool CommunicationBluetooth::erreurRobotDetecte
```

Référencé par [recevoirTramePeripheriqueTTPA\(\)](#), et [reprendreLaSeance\(\)](#).

9.2.6.3 etatRecherche

```
bool CommunicationBluetooth::etatRecherche
```

Référencé par [arreterRecherche\(\)](#), [rechercherPeripheriquesTTPA\(\)](#), et [rechercheTerminee\(\)](#).

9.2.6.4 nomPeripheriqueBluetoothLocal

```
QString CommunicationBluetooth::nomPeripheriqueBluetoothLocal
```

Référencé par [CommunicationBluetooth\(\)](#), et [getNomPeripheriqueBluetoothLocal\(\)](#).

9.2.6.5 numeroTable

```
int CommunicationBluetooth::numeroTable
```

Référencé par [ajouterPeripheriqueTTPA\(\)](#), [arreterSeance\(\)](#), [demarrerSeanceEcran\(\)](#), [demarrerSeanceRobot\(\)](#), [demarrerSeanceTable\(\)](#), [envoyerErreurRobot\(\)](#), [envoyerZoneImpactee\(\)](#), [finirSeance\(\)](#), [rechercherPeripheriquesTTPA\(\)](#), et [reprendreLaSeance\(\)](#).

9.2.6.6 peripheriqueEcranTTPACconnecte

```
bool CommunicationBluetooth::peripheriqueEcranTTPACconnecte
```

Référencé par [ajouterPeripheriqueTTPA\(\)](#), et [recevoirConnecte\(\)](#).

9.2.6.7 peripheriqueEcranTTPADeconnecte

```
bool CommunicationBluetooth::peripheriqueEcranTTPADeconnecte
```

Référencé par [recevoirDeconnecte\(\)](#).

9.2.6.8 peripheriqueEcranTTPADetecte

```
bool CommunicationBluetooth::peripheriqueEcranTTPADetecte
```

Référencé par [rechercherPeripheriquesTTPA\(\)](#).

9.2.6.9 peripheriqueRobotTTPACconnecte

```
bool CommunicationBluetooth::peripheriqueRobotTTPACconnecte
```

Référencé par [ajouterPeripheriqueTTPA\(\)](#), et [recevoirConnecte\(\)](#).

9.2.6.10 peripheriqueRobotTTPADeconnecte

```
bool CommunicationBluetooth::peripheriqueRobotTTPADeconnecte
```

Référencé par [recevoirDeconnecte\(\)](#).

9.2.6.11 peripheriqueRobotTTPADetecte

```
bool CommunicationBluetooth::peripheriqueRobotTTPADetecte
```

Référencé par [ajouterPeripheriqueTTPA\(\)](#), et [rechercherPeripheriquesTTPA\(\)](#).

9.2.6.12 peripheriquesTTPA

`QVariant CommunicationBluetooth::peripheriquesTTPA [read]`

Référencé par [ajouterPeripheriqueTTPA\(\)](#), [CommunicationBluetooth\(\)](#), [connecterPeripheriquesTTPA\(\)](#), [deconnecterPeripheriquesTTPA\(\)](#), [envoyerPeripheriqueTTPA\(\)](#), [getPeripheriquesTTPA\(\)](#), [getPeripheriqueTTPA\(\)](#), et [rechercherPeripheriquesTTPA\(\)](#).

9.2.6.13 peripheriqueTableTTPAConnecte

`bool CommunicationBluetooth::peripheriqueTableTTPAConnecte`

Référencé par [ajouterPeripheriqueTTPA\(\)](#), et [recevoirConnecte\(\)](#).

9.2.6.14 peripheriqueTableTTPADeconnecte

`bool CommunicationBluetooth::peripheriqueTableTTPADeconnecte`

Référencé par [recevoirDeconnecte\(\)](#).

9.2.6.15 peripheriqueTableTTPADetecte

`bool CommunicationBluetooth::peripheriqueTableTTPADetecte`

Référencé par [ajouterPeripheriqueTTPA\(\)](#), et [rechercherPeripheriquesTTPA\(\)](#).

9.2.6.16 seanceTerminee

`bool CommunicationBluetooth::seanceTerminee`

Référencé par [finirSeance\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

- [ecran-TTPA/communicationbluetooth.h](#)
- [ecran-TTPA/communicationbluetooth.cpp](#)

9.3 Référence de la classe FenetreAccueil

La fenêtre d'accueil.

Graphe de collaboration de FenetreAccueil :

FenetreAccueil
<ul style="list-style-type: none"> + recherche + robotDetecte + robotConnecte + robotDeconnecte + ecranDetecte + ecranConnecte + ecranDeconnecte + tableDetecte + tableConnecte + tableDeconnecte + erreurrobotDetectee + seanceTerminee - onRechercheChanged - onRobotDetecteChanged - onRobotConnecteChanged - onRobotDeconnecteChanged - onTableDetecteChanged - onTableConnecteChanged - onTableDeconnecteChanged - onEcranDetecteChanged - onEcranConnecteChanged - onEcranDeconnecteChanged et 22 de plus...

Propriétés

- bool [recherche](#)
- bool [robotDetecte](#)
- bool [robotConnecte](#)
- bool [robotDeconnecte](#)
- bool [ecranDetecte](#)
- bool [ecranConnecte](#)
- bool [ecranDeconnecte](#)
- bool [tableDetecte](#)
- bool [tableConnecte](#)
- bool [tableDeconnecte](#)
- bool [erreurrobotDetectee](#)
- bool [seanceTerminee](#)

Attributs privés

- var [onRechercheChanged](#)
- var [onRobotDetecteChanged](#)
- var [onRobotConnecteChanged](#)
- var [onRobotDeconnecteChanged](#)
- var [onTableDetecteChanged](#)

- var [onTableConnecteChanged](#)
- var [onTableDeconnecteChanged](#)
- var [onEcranDetecteChanged](#)
- var [onEcranConnecteChanged](#)
- var [onEcranDeconnecteChanged](#)
- var [onErreurrobotDetecteeChanged](#)
- var [onSeanceTermineeChanged](#)
- [AlertDialog](#) [alerteNomVide](#)
- [AlertDialog](#) [alertePeripherique](#)
- [Label](#) [messageErreurRobot](#)
- [ComboBox](#) [listeJoueurs](#)
- [TextField](#) [nomJoueur](#)
- [Button](#) [boutonJouer](#)
- [Button](#) [boutonArreter](#)
- [ToolButton](#) [toolButton2](#)
- [Dialog](#) [bluetooth](#)
- [Text](#) [labelNumTable](#)
- [TextField](#) [numTable](#)
- [Switch](#) [switchBluetooth](#)
- [BusyIndicator](#) [indicateur](#)
- [Rectangle](#) [etatTable](#)
- [Rectangle](#) [etatEcran](#)
- [Rectangle](#) [etatRobot](#)
- [Button](#) [table](#)
- [Button](#) [ecran](#)
- [Button](#) [robot](#)
- [Image](#) [logo](#)

9.3.1 Description détaillée

Auteur

Youssef Hammouma

Version

1.4

9.3.2 Documentation des données membres

9.3.2.1 [alerteNomVide](#)

`AlertDialog FenetreAccueil::alerteNomVide [private]`

9.3.2.2 [alertePeripherique](#)

`AlertDialog FenetreAccueil::alertePeripherique [private]`

9.3.2.3 [bluetooth](#)

`Dialog FenetreAccueil::bluetooth [private]`

9.3.2.4 boutonArreter

Button FenetreAccueil::boutonArreter [private]

9.3.2.5 boutonJouer

Button FenetreAccueil::boutonJouer [private]

9.3.2.6 ecran

Button FenetreAccueil::ecran [private]

9.3.2.7 etatEcran

Rectangle FenetreAccueil::etatEcran [private]

9.3.2.8 etatRobot

Rectangle FenetreAccueil::etatRobot [private]

9.3.2.9 etatTable

Rectangle FenetreAccueil::etatTable [private]

9.3.2.10 indicateur

BusyIndicator FenetreAccueil::indicateur [private]

9.3.2.11 labelnumTable

Text FenetreAccueil::labelnumTable [private]

9.3.2.12 listeJoueurs

ComboBox FenetreAccueil::listeJoueurs [private]

9.3.2.13 logo

```
Image FenetreAccueil::logo [private]
```

9.3.2.14 messageErreurRobot

```
Label FenetreAccueil::messageErreurRobot [private]
```

9.3.2.15 nomJoueur

```
TextField FenetreAccueil::nomJoueur [private]
```

9.3.2.16 numTable

```
TextField FenetreAccueil::numTable [private]
```

9.3.2.17 onEcranConnecteChanged

```
var FenetreAccueil::onEcranConnecteChanged [private]
```

9.3.2.18 onEcranDeconnecteChanged

```
var FenetreAccueil::onEcranDeconnecteChanged [private]
```

9.3.2.19 onEcranDetecteChanged

```
var FenetreAccueil::onEcranDetecteChanged [private]
```

9.3.2.20 onErreurrobotDetecteeChanged

```
var FenetreAccueil::onErreurrobotDetecteeChanged [private]
```

9.3.2.21 onRechercheChanged

```
var FenetreAccueil::onRechercheChanged [private]
```

9.3.2.22 onRobotConnecteChanged

```
var FenetreAccueil::onRobotConnecteChanged [private]
```

9.3.2.23 onRobotDeconnecteChanged

```
var FenetreAccueil::onRobotDeconnecteChanged [private]
```

9.3.2.24 onRobotDetecteChanged

```
var FenetreAccueil::onRobotDetecteChanged [private]
```

9.3.2.25 onSeanceTermineeChanged

```
var FenetreAccueil::onSeanceTermineeChanged [private]
```

9.3.2.26 onTableConnecteChanged

```
var FenetreAccueil::onTableConnecteChanged [private]
```

9.3.2.27 onTableDeconnecteChanged

```
var FenetreAccueil::onTableDeconnecteChanged [private]
```

9.3.2.28 onTableDetecteChanged

```
var FenetreAccueil::onTableDetecteChanged [private]
```

9.3.2.29 robot

```
Button FenetreAccueil::robot [private]
```

9.3.2.30 switchBluetooth

```
Switch FenetreAccueil::switchBluetooth [private]
```

9.3.2.31 table

```
Button FenetreAccueil::table [private]
```

9.3.2.32 toolButton2

```
ToolButton FenetreAccueil::toolButton2 [private]
```

9.3.3 Documentation des propriétés

9.3.3.1 écranConnecte

```
bool FenetreAccueil::écranConnecte
```

9.3.3.2 écranDeconnecte

```
bool FenetreAccueil::écranDeconnecte
```

9.3.3.3 écranDetecte

```
bool FenetreAccueil::écranDetecte
```

9.3.3.4 erreurrobotDetectee

```
bool FenetreAccueil::erreurrobotDetectee
```

9.3.3.5 recherche

```
bool FenetreAccueil::recherche
```

9.3.3.6 robotConnecte

```
bool FenetreAccueil::robotConnecte
```


9.3.3.7 robotDeconnecte

```
bool FenetreAccueil::robotDeconnecte
```

9.3.3.8 robotDetecte

```
bool FenetreAccueil::robotDetecte
```

9.3.3.9 seanceTerminee

```
bool FenetreAccueil::seanceTerminee
```

9.3.3.10 tableConnecte

```
bool FenetreAccueil::tableConnecte
```

9.3.3.11 tableDeconnecte

```
bool FenetreAccueil::tableDeconnecte
```

9.3.3.12 tableDetecte

```
bool FenetreAccueil::tableDetecte
```

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

— [FenetreAccueil.qml](#)

9.4 Référence de la classe FenetreReglages

La fenêtre de réglages.

Graphe de collaboration de FenetreReglages :

FenetreReglages
<ul style="list-style-type: none"> + frequenceBalles + balles + nbBalles + effet + puissanceBalles + switchPosition + nbZonesSelectionee + nomSeance + nbClics + numeroZoneObjectif + numeroZoneRobot + idParametres + listeProfils - frequenceBalles - sliderFrequenceBalles - nombreFrequenceBalles - balle - nombreBalles - sliderNombreBalles - effets - puissanceBalles - sliderPuissanceBalles - valeurPuissanceBalles et 13 de plus...

Propriétés

- int [frequenceBalles](#)
- string [balles](#)
- int [nbBalles](#)
- int [effet](#)
- int [puissanceBalles](#)
- bool [switchPosition](#)
- int [nbZonesSelectionee](#)
- TextField [nomSeance](#)
- bool [nbClics](#)
- string [numeroZoneObjectif](#)
- string [numeroZoneRobot](#)
- string [idParametres](#)
- ComboBox [listeProfils](#)

Attributs privés

- Label [frequenceBalles](#)
- Slider [sliderFrequenceBalles](#)
- Label [nombreFrequenceBalles](#)

- Label [balle](#)
- Label [nombreBalles](#)
- Slider [sliderNombreBalles](#)
- ComboBox [effets](#)
- Label [puissanceBalles](#)
- Slider [sliderPuissanceBalles](#)
- Label [valeurPuissanceBalles](#)
- Switch [rotation](#)
- Label [zoneObjectif](#)
- Label [zoneRobot](#)
- Button [boutonPositionRobot](#)
- Dialog [positionRobot](#)
- ButtonGroup [zones](#)
- Button [zone1](#)
- Button [zone2](#)
- Button [zone3](#)
- Button [zone4](#)
- Button [zone5](#)
- Button [zone6](#)
- Button [zone7](#)

9.4.1 Description détaillée

Auteur

Youssef Hammouma

Version

1.4

9.4.2 Documentation des données membres

9.4.2.1 [balle](#)

```
Label FenetreReglages::balle [private]
```

9.4.2.2 [boutonPositionRobot](#)

```
Button FenetreReglages::boutonPositionRobot [private]
```

9.4.2.3 [effets](#)

```
ComboBox FenetreReglages::effets [private]
```

9.4.2.4 [frequenceBalles](#)

```
Label FenetreReglages::frequenceBalles [private]
```

9.4.2.5 nombreBalles

```
Label FenetreReglages::nombreBalles [private]
```

9.4.2.6 nombreFrequenceBalles

```
Label FenetreReglages::nombreFrequenceBalles [private]
```

9.4.2.7 positionRobot

```
Dialog FenetreReglages::positionRobot [private]
```

A faire Faire en sorte que le bouton cancel reset le Dialog positionRobot et que le bouton Save valide les paramètres

9.4.2.8 puissanceBalles

```
Label FenetreReglages::puissanceBalles [private]
```

9.4.2.9 rotation

```
Switch FenetreReglages::rotation [private]
```

9.4.2.10 sliderFrequenceBalles

```
Slider FenetreReglages::sliderFrequenceBalles [private]
```

9.4.2.11 sliderNombreBalles

```
Slider FenetreReglages::sliderNombreBalles [private]
```

9.4.2.12 sliderPuissanceBalles

```
Slider FenetreReglages::sliderPuissanceBalles [private]
```

9.4.2.13 valeurPuissanceBalles

```
Label FenetreReglages::valeurPuissanceBalles [private]
```

9.4.2.14 zone1

Button FenetreReglages::zone1 [private]

9.4.2.15 zone2

Button FenetreReglages::zone2 [private]

9.4.2.16 zone3

Button FenetreReglages::zone3 [private]

9.4.2.17 zone4

Button FenetreReglages::zone4 [private]

9.4.2.18 zone5

Button FenetreReglages::zone5 [private]

9.4.2.19 zone6

Button FenetreReglages::zone6 [private]

9.4.2.20 zone7

Button FenetreReglages::zone7 [private]

9.4.2.21 zoneObjectif

Label FenetreReglages::zoneObjectif [private]

9.4.2.22 zoneRobot

Label FenetreReglages::zoneRobot [private]

9.4.2.23 zones

```
ButtonGroup FenetreReglages::zones [private]
```

9.4.3 Documentation des propriétés

9.4.3.1 balles

```
string FenetreReglages::balles
```

9.4.3.2 effet

```
int FenetreReglages::effet
```

9.4.3.3 frequenceBalles

```
int FenetreReglages::frequenceBalles
```

9.4.3.4 idParametres

```
string FenetreReglages::idParametres
```

9.4.3.5 listeProfils

```
ComboBox FenetreReglages::listeProfils
```

9.4.3.6 nbBalles

```
int FenetreReglages::nbBalles
```

9.4.3.7 nbClics

```
bool FenetreReglages::nbClics
```

9.4.3.8 nbZonesSelectionee

```
int FenetreReglages::nbZonesSelectionee
```

9.4.3.9 nomSeance

```
TextField FenetreReglages::nomSeance
```

9.4.3.10 numeroZoneObjectif

```
string FenetreReglages::numeroZoneObjectif
```

9.4.3.11 numeroZoneRobot

```
string FenetreReglages::numeroZoneRobot
```

9.4.3.12 puissanceBalles

```
int FenetreReglages::puissanceBalles
```

9.4.3.13 switchPosition

```
bool FenetreReglages::switchPosition
```

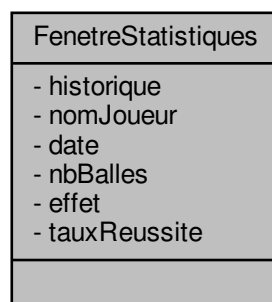
La documentation de cette classe a été générée à partir du fichier suivant :

— [FenetreReglages.qml](#)

9.5 Référence de la classe FenetreStatistiques

La fenêtre de statistiques.

Graphe de collaboration de FenetreStatistiques :



Attributs privés

- Label [historique](#)
- ComboBox [nomJoueur](#)
- Label [date](#)
- Label [nbBalles](#)
- Label [effet](#)
- Label [tauxReussite](#)

9.5.1 Description détaillée**Auteur**

Youssef Hammouma

Version

0.1

9.5.2 Documentation des données membres**9.5.2.1 date**

Label FenetreStatistiques::date [private]

9.5.2.2 effet

Label FenetreStatistiques::effet [private]

9.5.2.3 historique

Label FenetreStatistiques::historique [private]

9.5.2.4 nbBalles

Label FenetreStatistiques::nbBalles [private]

9.5.2.5 nomJoueur

ComboBox FenetreStatistiques::nomJoueur [private]

9.5.2.6 tauxReussite

```
Label FenetreStatistiques::tauxReussite [private]
```

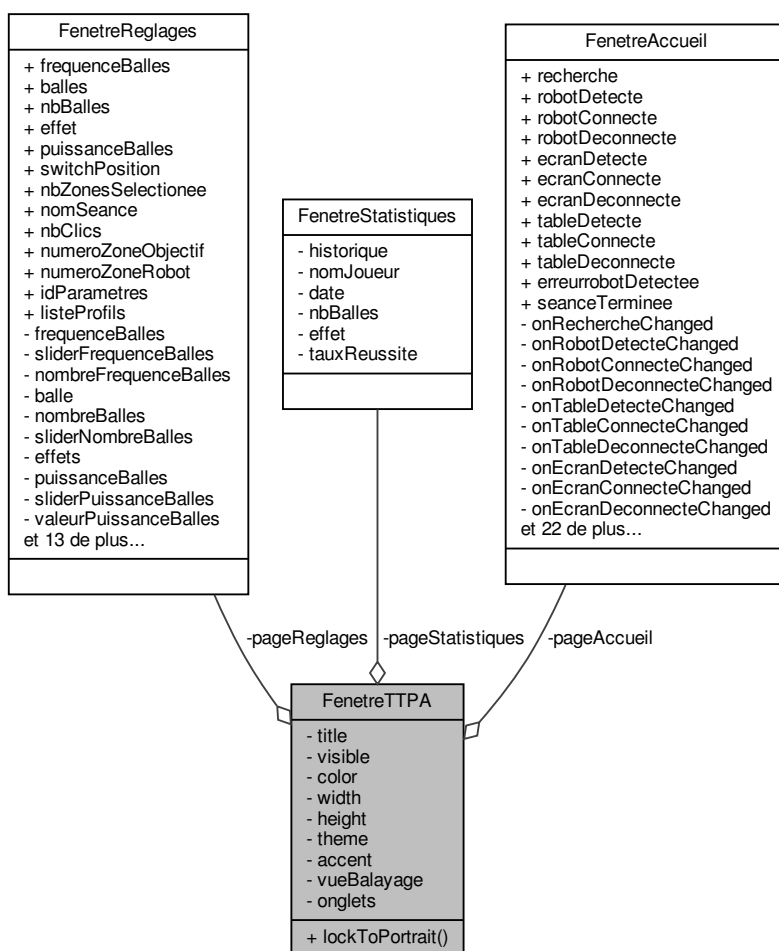
La documentation de cette classe a été générée à partir du fichier suivant :

— [FenetreStatistiques.qml](#)

9.6 Référence de la classe FenetreTTPA

La fenêtre principale de l'application terminal mobile.

Grphe de collaboration de FenetreTTPA :



Fonctions membres publiques

— void [lockToPortrait](#) ()

Attributs privés

- var [title](#)
- var [visible](#)
- var [color](#)
- var [width](#)
- var [height](#)
- var Material [theme](#)
- var Material [accent](#)
- SwipeView [vueBalayage](#)
- FenetreAccueil [pageAccueil](#)
- FenetreReglages [pageReglages](#)
- FenetreStatistiques [pageStatistiques](#)
- TabBar [onglets](#)

9.6.1 Description détaillée**Auteur**

Youssef Hammouma

Version

1.4

9.6.2 Documentation des fonctions membres**9.6.2.1 lockToPortrait()**

```
void FenetreTTPA::lockToPortrait ( )
```

9.6.3 Documentation des données membres**9.6.3.1 accent**

```
var Material FenetreTTPA::accent [private]
```

9.6.3.2 color

```
var FenetreTTPA::color [private]
```

9.6.3.3 height

```
var FenetreTTPA::height [private]
```

9.6.3.4 onglets

`TabBar FenetreTPA::onglets [private]`

9.6.3.5 pageAccueil

`FenetreAccueil FenetreTPA::pageAccueil [private]`

9.6.3.6 pageReglages

`FenetreReglages FenetreTPA::pageReglages [private]`

9.6.3.7 pageStatistiques

`FenetreStatistiques FenetreTPA::pageStatistiques [private]`

9.6.3.8 theme

`var Material FenetreTPA::theme [private]`

9.6.3.9 title

`var FenetreTPA::title [private]`

9.6.3.10 visible

`var FenetreTPA::visible [private]`

9.6.3.11 vueBalayage

`SwipeView FenetreTPA::vueBalayage [private]`

9.6.3.12 width

`var FenetreTPA::width [private]`

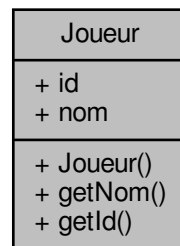
La documentation de cette classe a été générée à partir du fichier suivant :
— [FenetreTPA.qml](#)

9.7 Référence de la classe Joueur

Déclaration de la classe `Joueur`.

```
#include <Joueur.h>
```

Graphes de collaboration de `Joueur` :



Signaux

- void `nomChanged` ()
- void `idChanged` ()

Fonctions membres publiques

- `Joueur` (QString `id`="", QString `nom`="", QObject *parent=nullptr)
- QString `getNom` ()
- QString `getId` ()

Propriétés

- QString `id`
- QString `nom`

9.7.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

9.7.2 Documentation des constructeurs et destructeur

9.7.2.1 Joueur()

```
Joueur::Joueur (
    QString id = "",
    QString nom = "",
    QObject * parent = nullptr ) [explicit]
```

Références [nom](#).

```
00016                                     : QObject(parent), id(id),
    nom(nom)
00017 {
00018     qDebug() << Q_FUNC_INFO << "id" << id << "nom" << nom;
00019 }
```

9.7.3 Documentation des fonctions membres

9.7.3.1 getId()

```
QString Joueur::getId ( )
```

Références [id](#).

```
00027 {
00028     return id;
00029 }
```

9.7.3.2 getNom()

```
QString Joueur::getNom ( )
```

Références [nom](#).

```
00022 {
00023     return nom;
00024 }
```

9.7.3.3 idChanged

```
void Joueur::idChanged ( ) [signal]
```

9.7.3.4 nomChanged

```
void Joueur::nomChanged ( ) [signal]
```

9.7.4 Documentation des propriétés

9.7.4.1 id

```
QString Joueur::id [read]
```

Référencé par [getId\(\)](#).

9.7.4.2 nom

```
QString Joueur::nom [read]
```

Référencé par [getNom\(\)](#), et [Joueur\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

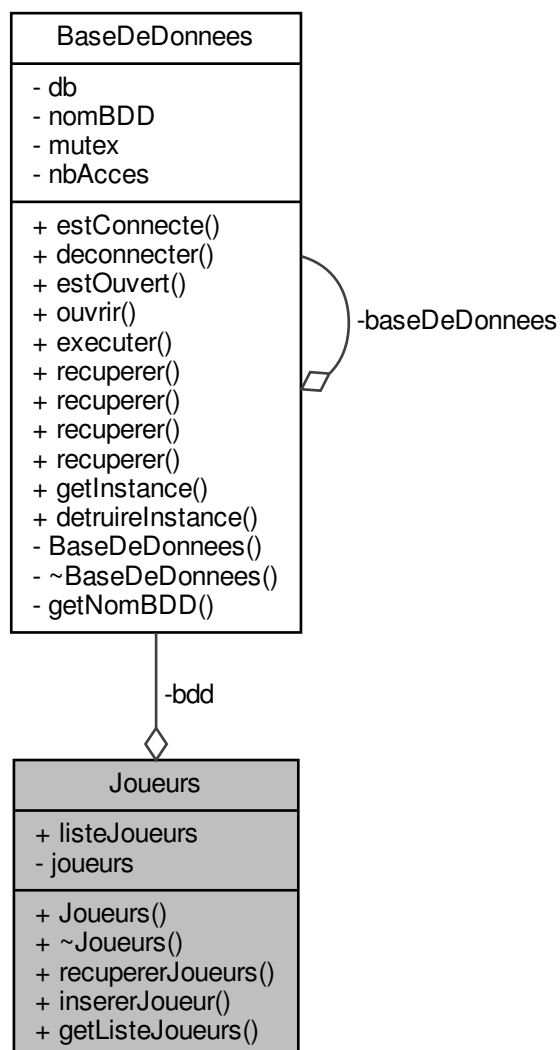
- [Joueur.h](#)
- [Joueur.cpp](#)

9.8 Référence de la classe Joueurs

Déclaration de la classe [Joueurs](#).

```
#include <Joueurs.h>
```

Graphe de collaboration de Joueurs :



Signaux

— void **listeJoueursUpdated** ()

Fonctions membres publiques

— **Joueurs** (QObject *parent=nullptr)
 — **~Joueurs** ()
 — Q_INVOKABLE bool **recupererJoueurs** ()
 — Q_INVOKABLE QString **insererJoueur** (QString nomJoueur)
 — QVariant **getListeJoueurs** ()

Propriétés

— QVariant **listeJoueurs**

Attributs privés

- `QList< QObject * > joueurs`
- `BaseDeDonnees * bdd`

9.8.1 Description détaillée**Auteur**

Hammouma Youssef

Version

1.4

9.8.2 Documentation des constructeurs et destructeur**9.8.2.1 Joueurs()**

```
Joueurs::Joueurs (
    QObject * parent = nullptr ) [explicit]
```

Références [bdd](#), [BaseDeDonnees : :getInstance\(\)](#), [BaseDeDonnees : :ouvrir\(\)](#), et [recupererJoueurs\(\)](#).

```
00017                                     : QObject (parent)
00018 {
00019     bdd = BaseDeDonnees::getInstance();
00020     bdd->ouvrir("ttpa.sqlite");
00021     recupererJoueurs();
00022 }
```

9.8.2.2 ~Joueurs()

```
Joueurs::~~Joueurs ( )
```

Références [BaseDeDonnees : :destruireInstance\(\)](#).

```
00025 {
00026     BaseDeDonnees::destruireInstance();
00027 }
```

9.8.3 Documentation des fonctions membres**9.8.3.1 getListeJoueurs()**

```
QVariant Joueurs::getListeJoueurs ( )
```

Références [joueurs](#).

```
00067 {
00068     return QVariant::fromValue(joueurs);
00069 }
```


9.8.3.2 insererJoueur()

```
QString Joueurs::insererJoueur (
    QString nomJoueur )
```

Références [bdd](#), [BaseDeDonnees : :executer\(\)](#), [BaseDeDonnees : :recuperer\(\)](#), et [recupererJoueurs\(\)](#).

```
00052 {
00053     QString idJoueur;
00054     QString requete = "INSERT INTO joueurs(nom) VALUES( ' " + nomJoueur + "' )";
00055
00056     bool etat = bdd->executer(requete);
00057     if(etat)
00058     {
00059         requete = "SELECT last_insert_rowid() FROM parametres";
00060         etat = bdd->recuperer(requete, idJoueur);
00061         recupererJoueurs();
00062     }
00063     return idJoueur;
00064 }
```

9.8.3.3 listeJoueursUpdated

```
void Joueurs::listeJoueursUpdated ( ) [signal]
```

Référencé par [recupererJoueurs\(\)](#).

9.8.3.4 recupererJoueurs()

```
bool Joueurs::recupererJoueurs ( )
```

Références [bdd](#), [joueurs](#), [listeJoueursUpdated\(\)](#), et [BaseDeDonnees : :recuperer\(\)](#).

Référencé par [insererJoueur\(\)](#), et [Joueurs\(\)](#).

```
00030 {
00031     QString requete = "SELECT idJoueur, nom FROM joueurs ORDER BY nom ASC";
00032     QVector<QStringList> lesJoueurs;
00033
00034     bool etat = bdd->recuperer(requete, lesJoueurs);
00035     if(etat)
00036     {
00037         qDebug() << Q_FUNC_INFO << lesJoueurs;
00038         joueurs.clear();
00039         joueurs.append(new Joueur());
00040         for(int i=0;i<lesJoueurs.size();i++)
00041         {
00042             Joueur *joueur = new Joueur(lesJoueurs.at(i).at(0), lesJoueurs.at(i).at(1), this);
00043             joueurs.append(joueur);
00044         }
00045         emit listeJoueursUpdated();
00046         return true;
00047     }
00048     return false;
00049 }
```

9.8.4 Documentation des données membres

9.8.4.1 bdd

```
BaseDeDonnees* Joueurs::bdd [private]
```

Référencé par [insérerJoueur\(\)](#), [Joueurs\(\)](#), et [recupererJoueurs\(\)](#).

9.8.4.2 joueurs

```
QList<QObject*> Joueurs::joueurs [private]
```

Référencé par [getListeJoueurs\(\)](#), et [recupererJoueurs\(\)](#).

9.8.5 Documentation des propriétés

9.8.5.1 listeJoueurs

```
QVariant Joueurs::listeJoueurs [read]
```

La documentation de cette classe a été générée à partir des fichiers suivants :

- [Joueurs.h](#)
- [Joueurs.cpp](#)

9.9 Référence de la classe Parametre

Déclaration de la classe [Parametre](#).

```
#include <Parametre.h>
```

Graphe de collaboration de Parametre :

Parametre
+ id + nbBalles + nbBallesMinute + effet + puissance + rotation + zoneObjectif + zoneRobot
+ Parametre() + getId() + getNbBalles() + getNbBallesMinute() + getEffet() + getPuissance() + getRotation() + getZoneObjectif() + getZoneRobot()

Signaux

- void `idChanged` ()
- void `nbBallesChanged` ()
- void `nbBallesMinuteChanged` ()
- void `effetChanged` ()
- void `puissanceChanged` ()
- void `rotationChanged` ()
- void `zoneObjectifChanged` ()
- void `zoneRobotChanged` ()

Fonctions membres publiques

- `Parametre` (QString `id`="", QString `nbBalles`="", QString `nbBallesMinute`="", QString `effet`="", QString `puissance`="", QString `rotation`="", QString `zoneObjectif`="", QString `zoneRobot`="", QObject `*parent`=nullptr)
- QString `getId` ()
- QString `getNbBalles` ()
- QString `getNbBallesMinute` ()
- QString `getEffet` ()
- QString `getPuissance` ()
- QString `getRotation` ()
- QString `getZoneObjectif` ()
- QString `getZoneRobot` ()

Propriétés

- QString `id`
- QString `nbBalles`
- QString `nbBallesMinute`
- QString `effet`
- QString `puissance`
- QString `rotation`
- QString `zoneObjectif`
- QString `zoneRobot`

9.9.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

9.9.2 Documentation des constructeurs et destructeur

9.9.2.1 Parametre()

```
Parametre::Parametre (
    QString id = "",
    QString nbBalles = "",
    QString nbBallesMinute = "",
    QString effet = "",
    QString puissance = "",
    QString rotation = "",
    QString zoneObjectif = "",
    QString zoneRobot = "",
    QObject * parent = nullptr ) [explicit]
```

Références [zoneRobot](#).

```
00015                                     : QObject(parent),
    id(id), nbBalles(nbBalles), nbBallesMinute(
    nbBallesMinute), effet(effet), puissance(
    puissance), rotation(rotation), zoneObjectif(
    zoneObjectif), zoneRobot(zoneRobot)
00016 {
00017     qDebug() << Q_FUNC_INFO << "id" << id << "nbBalles" << nbBalles << "nbBallesMinute" <<
    nbBallesMinute << "effet" << effet << "puissance" <<
    puissance << "rotation" << rotation << "zoneObjectif" <<
    zoneObjectif << "zoneRobot" << zoneRobot;
00018 }
```

9.9.3 Documentation des fonctions membres

9.9.3.1 effetChanged

```
void Parametre::effetChanged ( ) [signal]
```

9.9.3.2 getEffet()

```
QString Parametre::getEffet ( )
```

Références [effet](#).

```
00036 {
00037     return effet;
00038 }
```

9.9.3.3 getId()

```
QString Parametre::getId ( )
```

Références [id](#).

```
00021 {
00022     return id;
00023 }
```

9.9.3.4 getNbBalles()

QString Parametre::getNbBalles ()

Références [nbBalles](#).

```
00026 {  
00027     return nbBalles;  
00028 }
```

9.9.3.5 getNbBallesMinute()

QString Parametre::getNbBallesMinute ()

Références [nbBallesMinute](#).

```
00031 {  
00032     return nbBallesMinute;  
00033 }
```

9.9.3.6 getPuissance()

QString Parametre::getPuissance ()

Références [puissance](#).

```
00041 {  
00042     return puissance;  
00043 }
```

9.9.3.7 getRotation()

QString Parametre::getRotation ()

Références [rotation](#).

```
00046 {  
00047     return rotation;  
00048 }
```

9.9.3.8 getZoneObjectif()

QString Parametre::getZoneObjectif ()

Références [zoneObjectif](#).

```
00051 {  
00052     return zoneObjectif;  
00053 }
```

9.9.3.9 getZoneRobot()

```
QString Parametre::getZoneRobot ( )
```

Références [zoneRobot](#).

```
00056 {  
00057     return zoneRobot;  
00058 }
```

9.9.3.10 idChanged

```
void Parametre::idChanged ( ) [signal]
```

9.9.3.11 nbBallesChanged

```
void Parametre::nbBallesChanged ( ) [signal]
```

9.9.3.12 nbBallesMinuteChanged

```
void Parametre::nbBallesMinuteChanged ( ) [signal]
```

9.9.3.13 puissanceChanged

```
void Parametre::puissanceChanged ( ) [signal]
```

9.9.3.14 rotationChanged

```
void Parametre::rotationChanged ( ) [signal]
```

9.9.3.15 zoneObjectifChanged

```
void Parametre::zoneObjectifChanged ( ) [signal]
```

9.9.3.16 zoneRobotChanged

```
void Parametre::zoneRobotChanged ( ) [signal]
```

9.9.4 Documentation des propriétés

9.9.4.1 effet

QString Parametre::effet [read]

Référencé par [getEffet\(\)](#).

9.9.4.2 id

QString Parametre::id [read]

Référencé par [getId\(\)](#).

9.9.4.3 nbBalles

QString Parametre::nbBalles [read]

Référencé par [getNbBalles\(\)](#).

9.9.4.4 nbBallesMinute

QString Parametre::nbBallesMinute [read]

Référencé par [getNbBallesMinute\(\)](#).

9.9.4.5 puissance

QString Parametre::puissance [read]

Référencé par [getPuissance\(\)](#).

9.9.4.6 rotation

QString Parametre::rotation [read]

Référencé par [getRotation\(\)](#).

9.9.4.7 zoneObjectif

```
QString Parametre::zoneObjectif [read]
```

Référencé par [getZoneObjectif\(\)](#).

9.9.4.8 zoneRobot

```
QString Parametre::zoneRobot [read]
```

Référencé par [getZoneRobot\(\)](#), et [Parametre\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

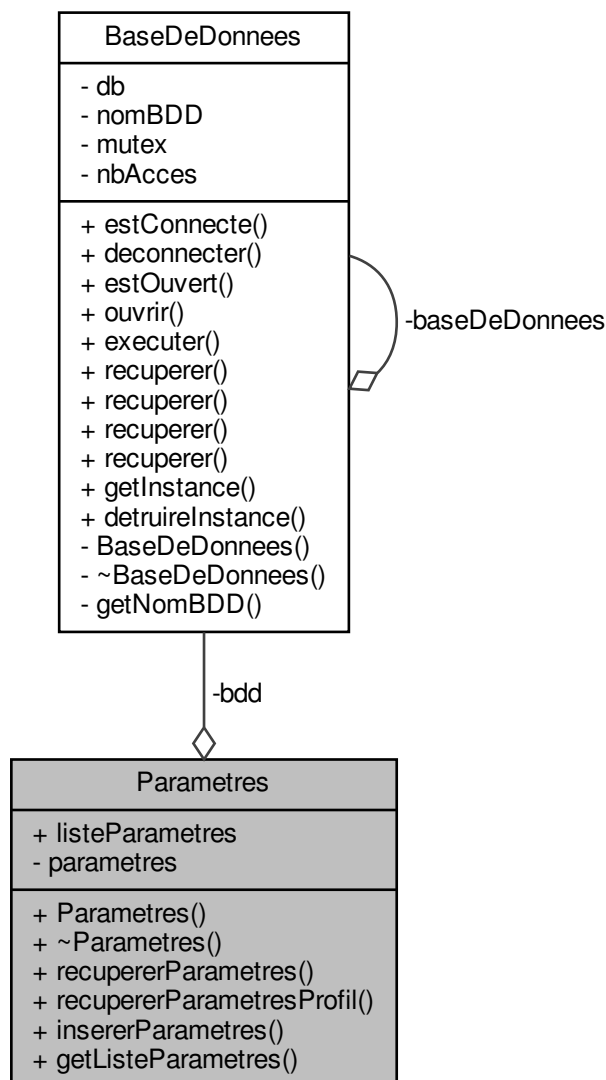
- [Parametre.h](#)
- [Parametre.cpp](#)

9.10 Référence de la classe Parametres

Déclaration de la classe [Parametres](#).

```
#include <Parametres.h>
```


Graphe de collaboration de Parametres :



Signaux

— void `listeParametresUpdated()`

Fonctions membres publiques

— `Parametres` (QObject *parent=nullptr)
 — `~Parametres()`
 — Q_INVOKABLE bool `recupererParametres()`
 — Q_INVOKABLE QStringList `recupererParametresProfil` (QString idParametres)
 — Q_INVOKABLE QString `insérerParametres` (QString nbBalles, QString nbBallesMinute, QString effet, QString puissance, Q←
 String rotation, QString zoneObjectif, QString zoneRobot)
 — QVariant `getListeParametres()`

Propriétés

— QVariant [listeParametres](#)

Attributs privés

— QList< QObject * > [parametres](#)
— [BaseDeDonnees](#) * [bdd](#)

9.10.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

9.10.2 Documentation des constructeurs et destructeur

9.10.2.1 Parametres()

```
Parametres::Parametres (
    QObject * parent = nullptr ) [explicit]
```

Références [bdd](#), [BaseDeDonnees](#) : [getInstance\(\)](#), [BaseDeDonnees](#) : [ouvrir\(\)](#), et [recupererParametres\(\)](#).

```
00017                                     : QObject (parent)
00018 {
00019     bdd = BaseDeDonnees::getInstance\(\);
00020     bdd->ouvrir("ttpa.sqlite");
00021     recupererParametres();
00022 }
```

9.10.2.2 ~Parametres()

```
Parametres::~~Parametres ( )
```

Références [BaseDeDonnees](#) : [detruireInstance\(\)](#).

```
00025 {
00026     BaseDeDonnees::detruireInstance();
00027 }
```

9.10.3 Documentation des fonctions membres

9.10.3.1 getListeParametres()

```
QVariant Parametres::getListeParametres ( )
```

Références [parametres](#).

```
00086 {
00087     return QVariant::fromValue(parametres);
00088 }
```

9.10.3.2 insererParametres()

```
QString Parametres::insererParametres (
    QString nbBalles,
    QString nbBallesMinute,
    QString effet,
    QString puissance,
    QString rotation,
    QString zoneObjectif,
    QString zoneRobot )
```

Références [bdd](#), [BaseDeDonnees : :executer\(\)](#), [BaseDeDonnees : :recuperer\(\)](#), et [recupererParametres\(\)](#).

```
00068 {
00069     if(nbBalles.isEmpty() || nbBallesMinute.isEmpty() || effet.isEmpty() || puissance.isEmpty() || rotation
.isEmpty() || zoneObjectif.isEmpty() || zoneRobot.isEmpty())
00070         return QString("");
00071
00072     QString idParametres;
00073     QString requete = "INSERT INTO parametres(nbBalles, nbBallesMinute, effet, puissance, rotation,
zoneObjectif, zoneRobot) VALUES("+ nbBalles + "," + nbBallesMinute + "," + effet + "," + puissance + "," + rotation
+ "," + zoneObjectif + "," + zoneRobot + ")";
00074
00075     bool etat = bdd->executer(requete);
00076     if(etat)
00077     {
00078         requete = "SELECT last_insert_rowid() FROM parametres";
00079         etat = bdd->recuperer(requete, idParametres);
00080         recupererParametres();
00081     }
00082     return idParametres;
00083 }
```

9.10.3.3 listeParametresUpdated

```
void Parametres::listeParametresUpdated ( ) [signal]
```

Référencé par [recupererParametres\(\)](#).

9.10.3.4 recupererParametres()

```
bool Parametres::recupererParametres ( )
```

Références [bdd](#), [listeParametresUpdated\(\)](#), [parametres](#), et [BaseDeDonnees : :recuperer\(\)](#).

Référencé par [insererParametres\(\)](#), et [Parametres\(\)](#).

```
00030 {
00031     QString requete = "SELECT idParametres, nbBalles, nbBallesMinute, effet, puissance, rotation,
00032     zoneObjectif, zoneRobot FROM parametres";
00033     QVector<QStringList> lesParametres;
00034     bool etat = bdd->recuperer(requete, lesParametres);
00035     if(etat)
00036     {
00037         qDebug() << Q_FUNC_INFO << lesParametres;
00038         lesParametres.clear();
00039         for(int i=0; i < lesParametres.size(); i++)
00040         {
00041             Parametre *parametre = new Parametre(lesParametres.at(i).at(0), lesParametres
00042             .at(i).at(1),
00043             lesParametres.at(i).at(2), lesParametres.at(i).at(3),
00044             lesParametres.at(i).at(4), lesParametres.at(i).at(5),
00045             lesParametres.at(i).at(6), lesParametres.at(i).at(7));
00046             parametres.append(parametre);
00047         }
00048         emit listeParametresUpdated();
00049         return true;
00050     }
00051     return false;
00052 }
```

9.10.3.5 recupererParametresProfil()

```
QStringList Parametres::recupererParametresProfil (
    QString idParametres )
```

Références [bdd](#), et [BaseDeDonnees : :recuperer\(\)](#).

```
00054 {
00055     QString requete = "SELECT parametres.nbBalles, parametres.nbBallesMinute, parametres.effet,
00056     parametres.puissance, parametres.rotation, parametres.zoneObjectif, parametres.zoneRobot FROM parametres INNER JOIN
00057     profils ON parametres.idParametres = profils.idParametres WHERE parametres.idParametres =" + idParametres;
00058     QStringList lesParametres;
00059     bool etat = bdd->recuperer(requete, lesParametres);
00060     if(etat)
00061     {
00062         qDebug() << Q_FUNC_INFO << lesParametres;
00063     }
00064     return lesParametres;
00065 }
```

9.10.4 Documentation des données membres

9.10.4.1 bdd

```
BaseDeDonnees* Parametres::bdd [private]
```

Référencé par [insererParametres\(\)](#), [Parametres\(\)](#), [recupererParametres\(\)](#), et [recupererParametresProfil\(\)](#).

9.10.4.2 parametres

```
QList<QObject*> Parametres::parametres [private]
```

Référencé par [getListeParametres\(\)](#), et [recupererParametres\(\)](#).

9.10.5 Documentation des propriétés

9.10.5.1 listeParametres

```
QVariant Parametres::listeParametres [read]
```

La documentation de cette classe a été générée à partir des fichiers suivants :

- [Parametres.h](#)
- [Parametres.cpp](#)

9.11 Référence de la classe PeripheriqueTTPA

Permet la communication avec les appareils Bluetooth TTPA.

```
#include <peripheriqueTTPA.h>
```

Graphe de collaboration de PeripheriqueTTPA :

PeripheriqueTTPA
+ nom + adresseMAC + etatConnexion + connexionErreur - socket
+ PeripheriqueTTPA() + getNom() + getAdresseMAC() + connecter() + deconnecter() + envoyer() + estConnecte() + socketConnecte() + socketDeconnecte() + socketReadyRead() + socketErreur()

Connecteurs publics

- void [socketConnecte](#) ()
slot exécuté quand la socket de communication est connecté
- void [socketDeconnecte](#) ()
slot exécuté quand la socket de communication est déconnecté
- void [socketReadyRead](#) ()
slot exécuté quand des données sont prêtes à être lues
- void [socketErreur](#) (QBluetoothSocket : :SocketError [erreur](#))
slot exécuté quand la socket est en erreur

Signaux

- void [peripheriqueTTPAChanged](#) ()
- void [connecte](#) ()
signal un périphérique est connecté
- void [deconnecte](#) ()
signal un périphérique est déconnecté
- void [erreur](#) ()
signal une erreur
- void [trameRecue](#) (QString nomPeripherique, QString trame)
signal indiquant qu'une trame a été reçue

Fonctions membres publiques

- [PeripheriqueTTPA](#) (QString [nom](#), QString [adresseMAC](#), QObject *parent=nullptr)
- QString [getNom](#) () const
- QString [getAdresseMAC](#) () const
- Q_INVOKABLE void [connecter](#) ()
Connecte le terminal mobile à l'appareil "TTPA".
- Q_INVOKABLE void [deconnecter](#) ()
Deconnecte l'appareil "TTPA" connecté
- Q_INVOKABLE bool [envoyer](#) (QString trame)
Envoie une trame à l'appareil "TTPA" connecté
- Q_INVOKABLE bool [estConnecte](#) ()
Retourne vrai si l'appareil est connecté

Propriétés

- QString [nom](#)
nom du périphérique Bluetooth
- QString [adresseMAC](#)
adresse MAS du périphérique Bluetooth
- bool [etatConnexion](#)
état de connexion périphérique Bluetooth.
- bool [connexionErreur](#)
état d'erreur du périphérique Bluetooth.

Attributs privés

- QBluetoothSocket * [socket](#)
une socket de communication

9.11.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

9.11.2 Documentation des constructeurs et destructeur

9.11.2.1 PeripheriqueTTPA()

```
PeripheriqueTTPA::PeripheriqueTTPA (
    QString nom,
    QString adresseMAC,
    QObject * parent = nullptr )
```

Références [adresseMAC](#).

```
00015                                     : QObject (parent),
    nom(nom), adresseMAC(adresseMAC), etatConnexion(false),
    connexionErreur(false), socket(NULL)
00016 {
00017     qDebug() << Q_FUNC_INFO << nom << adresseMAC;
00018 }
```

9.11.3 Documentation des fonctions membres

9.11.3.1 connecte

```
void PeripheriqueTTPA::connecte ( ) [signal]
```

Référencé par [socketConnecte\(\)](#), et [socketErreur\(\)](#).

9.11.3.2 connecter()

```
PeripheriqueTTPA::connecter ( )
```

Références [adresseMAC](#), [nom](#), [socket](#), [socketConnecte\(\)](#), [socketDeconnecte\(\)](#), [socketErreur\(\)](#), et [socketReadyRead\(\)](#).

Référencé par [CommunicationBluetooth : :connecterPeripheriquesTTPA\(\)](#).

```
00031 {
00032     qDebug() << Q_FUNC_INFO << nom << adresseMAC;
00033
00034     if (!socket)
00035     {
00036         socket = new QBluetoothSocket(QBluetoothServiceInfo::RfcommProtocol);
00037         connect(socket, SIGNAL(connected()), this, SLOT(socketConnecte()));
00038         connect(socket, SIGNAL(disconnected()), this, SLOT(
00039             socketDeconnecte()));
00039         connect(socket, SIGNAL(readyRead()), this, SLOT(socketReadyRead()));
00040         connect(socket, SIGNAL(error(QBluetoothSocket::SocketError)), this, SLOT(
00041             socketErreur(QBluetoothSocket::SocketError)));
00041     }
00042     else if (socket->isOpen())
00043     {
00044         socket->close();
00045     }
00046     QBluetoothUuid uuid = QBluetoothUuid(QBluetoothUuid::SerialPort);
00047     socket->connectToService(QBluetoothAddress(adresseMAC), uuid);
00048     socket->open(QIODevice::ReadWrite);
00049 }
```

9.11.3.3 deconnecte

```
void PeripheriqueTTPA::deconnecte ( ) [signal]
```

Référencé par [socketDeconnecte\(\)](#).

9.11.3.4 deconnecter()

```
PeripheriqueTTPA::deconnecter ( )
```

Références [adresseMAC](#), [nom](#), et [socket](#).

Référencé par [CommunicationBluetooth : :deconnecterPeripheriquesTTPA\(\)](#).

```
00052 {
00053     qDebug() << Q_FUNC_INFO << nom << adresseMAC;
00054
00055     if (!socket)
00056         return;
00057     if (socket->isOpen())
00058         socket->close();
00059 }
```

9.11.3.5 envoyer()

```
PeripheriqueTTPA::envoyer (
    QString trame )
```

Paramètres

<i>trame</i>	QString Trame à envoyer à l'appareil
--------------	--------------------------------------

Références [adresseMAC](#), [nom](#), et [socket](#).

Référencé par [CommunicationBluetooth : :envoyerPeripheriqueTTPA\(\)](#).

```
00102 {
00103     if (!socket || !socket->isOpen())
00104     {
00105         return false;
00106     }
00107     trame = "$" + trame;
00108     qDebug() << Q_FUNC_INFO << nom << adresseMAC << "trame" << trame;
00109     qint64 nbOctets = socket->write(trame.toLatin1());
00110     if (nbOctets > 0)
00111         return true;
00112     return false;
00113 }
```

9.11.3.6 erreur

```
void PeripheriqueTTPA::erreur ( ) [signal]
```

Référencé par [socketConnecte\(\)](#), [socketDeconnecte\(\)](#), et [socketErreur\(\)](#).

9.11.3.7 estConnecte()

```
bool PeripheriqueTPA::estConnecte ( )
```

Renvoie

bool L'etat de connexion de l'appareil

Références [adresseMAC](#), [nom](#), et [socket](#).

Référencé par [CommunicationBluetooth : :connecterPeripheriquesTPA\(\)](#).

```
00116 {
00117     if (!socket)
00118         return false;
00119     qDebug() << Q_FUNC_INFO << nom << adresseMAC << "connecte" <<
        socket->isOpen();
00120     return socket->isOpen();
00121 }
```

9.11.3.8 getAdresseMAC()

```
QString PeripheriqueTPA::getAdresseMAC ( ) const
```

Références [adresseMAC](#).

```
00026 {
00027     return adresseMAC;
00028 }
```

9.11.3.9 getNom()

```
QString PeripheriqueTPA::getNom ( ) const
```

Références [nom](#).

Référencé par [CommunicationBluetooth : :connecterPeripheriquesTPA\(\)](#), [CommunicationBluetooth : :deconnecterPeripheriquesTPA\(\)](#), [CommunicationBluetooth : :envoyerPeripheriqueTPA\(\)](#), [CommunicationBluetooth : :getPeripheriqueTPA\(\)](#), [CommunicationBluetooth : :recevoirConnecte\(\)](#), et [CommunicationBluetooth : :recevoirDeconnecte\(\)](#).

```
00021 {
00022     return nom;
00023 }
```

9.11.3.10 peripheriqueTPAChanged

```
void PeripheriqueTPA::peripheriqueTPAChanged ( ) [signal]
```

9.11.3.11 socketConnecte

```
void PeripheriqueTTPA::socketConnecte ( ) [slot]
```

Références [adresseMAC](#), [connecte\(\)](#), [connexionErreur](#), [erreur\(\)](#), [etatConnexion](#), et [nom](#).

Référencé par [connecter\(\)](#).

```
00062 {  
00063     qDebug() << Q_FUNC_INFO << nom << adresseMAC;  
00064     etatConnexion = true;  
00065     connexionErreur = false;  
00066     emit connecte();  
00067     emit erreur();  
00068 }
```

9.11.3.12 socketDeconnecte

```
void PeripheriqueTTPA::socketDeconnecte ( ) [slot]
```

Références [adresseMAC](#), [connexionErreur](#), [deconnecte\(\)](#), [erreur\(\)](#), [etatConnexion](#), et [nom](#).

Référencé par [connecter\(\)](#).

```
00071 {  
00072     qDebug() << Q_FUNC_INFO << nom << adresseMAC;  
00073     etatConnexion = false;  
00074     connexionErreur = false;  
00075     emit deconnecte();  
00076     emit erreur();  
00077 }
```

9.11.3.13 socketErreur

```
void PeripheriqueTTPA::socketErreur (  
    QBluetoothSocket::SocketError erreur ) [slot]
```

Références [adresseMAC](#), [connecte\(\)](#), [connexionErreur](#), [erreur\(\)](#), [etatConnexion](#), et [nom](#).

Référencé par [connecter\(\)](#).

```
00093 {  
00094     qDebug() << Q_FUNC_INFO << nom << adresseMAC << "erreur" << error;  
00095     etatConnexion = false;  
00096     connexionErreur = true;  
00097     emit connecte();  
00098     emit erreur();  
00099 }
```

9.11.3.14 socketReadyRead

```
void PeripheriqueTTPA::socketReadyRead ( ) [slot]
```

Références [adresseMAC](#), [nom](#), [socket](#), et [trameRecue\(\)](#).

Référencé par [connecter\(\)](#).

```
00080 {
00081     qDebug() << Q_FUNC_INFO << nom << adresseMAC;
00082     QByteArray trame;
00083
00084     while (socket->canReadLine())
00085     {
00086         trame = socket->readLine();
00087         qDebug() << Q_FUNC_INFO << nom << adresseMAC << "trame" << trame;
00088         emit trameRecue(nom, QString(trame));
00089     }
00090 }
```

9.11.3.15 trameRecue

```
void PeripheriqueTTPA::trameRecue (
    QString nomPeripherique,
    QString trame ) [signal]
```

Référencé par [socketReadyRead\(\)](#).

9.11.4 Documentation des données membres

9.11.4.1 socket

```
QBluetoothSocket* PeripheriqueTTPA::socket [private]
```

Référencé par [connecter\(\)](#), [deconnecter\(\)](#), [envoyer\(\)](#), [estConnecte\(\)](#), et [socketReadyRead\(\)](#).

9.11.5 Documentation des propriétés

9.11.5.1 adresseMAC

```
QString PeripheriqueTTPA::adresseMAC [read]
```

Référencé par [connecter\(\)](#), [deconnecter\(\)](#), [envoyer\(\)](#), [estConnecte\(\)](#), [getAdresseMAC\(\)](#), [PeripheriqueTTPA\(\)](#), [socketConnecte\(\)](#), [socketDeconnecte\(\)](#), [socketErreur\(\)](#), et [socketReadyRead\(\)](#).

9.11.5.2 connexionErreur

```
bool PeripheriqueTTPA::connexionErreur
```

Référencé par [socketConnecte\(\)](#), [socketDeconnecte\(\)](#), et [socketErreur\(\)](#).

9.11.5.3 etatConnexion

```
bool PeripheriqueTTPA::etatConnexion
```

Référencé par [socketConnecte\(\)](#), [socketDeconnecte\(\)](#), et [socketErreur\(\)](#).

9.11.5.4 nom

```
QString PeripheriqueTTPA::nom [read]
```

Référencé par [connecter\(\)](#), [deconnecter\(\)](#), [envoyer\(\)](#), [estConnecte\(\)](#), [getNom\(\)](#), [socketConnecte\(\)](#), [socketDeconnecte\(\)](#), [socket←Erreur\(\)](#), et [socketReadyRead\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

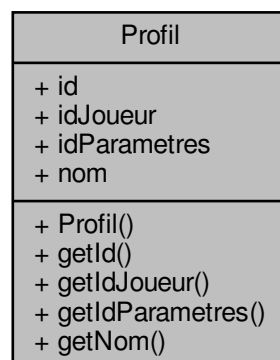
- [peripheriqueTTPA.h](#)
- [peripheriqueTTPA.cpp](#)

9.12 Référence de la classe Profil

Déclaration de la classe [Profil](#).

```
#include <Profil.h>
```

Graphe de collaboration de Profil :



Signaux

- void `idChanged ()`
- void `idJoueurChanged ()`
- void `idParametresChanged ()`
- void `nomChanged ()`

Fonctions membres publiques

- `Profil (QString id="", QString idJoueur="", QString idParametres="", QString nom="", QObject *parent=nullptr)`
- `QString getId ()`
- `QString getIdJoueur ()`
- `QString getIdParametres ()`
- `QString getNom ()`

Propriétés

- `QString id`
- `QString idJoueur`
- `QString idParametres`
- `QString nom`

9.12.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

9.12.2 Documentation des constructeurs et destructeur

9.12.2.1 Profil()

```
Profil::Profil (
    QString id = "",
    QString idJoueur = "",
    QString idParametres = "",
    QString nom = "",
    QObject * parent = nullptr ) [explicit]
```

Références `nom`.

```
00015                                     : QObject (
    parent), id(id), idJoueur(idJoueur), idParametres(
    idParametres), nom(nom)
00016 {
00017     qDebug() << Q_FUNC_INFO << "id" << id << "idJoueur" << idJoueur << "idParametres" <<
    idParametres << "nom" << nom;
00018 }
```

9.12.3 Documentation des fonctions membres

9.12.3.1 getId()

```
QString Profil::getId ( )
```

Références [id](#).

```
00021 {  
00022     return id;  
00023 }
```

9.12.3.2 getIdJoueur()

```
QString Profil::getIdJoueur ( )
```

Références [idJoueur](#).

```
00026 {  
00027     return idJoueur;  
00028 }
```

9.12.3.3 getIdParametres()

```
QString Profil::getIdParametres ( )
```

Références [idParametres](#).

```
00031 {  
00032     return idParametres;  
00033 }
```

9.12.3.4 getNom()

```
QString Profil::getNom ( )
```

Références [nom](#).

```
00036 {  
00037     return nom;  
00038 }
```

9.12.3.5 idChanged

```
void Profil::idChanged ( ) [signal]
```

9.12.3.6 idJoueurChanged

```
void Profil::idJoueurChanged ( ) [signal]
```

9.12.3.7 idParametresChanged

```
void Profil::idParametresChanged ( ) [signal]
```

9.12.3.8 nomChanged

```
void Profil::nomChanged ( ) [signal]
```

9.12.4 Documentation des propriétés

9.12.4.1 id

```
QString Profil::id [read]
```

Référencé par [getId\(\)](#).

9.12.4.2 idJoueur

```
QString Profil::idJoueur [read]
```

Référencé par [getIdJoueur\(\)](#).

9.12.4.3 idParametres

```
QString Profil::idParametres [read]
```

Référencé par [getIdParametres\(\)](#).

9.12.4.4 nom

```
QString Profil::nom [read]
```

Référencé par [getNom\(\)](#), et [Profil\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

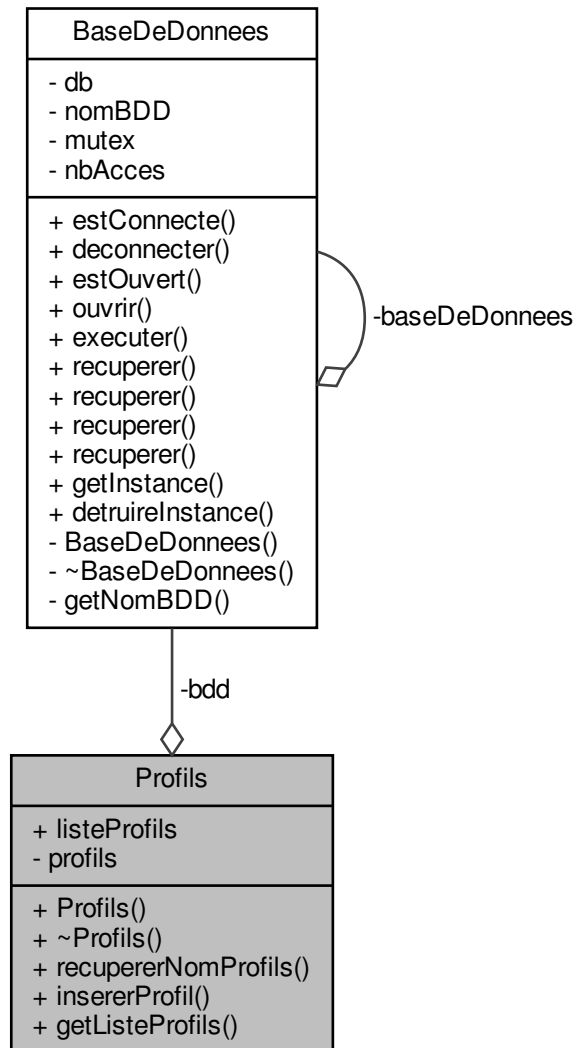
- [Profil.h](#)
- [Profil.cpp](#)

9.13 Référence de la classe Profils

Déclaration de la classe [Profils](#).

```
#include <Profils.h>
```

Graphique de collaboration de Profils :



Signaux

— void [listeProfilsUpdated](#) ()

Fonctions membres publiques

— [Profils](#) (QObject *parent=nullptr)
 — [~Profils](#) ()
 — Q_INVOKABLE bool [recupererNomProfils](#) (QString idJoueur="")
 — Q_INVOKABLE bool [insererProfil](#) (QString nom, QString idJoueur, QString idParametres)
 — QVariant [getListeProfils](#) ()

Propriétés

— QVariant [listeProfils](#)

Attributs privés

— QList< QObject * > [profils](#)
 — [BaseDeDonnees](#) * [bdd](#)

9.13.1 Description détaillée**Auteur**

Hammouma Youssef

Version

1.4

9.13.2 Documentation des constructeurs et destructeur**9.13.2.1 Profils()**

```
Profils::Profils (
    QObject * parent = nullptr ) [explicit]
```

Références [bdd](#), [BaseDeDonnees : :getInstance\(\)](#), et [BaseDeDonnees : :ouvrir\(\)](#).

```
00018                                     : QObject (parent)
00019 {
00020     bdd = BaseDeDonnees : :getInstance\(\);
00021     bdd->ouvrir("ttpa.sqlite");
00022 }
```

9.13.2.2 ~Profils()

```
Profils::~~Profils ( )
```

Références [BaseDeDonnees : :destruireInstance\(\)](#).

```
00025 {
00026     BaseDeDonnees : :destruireInstance\(\);
00027 }
```

9.13.3 Documentation des fonctions membres

9.13.3.1 getListeProfils()

```
QVariant Profils::getListeProfils ( )
```

Références [profils](#).

```
00073 {  
00074     return QVariant::fromValue(profils);  
00075 }
```

9.13.3.2 insererProfil()

```
bool Profils::insererProfil (   
    QString nom,  
    QString idJoueur,  
    QString idParametres )
```

Références [bdd](#), [BaseDeDonnees : :executer\(\)](#), et [recupererNomProfils\(\)](#).

```
00059 {  
00060     qDebug() << Q_FUNC_INFO ;  
00061     QString requete = "INSERT INTO profils(idJoueur,idParametres,nom) VALUES('" + idJoueur + "', '" +  
        idParametres + "', '" + nom + "')";  
00062     bool etat = bdd->executer(requete);  
00063     qDebug() << Q_FUNC_INFO << "requete" << requete;  
00064     if(etat)  
00065     {  
00066         recupererNomProfils(idJoueur);  
00067         return true;  
00068     }  
00069     return false;  
00070 }
```

9.13.3.3 listeProfilsUpdated

```
void Profils::listeProfilsUpdated ( ) [signal]
```

Référencé par [recupererNomProfils\(\)](#).

9.13.3.4 recupererNomProfils()

```
bool Profils::recupererNomProfils (   
    QString idJoueur = "" )
```

Références [bdd](#), [listeProfilsUpdated\(\)](#), [profils](#), et [BaseDeDonnees : :recuperer\(\)](#).

Référencé par [insererProfil\(\)](#).

```

00030 {
00031     QString requete = "SELECT idProfil, idJoueur, idParametres, nom FROM profils WHERE idJoueur =" +
    idJoueur + "'";
00032     QVector<QStringList> lesProfils;
00033
00034     if(idJoueur.isEmpty())
00035     {
00036         profils.clear();
00037         emit listeProfilsUpdated();
00038         return true;
00039     }
00040
00041     bool etat = bdd->recuperer(requete, lesProfils);
00042     if(etat)
00043     {
00044         qDebug() << Q_FUNC_INFO << lesProfils;
00045         profils.clear();
00046         profils.append(new Profil());
00047         for(int i = 0; i < lesProfils.size(); i++)
00048         {
00049             Profil *profil = new Profil(lesProfils.at(i).at(0), lesProfils.at(i).at(1),
    lesProfils.at(i).at(2), lesProfils.at(i).at(3), this);
00050             profils.append(profil);
00051         }
00052         emit listeProfilsUpdated();
00053         return true;
00054     }
00055     return false;
00056 }

```

9.13.4 Documentation des données membres

9.13.4.1 bdd

`BaseDeDonnees* Profils::bdd` [private]

Référencé par [insérerProfil\(\)](#), [Profils\(\)](#), et [recupererNomProfils\(\)](#).

9.13.4.2 profils

`QList<QObject*> Profils::profils` [private]

Référencé par [getListeProfils\(\)](#), et [recupererNomProfils\(\)](#).

9.13.5 Documentation des propriétés

9.13.5.1 listeProfils

`QVariant Profils::listeProfils` [read]

La documentation de cette classe a été générée à partir des fichiers suivants :

- [Profils.h](#)
- [Profils.cpp](#)

9.14 Référence de la classe StatistiquesSeance

Déclaration de la classe [StatistiquesSeance](#).

```
#include <statistiquesseance.h>
```

Graphique de collaboration de StatistiquesSeance :

StatistiquesSeance
<ul style="list-style-type: none"> - numeroZoneObjectif - numeroZoneRobot - nbImpactsZone - nbBallesBonnes - nbBallesTotales - nbBallesTotalesTable - nbBallesEnchainees - nbBallesEnchaineesMax - balleToucheTable - ballePrecedenteToucheTable - balleToucheRobot
<ul style="list-style-type: none"> + StatistiquesSeance() + ~StatistiquesSeance() + impacterZone() + initialiserStatistiques() + getBallesBonnes() + getBallesTotal() + getBallesHorsTable() + getballesPrecedentToucheTable() + getballesToucheTable() + getBallesEnchainees() et 10 de plus... + finirSeance() - calculerStatistiques()

Types publics

```
— enum Zone {
    ZoneNonDefinie = -1, Zone1 = 0, Zone2, Zone3,
    Zone4, Zone5, Zone6, Zone7,
    NbZonesImpact, ZoneHorsTable, NbZonesTotales }
    les zones numérotés de 1 à 7 de la table plus une zone non définie et le nombre de zone total
```

Connecteurs publics

```
— void finirSeance ()
    fini la seance
```

Signaux

- void `affichageZone` (int numeroZone, int nbImpacts, QString pourcentage)
- void `ballesTotales` (int `nbBallesTotales`, int `nbBallesTotalesTable`)
- void `ballesTotalesHorsTable` (int `ballesToucheTable`)
- void `ballesBonnes` (int `nbBallesBonnes`)

Fonctions membres publiques

- `StatistiquesSeance` (QObject *parent=nullptr)
- `~StatistiquesSeance` ()
Destructeur de la seance.
- bool `impacterZone` (uint8_t numeroZone)
Calcul et affiche l'impact sur l'IHM et la table.
- void `initialiserStatistiques` ()
initialise les statistiques
- int `getBallesBonnes` ()
- int `getBallesTotal` ()
recupère les données de la zone du robot
- int `getBallesHorsTable` ()
recupère les données des balles hors table
- int `getballesPrecedentToucheTable` ()
recupère les données des balles ayant touché précédemment la table
- int `getballesToucheTable` ()
recupère les données des balles recupère les données des balles ayant touché la table
- int `getBallesEnchainees` ()
- int `getZoneObjectif` ()
recupère les données de la zone de l'objectif
- void `setZoneObjectif` (int numeroZone)
définie la Zone où sera la zone objectif
- int `getZoneRobot` ()
- void `setZoneRobot` (int numeroZone)
définie la Zone où sera la zone robot
- double `calculerPourcentage` (int `nbImpactsZone`)
calcul le pourcentage de chaque zone sur le nombre de balle
- double `calculerPourcentageSerieMax` (int `nbImpactsZone`)
calcul le pourcentage du nombre de série max sur le nombre de balle dans l'objectif
- void `afficherStatistiquesZones` ()
affiche les statistiques (impact et pourcentage) de toutes les zones
- void `afficherStatistiquesBallesTotales` ()
affiche le nombre de balle total et nombre total de balle ayant touché la table
- void `afficherStatistiquesBallesTotalesHorsTable` ()
affiche le nombre de balle hors table
- void `afficherStatistiqueBallesBonnes` ()
affiche le nombre de balle ayant atteint la zone objectifs

Fonctions membres privées

- void `calculerStatistiques` (uint8_t numeroZone)

Attributs privés

- int `numeroZoneObjectif`
Placement de la zone de l'objectif.
- int `numeroZoneRobot`
Placement de la zone d'objectif.
- QVector< int > `nbImpactsZone`
Un vecteur avec le nombre d'impacts dans chaque zone.
- int `nbBallesBonnes`
Nombre de balles ayant touché la zone objectifs.
- int `nbBallesTotales`
Nombre de balles total tirées par le robot et jouables.
- int `nbBallesTotalesTable`
Nombre de balles totales tirées par le robot et ayant impacté la table.
- int `nbBallesEnchainees`
Série de balles dans la zone objectif.

- int [nbBallesEnchaineesMax](#)
Série max de balles dans la zone objectif.
- bool [balleToucheTable](#)
Balle ayant touché la table.
- bool [ballePrecedenteToucheTable](#)
Balle précédente qui a touché la table.
- bool [balleToucheRobot](#)
Balle ayant touché la table du robot.

9.14.1 Description détaillée

Auteur

Botella-Broc Yohann

Version

1.4

9.14.2 Documentation des énumérations membres

9.14.2.1 Zone

enum [StatistiquesSeance::Zone](#)

Valeurs énumérées

ZoneNonDefinie	
Zone1	
Zone2	
Zone3	
Zone4	
Zone5	
Zone6	
Zone7	
NbZonesImpact	
ZoneHorsTable	
NbZonesTotales	

```
00045      {
00046          ZoneNonDefinie = -1,
00047          Zone1 = 0,
00048          Zone2,
00049          Zone3,
00050          Zone4,
00051          Zone5,
00052          Zone6,
00053          Zone7,
00054          NbZonesImpact, // 7
00055          ZoneHorsTable, // 8
00056          NbZonesTotales, // 9
00057      };
```

9.14.3 Documentation des constructeurs et destructeur

9.14.3.1 StatistiquesSeance()

```
StatistiquesSeance::StatistiquesSeance (
    QObject * parent = nullptr )
```

Références [initialiserStatistiques\(\)](#).

```
00023                                     : QObject (parent),
    numeroZoneObjectif (StatistiquesSeance::ZoneNonDefinie),
    numeroZoneRobot (StatistiquesSeance::ZoneNonDefinie),
    nbImpactsZone (StatistiquesSeance::NbZonesTotales, 0),
    nbBallesBonnes (0), nbBallesTotales (0),
    nbBallesTotalesTable (0), nbBallesEnchainees (0),
    nbBallesEnchaineesMax (0)
00024 {
00025     qDebug () << Q_FUNC_INFO;
00026     initialiserStatistiques ();
00027 }
```

9.14.3.2 ~StatistiquesSeance()

```
StatistiquesSeance::~~StatistiquesSeance ( )
```

```
00036 {
00037     qDebug () << Q_FUNC_INFO;
00038 }
```

9.14.4 Documentation des fonctions membres

9.14.4.1 affichageZone

```
void StatistiquesSeance::affichageZone (
    int numeroZone,
    int nbImpacts,
    QString pourcentage ) [signal]
```

Référencé par [afficherStatistiquesZones\(\)](#).

9.14.4.2 afficherStatistiqueBallesBonnes()

```
void StatistiquesSeance::afficherStatistiqueBallesBonnes ( )
```

Références [ballesBonnes\(\)](#), et [getBallesBonnes\(\)](#).

Référencé par [impacterZone\(\)](#).

```
00225 {
00226     emit ballesBonnes (getBallesBonnes ());
00227 }
```

9.14.4.3 afficherStatistiquesBallesTotales()

```
void StatistiquesSeance::afficherStatistiquesBallesTotales ( )
```

Références [ballesTotales\(\)](#), [getBallesTotal\(\)](#), et [getballesToucheTable\(\)](#).

Référencé par [Ttpalhm : commencerSeance\(\)](#), et [impacterZone\(\)](#).

```
00203 {
00204     emit ballesTotales(getBallesTotal(),
00205                       getballesToucheTable());
00205 }
```

9.14.4.4 afficherStatistiquesBallesTotalesHorsTable()

```
void StatistiquesSeance::afficherStatistiquesBallesTotalesHorsTable ( )
```

Références [ballesTotalesHorsTable\(\)](#), et [getballesToucheTable\(\)](#).

```
00214 {
00215     emit ballesTotalesHorsTable(getballesToucheTable());
00216 }
```

9.14.4.5 afficherStatistiquesZones()

```
void StatistiquesSeance::afficherStatistiquesZones ( )
```

Références [affichageZone\(\)](#), [calculerPourcentage\(\)](#), [nbImpactsZone](#), [NbZonesImpact](#), [NbZonesTotales](#), et [Zone1](#).

Référencé par [Ttpalhm : commencerSeance\(\)](#), et [impacterZone\(\)](#).

```
00183 {
00184     double pourcentage;
00185     for(uint8_t numeroZone = Zone1; numeroZone < NbZonesTotales; ++numeroZone)
00186     {
00187         if(numeroZone == NbZonesImpact)
00188             continue;
00189
00190         pourcentage = calculerPourcentage(nbImpactsZone[numeroZone]);
00191         emit affichageZone(numeroZone, nbImpactsZone[numeroZone], QString::number
00192 (pourcentage, 'f', 0) + " %");
00192         qDebug() << Q_FUNC_INFO << "numeroZone=" << numeroZone << "nbImpactsZone" <<
00193         nbImpactsZone[numeroZone] << "pourcentage" << pourcentage;
00193     }
00194 }
```

9.14.4.6 ballesBonnes

```
void StatistiquesSeance::ballesBonnes (
    int nbBallesBonnes ) [signal]
```

Référencé par [afficherStatistiqueBallesBonnes\(\)](#), et [finirSeance\(\)](#).

9.14.4.7 ballesTotales

```
void StatistiquesSeance::ballesTotales (
    int nbBallesTotales,
    int nbBallesTotalesTable ) [signal]
```

Référencé par [afficherStatistiquesBallesTotales\(\)](#).

9.14.4.8 ballesTotalesHorsTable

```
void StatistiquesSeance::ballesTotalesHorsTable (
    int ballesToucheTable ) [signal]
```

Référencé par [afficherStatistiquesBallesTotalesHorsTable\(\)](#), et [finirSeance\(\)](#).

9.14.4.9 calculerPourcentage()

```
double StatistiquesSeance::calculerPourcentage (
    int nbImpactsZone )
```

Paramètres

<i>nbImpactsZone</i>	int le numéro de zone impacté
----------------------	-------------------------------

Renvoie

double le pourcentage calculé

Références [nbBallesTotales](#).

Référencé par [Ttpalhm : :afficherStatistiquePourcentageBallesHorstable\(\)](#), [Ttpalhm : :afficherStatistiquePourcentageBallesObjectif\(\)](#), [Ttpalhm : :afficherStatistiquePourcentageBallesSerie\(\)](#), et [afficherStatistiquesZones\(\)](#).

```
00155 {
00156     if (nbImpactsZone == 0)
00157         return 0.;
00158
00159     return (double(nbImpactsZone) / double(nbBallesTotales))*100.;
00160 }
```

9.14.4.10 calculerPourcentageSerieMax()

```
double StatistiquesSeance::calculerPourcentageSerieMax (
    int nbImpactsZone )
```

Paramètres

<i>nbImpactsZone</i>	int le numéro de zone impacté
----------------------	-------------------------------

Renvoie

QString le pourcentage formaté pour l'affichage "X%"

Références [nbBallesBonnes](#), et [nbBallesEnchaineesMax](#).

Référencé par [Ttpalm : :afficherStatistiquePourcentageBallesSerie\(\)](#).

```
00170 {
00171     if (nbImpactsZone == 0)
00172         return 0.;
00173
00174     return (double(nbBallesEnchaineesMax) / double(
00175         nbBallesBonnes)) * 100.;
00176 }
```

9.14.4.11 calculerStatistiques()

```
void StatistiquesSeance::calculerStatistiques (
    uint8_t numeroZone ) [private]
```

Références [ballePrecedenteToucheTable](#), [balleToucheTable](#), [nbBallesBonnes](#), [nbBallesEnchainees](#), [nbBallesEnchaineesMax](#), [nbBallesTotales](#), [nbBallesTotalesTable](#), [nbImpactsZone](#), [NbZonesImpact](#), [NbZonesTotales](#), [numeroZoneObjectif](#), [numeroZoneRobot](#), [Zone1](#), [ZoneHorsTable](#), et [ZoneNonDefinie](#).

Référencé par [impacterZone\(\)](#).

```
00072 {
00073     qDebug() << Q_FUNC_INFO << "numeroZone = " << numeroZone << "numeroZoneObjectif = " <<
00074         numeroZoneObjectif;
00075
00076     if (numeroZone == numeroZoneRobot)
00077         nbImpactsZone[ZoneHorsTable] += 1;
00078     else
00079         nbImpactsZone[numeroZone] += 1;
00080     if (numeroZone == numeroZoneObjectif)
00081     {
00082         nbBallesBonnes++;
00083     }
00084     qDebug() << Q_FUNC_INFO << "nbBalles zone = " << nbImpactsZone[numeroZone];
00085
00086     ballePrecedenteToucheTable = balleToucheTable;
00087     balleToucheTable = false;
00088     ballePrecedenteToucheTable = balleToucheTable;
00089     balleToucheTable = numeroZone;
00090
00091     nbBallesTotales++;
00092     qDebug() << Q_FUNC_INFO << "nbBallesTotales = " << nbBallesTotales;
00093
00094     nbBallesTotalesTable = 0;
00095     for (int numeroZone = Zone1; numeroZone < NbZonesTotales; ++numeroZone)
00096     {
00097         if (numeroZone == NbZonesImpact)
00098             continue;
00099         if (numeroZone < NbZonesImpact)
00100             nbBallesTotalesTable += nbImpactsZone[numeroZone];
00101     }
00102     qDebug() << Q_FUNC_INFO << "nbBallesTotalesTable = " << nbBallesTotalesTable << "
00103     nbBallesBonnes = " << nbBallesBonnes;
00104
00105     if (numeroZone != ZoneNonDefinie)
00106     {
00107         if (numeroZone == numeroZoneObjectif /*|| numeroZoneObjectif == ZoneNonDefinie*/)
00108         {
00109             nbBallesEnchainees++;
00110         }
00111         else
00112         {
00113             nbBallesEnchainees = 0;
00114         }
00115     }
00116     else
00117         nbBallesEnchainees = 0;
00118
00119     if (nbBallesEnchainees > nbBallesEnchaineesMax)
00120         nbBallesEnchaineesMax = nbBallesEnchainees;
00121     qDebug() << Q_FUNC_INFO << "nbBallesEnchainees = " << nbBallesEnchainees << "
00122     nbBallesEnchaineesMax = " << nbBallesEnchaineesMax;
00123 }
```

9.14.4.12 finirSeance

```
void StatistiquesSeance::finirSeance ( ) [slot]
```

Références [ballesBonnes\(\)](#), [ballesTotalesHorsTable\(\)](#), [getBallesBonnes\(\)](#), et [getBallesHorsTable\(\)](#).

```
00141 {
00142     //...
00143     emit ballesBonnes(getBallesBonnes());
00144     emit ballesTotalesHorsTable(getBallesHorsTable());
00145 }
```

9.14.4.13 getBallesBonnes()

```
int StatistiquesSeance::getBallesBonnes ( )
```

Références [nbBallesBonnes](#).

Référencé par [afficherStatistiqueBallesBonnes\(\)](#), [Ttpalhm : :afficherStatistiqueEcranResultats\(\)](#), [Ttpalhm : :afficherStatistiquePourcentageBallesObjectif\(\)](#), et [finirSeance\(\)](#).

```
00317 {
00318     return nbBallesBonnes;
00319 }
```

9.14.4.14 getBallesEnchainees()

```
int StatistiquesSeance::getBallesEnchainees ( )
```

Références [nbBallesEnchaineesMax](#).

Référencé par [Ttpalhm : :afficherStatistiqueEcranResultats\(\)](#), et [Ttpalhm : :afficherStatistiquePourcentageBallesSerie\(\)](#).

```
00322 {
00323     return nbBallesEnchaineesMax;
00324 }
```

9.14.4.15 getBallesHorsTable()

```
int StatistiquesSeance::getBallesHorsTable ( )
```

Renvoie

Références [nbImpactsZone](#), et [ZoneHorsTable](#).

Référencé par [Ttpalhm : :afficherStatistiqueEcranResultats\(\)](#), [Ttpalhm : :afficherStatistiquePourcentageBallesHorstable\(\)](#), [Ttpalhm : :afficherStatistiquePourcentageBallesSerie\(\)](#), et [finirSeance\(\)](#).

```
00268 {
00269     return nbImpactsZone[ZoneHorsTable];
00270 }
```

9.14.4.16 getballesPrecedentToucheTable()

```
int StatistiquesSeance::getballesPrecedentToucheTable ( )
```

Renvoie

Références [ballePrecedenteToucheTable](#).

```
00279 {  
00280     return ballePrecedenteToucheTable;  
00281 }
```

9.14.4.17 getBallesTotal()

```
int StatistiquesSeance::getBallesTotal ( )
```

recupère les données des balles tiré

Renvoie

Références [nbBallesTotales](#).

Référencé par [Ttpalhm : :afficherStatistiqueEcranResultats\(\)](#), et [afficherStatistiquesBallesTotales\(\)](#).

```
00257 {  
00258     return nbBallesTotales;  
00259 }
```

9.14.4.18 getballesToucheTable()

```
int StatistiquesSeance::getballesToucheTable ( )
```

Renvoie

Références [nbBallesTotalesTable](#).

Référencé par [afficherStatistiquesBallesTotales\(\)](#), et [afficherStatistiquesBallesTotalesHorsTable\(\)](#).

```
00290 {  
00291     return nbBallesTotalesTable;  
00292 }
```

9.14.4.19 `getZoneObjectif()`

```
int StatistiquesSeance::getZoneObjectif ( )
```

Renvoie

Références [numeroZoneObjectif](#).

Référencé par [Ttpalhm : :afficherImpactCouleurs\(\)](#), et [Ttpalhm : :effacerImpactZone\(\)](#).

```
00301 {
00302     return numeroZoneObjectif;
00303 }
```

9.14.4.20 `getZoneRobot()`

```
int StatistiquesSeance::getZoneRobot ( )
```

Références [numeroZoneRobot](#).

Référencé par [Ttpalhm : :afficherImpactCouleurs\(\)](#), et [Ttpalhm : :afficherZone\(\)](#).

```
00235 {
00236     return numeroZoneRobot;
00237 }
```

9.14.4.21 `impacterZone()`

```
bool StatistiquesSeance::impacterZone (
    uint8_t numeroZone )
```

Paramètres

<i>numeroZone</i>	int le numéro de zone de l'impact
-------------------	-----------------------------------

Renvoie

Références [afficherStatistiqueBallesBonnes\(\)](#), [afficherStatistiquesBallesTotales\(\)](#), [afficherStatistiquesZones\(\)](#), et [calculerStatistiques\(\)](#).

Référencé par [Ttpalhm : :recevoirNumeroZone\(\)](#), et [Ttpalhm : :saisirNumeroZone\(\)](#).

```
00123 {
00124     qDebug() << Q_FUNC_INFO << "numeroZone=" << numeroZone /*<< "numeroZoneRobot" << numeroZoneRobot*/;
00125
00126     calculerStatistiques(numeroZone);
00127     afficherStatistiquesZones();
00128     afficherStatistiquesBallesTotales();
00129     afficherStatistiqueBallesBonnes();
00130
00131     return true;
00132 }
```

9.14.4.22 initialiserStatistiques()

```
void StatistiquesSeance::initialiserStatistiques ( )
```

Références [ballePrecedenteToucheTable](#), [balleToucheRobot](#), [balleToucheTable](#), [nbBallesBonnes](#), [nbBallesEnchainees](#), [nbBallesEnchaineesMax](#), [nbBallesTotales](#), [nbBallesTotalesTable](#), [nbImpactsZone](#), [numeroZoneObjectif](#), [numeroZoneRobot](#), et [ZoneNonDefinie](#).

Référencé par [Ttpalhm : :resetSeance\(\)](#), [StatistiquesSeance\(\)](#), et [Ttpalhm : :Ttpalhm\(\)](#).

```
00047 {
00048     for(int i=0; i < nbImpactsZone.size(); i++)
00049     {
00050         nbImpactsZone[i] = 0;
00051     }
00052     numeroZoneObjectif = numeroZoneRobot =
StatistiquesSeance::ZoneNonDefinie;
00053     nbBallesBonnes = 0;
00054     nbBallesTotales = 0;
00055     nbBallesTotalesTable = 0;
00056     nbBallesEnchainees = 0;
00057     nbBallesEnchaineesMax = 0;
00058     balleToucheTable = true;
00059     ballePrecedenteToucheTable = true;
00060     balleToucheRobot = true;
00061 }
```

9.14.4.23 setZoneObjectif()

```
void StatistiquesSeance::setZoneObjectif (
    int numeroZone )
```

Références [numeroZoneObjectif](#).

Référencé par [Ttpalhm : :setZoneObjectif\(\)](#).

```
00311 {
00312     numeroZoneObjectif = numeroZone;
00313     qDebug() << Q_FUNC_INFO << "numeroZoneObjectif" << numeroZoneObjectif;
00314 }
```

9.14.4.24 setZoneRobot()

```
void StatistiquesSeance::setZoneRobot (
    int numeroZone )
```

Références [numeroZoneRobot](#).

Référencé par [Ttpalhm : :setZoneRobot\(\)](#).

```
00245 {
00246     numeroZoneRobot = numeroZone;
00247     qDebug() << Q_FUNC_INFO << "numeroZoneRobot" << numeroZoneRobot;
00248 }
```

9.14.5 Documentation des données membres

9.14.5.1 `ballePrecedenteToucheTable`

```
bool StatistiquesSeance::ballePrecedenteToucheTable [private]
```

Référencé par [calculerStatistiques\(\)](#), [getballesPrecedentToucheTable\(\)](#), et [initialiserStatistiques\(\)](#).

9.14.5.2 `balleToucheRobot`

```
bool StatistiquesSeance::balleToucheRobot [private]
```

Référencé par [initialiserStatistiques\(\)](#).

9.14.5.3 `balleToucheTable`

```
bool StatistiquesSeance::balleToucheTable [private]
```

Référencé par [calculerStatistiques\(\)](#), et [initialiserStatistiques\(\)](#).

9.14.5.4 `nbBallesBonnes`

```
int StatistiquesSeance::nbBallesBonnes [private]
```

Référencé par [calculerPourcentageSerieMax\(\)](#), [calculerStatistiques\(\)](#), [getBallesBonnes\(\)](#), et [initialiserStatistiques\(\)](#).

9.14.5.5 `nbBallesEnchainees`

```
int StatistiquesSeance::nbBallesEnchainees [private]
```

Référencé par [calculerStatistiques\(\)](#), et [initialiserStatistiques\(\)](#).

9.14.5.6 `nbBallesEnchaineesMax`

```
int StatistiquesSeance::nbBallesEnchaineesMax [private]
```

Référencé par [calculerPourcentageSerieMax\(\)](#), [calculerStatistiques\(\)](#), [getBallesEnchainees\(\)](#), et [initialiserStatistiques\(\)](#).

9.14.5.7 nbBallesTotales

```
int StatistiquesSeance::nbBallesTotales [private]
```

Référencé par [calculerPourcentage\(\)](#), [calculerStatistiques\(\)](#), [getBallesTotal\(\)](#), et [initialiserStatistiques\(\)](#).

9.14.5.8 nbBallesTotalesTable

```
int StatistiquesSeance::nbBallesTotalesTable [private]
```

Référencé par [calculerStatistiques\(\)](#), [getballesToucheTable\(\)](#), et [initialiserStatistiques\(\)](#).

9.14.5.9 nbImpactsZone

```
QVector<int> StatistiquesSeance::nbImpactsZone [private]
```

Référencé par [afficherStatistiquesZones\(\)](#), [calculerStatistiques\(\)](#), [getBallesHorsTable\(\)](#), et [initialiserStatistiques\(\)](#).

9.14.5.10 numeroZoneObjectif

```
int StatistiquesSeance::numeroZoneObjectif [private]
```

Référencé par [calculerStatistiques\(\)](#), [getZoneObjectif\(\)](#), [initialiserStatistiques\(\)](#), et [setZoneObjectif\(\)](#).

9.14.5.11 numeroZoneRobot

```
int StatistiquesSeance::numeroZoneRobot [private]
```

Référencé par [calculerStatistiques\(\)](#), [getZoneRobot\(\)](#), [initialiserStatistiques\(\)](#), et [setZoneRobot\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

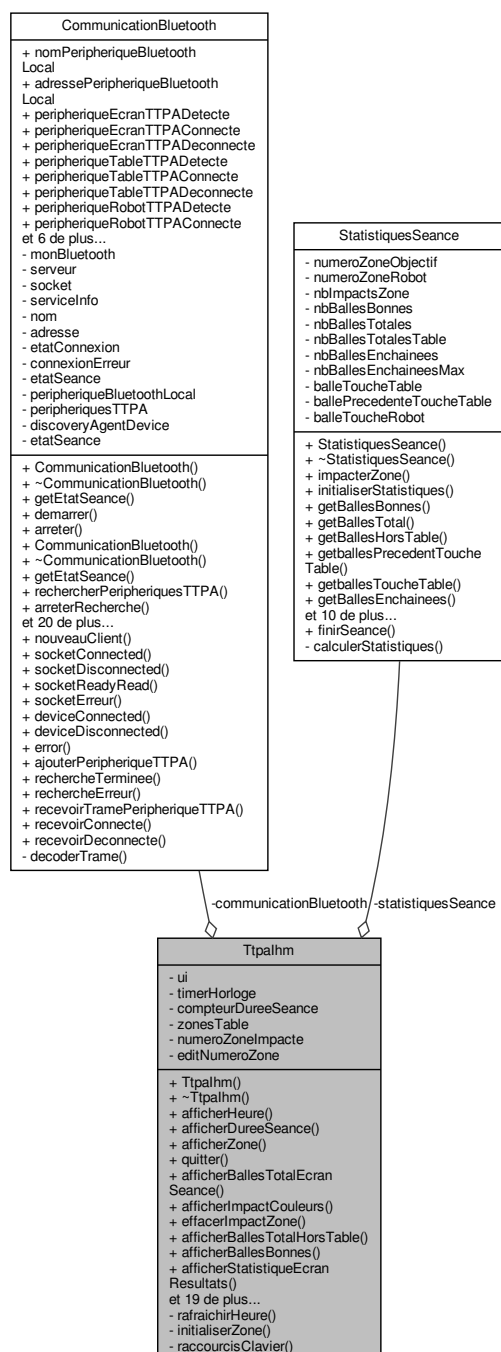
- [statistiquesseance.h](#)
- [statistiquesseance.cpp](#)

9.15 Référence de la classe Ttpalhm

La fenêtre principale de l'application.

```
#include <ttpaihm.h>
```

Graphe de collaboration de Ttpalhm :



Connecteurs publics

— void [afficherHeure](#) ()

- *Affiche l'heure périodiquement (slot)*
- void `afficherDureeSeance` ()
- *Affiche la durée d'une Seance(slot)*
- void `afficherZone` (int numeroZone, int nblImpacts, QString pourcentage)
- *Affiche le nombre d'impacts et le pourcentage dans une zone.*
- void `quitter` ()
- *Permet de quitter l'application.*
- void `afficherBallesTotalEcranSeance` (int nombreTotalDeBalle, int nombreTotalDeBalleTable)
- *Affiche le nombre de balle ayant toucher la table sur le nombre total de balle jouer par le robot sur l'écran de séance.*
- void `afficherImpactCouleurs` (int numeroZone)
- *Affiche la couleur d'une zone lors qu'elle est impacter.*
- void `effacerImpactZone` ()
- *Efface la couleur émise par afficherImpactCouleurs pour la remettre de la couleur habituel au bout de 500 ms(bleu)*
- void `afficherBallesTotalHorsTable` (int nombreTotalDeBalleHorsTable)
- *Affiche le nombre de balle hors zone durant toute la partie et l'affiche sur l'écran résultats(surcharge)*
- void `afficherBallesBonnes` (int nombreDeBallesBonnes)
- *Affiche le nombre de balle ayant atteints l'objectifs sur l'écran résultats.*
- void `afficherStatistiqueEcranResultats` ()
- *Affiche sur l'écran résultats toutes les statistique nécessaire comme balles ayant atteint l'objectifs, balles hors de la table et le nombre de série de balle sur l'objectifs.*
- void `afficherStatistiquePourcentageBallesHorstable` ()
- void `afficherStatistiquePourcentageBallesSerie` ()
- void `afficherStatistiquePourcentageBallesObjectif` ()
- void `setZoneRobot` (QString numeroZone)
- *Place le robot sur la table.*
- void `setZoneObjectif` (QString numeroZone)
- *Place le robot sur la table.*
- void `recevoirNumeroZone` (QString numeroZone)
- void `afficherConnexion` ()
- void `connecterJoueur` (QString nomJoueur)
- *Actions nécessaires a la connexion du joueur (slot)*
- void `deconnecterJoueur` ()
- *Actions nécessaires a la deconnexion du joueur (slot)*
- void `commencerSeance` (QString nomJoueur, QString zoneObjectif, QString zoneRobot)
- void `mettreEnPauseSeance` ()
- void `repandreSeance` ()
- void `finirSeance` ()
- void `resetSeance` ()
- void `erreurRobot` (QString infoDuRobot)
- void `InfoRobot` (QString infoDuRobot)
- void `saisirNumeroZone` (const QString)
- *Permet de simuler un impact de balle à partir du clavier numérique (mode debug)*
- void `allerSuivant` ()
- *Permet d'aller à l'écran suivant (mode debug)*
- void `revenirPrecedent` ()
- *Permet de revenir à l'écran précédent (mode debug)*

Fonctions membres publiques

- `Ttpalhm` (QWidget *parent=nullptr)
- *Constructeur des statistique d'une Seance.*
- `~Ttpalhm` ()
- *Destructeur de la fenêtre principale.*

Types privés

- enum `Page` { `PageAttente` = 0, `PageSeance`, `PageResultats`, `NbPages` }
- *Enumeration des fenetres de l'IHM.*

Fonctions membres privées

- void `rafraichirHeure` ()
- void `initialiserZone` ()
- *Initialise le vecteur des zones de la table.*
- void `raccourcisClavier` ()
- *Action ctrl+Q pour Quitte.*

Attributs privés

- `Ui : Ttpalhm * ui`
agrégation de la partie graphique de l'IHM
- `QTimer * timerHorloge`
temporisateur pour le rafraichissement de l'heure
- `int compteurDureeSeance`
compteur de secondes pour la durée d'une séance
- `QVector< QLabel * > zonesTable`
vecteur pour les labels de zone
- `StatistiquesSeance * statistiquesSeance`
les statistiques d'une séance
- `CommunicationBluetooth * communicationBluetooth`
pour communiquer en Bluetooth avec la tablette
- `int numeroZoneImpacte`
le numero de la zone qui a été impacté par la balle
- `QLineEdit * editNumeroZone`
pour la simulation d'un impact de balle à partir du clavier numérique (mode debug)

9.15.1 Description détaillée

Auteur

Botella-Broc Yohann

Version

1.4

9.15.2 Documentation des énumérations membres

9.15.2.1 Page

```
enum Ttpalhm::Page [private]
```

Valeurs énumérées

PageAttente	
PageSeance	
PageResultats	
NbPages	

```
00069      {
00070          PageAttente = 0,
00071          PageSeance,
00072          PageResultats,
00073          NbPages
00074      };
```

9.15.3 Documentation des constructeurs et destructeur

9.15.3.1 Ttpalhm()

```
Ttpalhm::Ttpalhm (
    QWidget * parent = nullptr ) [explicit]
```

Constructeur de la fenêtre principale.

Paramètres

<i>parent</i>	QObject Adresse de l'objet Qt parent numeroZoneObjectif définit la zone de l'objectif numeroZoneRobot définit la zone du robot nbImpactsZone définit le nombre d'impacte dans une zone (1 à 7) nbBallesTotales tiré par le robot.
<i>parent</i>	QObject Adresse de l'objet Qt parent (ici 0 car c'est la fenêtre principale)

ajout d'un bouton dans chaque écran peut être désactiver avec le .pro

Références [afficherBallesBonnes\(\)](#), [afficherBallesTotalEcranSeance\(\)](#), [afficherBallesTotalHorsTable\(\)](#), [afficherConnexion\(\)](#), [afficherHeure\(\)](#), [afficherZone\(\)](#), [allerSuivant\(\)](#), [commencerSeance\(\)](#), [communicationBluetooth](#), [connecterJoueur\(\)](#), [CommunicationBluetooth::demarrer\(\)](#), [editNumeroZone](#), [erreurRobot\(\)](#), [finirSeance\(\)](#), [StatistiquesSeance::initialiserStatistiques\(\)](#), [mettreEnPauseSeance\(\)](#), [StatistiquesSeance::NbZonesImpact](#), [PageAttente](#), [PERIODE_HORLOGE](#), [raccourcisClavier\(\)](#), [recevoirNumeroZone\(\)](#), [reprendreSeance\(\)](#), [resetSeance\(\)](#), [revenirPrecedent\(\)](#), [saisirNumeroZone\(\)](#), [setZoneObjectif\(\)](#), [setZoneRobot\(\)](#), [statistiquesSeance](#), [timerHorloge](#), et [ui](#).

```
00031                                     : QWidget(parent), ui(new Ui::Ttpalhm),
    timerHorloge(nullptr), compteurDureeSeance(0),
    zonesTable(StatistiquesSeance::NbZonesTotales, nullptr)
00032 {
00033     qDebug() << Q_FUNC_INFO;
00034     ui->setupUi(this);
00038 #ifndef QT_NO_DEBUG_OUTPUT
00039     // Pour simuler un impact
00040     editNumeroZone = new QLineEdit(this);
00041     editNumeroZone->setFocus();
00042     editNumeroZone->setFixedSize(0,0);
00043     ui->hLayoutVersEcranResultats->addWidget(editNumeroZone);
00044 #endif
00045     // Choix de l'écran de départ
00046     ui->ecranTtpa->setCurrentIndex(Ttpalhm::PageAttente);
00047     //ui->ecranTtpa->setCurrentIndex(Ttpalhm::PageSeance);
00048     //ui->ecranTtpa->setCurrentIndex(Ttpalhm::PageResultats);
00049
00050     //Plein écran
00051     //const int largeur = qApp->desktop()->availableGeometry(this).width();
00052     //const int hauteur = qApp->desktop()->availableGeometry(this).height();
00053     //resize(largeur, hauteur);
00054     //showMaximized();
00055     showFullScreen();
00056
00057     raccourcisClavier();
00058
00059     //Timer
00060     timerHorloge = new QTimer(this); // Instancie dynamiquement le temporisateur du
    rafraichissement de l'heure
00061     connect(timerHorloge, SIGNAL(timeout()),this,SLOT(afficherHeure())); // Pour
    le déclenchement périodique de l'affichage de l'heure
00062     timerHorloge->start(PERIODE_HORLOGE); // Toutes les secondes (1000 ms)
00063
00064 #ifndef QT_NO_DEBUG_OUTPUT
00065     QAction *actionAllerDroite = new QAction(this);
00066     actionAllerDroite->setShortcut(QKeySequence(Qt::Key_F3));
00067     addAction(actionAllerDroite);
00068     connect(actionAllerDroite, SIGNAL(triggered()), this, SLOT(allerSuivant())); // Pour passer
    à l'écran suivant
00069     QAction *actionAllerGauche = new QAction(this);
00070     actionAllerGauche->setShortcut(QKeySequence(Qt::SHIFT + Qt::Key_F3));
00071     addAction(actionAllerGauche);
00072     connect(actionAllerGauche, SIGNAL(triggered()), this, SLOT(revenirPrecedent())); // Pour
    revenir à l'écran précédent
00073 #endif
00074
00075     statistiquesSeance = new StatistiquesSeance(this);
00076     connect(statistiquesSeance, SIGNAL(affichageZone(int, int, QString)), this, SLOT(
    afficherZone(int, int, QString)));
00077     connect(statistiquesSeance, SIGNAL(ballesTotales(int, int)), this, SLOT(
    afficherBallesTotalEcranSeance(int, int)));
00078     connect(statistiquesSeance, SIGNAL(ballesTotalesHorsTable(int)),this ,SLOT(
    afficherBallesTotalHorsTable(int)));
```

```

00079     connect(statistiquesSeance, SIGNAL(ballesBonnes(int)), this, SLOT(
00080         afficherBallesBonnes(int));
00081
00082     statistiquesSeance->initialiserStatistiques();
00083 #ifdef BLUETOOTH
00084     communicationBluetooth = new CommunicationBluetooth(this);
00085     communicationBluetooth->demarrer();
00086     connect(communicationBluetooth, SIGNAL(impactZone(QString)), this, SLOT(
00087         recevoirNumeroZone(QString));
00088     connect(communicationBluetooth, SIGNAL(departSeance(QString,QString,QString)),
00089         this, SLOT(commencerSeance(QString,QString,QString)); // Pour commencer la seance
00090     connect(communicationBluetooth, SIGNAL(
00091         mettreEnPauseSeance()), this, SLOT(mettreEnPauseSeance()));
00092     connect(communicationBluetooth, SIGNAL(repriseSeance()), this, SLOT(
00093         reprendreSeance()));
00094     connect(communicationBluetooth, SIGNAL(finSeance()), this, SLOT(
00095         finirSeance()));
00096     connect(communicationBluetooth, SIGNAL(resetSeance()), this, SLOT(
00097         resetSeance()));
00098     connect(communicationBluetooth, SIGNAL(erreurRobot(QString)), this,
00099         SLOT(erreurRobot(QString));
00100     connect(communicationBluetooth, SIGNAL(connecte()), this, SLOT(
00101         afficherConnexion()));
00102 #else
00103     communicationBluetooth = nullptr;
00104 #endif
00105
00106     // slot pour les boutons dans chaque écran peut être désactiver avec le .pro ou rajouter le ifndef
00107 #ifndef QT_NO_DEBUG_OUTPUT
00108     connect(editNumeroZone, SIGNAL(textChanged(const QString)), this, SLOT(
00109         saisirNumeroZone(const QString));
00110     //editNumeroZone->setFocus();
00111 #endif
00112 #ifdef SIMULATION
00113     // Simulation en attendant la communication avec le terminal mobile
00114     qsrand(QTime::currentTime().msec());
00115     connecterJoueur("Simon GAUZY");
00116     setZoneRobot(qrand()%int(StatistiquesSeance::NbZonesImpact
00117 ));
00118     setZoneObjectif(qrand()%int(StatistiquesSeance::NbZonesImpact
00119 ));
00120 #endif
00121 }

```

9.15.3.2 ~Ttpalm()

TtpaIhm::~TtpaIhm ()

Références [ui](#).

```

00118 {
00119     // arrêt du timerSequence
00120     //emit arret();
00121     delete ui;
00122     qDebug() << Q_FUNC_INFO;
00123 }

```

9.15.4 Documentation des fonctions membres

9.15.4.1 afficherBallesBonnes

```

void TtpaIhm::afficherBallesBonnes (
    int nombreDeBallesBonnes ) [slot]

```

Paramètres

<i>nombreDeBallesBonnes</i>	et le nombre total ayant atteint la zone objectif
-----------------------------	---

Références [ui](#).

Référencé par [Ttpalhm\(\)](#).

```
00414 {
00415     ui->labelZoneObjectifs->setStyleSheet("color: #FFFFFF;");
00416     QString nbBallesBonnes = QString::number(nombreDeBallesBonnes);
00417     qDebug() << Q_FUNC_INFO << "nbBallesBonnes=" << nbBallesBonnes;
00418     QString affichageBallesBonnes = "<span style=\"font-size:32pt; font-weight:600;\">" + nbBallesBonnes +
    "</span>";
00419     ui->labelZoneObjectifs->setText(affichageBallesBonnes);
00420     ui->labelZoneObjectifs->setStyleSheet("color: #FFFFFF;");
00421 }
```

9.15.4.2 afficherBallesTotalEcranSeance

```
void TtpaIhm::afficherBallesTotalEcranSeance (
    int nombreTotalDeBalle,
    int nombreTotalDeBalleTable ) [slot]
```

Paramètres

<i>nombreTotalDeBalle</i>	et le nombre total jouer par le robot
<i>nombreTotalDeBalleTable</i>	et le nombre total ayant toucher la table (hors table et zoneRobot non comprise)

Références [ui](#).

Référencé par [Ttpalhm\(\)](#).

```
00380 {
00381     QString nbBallesTotales = QString::number(nombreTotalDeBalle);
00382     QString nbBallesTotalesTable = QString::number(nombreTotalDeBalleTable);
00383     qDebug() << Q_FUNC_INFO << "nbBallesTotales=" << nbBallesTotales << "nbBallesTotalesTable=" <<
    nbBallesTotalesTable;
00384     QString affichage = "<span style=\"font-size:32pt; font-weight:600;\">" + nbBallesTotalesTable + " / "
    + nbBallesTotales + "</span>";
00385     ui->labelNbBallesTotales->setText(affichage);
00386     ui->labelNbBallesTotales->setStyleSheet("color: #FFFFFF;");
00387 }
```

9.15.4.3 afficherBallesTotalHorsTable

```
void TtpaIhm::afficherBallesTotalHorsTable (
    int nombreTotalDeBalleHorsTable ) [slot]
```

Paramètres

<i>nombreTotalDeBalleHorsTable</i>	et le nombre total de balle hors table (zoneRobot + hors table)
------------------------------------	---

Références [ui](#).

Référencé par [Ttpalhm\(\)](#).

```
00397 {
```

```

00398     ui->labelNbBallesHorsTable->setStyleSheet("color: #FFFFFF;");
00399     QString nbBallesTotalesHorsTable = QString::number(nombreTotalDeBalleHorsTable);
00400     qDebug() << Q_FUNC_INFO << "nbBallesTotalesTable=" << nbBallesTotalesHorsTable;
00401     QString affichage = "<span style=\"font-size:32pt; font-weight:600;\">" + nbBallesTotalesHorsTable + "
</span>";
00402     ui->labelNbBallesHorsTable->setText(affichage);
00403     ui->labelNbBallesHorsTable->setStyleSheet("color: #FFFFFF;");
00404 }

```

9.15.4.4 afficherConnexion

```
void TtpaIhm::afficherConnexion ( ) [slot]
```

Références [ui](#).

Référencé par [TtpaIhm\(\)](#).

```

00225 {
00226     QString etat;
00227     etat = "<span style=\"font-size:32pt; font-weight:600;\">Connecté</span>";
00228     ui->labelEtatConnexion->setText(etat);
00229 }

```

9.15.4.5 afficherDureeSeance

```
void TtpaIhm::afficherDureeSeance ( ) [slot]
```

Références [compteurDureeSeance](#), et [ui](#).

Référencé par [commencerSeance\(\)](#), [erreurRobot\(\)](#), [finirSeance\(\)](#), [mettreEnPauseSeance\(\)](#), [reprendreSeance\(\)](#), et [resetSeance\(\)](#).

```

00262 {
00263     QString affichageDuree;
00264     QString affichageFinDuree;
00265     QTime duree(0, 0);
00266     compteurDureeSeance++;
00267     QTime dureeSeance = duree.addSecs(compteurDureeSeance);
00268     if(compteurDureeSeance >= 3600)
00269     {
00270         affichageDuree = "<span style=\"font-size:32pt; font-weight:600;\">" + dureeSeance.toString("hh :
mm : ss") + "</span>";
00271         affichageFinDuree = "<span style=\"font-size:32pt; font-weight:600;\">" /**
QString::fromUtf8("Durée de la séance")*/ + dureeSeance.toString("mm : ss") + "</span>";
00272         ui->labelDureeSeance->setText(affichageDuree);
00273         ui->labelDureeSeance->setStyleSheet("color: #FFFFFF;");
00274         ui->labelTimeFinSeance->setText(affichageFinDuree);
00275         ui->labelTimeFinSeance->setStyleSheet("color: #FFFFFF;");
00276     }
00277     else
00278     {
00279         affichageDuree = "<span style=\"font-size:32pt; font-weight:600;\">" + dureeSeance.toString("mm :
ss") + "</span>";
00280         affichageFinDuree = "<span style=\"font-size:32pt; font-weight:600;\">" /**
QString::fromUtf8("Durée de la séance : ")*/ + dureeSeance.toString("mm : ss") + "</span>";
00281         ui->labelDureeSeance->setText(affichageDuree);
00282         ui->labelDureeSeance->setStyleSheet("color: #FFFFFF;");
00283         ui->labelTimeFinSeance->setText(affichageFinDuree);
00284         ui->labelTimeFinSeance->setStyleSheet("color: #FFFFFF;");
00285     }
00286 }

```

9.15.4.6 afficherHeure

```
void TtpaIhm::afficherHeure ( ) [slot]
```

Références [ui](#).

Référencé par [Ttpalhm\(\)](#).

```
00197 {
00198     QString affichageHeure;
00199     QTime heure = QTime::currentTime();
00200     affichageHeure = "<span style=\"font-size:42pt; font-weight:600;\">" + heure.toString("hh : mm : ss") +
    "</span>";
00201     ui->labelHorlogeEcranSeance->setText(affichageHeure);
00202     ui->labelHorlogeEcranResultats->setText(affichageHeure);
00203     ui->labelHorlogeEcranSeance->setStyleSheet("color: rgb(250, 142, 7)");
00204     ui->labelHorlogeEcranResultats->setStyleSheet("color: rgb(250, 142, 7)");
00205     //ui->labelHorlogeEcranSeance->setText(heure.toString("hh : mm : ss"));
00206     //ui->labelHorlogeEcranResultats->setText(heure.toString("hh : mm : ss"));
00207 }
```

9.15.4.7 afficherImpactCouleurs

```
void TtpaIhm::afficherImpactCouleurs (
    int numeroZone ) [slot]
```

Paramètres

<i>numeroZone</i>	et le numéro de la zone toucher par ImpacterZone et l'affiche en couleur en temps réel écran Seance
-------------------	---

Références [effacerImpactZone\(\)](#), [StatistiquesSeance : :getZoneObjectif\(\)](#), [StatistiquesSeance : :getZoneRobot\(\)](#), [numeroZone←Impacte](#), [statistiquesSeance](#), [TEMPS_IMPACT_ZONE](#), [StatistiquesSeance : :ZoneHorsTable](#), et [zonesTable](#).

Référencé par [recevoirNumeroZone\(\)](#), et [saisirNumeroZone\(\)](#).

```
00431 {
00432     // TODO vérifier validité du numeroZone
00433     if(numeroZone == statistiquesSeance->getZoneObjectif())
00434         zonesTable[numeroZone]->setStyleSheet("background-color: rgb(3, 254, 7);color: rgb(255,
    255, 255);border: 2px solid rgb(255, 255, 255);");
00435     else if(numeroZone == statistiquesSeance->getZoneRobot())
00436         zonesTable[numeroZone]->setStyleSheet("background-color: rgb(250, 254, 125);color:
    rgb(255, 255, 255);border: 2px solid rgb(255, 255, 255);");
00437     else
00438         zonesTable[numeroZone]->setStyleSheet("background-color: #ff5e30;color: rgb(255, 255,
    255);border: 2px solid rgb(255, 255, 255);");
00439     if(numeroZone == statistiquesSeance->getZoneRobot())
00440         zonesTable[StatistiquesSeance::ZoneHorsTable]->
    setStyleSheet("background-color: rgb(250, 254, 125);rgb(255, 255, 255);border: 2px solid rgb(255, 255, 255);");
00441     numeroZoneImpacte = numeroZone;
00442     QTimer::singleShot(TEMPS_IMPACT_ZONE, this, SLOT(
    effacerImpactZone()));
00443 }
```

9.15.4.8 afficherStatistiqueEcranResultats

```
void TtpaIhm::afficherStatistiqueEcranResultats ( ) [slot]
```

Références [afficherStatistiquePourcentageBallesHorstable\(\)](#), [afficherStatistiquePourcentageBallesObjectif\(\)](#), [afficherStatistique←PourcentageBallesSerie\(\)](#), [StatistiquesSeance : :getBallesBonnes\(\)](#), [StatistiquesSeance : :getBallesEnchainees\(\)](#), [Statistiques←Seance : :getBallesHorsTable\(\)](#), [StatistiquesSeance : :getBallesTotal\(\)](#), [statistiquesSeance](#), et [ui](#).

Référencé par [finirSeance\(\)](#).


```

00464 {
00465     QString finSeance;
00466     QString zoneObjectif;
00467     QString balleObjectif;
00468     QString zoneHorsTable;
00469     QString balleHorsTable;
00470     QString serieMax;
00471     QString balleEnchainees;
00472     QString dureeSeance;
00473
00474     dureeSeance = "<span style=\"font-size:32pt; font-weight:600;\">" + QString::fromUtf8("Durée de la
séance ") + "</span>";
00475     ui->labelDureeSeanceEcrie->setText(dureeSeance);
00476     ui->labelDureeSeanceEcrie->setStyleSheet("color: #FFFFFF;");
00477     finSeance = "<span style=\"font-size:42pt; font-weight:600;\">" + QString::fromUtf8(" Fin de la Séance
") + "</span>";
00478     ui->labelNbBallesTotalesFin->setText(finSeance);
00479     ui->labelNbBallesTotalesFin->setStyleSheet("color: #FFFFFF;");
00480     zoneObjectif = "<span style=\"font-size:32pt; font-weight:600;\">" + QString::number(
statistiquesSeance->getBallesBonnes()) + " / " + QString::number(
statistiquesSeance->getBallesTotal()) + "</span>";
00481     ui->labelZoneObjectifs->setText(zoneObjectif);
00482     ui->labelZoneObjectifs->setStyleSheet("color: #FFFFFF;");
00483     balleObjectif = "<span style=\"font-size:32pt; font-weight:600;\">" + QString::fromUtf8("Balles ayant
atteint l'objectif ") + "</span>";
00484     ui->labelBallesObjectifs->setText(balleObjectif);
00485     ui->labelBallesObjectifs->setStyleSheet("color: #FFFFFF;");
00486     zoneHorsTable = "<span style=\"font-size:32pt; font-weight:600;\">" + QString::number(
statistiquesSeance->getBallesHorsTable()) + " / " + QString::number(
statistiquesSeance->getBallesTotal()) + "</span>";
00487     ui->labelNbBallesHorsTable->setText(zoneHorsTable);
00488     ui->labelNbBallesHorsTable->setStyleSheet("color: #FFFFFF;");
00489     balleHorsTable = "<span style=\"font-size:32pt; font-weight:600;\">" + QString::fromUtf8("Balles Hors
de la Table ") + "</span>";
00490     ui->labelBallesHorsTable->setText(balleHorsTable);
00491     ui->labelBallesHorsTable->setStyleSheet("color: #FFFFFF;");
00492     balleEnchainees = "<span style=\"font-size:32pt; font-weight:600;\">" + QString::number(
statistiquesSeance->getBallesEnchainees()) + " / " + QString::number(
statistiquesSeance->getBallesBonnes()) + "</span>";
00493     ui->labelNbSerieMax->setText(balleEnchainees);
00494     ui->labelNbSerieMax->setStyleSheet("color: #FFFFFF;");
00495     serieMax = "<span style=\"font-size:32pt; font-weight:600;\">" + QString::fromUtf8("Nombre série de
balles sur l'objectif ") + "</span>";
00496     ui->labelNbSerieMax->setText(serieMax);
00497     ui->labelNbSerieMax->setStyleSheet("color: #FFFFFF;");
00498     afficherStatistiquePourcentageBallesObjectif();
00499     afficherStatistiquePourcentageBallesHorsTable();
00500     afficherStatistiquePourcentageBallesSerie();
00501 }

```

9.15.4.9 afficherStatistiquePourcentageBallesHorstable

```
void TtpaIhm::afficherStatistiquePourcentageBallesHorstable ( ) [slot]
```

Références [StatistiquesSeance : :calculerPourcentage\(\)](#), [StatistiquesSeance : :getBallesHorsTable\(\)](#), [statistiquesSeance](#), et [ui](#).

Référencé par [afficherStatistiqueEcranResults\(\)](#).

```

00522 {
00523     QString pourcentageBallesHorsTable;
00524
00525     pourcentageBallesHorsTable = "<span style=\"font-size:32pt; font-weight:600;\">" + QString::number((
statistiquesSeance->calculerPourcentage(
statistiquesSeance->getBallesHorsTable()), 'f', 0) + "%</span>";
00526     qDebug() << Q_FUNC_INFO << "pourcentageBallesHorsTable=" << pourcentageBallesHorsTable;
00527     if (statistiquesSeance->calculerPourcentage(
statistiquesSeance->getBallesHorsTable()) >= 50.)
00528     {
00529         ui->labelPourcentageBallesHorsTable->setText(pourcentageBallesHorsTable);
00530         ui->labelPourcentageBallesHorsTable->setStyleSheet("color: #FF0000;");
00531     }
00532     else
00533     {
00534         ui->labelPourcentageBallesHorsTable->setText(pourcentageBallesHorsTable);
00535         ui->labelPourcentageBallesHorsTable->setStyleSheet("color: #00FF00;");
00536     }
00537 }

```

9.15.4.10 afficherStatistiquePourcentageBallesObjectif

```
void TtpaIhm::afficherStatistiquePourcentageBallesObjectif ( ) [slot]
```

Références [StatistiquesSeance : :calculerPourcentage\(\)](#), [StatistiquesSeance : :getBallesBonnes\(\)](#), [statistiquesSeance](#), et [ui](#).

Référencé par [afficherStatistiqueEcranResultats\(\)](#).

```
00504 {
00505     QString pourcentageBallesObjectif;
00506
00507     pourcentageBallesObjectif = QString("<span style=\"font-size:32pt; font-weight:600;\">") +
        QString::number(statistiquesSeance->calculerPourcentage(
statistiquesSeance->getBallesBonnes()),'f',0) + " %</span>";
00508     qDebug() << Q_FUNC_INFO << "pourcentageBallesObjectif=" << pourcentageBallesObjectif;
00509     if(statistiquesSeance->calculerPourcentage(
statistiquesSeance->getBallesBonnes()) >= 50.)
00510     {
00511         ui->labelPourcentageBallesObjectif->setText(pourcentageBallesObjectif);
00512         ui->labelPourcentageBallesObjectif->setStyleSheet("color: #00FF00;");
00513     }
00514     else
00515     {
00516         ui->labelPourcentageBallesObjectif->setText(pourcentageBallesObjectif);
00517         ui->labelPourcentageBallesObjectif->setStyleSheet("color: #FF0000;");
00518     }
00519 }
```

9.15.4.11 afficherStatistiquePourcentageBallesSerie

```
void TtpaIhm::afficherStatistiquePourcentageBallesSerie ( ) [slot]
```

Références [StatistiquesSeance : :calculerPourcentage\(\)](#), [StatistiquesSeance : :calculerPourcentageSerieMax\(\)](#), [StatistiquesSeance : :getBallesEnchainees\(\)](#), [StatistiquesSeance : :getBallesHorsTable\(\)](#), [statistiquesSeance](#), et [ui](#).

Référencé par [afficherStatistiqueEcranResultats\(\)](#).

```
00540 {
00541     QString pourcentageBallesSerie;
00542
00543     pourcentageBallesSerie = QString("<span style=\"font-size:32pt; font-weight:600;\">") + QString::number
        ((statistiquesSeance->calculerPourcentageSerieMax(
statistiquesSeance->getBallesEnchainees()),'f',0) + "%</span>";
00544     qDebug() << Q_FUNC_INFO << "pourcentageBallesHorsTable=" << pourcentageBallesSerie;
00545     if(statistiquesSeance->calculerPourcentage(
statistiquesSeance->getBallesHorsTable()) >= 50.)
00546     {
00547         ui->labelPourcentageBallesSerie->setText(pourcentageBallesSerie);
00548         ui->labelPourcentageBallesSerie->setStyleSheet("color: #00FF00;");
00549     }
00550     else
00551     {
00552         ui->labelPourcentageBallesSerie->setText(pourcentageBallesSerie);
00553         ui->labelPourcentageBallesSerie->setStyleSheet("color: #FF0000;");
00554     }
00555 }
```

9.15.4.12 afficherZone

```
void TtpaIhm::afficherZone (
    int numeroZone,
    int nbImpacts,
    QString pourcentage ) [slot]
```

Paramètres

<i>numeroZone</i>	int numéro de la zone compris entre 1 et 7
<i>nbImpacts</i>	int nombre d'impacts dans la zone
<i>pourcentage</i>	QString calculer pour chaque zone

Références [StatistiquesSeance : :getZoneRobot\(\)](#), [statistiquesSeance](#), [StatistiquesSeance : :Zone1](#), [StatistiquesSeance : :Zone7](#), [StatistiquesSeance : :ZoneHorsTable](#), et [zonesTable](#).

Référencé par [Ttpalhm\(\)](#).

```

00328 {
00329     qDebug() << Q_FUNC_INFO << "numeroZone=" << numeroZone << "pourcentage" << pourcentage;
00330     if(numeroZone == statistiquesSeance->getZoneRobot())
00331         return;
00332     if(numeroZone >= StatistiquesSeance::Zone1 && numeroZone <=
00333         StatistiquesSeance::Zone7)
00334     {
00335         QString affichageZone = "<span style=\"font-size:32pt; font-weight:600;\">" + QString::number(
nbImpacts) + "<br />" + pourcentage + "</span>";
00336         zonesTable[numeroZone]->setText(affichageZone);
00337     }
00338     else if(numeroZone == StatistiquesSeance::ZoneHorsTable)
00339     {
00340         QString affichageZone = "<span style=\"font-size:32pt; font-weight:600;\">Hors Table : " +
QString::number(nbImpacts) + " (" + pourcentage + "</span>";
00341         zonesTable[numeroZone]->setText(affichageZone);
00342     }
00343 }
```

9.15.4.13 allerSuivant

```
void TtpaIhm::allerSuivant ( ) [slot]
```

Références [NbPages](#), et [ui](#).

Référencé par [Ttpalhm\(\)](#).

```

00686 {
00687     int ecranCourant = Page(ui->ecranTtpa->currentIndex());
00688     int ecranSuivant = (ecranCourant+1)%int(TtpaIhm::NbPages);
00689     ui->ecranTtpa->setCurrentIndex(ecranSuivant);
00690 }
```

9.15.4.14 commencerSeance

```

void TtpaIhm::commencerSeance (
    QString nomJoueur,
    QString zoneObjectif,
    QString zoneRobot ) [slot]
```

Références [afficherDureeSeance\(\)](#), [StatistiquesSeance : :afficherStatistiquesBallesTotales\(\)](#), [StatistiquesSeance : :afficherStatistiquesZones\(\)](#), [compteurDureeSeance](#), [connecterJoueur\(\)](#), [initialiserZone\(\)](#), [PageSeance](#), [resetSeance\(\)](#), [setZoneObjectif\(\)](#), [setZoneRobot\(\)](#), [statistiquesSeance](#), [timerHorloge](#), et [ui](#).

Référencé par [Ttpalhm\(\)](#).

```

00126 {
00127     qDebug() << Q_FUNC_INFO;
00128     resetSeance();
00129     initialiserZone();
00130     connecterJoueur(nomJoueur);
00131     setZoneObjectif(zoneObjectif);
00132     setZoneRobot(zoneRobot);
00133     statistiquesSeance->afficherStatistiquesZones();
00134     statistiquesSeance->afficherStatistiquesBallesTotales
    ();
00135     compteurDureeSeance = 0;
00136     connect(timerHorloge, SIGNAL(timeout()),this,SLOT(
    afficherDureeSeance())); // Pour le comptage et l'affichage de la durée d'une séance
00137     //QPixmap SystemeOn(":/play-button.png");
00138     //ui->labelSystemeOn->setPixmap(SystemeOn);
00139     ui->ecranTtpa->setCurrentIndex(Ttpalhm::PageSeance);
00140 }

```

9.15.4.15 connecterJoueur

```

void Ttpalhm::connecterJoueur (
    QString nomJoueur ) [slot]

```

Références [ui](#).

Référencé par [commencerSeance\(\)](#), et [Ttpalhm\(\)](#).

```

00238 {
00239     QString nomDuJoueur;
00240     nomDuJoueur = "<span style=\"font-size:32pt; font-weight:600;\">" + QString::fromUtf8("Joueur : %1").
    arg(nomJoueur) + "</span>";
00241     ui->labelNomJoueur->setText(nomDuJoueur.toUtf8());
00242     ui->labelNomJoueur->setStyleSheet("color: rgb(250, 142, 7);");
00243     ui->labelNomJoueur_2->setText(nomDuJoueur.toUtf8());
00244     ui->labelNomJoueur_2->setStyleSheet("color: rgb(250, 142, 7);");
00245 }

```

9.15.4.16 deconnecterJoueur

```

void Ttpalhm::deconnecterJoueur ( ) [slot]

```

Références [ui](#).

Référencé par [resetSeance\(\)](#).

```

00217 {
00218     qDebug() << Q_FUNC_INFO;
00219     ui->labelEtatConnexion->setText("En attente de connexion");
00220     ui->labelNomJoueur->setText("");
00221     //resetSeance();
00222 }

```

9.15.4.17 effacerImpactZone

```
void TtpaIhm::effacerImpactZone ( ) [slot]
```

Références [StatistiquesSeance : :getZoneObjectif\(\)](#), [numeroZoneImpacte](#), [statistiquesSeance](#), [StatistiquesSeance : :ZoneHorsTable](#), et [zonesTable](#).

Référencé par [afficherImpactCouleurs\(\)](#).

```
00451 {
00452     zonesTable[numeroZoneImpacte]->setStyleSheet("color: rgb(255, 255,
255);background-color: rgb(35, 93,166);border: 2px solid rgb(255, 255, 255);");
00453     zonesTable[StatistiquesSeance::ZoneHorsTable]->setStyleSheet
("color: #FFFFFF;background-color : rgb(35, 93,166);");
00454     zonesTable[statistiquesSeance->getZoneObjectif()]->
setStyleSheet(("background-color: rgb(25,25,255);color: rgb(110, 254, 7);border: 2px solid rgb(255, 255, 255);"));
00455 }
```

9.15.4.18 erreurRobot

```
void TtpaIhm::erreurRobot (
    QString infoDuRobot ) [slot]
```

Références [afficherDureeSeance\(\)](#), [InfoRobot\(\)](#), [timerHorloge](#), et [ui](#).

Référencé par [TtpaIhm\(\)](#).

```
00180 {
00181     qDebug() << Q_FUNC_INFO;
00182     disconnect(timerHorloge, SIGNAL(timeout()),this,SLOT(
afficherDureeSeance())); // Pour le comptage et l'affichage de la durée d'une séance
00183     QPixmap systemeOff(":/pause.png" );
00184     InfoRobot (infoDuRobot);
00185     ui->labelErreurRobot->setVisible(true);
00186     ui->labelSystemeOff->setPixmap(systemeOff);
00187     ui->labelSystemeOff->setVisible(true);
00188     //ui->labelErreurRobot->setCurrentIndex(TtpaIhm::PageAttente);
00189 }
```

9.15.4.19 finirSeance

```
void TtpaIhm::finirSeance ( ) [slot]
```

Références [afficherDureeSeance\(\)](#), [afficherStatistiqueEcranResultats\(\)](#), [PageResultats](#), [timerHorloge](#), et [ui](#).

Référencé par [TtpaIhm\(\)](#).

```
00163 {
00164     qDebug() << Q_FUNC_INFO;
00165     disconnect(timerHorloge, SIGNAL(timeout()),this,SLOT(
afficherDureeSeance())); // Pour le comptage et l'affichage de la durée d'une séance
00166     ui->ecranTtpa->setCurrentIndex(TtpaIhm::PageResultats);
00167     afficherStatistiqueEcranResultats();
00168 }
```

9.15.4.20 InfoRobot

```
void Ttpalhm::InfoRobot (
    QString infoDuRobot ) [slot]
```

Références [ui](#).

Référencé par [erreurRobot\(\)](#).

```
00248 {
00249     QString infoRobot;
00250     infoRobot = "<span style=\"font-size:18pt; font-weight:600;\">" + QString::fromUtf8("Info : %1").arg(
infoDuRobot) + "</span>";
00251     ui->labelErreurRobot->setText(infoRobot.toUtf8());
00252     ui->labelErreurRobot->setStyleSheet("color: rgb(250, 0, 0);");
00253 }
00254 }
```

9.15.4.21 initialiserZone()

```
void Ttpalhm::initialiserZone ( ) [private]
```

Références [ui](#), [StatistiquesSeance : :Zone1](#), [StatistiquesSeance : :Zone2](#), [StatistiquesSeance : :Zone3](#), [StatistiquesSeance : :Zone4](#), [StatistiquesSeance : :Zone5](#), [StatistiquesSeance : :Zone6](#), [StatistiquesSeance : :Zone7](#), [StatistiquesSeance : :ZoneHorsTable](#), et [zonesTable](#).

Référencé par [commencerSeance\(\)](#).

```
00295 {
00296     // Initialise le vecteur des zones de la table
00297     zonesTable[StatistiquesSeance::Zone1] =
ui->labelZone1;
00298     zonesTable[StatistiquesSeance::Zone2] =
ui->labelZone2;
00299     zonesTable[StatistiquesSeance::Zone3] =
ui->labelZone3;
00300     zonesTable[StatistiquesSeance::Zone4] =
ui->labelZone4;
00301     zonesTable[StatistiquesSeance::Zone5] =
ui->labelZone5;
00302     zonesTable[StatistiquesSeance::Zone6] =
ui->labelZone6;
00303     zonesTable[StatistiquesSeance::Zone7] =
ui->labelZone7;
00304     zonesTable[StatistiquesSeance::ZoneHorsTable] =
ui->labelZoneHorsTable;
00305
00306     QPixmap aucune;
00307     for(int i=0; i <= StatistiquesSeance::Zone7; i++)
00308     {
00309         zonesTable[i]->clear();
00310         zonesTable[i]->setPixmap(aucune);
00311         zonesTable[i]->setStyleSheet("color: rgb(255, 255, 255);background-color: rgb(35,
93,166);border: 2px solid rgb(255, 255, 255);");
00312     }
00313     zonesTable[StatistiquesSeance::ZoneHorsTable]->setStyleSheet
("color: #FFFFFF;");
00314
00315     ui->labelDureeSeance->setStyleSheet("color: #FFFFFF;");
00316 }
```

9.15.4.22 mettreEnPauseSeance

```
void TtpaIhm::mettreEnPauseSeance ( ) [slot]
```

Références [afficherDureeSeance\(\)](#), [timerHorloge](#), et [ui](#).

Référencé par [Ttpalhm\(\)](#).

```
00143 {
00144     qDebug() << Q_FUNC_INFO;
00145     disconnect(timerHorloge, SIGNAL(timeout()),this,SLOT(
    afficherDureeSeance())); // Pour le comptage et l'affichage de la durée d'une séance
00146     QPixmap systemeOff(":/pause.png" );
00147     //ui->labelSystemeOff->setAlignment(Qt::AlignHCenter);
00148     //ui->labelSystemeOff->setAlignment(Qt::AlignRight);
00149     //SystemeOff = SystemeOff.scaled(QSize(150, 150),Qt::IgnoreAspectRatio);
00150     ui->labelSystemeOff->setPixmap(systemeOff);
00151     ui->labelSystemeOff->setVisible(true);
00152 }
```

9.15.4.23 quitter

```
void TtpaIhm::quitter ( ) [slot]
```

Référencé par [raccourcisClavier\(\)](#).

```
00366 {
00367     // on ferme la fenêtre
00368     close();
00369 }
```

9.15.4.24 raccourcisClavier()

```
void TtpaIhm::raccourcisClavier ( ) [private]
```

Références [quitter\(\)](#).

Référencé par [Ttpalhm\(\)](#).

```
00352 {
00353     QAction *actionQuitter = new QAction("&Quitter", this);
00354     actionQuitter->setShortcut(QKeySequence(QKeySequence::Quit)); // Ctrl+Q
00355     addAction(actionQuitter);
00356     connect(actionQuitter, SIGNAL(triggered()), this, SLOT(quitter())); // Pour le déclenchement
    Ctrl+Q pour quitter l'application
00357 }
```

9.15.4.25 rafraichirHeure()

```
void TtpaIhm::rafraichirHeure ( ) [private]
```

9.15.4.26 recevoirNumeroZone

```
void Ttpalhm::recevoirNumeroZone (
    QString numeroZone ) [slot]
```

Références [afficherImpactCouleurs\(\)](#), [StatistiquesSeance : :impacterZone\(\)](#), et [statistiquesSeance](#).

Référencé par [Ttpalhm\(\)](#).

```
00607 {
00608     bool ok = false;
00609     int valeur = numeroZone.toInt(&ok);
00610     // une valeur entière seulement pour la simulation ?
00611     if(!ok)
00612         return;
00613     statistiquesSeance->impacterZone(
00614         StatistiquesSeance::Zone(valeur));
00615     afficherImpactCouleurs(StatistiquesSeance::Zone(valeur));
00616 }
```

9.15.4.27 reprendreSeance

```
void Ttpalhm::reprendreSeance ( ) [slot]
```

Références [afficherDureeSeance\(\)](#), [timerHorloge](#), et [ui](#).

Référencé par [Ttpalhm\(\)](#).

```
00155 {
00156     qDebug() << Q_FUNC_INFO;
00157     connect(timerHorloge, SIGNAL(timeout()),this,SLOT(
00158         afficherDureeSeance())); // Pour le comptage et l'affichage de la durée d'une séance
00159     ui->labelSystemeOff->setVisible(false);
00160     ui->labelErreurRobot->setVisible(false);
00161 }
```

9.15.4.28 resetSeance

```
void Ttpalhm::resetSeance ( ) [slot]
```

Références [afficherDureeSeance\(\)](#), [deconnecterJoueur\(\)](#), [StatistiquesSeance : :initialiserStatistiques\(\)](#), [PageAttente](#), [statistiquesSeance](#), [timerHorloge](#), et [ui](#).

Référencé par [commencerSeance\(\)](#), et [Ttpalhm\(\)](#).

```
00170 {
00171     deconnecterJoueur();
00172     disconnect(timerHorloge, SIGNAL(timeout()),this,SLOT(
00173         afficherDureeSeance())); // Pour le comptage et l'affichage de la durée d'une séance
00174     statistiquesSeance->initialiserStatistiques();
00175     ui->labelSystemeOff->setVisible(false);
00176     ui->labelErreurRobot->setVisible(false);
00177     ui->ecranTtpa->setCurrentIndex(Ttpalhm::PageAttente);
00178 }
```


9.15.4.29 revenirPrecedent

```
void TtpaIhm::revenirPrecedent ( ) [slot]
```

Références [NbPages](#), et [ui](#).

Référencé par [Ttpalhm\(\)](#).

```
00699 {
00700     int ecranCourant = ui->ecranTtpa->currentIndex();
00701     int ecranPrecedent = (ecranCourant-1)%int(TtpaIhm::NbPages);
00702     if(ecranPrecedent == -1)
00703         ecranPrecedent = NbPages-1;
00704     ui->ecranTtpa->setCurrentIndex(ecranPrecedent);
00705 }
```

9.15.4.30 saisirNumeroZone

```
void TtpaIhm::saisirNumeroZone (
    const QString numeroZone ) [slot]
```

Paramètres

<i>numeroZone</i>	QString
-------------------	---------

Références [afficherImpactCouleurs\(\)](#), [editNumeroZone](#), [StatistiquesSeance : :impacterZone\(\)](#), [statistiquesSeance](#), [StatistiquesSeance : :Zone1](#), [StatistiquesSeance : :Zone2](#), [StatistiquesSeance : :Zone3](#), [StatistiquesSeance : :Zone4](#), [StatistiquesSeance : :Zone5](#), [StatistiquesSeance : :Zone6](#), [StatistiquesSeance : :Zone7](#), et [StatistiquesSeance : :ZoneHorsTable](#).

Référencé par [Ttpalhm\(\)](#).

```
00625 {
00626     bool ok = false;
00627     int valeur = numeroZone.toInt(&ok);
00628     // une valeur entière seulement pour la simulation ?
00629     if(!ok)
00630         return;
00631
00632     switch(valeur)
00633     {
00634         case 7:
00635             statistiquesSeance->impacterZone(
StatistiquesSeance::Zone1);
00636             afficherImpactCouleurs(
StatistiquesSeance::Zone1);
00637             editNumeroZone->clear();
00638             break;
00639         case 8:
00640             statistiquesSeance->impacterZone(
StatistiquesSeance::Zone2);
00641             afficherImpactCouleurs(
StatistiquesSeance::Zone2);
00642             editNumeroZone->clear();
00643             break;
00644         case 9:
00645             statistiquesSeance->impacterZone(
StatistiquesSeance::Zone3);
00646             afficherImpactCouleurs(
StatistiquesSeance::Zone3);
00647             editNumeroZone->clear();
00648             break;
00649         case 4:
00650         case 5:
00651         case 6:
00652             statistiquesSeance->impacterZone(
StatistiquesSeance::Zone4);
00653             afficherImpactCouleurs(
StatistiquesSeance::Zone4);
```

```

00654         editNumeroZone->clear();
00655         break;
00656         case 1:
00657             statistiquesSeance->impacterZone(
StatistiquesSeance::Zone5);
00658             afficherImpactCouleurs(
StatistiquesSeance::Zone5);
00659             editNumeroZone->clear();
00660             break;
00661             case 2:
00662                 statistiquesSeance->impacterZone(
StatistiquesSeance::Zone6);
00663                 afficherImpactCouleurs(
StatistiquesSeance::Zone6);
00664                 editNumeroZone->clear();
00665                 break;
00666                 case 3:
00667                     statistiquesSeance->impacterZone(
StatistiquesSeance::Zone7);
00668                     afficherImpactCouleurs(
StatistiquesSeance::Zone7);
00669                     editNumeroZone->clear();
00670                     break;
00671                     case 0:
00672                         statistiquesSeance->impacterZone(
StatistiquesSeance::ZoneHorsTable);
00673                         afficherImpactCouleurs(
StatistiquesSeance::ZoneHorsTable);
00674                         editNumeroZone->clear();
00675                         break;
00676                     }
00677 }

```

9.15.4.31 setZoneObjectif

```

void TtpaIhm::setZoneObjectif (
    QString numeroZone ) [slot]

```

Paramètres

<i>numeroZone</i>	indique le numéro de la zone
-------------------	------------------------------

Références [StatistiquesSeance : :NbZonesTotales](#), [StatistiquesSeance : :setZoneObjectif\(\)](#), [statistiquesSeance](#), [StatistiquesSeance← : :ZoneNonDefinie](#), et [zonesTable](#).

Référencé par [commencerSeance\(\)](#), et [Ttpalhm\(\)](#).

```

00564 {
00565     bool ok = false;
00566     int valeur = numeroZone.toInt(&ok);
00567     // une valeur entière seulement pour la simulation ?
00568     if(!ok)
00569         return;
00570     qDebug() << Q_FUNC_INFO << "numeroZoneObjectif" << valeur;
00571     statistiquesSeance->setZoneObjectif(valeur);
00572     zonesTable[valeur]->setStyleSheet("background-color: rgb(25,25,255);color: rgb(110, 254,
7);border: 2px solid rgb(255, 255, 255);");
00573     if (((valeur) >= StatistiquesSeance::NbZonesTotales && (valeur) !=
StatistiquesSeance::ZoneNonDefinie))
00574     {
00575         qDebug() << " !\\ Erreur Zone !\\ ";
00576         return;
00577     }
00578 }

```

9.15.4.32 setZoneRobot

```

void TtpaIhm::setZoneRobot (
    QString numeroZone ) [slot]

```

Paramètres

<code>numeroZone</code>	indique le numéro de la zone
-------------------------	------------------------------

Références [StatistiquesSeance : :NbZonesTotales](#), [StatistiquesSeance : :setZoneRobot\(\)](#), [statistiquesSeance](#), [StatistiquesSeance← : :ZoneNonDefinie](#), et [zonesTable](#).

Référencé par [commencerSeance\(\)](#), et [Ttpalhm\(\)](#).

```

00588 {
00589     bool ok = false;
00590     int valeur = numeroZone.toInt(&ok);
00591     // une valeur entière seulement pour la simulation ?
00592     if(!ok)
00593         return;
00594     qDebug() << Q_FUNC_INFO << "numeroZoneRobot" << valeur;
00595     statistiquesSeance->setZoneRobot(valeur);
00596     QPixmap imgRobot(":/robot.png");
00597     QPixmap imageRobot = imgRobot.scaled(QSize(170, 170),Qt::IgnoreAspectRatio);
00598     zonesTable[valeur]->setPixmap(imageRobot);
00599     if ((valeur) >= StatistiquesSeance::NbZonesTotales && (valeur) !=
StatistiquesSeance::ZoneNonDefinie))
00600     {
00601         qDebug() << " /\ \ Erreur Zone /\ \ ";
00602         return;
00603     }
00604 }
```

9.15.5 Documentation des données membres

9.15.5.1 communicationBluetooth

`CommunicationBluetooth*` `TtpaIhm::communicationBluetooth` [private]

Référencé par [Ttpalhm\(\)](#).

9.15.5.2 compteurDureeSeance

`int` `TtpaIhm::compteurDureeSeance` [private]

Référencé par [afficherDureeSeance\(\)](#), et [commencerSeance\(\)](#).

9.15.5.3 editNumeroZone

`QLineEdit*` `TtpaIhm::editNumeroZone` [private]

Référencé par [saisirNumeroZone\(\)](#), et [Ttpalhm\(\)](#).

9.15.5.4 numeroZoneImpacte

`int` `TtpaIhm::numeroZoneImpacte` [private]

Référencé par [afficherImpactCouleurs\(\)](#), et [effacerImpactZone\(\)](#).

9.15.5.5 statistiquesSeance

```
StatistiquesSeance* TtpaIhm::statistiquesSeance [private]
```

Référencé par [afficherImpactCouleurs\(\)](#), [afficherStatistiqueEcranResultats\(\)](#), [afficherStatistiquePourcentageBallesHorstable\(\)](#), [afficherStatistiquePourcentageBallesObjectif\(\)](#), [afficherStatistiquePourcentageBallesSerie\(\)](#), [afficherZone\(\)](#), [commencerSeance\(\)](#), [effacerImpactZone\(\)](#), [recevoirNumeroZone\(\)](#), [resetSeance\(\)](#), [saisirNumeroZone\(\)](#), [setZoneObjectif\(\)](#), [setZoneRobot\(\)](#), et [TtpaIhm\(\)](#).

9.15.5.6 timerHorloge

```
QTimer* TtpaIhm::timerHorloge [private]
```

Référencé par [commencerSeance\(\)](#), [erreurRobot\(\)](#), [finirSeance\(\)](#), [mettreEnPauseSeance\(\)](#), [reprendreSeance\(\)](#), [resetSeance\(\)](#), et [TtpaIhm\(\)](#).

9.15.5.7 ui

```
Ui::TtpaIhm* TtpaIhm::ui [private]
```

Référencé par [afficherBallesBonnes\(\)](#), [afficherBallesTotalEcranSeance\(\)](#), [afficherBallesTotalHorsTable\(\)](#), [afficherConnexion\(\)](#), [afficherDureeSeance\(\)](#), [afficherHeure\(\)](#), [afficherStatistiqueEcranResultats\(\)](#), [afficherStatistiquePourcentageBallesHorstable\(\)](#), [afficherStatistiquePourcentageBallesObjectif\(\)](#), [afficherStatistiquePourcentageBallesSerie\(\)](#), [allerSuivant\(\)](#), [commencerSeance\(\)](#), [connecterJoueur\(\)](#), [deconnecterJoueur\(\)](#), [erreurRobot\(\)](#), [finirSeance\(\)](#), [InfoRobot\(\)](#), [initialiserZone\(\)](#), [mettreEnPauseSeance\(\)](#), [reprendreSeance\(\)](#), [resetSeance\(\)](#), [revenirPrecedent\(\)](#), [TtpaIhm\(\)](#), et [~TtpaIhm\(\)](#).

9.15.5.8 zonesTable

```
QVector<QLabel *> TtpaIhm::zonesTable [private]
```

Référencé par [afficherImpactCouleurs\(\)](#), [afficherZone\(\)](#), [effacerImpactZone\(\)](#), [initialiserZone\(\)](#), [setZoneObjectif\(\)](#), et [setZoneRobot\(\)](#).

La documentation de cette classe a été générée à partir des fichiers suivants :

- [ttpaihm.h](#)
- [statistiquesseance.cpp](#)
- [ttpaihm.cpp](#)

10 Documentation des fichiers

10.1 Référence du fichier BaseDeDonnees.cpp

Définition de la classe [BaseDeDonnees](#) pour Android.

```
#include "BaseDeDonnees.h"
#include <QDebug>
```

10.1.1 Description détaillée

Auteur

Thierry Vaira

Version

1.1

10.2 Référence du fichier BaseDeDonnees.h

Déclaration de la classe [BaseDeDonnees](#) pour Android.

```
#include <QObject>
#include <QtSql/QtSql>
#include <QSqlDatabase>
#include <QMutex>
#include <QString>
```

Classes

— class [BaseDeDonnees](#)

Déclaration de la classe [BaseDeDonnees](#) pour Android.

10.2.1 Description détaillée

Auteur

Thierry VAIRA

Version

1.1

10.3 Référence du fichier Changelog.md

10.4 Référence du fichier communicationbluetooth.cpp

Définition de la classe [CommunicationBluetooth](#).

```
#include "communicationbluetooth.h"
#include <QDebug>
```

10.4.1 Description détaillée

Auteur

Botella-Broc Yohann

Version

1.4

10.5 Référence du fichier CommunicationBluetooth.cpp

Définition de la classe [CommunicationBluetooth](#).

```
#include "CommunicationBluetooth.h"
#include "peripheriqueTPA.h"
#include <QBluetoothDeviceInfo>
#include <QDebug>
#include <QtEndian>
```

10.5.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.6 Référence du fichier communicationbluetooth.h

Déclaration de la classe [CommunicationBluetooth](#).

```
#include <qglobal.h>
#include <QObject>
#include <QLabel>
#include <QBluetoothLocalDevice>
#include <QBluetoothDeviceInfo>
#include <QBluetoothDeviceDiscoveryAgent>
#include <QBluetoothSocket>
#include <QBluetoothServer>
#include <unistd.h>
```

Classes

- class [CommunicationBluetooth](#)
Déclaration de la classe [CommunicationBluetooth](#).

Fonctions

- static const QString [serviceUuid](#) (QStringLiteral("00001101-0000-1000-8000-00805F9B34FB"))
- static const QString [serviceNom](#) (QStringLiteral("raspberry"))

10.6.1 Description détaillée

Auteur

Botella-Broc Yohann

Version

1.4

10.6.2 Documentation des fonctions

10.6.2.1 serviceNom()

```
static const QString serviceNom (
    QStringLiteral("raspberry") ) [static]
```

Référencé par [CommunicationBluetooth : :demarrer\(\)](#).

10.6.2.2 serviceUuid()

```
static const QString serviceUuid (
    QStringLiteral("00001101-0000-1000-8000-00805F9B34FB") ) [static]
```

Référencé par [CommunicationBluetooth : :demarrer\(\)](#).

10.7 Référence du fichier CommunicationBluetooth.h

Déclaration de la classe [CommunicationBluetooth](#).

```
#include <QBluetoothLocalDevice>
#include <QBluetoothAddress>
#include <QBluetoothUuid>
#include <QBluetoothDeviceInfo>
#include <QBluetoothServiceInfo>
#include <QBluetoothDeviceDiscoveryAgent>
```

Classes

- class [CommunicationBluetooth](#)
Déclaration de la classe [CommunicationBluetooth](#).

Macros

- #define [SERVICE_UUID](#) "{00001101-0000-1000-8000-00805F9B34FB}"
- #define [NOM_BLUETOOTH_ROBOT](#) "robot-tpa-"
- #define [NOM_BLUETOOTH_TABLE](#) "table-tpa-"
- #define [NOM_BLUETOOTH_ECRAN](#) "ecran-tpa-"

10.7.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.7.2 Documentation des macros

10.7.2.1 NOM_BLUETOOTH_ECRAN

```
#define NOM_BLUETOOTH_ECRAN "ecran-ttpa-"
```

Référencé par [CommunicationBluetooth :ajouterPeripheriqueTTPA\(\)](#), [CommunicationBluetooth :arreterSeance\(\)](#), [CommunicationBluetooth :demarrerSeanceEcran\(\)](#), [CommunicationBluetooth :envoyerErreurRobot\(\)](#), [CommunicationBluetooth :envoyerZoneImpactee\(\)](#), [CommunicationBluetooth :finirSeance\(\)](#), [CommunicationBluetooth :rechercherPeripheriquesTTPA\(\)](#), et [CommunicationBluetooth :reprendreLaSeance\(\)](#).

10.7.2.2 NOM_BLUETOOTH_ROBOT

```
#define NOM_BLUETOOTH_ROBOT "robot-ttpa-"
```

Référencé par [CommunicationBluetooth :ajouterPeripheriqueTTPA\(\)](#), [CommunicationBluetooth :arreterSeance\(\)](#), [CommunicationBluetooth :demarrerSeanceRobot\(\)](#), [CommunicationBluetooth :finirSeance\(\)](#), et [CommunicationBluetooth :reprendreLaSeance\(\)](#).

10.7.2.3 NOM_BLUETOOTH_TABLE

```
#define NOM_BLUETOOTH_TABLE "table-ttpa-"
```

Référencé par [CommunicationBluetooth :ajouterPeripheriqueTTPA\(\)](#), [CommunicationBluetooth :arreterSeance\(\)](#), [CommunicationBluetooth :demarrerSeanceTable\(\)](#), [CommunicationBluetooth :finirSeance\(\)](#), et [CommunicationBluetooth :reprendreLaSeance\(\)](#).

10.7.2.4 SERVICE_UUID

```
#define SERVICE_UUID "{00001101-0000-1000-8000-00805F9B34FB}"
```

10.8 Référence du fichier FenetreAccueil.qml

Définition de la fenêtre d'accueil.

Classes

— class [FenetreAccueil](#)
La fenêtre d'accueil.

10.8.1 Description détaillée

Auteur

Youssef Hammouma

Version

1.4

10.9 Référence du fichier FenetreReglages.qml

Définition de la fenêtre de réglages.

Classes

- class [FenetreReglages](#)
La fenêtre de réglages.

10.9.1 Description détaillée

Auteur

Youssef Hammouma

Version

1.4

10.10 Référence du fichier FenetreStatistiques.qml

Définition de la fenêtre de statistiques.

Classes

- class [FenetreStatistiques](#)
La fenêtre de statistiques.

10.10.1 Description détaillée

Auteur

Youssef Hammouma

Version

0.1

10.11 Référence du fichier FenetreTTPA.qml

Définition de la fenêtre principale de l'application terminal mobile.

Classes

- class [FenetreTTPA](#)
La fenêtre principale de l'application terminal mobile.

10.11.1 Description détaillée

Auteur

Youssef Hammouma

Version

1.4

10.12 Référence du fichier INSTALL.md

10.13 Référence du fichier Joueur.cpp

Définition de la classe [Joueur](#).

```
#include "Joueur.h"  
#include <QDebug>
```

10.13.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.14 Référence du fichier Joueur.h

Déclaration de la classe [Joueur](#).

```
#include <QObject>
```

Classes

— class [Joueur](#)
Déclaration de la classe [Joueur](#).

10.14.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.15 Référence du fichier Joueurs.cpp

Définition de la classe [Joueurs](#).

```
#include "Joueurs.h"  
#include "Joueur.h"  
#include "BaseDeDonnees.h"  
#include <QDebug>
```

10.15.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.16 Référence du fichier Joueurs.h

Déclaration de la classe [Joueurs](#).

```
#include <QObject>  
#include <QVariant>
```

Classes

— class [Joueurs](#)
Déclaration de la classe [Joueurs](#).

10.16.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.17 Référence du fichier main.cpp

Programme principal ecran-TTPA.

```
#include "ttpaihm.h"  
#include <QApplication>
```

Fonctions

— int `main` (int argc, char *argv[])

10.17.1 Description détaillée

Crée et affiche la fenêtre principale de l'application ecran-TTPA

Auteur

Botella-Broc Yohann

Version

1.4

10.17.2 Documentation des fonctions

10.17.2.1 `main()`

```
main (
    int argc,
    char * argv[] )
```

Paramètres

<code>argc</code>	
<code>argv[]</code>	

Renvoie

int

```
00023 {
00024     QApplication a(argc, argv);
00025
00026     TtpaIhm w; // crée la fenêtre principale de l'application
00027     w.show(); // affiche la fenêtre principale de l'application
00028
00029     /*// Le splash
00030     QSplashScreen splash;
00031     // En avant-plan et sans décoration (titre ...)
00032     splash.setWindowFlags(Qt::WindowStaysOnTopHint | Qt::FramelessWindowHint);
00033     // L'image de l'écran de démarrage
00034     splash.setPixmap(QPixmap(":/logo-la-salle.png"));
00035     // On affiche
00036     splash.show();
00037     // On peut afficher un message sur l'image
00038     splash.showMessage(QString::fromUtf8("Lancement de l'application ..."), Qt::AlignHCenter |
Qt::AlignTop, Qt::black);
00039     // On ferme le splash après un certain temps (ici 2 s)
00040     QTimer::singleShot(2000, &splash, SLOT(close()));
00041     // On affiche l'application
00042     QTimer::singleShot(2000, &w, SLOT(show()));*/
00043
00044
00045     return a.exec();
00046 }
```

10.18 Référence du fichier main.cpp

Programme principal terminal-TTPA.

```
#include <QApplication>
#include <QQmlApplicationEngine>
#include <QQuickStyle>
#include <QIcon>
#include <QQmlContext>
#include <QtSql/QtSql>
#include "CommunicationBluetooth.h"
#include "peripheriqueTTPA.h"
#include "Joueurs.h"
#include "Parametres.h"
#include "Profils.h"
```

Fonctions

— int `main` (int argc, char *argv[])

10.18.1 Description détaillée

Crée et affiche la fenêtre principale de l'application

Auteur

Youssef Hammouma

Version

1.4

10.18.2 Documentation des fonctions

10.18.2.1 main()

```
int main (
    int argc,
    char * argv[] )

00032 {
00033     QCoreApplication::setApplicationName("TTPA");
00034     QCoreApplication::setAttribute(Qt::AA_EnableHighDpiScaling);
00035
00036     QApplication app(argc, argv);
00037     QQmlApplicationEngine engine;
00038
00039     QQuickStyle::setStyle("Material");
00040     QIcon::setThemeName("terminal-TTPA");
00041
00042     CommunicationBluetooth *communicationBluetooth = new
CommunicationBluetooth;
00043     engine.rootContext()->setContextProperty("communicationBluetooth", communicationBluetooth);
00044     qmlRegisterType<PeripheriqueTTPA>();
00045
00046     Joueurs *joueurs = new Joueurs;
00047     engine.rootContext()->setContextProperty("joueurs", joueurs);
00048
00049     Parametres *parametres = new Parametres;
00050     engine.rootContext()->setContextProperty("parametres", parametres);
00051
00052     Profils *profils = new Profils;
00053     engine.rootContext()->setContextProperty("profils", profils);
00054
00055     engine.load(QUrl(QStringLiteral("qrc:/FenetreTTPA.qml")));
00056
00057     if (engine.rootObjects().isEmpty())
00058         return -1;
00059
00060     return app.exec();
00061 }
```

10.19 Référence du fichier Parametre.cpp

Définition de la classe [Parametre](#).

```
#include "Parametre.h"  
#include <QDebug>
```

10.19.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.20 Référence du fichier Parametre.h

Déclaration de la classe [Parametre](#).

```
#include <QObject>
```

Classes

— class [Parametre](#)
Déclaration de la classe [Parametre](#).

10.20.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.21 Référence du fichier Parametres.cpp

Définition de la classe [Parametres](#).

```
#include "Parametres.h"  
#include "Parametre.h"  
#include "BaseDeDonnees.h"  
#include <QDebug>
```

10.21.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.22 Référence du fichier Parametres.h

Déclaration de la classe [Parametres](#).

```
#include <QObject>
#include <QVariant>
```

Classes

— class [Parametres](#)
Déclaration de la classe [Parametres](#).

10.22.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.23 Référence du fichier peripheriqueTPPA.cpp

Définition de la classe [PeripheriqueTPPA](#).

```
#include "peripheriqueTPPA.h"
#include <unistd.h>
```

10.23.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.24 Référence du fichier peripheriqueTTPA.h

Déclaration de la classe [PeripheriqueTTPA](#).

```
#include <QObject>
#include <QBluetoothSocket>
#include <QBluetoothLocalDevice>
```

Classes

- class [PeripheriqueTTPA](#)
Permet la communication avec les appareils Bluetooth TTPA.

10.24.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.25 Référence du fichier Profil.cpp

Définition de la classe [Profil](#).

```
#include "Profil.h"
#include <QDebug>
```

10.25.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.26 Référence du fichier Profil.h

Déclaration de la classe [Profil](#).

```
#include <QObject>
```

Classes

- class [Profil](#)
Déclaration de la classe [Profil](#).

10.26.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.27 Référence du fichier Profils.cpp

Définition de la classe [Profils](#).

```
#include "Profils.h"
#include "BaseDeDonnees.h"
#include "Profil.h"
#include <QDebug>
```

10.27.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.28 Référence du fichier Profils.h

Déclaration de la classe [Profils](#).

```
#include <QObject>
#include <QVariant>
```

Classes

— class [Profils](#)
Déclaration de la classe [Profils](#).

10.28.1 Description détaillée

Auteur

Hammouma Youssef

Version

1.4

10.29 Référence du fichier README.md

10.30 Référence du fichier statistiquesseance.cpp

Définition de la classe [statistiquesseance.cpp](#).

```
#include "statistiquesseance.h"
#include <QDebug>
```

10.30.1 Description détaillée

Auteur

Botella-Broc Yohann

Version

1.4

10.31 Référence du fichier statistiquesseance.h

Déclaration de la classe [StatistiquesSeance](#).

```
#include <qglobal.h>
#include <QObject>
#include <QVector>
#include <QLabel>
```

Classes

— class [StatistiquesSeance](#)
Déclaration de la classe [StatistiquesSeance](#).

10.31.1 Description détaillée

Auteur

Botella-Broc Yohann

Version

1.4

10.32 Référence du fichier ttpaihm.cpp

Définition de la classe [Ttpaihm](#).

```
#include "ttpaihm.h"
#include "ui_ttpaihm.h"
#include "statistiquesseance.h"
#include "communicationbluetooth.h"
#include <QApplication>
#include <QDebug>
#include <sys/cdefs.h>
#include <QColor>
#include <QFrame>
```

10.32.1 Description détaillée

Auteur

Botella-Broc Yohann

Version

1.4

10.33 Référence du fichier ttpaihm.h

Déclaration de la classe [Ttpaihm](#).

```
#include <qglobal.h>
#include <QtWidgets>
```

Classes

- class [Ttpaihm](#)
La fenêtre principale de l'application.

Espaces de nommage

- [Ui](#)

Macros

- #define [BLUETOOTH](#)
si le Bluetooth est présent (Raspberry Pi)
- #define [PERIODE_HORLOGE](#) 1000
- #define [TEMPS_IMPACT_ZONE](#) 500

10.33.1 Description détaillée

Auteur

Botella-Broc Yohann

Version

1.4

10.33.2 Documentation des macros

10.33.2.1 BLUETOOTH

```
#define BLUETOOTH
```

10.33.2.2 PERIODE_HORLOGE

```
#define PERIODE_HORLOGE 1000
```

Référencé par [Ttpaihm](#) : [:Ttpaihm\(\)](#).

10.33.2.3 TEMPS_IMPACT_ZONE

```
#define TEMPS_IMPACT_ZONE 500
```

Référencé par [Ttpaihm](#) : [:afficherImpactCouleurs\(\)](#).

Index

- ~BaseDeDonnees
 - BaseDeDonnees, [12](#)
- ~CommunicationBluetooth
 - CommunicationBluetooth, [26](#), [28](#)
- ~Joueurs
 - Joueurs, [74](#)
- ~Parametres
 - Parametres, [84](#)
- ~Profils
 - Profils, [99](#)
- ~StatistiquesSeance
 - StatistiquesSeance, [105](#)
- ~Ttpalhm
 - Ttpalhm, [119](#)
- accent
 - FenetreTTPA, [68](#)
- adresse
 - CommunicationBluetooth, [48](#)
- adresseMAC
 - PeripheriqueTTPA, [93](#)
- adressePeripheriqueBluetoothLocal
 - CommunicationBluetooth, [50](#)
- affichageZone
 - StatistiquesSeance, [105](#)
- afficherBallesBonnes
 - Ttpalhm, [119](#)
- afficherBallesTotalEcranSeance
 - Ttpalhm, [120](#)
- afficherBallesTotalHorsTable
 - Ttpalhm, [120](#)
- afficherConnexion
 - Ttpalhm, [121](#)
- afficherDureeSeance
 - Ttpalhm, [121](#)
- afficherHeure
 - Ttpalhm, [121](#)
- afficherImpactCouleurs
 - Ttpalhm, [122](#)
- afficherStatistiqueBallesBonnes
 - StatistiquesSeance, [105](#)
- afficherStatistiqueEcranResultats
 - Ttpalhm, [122](#)
- afficherStatistiquePourcentageBallesHorstable
 - Ttpalhm, [123](#)
- afficherStatistiquePourcentageBallesObjectif
 - Ttpalhm, [123](#)
- afficherStatistiquePourcentageBallesSerie
 - Ttpalhm, [124](#)
- afficherStatistiquesBallesTotales
 - StatistiquesSeance, [105](#)
- afficherStatistiquesBallesTotalesHorsTable
 - StatistiquesSeance, [106](#)
- afficherStatistiquesZones
 - StatistiquesSeance, [106](#)
- afficherZone
 - Ttpalhm, [124](#)
- ajouterPeripheriqueTTPA

- CommunicationBluetooth, [28](#)
- alerteNomVide
 - FenetreAccueil, [54](#)
- alertePeripherique
 - FenetreAccueil, [54](#)
- allerSuivant
 - Ttpalhm, [125](#)
- arreter
 - CommunicationBluetooth, [28](#)
- arreterRecherche
 - CommunicationBluetooth, [29](#)
- arreterSeance
 - CommunicationBluetooth, [29](#)

BLUETOOTH

- ttpaih.h, [149](#)
- balle
 - FenetreReglages, [61](#)
- ballePrecedenteToucheTable
 - StatistiquesSeance, [113](#)
- balleToucheRobot
 - StatistiquesSeance, [113](#)
- balleToucheTable
 - StatistiquesSeance, [113](#)
- balles
 - FenetreReglages, [64](#)
- ballesBonnes
 - StatistiquesSeance, [106](#)
- ballesTotales
 - StatistiquesSeance, [106](#)
- ballesTotalesHorsTable
 - StatistiquesSeance, [107](#)
- BaseDeDonnees, [10](#)
 - ~BaseDeDonnees, [12](#)
 - BaseDeDonnees, [12](#)
 - baseDeDonnees, [20](#)
 - db, [20](#)
 - deconnecter, [12](#)
 - detruireInstance, [13](#)
 - estConnecte, [13](#)
 - estOuvert, [13](#)
 - executer, [14](#)
 - getInstance, [14](#)
 - getNomBDD, [15](#)
 - mutex, [20](#)
 - nbAcces, [20](#)
 - nomBDD, [20](#)
 - ouvrir, [15](#)
 - recuperer, [16–19](#)
- baseDeDonnees
 - BaseDeDonnees, [20](#)
- BaseDeDonnees.cpp, [134](#)
- BaseDeDonnees.h, [135](#)
- bdd
 - Joueurs, [75](#)
 - Parametres, [86](#)
 - Profils, [101](#)
- bluetooth

- FenetreAccueil, 54
- boutonArreter
 - FenetreAccueil, 54
- boutonJouer
 - FenetreAccueil, 55
- boutonPositionRobot
 - FenetreReglages, 61
- calculerPourcentage
 - StatistiquesSeance, 107
- calculerPourcentageSerieMax
 - StatistiquesSeance, 107
- calculerStatistiques
 - StatistiquesSeance, 108
- Changelog.md, 135
- color
 - FenetreTTPA, 68
- commencerSeance
 - Ttpalhm, 125
- CommunicationBluetooth, 21
 - ~CommunicationBluetooth, 26, 28
 - adresse, 48
 - adressePeripheriqueBluetoothLocal, 50
 - ajouterPeripheriqueTTPA, 28
 - arreter, 28
 - arreterRecherche, 29
 - arreterSeance, 29
 - CommunicationBluetooth, 26, 27
 - connecte, 29, 30
 - connecterPeripheriquesTTPA, 30
 - connexionErreur, 48
 - decoderTrame, 30
 - deconnecte, 31
 - deconnecterPeripheriquesTTPA, 32
 - demarrer, 32
 - demarrerSeance, 32
 - demarrerSeanceEcran, 33
 - demarrerSeanceRobot, 33
 - demarrerSeanceTable, 34
 - departSeance, 35
 - detecte, 35
 - deviceConnected, 35
 - deviceDisconnected, 35
 - discoveryAgentDevice, 48
 - ecranDetecte, 36
 - envoyerErreurRobot, 36
 - envoyerPeripheriqueTTPA, 36
 - envoyerZoneImpactee, 37
 - erreur, 37
 - erreurRobot, 37, 38
 - erreurRobotDetecte, 50
 - error, 38
 - estValide, 38
 - etatConnexion, 48
 - EtatDeLaSeance, 25
 - etatRecherche, 50
 - EtatSeance, 26
 - etatSeance, 48, 49
 - finSeance, 39
 - finirSeance, 38
 - getAdressePeripheriqueBluetoothLocal, 39

- getEtatSeance, 39
- getNomPeripheriqueBluetoothLocal, 40
- getPeripheriqueTTPA, 40
- getPeripheriquesTTPA, 40
- getZoneObjectif, 40
- getZoneRobot, 41
- impactZone, 41
- mettreEnPauseSeance, 41, 42
- monBluetooth, 49
- nom, 49
- nomPeripheriqueBluetoothLocal, 50
- nouveauClient, 42
- numeroTable, 50
- numeroTableUpdated, 42
- peripheriqueBluetoothLocal, 49
- peripheriqueEcranTTPAConnecte, 51
- peripheriqueEcranTTPADeconnecte, 51
- peripheriqueEcranTTPADetecte, 51
- peripheriqueRobotTTPAConnecte, 51
- peripheriqueRobotTTPADeconnecte, 51
- peripheriqueRobotTTPADetecte, 51
- peripheriqueTableTTPAConnecte, 52
- peripheriqueTableTTPADeconnecte, 52
- peripheriqueTableTTPADetecte, 52
- peripheriquesTTPAUpdated, 43
- peripheriquesTTPA, 49, 51
- recevoirConnecte, 43
- recevoirDeconnecte, 43
- recevoirTramePeripheriqueTTPA, 43
- recherche, 44
- rechercheErreur, 44
- rechercheTerminee, 45
- rechercherPeripheriquesTTPA, 44
- repandreLaSeance, 45
- repriseSeance, 46
- resetSeance, 46
- robotDetecte, 46
- seanceFinie, 46
- seanceTerminee, 52
- serveur, 49
- serviceInfo, 49
- socket, 50
- socketConnected, 46
- socketDisconnected, 47
- socketErreur, 47
- socketReadyRead, 47
- tableDetecte, 48
- communicationBluetooth
 - Ttpalhm, 133
- CommunicationBluetooth.cpp, 136
- CommunicationBluetooth.h, 137
- communicationbluetooth.cpp, 135
- communicationbluetooth.h, 136
- compteurDureeSeance
 - Ttpalhm, 133
- connecte
 - CommunicationBluetooth, 29, 30
 - PeripheriqueTTPA, 89
- connecter
 - PeripheriqueTTPA, 89
- connecterJoueur

- Ttpalhm, 126
- connecterPeripheriquesTTPA
 - CommunicationBluetooth, 30
- connexionErreur
 - CommunicationBluetooth, 48
 - PeripheriqueTTPA, 93
- date
 - FenetreStatistiques, 66
- db
 - BaseDeDonnees, 20
- decoderTrame
 - CommunicationBluetooth, 30
- deconnecte
 - CommunicationBluetooth, 31
 - PeripheriqueTTPA, 89
- deconnecter
 - BaseDeDonnees, 12
 - PeripheriqueTTPA, 90
- deconnecterJoueur
 - Ttpalhm, 126
- deconnecterPeripheriquesTTPA
 - CommunicationBluetooth, 32
- demarrer
 - CommunicationBluetooth, 32
- demarrerSeance
 - CommunicationBluetooth, 32
- demarrerSeanceEcran
 - CommunicationBluetooth, 33
- demarrerSeanceRobot
 - CommunicationBluetooth, 33
- demarrerSeanceTable
 - CommunicationBluetooth, 34
- departSeance
 - CommunicationBluetooth, 35
- detecte
 - CommunicationBluetooth, 35
- detruireInstance
 - BaseDeDonnees, 13
- deviceConnected
 - CommunicationBluetooth, 35
- deviceDisconnected
 - CommunicationBluetooth, 35
- discoveryAgentDevice
 - CommunicationBluetooth, 48
- ecran
 - FenetreAccueil, 55
- ecran-TTPA/communicationbluetooth.h
 - serviceNom, 137
 - serviceUuid, 137
- ecran-TTPA/main.cpp
 - main, 142
- ecranConnecte
 - FenetreAccueil, 58
- ecranDeconnecte
 - FenetreAccueil, 58
- ecranDetecte
 - CommunicationBluetooth, 36
 - FenetreAccueil, 58
- editNumeroZone
 - Ttpalhm, 133
- effacerImpactZone
 - Ttpalhm, 126
- effet
 - FenetreReglages, 64
 - FenetreStatistiques, 66
 - Parametre, 81
- effetChanged
 - Parametre, 78
- effets
 - FenetreReglages, 61
- envoyer
 - PeripheriqueTTPA, 90
- envoyerErreurRobot
 - CommunicationBluetooth, 36
- envoyerPeripheriqueTTPA
 - CommunicationBluetooth, 36
- envoyerZoneImpactee
 - CommunicationBluetooth, 37
- erreur
 - CommunicationBluetooth, 37
 - PeripheriqueTTPA, 90
- erreurRobot
 - CommunicationBluetooth, 37, 38
 - Ttpalhm, 127
- erreurRobotDetecte
 - CommunicationBluetooth, 50
- erreurrobotDetectee
 - FenetreAccueil, 58
- error
 - CommunicationBluetooth, 38
- estConnecte
 - BaseDeDonnees, 13
 - PeripheriqueTTPA, 90
- estOuvert
 - BaseDeDonnees, 13
- estValide
 - CommunicationBluetooth, 38
- etatConnexion
 - CommunicationBluetooth, 48
 - PeripheriqueTTPA, 94
- EtatDeLaSeance
 - CommunicationBluetooth, 25
- etatEcran
 - FenetreAccueil, 55
- etatRecherche
 - CommunicationBluetooth, 50
- etatRobot
 - FenetreAccueil, 55
- EtatSeance
 - CommunicationBluetooth, 26
- etatSeance
 - CommunicationBluetooth, 48, 49
- etatTable
 - FenetreAccueil, 55
- executer
 - BaseDeDonnees, 14
- FenetreAccueil, 53
 - alerteNomVide, 54
 - alertePeripherique, 54

- bluetooth, 54
- boutonArreter, 54
- boutonJouer, 55
- ecran, 55
- ecranConnecte, 58
- ecranDeconnecte, 58
- ecranDetecte, 58
- erreurrobotDetectee, 58
- etatEcran, 55
- etatRobot, 55
- etatTable, 55
- indicateur, 55
- labelnumTable, 55
- listeJoueurs, 55
- logo, 55
- messageErreurRobot, 56
- nomJoueur, 56
- numTable, 56
- onEcranConnecteChanged, 56
- onEcranDeconnecteChanged, 56
- onEcranDetecteChanged, 56
- onErreurrobotDetecteeChanged, 56
- onRechercheChanged, 56
- onRobotConnecteChanged, 56
- onRobotDeconnecteChanged, 57
- onRobotDetecteChanged, 57
- onSeanceTermineeChanged, 57
- onTableConnecteChanged, 57
- onTableDeconnecteChanged, 57
- onTableDetecteChanged, 57
- recherche, 58
- robot, 57
- robotConnecte, 58
- robotDeconnecte, 58
- robotDetecte, 59
- seanceTerminee, 59
- switchBluetooth, 57
- table, 57
- tableConnecte, 59
- tableDeconnecte, 59
- tableDetecte, 59
- toolButton2, 58
- FenetreAccueil.qml, 138
- FenetreReglages, 60
 - balle, 61
 - balles, 64
 - boutonPositionRobot, 61
 - effet, 64
 - effets, 61
 - frequenceBalles, 61, 64
 - idParametres, 64
 - listeProfils, 64
 - nbBalles, 64
 - nbClics, 64
 - nbZonesSelectionee, 64
 - nomSeance, 65
 - nombreBalles, 61
 - nombreFrequenceBalles, 62
 - numeroZoneObjectif, 65
 - numeroZoneRobot, 65
 - positionRobot, 62
 - puissanceBalles, 62, 65
 - rotation, 62
 - sliderFrequenceBalles, 62
 - sliderNombreBalles, 62
 - sliderPuissanceBalles, 62
 - switchPosition, 65
 - valeurPuissanceBalles, 62
 - zone1, 62
 - zone2, 63
 - zone3, 63
 - zone4, 63
 - zone5, 63
 - zone6, 63
 - zone7, 63
 - zoneObjectif, 63
 - zoneRobot, 63
 - zones, 63
- FenetreReglages.qml, 139
- FenetreStatistiques, 65
 - date, 66
 - effet, 66
 - historique, 66
 - nbBalles, 66
 - nomJoueur, 66
 - tauxReussite, 66
- FenetreStatistiques.qml, 139
- FenetreTTPA.qml, 139
- FenetreTTPA, 67
 - accent, 68
 - color, 68
 - height, 68
 - lockToPortrait, 68
 - onglets, 68
 - pageAccueil, 69
 - pageReglages, 69
 - pageStatistiques, 69
 - theme, 69
 - title, 69
 - visible, 69
 - vueBalayage, 69
 - width, 69
- finSeance
 - CommunicationBluetooth, 39
- finirSeance
 - CommunicationBluetooth, 38
 - StatistiquesSeance, 108
 - Ttpalhm, 127
- frequenceBalles
 - FenetreReglages, 61, 64
- getAdresseMAC
 - PeripheriqueTTPA, 91
- getAdressePeripheriqueBluetoothLocal
 - CommunicationBluetooth, 39
- getBallesBonnes
 - StatistiquesSeance, 109
- getBallesEnchainees
 - StatistiquesSeance, 109
- getBallesHorsTable
 - StatistiquesSeance, 109
- getBallesTotal

- StatistiquesSeance, 110
- getEffet
 - Parametre, 78
- getEtatSeance
 - CommunicationBluetooth, 39
- getId
 - Joueur, 71
 - Parametre, 78
 - Profil, 95
- getIdJoueur
 - Profil, 96
- getIdParametres
 - Profil, 96
- getInstance
 - BaseDeDonnees, 14
- getListeJoueurs
 - Joueurs, 74
- getListeParametres
 - Parametres, 84
- getListeProfils
 - Profils, 99
- getNbBalles
 - Parametre, 78
- getNbBallesMinute
 - Parametre, 79
- getNom
 - Joueur, 71
 - PeripheriqueTTPA, 91
 - Profil, 96
- getNomBDD
 - BaseDeDonnees, 15
- getNomPeripheriqueBluetoothLocal
 - CommunicationBluetooth, 40
- getPeripheriqueTTPA
 - CommunicationBluetooth, 40
- getPeripheriquesTTPA
 - CommunicationBluetooth, 40
- getPuissance
 - Parametre, 79
- getRotation
 - Parametre, 79
- getZoneObjectif
 - CommunicationBluetooth, 40
 - Parametre, 79
 - StatistiquesSeance, 110
- getZoneRobot
 - CommunicationBluetooth, 41
 - Parametre, 79
 - StatistiquesSeance, 111
- getballesPrecedentToucheTable
 - StatistiquesSeance, 109
- getballesToucheTable
 - StatistiquesSeance, 110
- height
 - FenetreTTPA, 68
- historique
 - FenetreStatistiques, 66
- INSTALL.md, 140
- id
 - Joueur, 72
 - Parametre, 81
 - Profil, 97
- idChanged
 - Joueur, 71
 - Parametre, 80
 - Profil, 96
- idJoueur
 - Profil, 97
- idJoueurChanged
 - Profil, 96
- idParametres
 - FenetreReglages, 64
 - Profil, 97
- idParametresChanged
 - Profil, 97
- impactZone
 - CommunicationBluetooth, 41
- impacterZone
 - StatistiquesSeance, 111
- indicateur
 - FenetreAccueil, 55
- InfoRobot
 - Ttpalhm, 127
- initialiserStatistiques
 - StatistiquesSeance, 112
- initialiserZone
 - Ttpalhm, 128
- insérerJoueur
 - Joueurs, 74
- insérerParametres
 - Parametres, 85
- insérerProfil
 - Profils, 100
- Joueur, 70
 - getId, 71
 - getNom, 71
 - id, 72
 - idChanged, 71
 - Joueur, 70
 - nom, 72
 - nomChanged, 71
- Joueur.cpp, 140
- Joueur.h, 140
- Joueurs, 72
 - ~Joueurs, 74
 - bdd, 75
 - getListeJoueurs, 74
 - insérerJoueur, 74
 - Joueurs, 74
 - joueurs, 76
 - listeJoueurs, 76
 - listeJoueursUpdated, 75
 - recupererJoueurs, 75
- joueurs
 - Joueurs, 76
- Joueurs.cpp, 141
- Joueurs.h, 141
- labelnumTable

- FenetreAccueil, 55
- listeJoueurs
 - FenetreAccueil, 55
 - Joueurs, 76
- listeJoueursUpdated
 - Joueurs, 75
- listeParametres
 - Parametres, 87
- listeParametresUpdated
 - Parametres, 85
- listeProfils
 - FenetreReglages, 64
 - Profils, 101
- listeProfilsUpdated
 - Profils, 100
- lockToPortrait
 - FenetreTTPA, 68
- logo
 - FenetreAccueil, 55
- main
 - ecran-TTPA/main.cpp, 142
 - terminal-TTPA/main.cpp, 143
- main.cpp, 141, 143
- messageErreurRobot
 - FenetreAccueil, 56
- mettreEnPauseSeance
 - CommunicationBluetooth, 41, 42
 - Ttpalhm, 128
- monBluetooth
 - CommunicationBluetooth, 49
- mutex
 - BaseDeDonnees, 20
- NOM_BLUETOOTH_ECRAN
 - terminal-TTPA/communicationbluetooth.h, 138
- NOM_BLUETOOTH_ROBOT
 - terminal-TTPA/communicationbluetooth.h, 138
- NOM_BLUETOOTH_TABLE
 - terminal-TTPA/communicationbluetooth.h, 138
- nbAcces
 - BaseDeDonnees, 20
- nbBalles
 - FenetreReglages, 64
 - FenetreStatistiques, 66
 - Parametre, 81
- nbBallesBonnes
 - StatistiquesSeance, 113
- nbBallesChanged
 - Parametre, 80
- nbBallesEnchainees
 - StatistiquesSeance, 113
- nbBallesEnchaineesMax
 - StatistiquesSeance, 113
- nbBallesMinute
 - Parametre, 81
- nbBallesMinuteChanged
 - Parametre, 80
- nbBallesTotales
 - StatistiquesSeance, 113
- nbBallesTotalesTable
 - StatistiquesSeance, 114
- nbClics
 - FenetreReglages, 64
- nbImpactsZone
 - StatistiquesSeance, 114
- nbZonesSelectionee
 - FenetreReglages, 64
- nom
 - CommunicationBluetooth, 49
 - Joueur, 72
 - PeripheriqueTTPA, 94
 - Profil, 97
- nomBDD
 - BaseDeDonnees, 20
- nomChanged
 - Joueur, 71
 - Profil, 97
- nomJoueur
 - FenetreAccueil, 56
 - FenetreStatistiques, 66
- nomPeripheriqueBluetoothLocal
 - CommunicationBluetooth, 50
- nomSeance
 - FenetreReglages, 65
- nombreBalles
 - FenetreReglages, 61
- nombreFrequenceBalles
 - FenetreReglages, 62
- nouveauClient
 - CommunicationBluetooth, 42
- numTable
 - FenetreAccueil, 56
- numeroTable
 - CommunicationBluetooth, 50
- numeroTableUpdated
 - CommunicationBluetooth, 42
- numeroZoneImpacte
 - Ttpalhm, 133
- numeroZoneObjectif
 - FenetreReglages, 65
 - StatistiquesSeance, 114
- numeroZoneRobot
 - FenetreReglages, 65
 - StatistiquesSeance, 114
- onEcranConnecteChanged
 - FenetreAccueil, 56
- onEcranDeconnecteChanged
 - FenetreAccueil, 56
- onEcranDetecteChanged
 - FenetreAccueil, 56
- onErreurrobotDetecteeChanged
 - FenetreAccueil, 56
- onRechercheChanged
 - FenetreAccueil, 56
- onRobotConnecteChanged
 - FenetreAccueil, 56
- onRobotDeconnecteChanged
 - FenetreAccueil, 57
- onRobotDetecteChanged
 - FenetreAccueil, 57

- onSeanceTermineeChanged
 - FenetreAccueil, 57
- onTableConnecteChanged
 - FenetreAccueil, 57
- onTableDeconnecteChanged
 - FenetreAccueil, 57
- onTableDetecteChanged
 - FenetreAccueil, 57
- onglets
 - FenetreTTPA, 68
- ouvrir
 - BaseDeDonnees, 15
- PERIODE_HORLOGE
 - ttpaihm.h, 149
- Page
 - Ttpalhm, 117
- pageAccueil
 - FenetreTTPA, 69
- pageReglages
 - FenetreTTPA, 69
- pageStatistiques
 - FenetreTTPA, 69
- Parametre, 76
 - effet, 81
 - effetChanged, 78
 - getEffet, 78
 - getId, 78
 - getNbBalles, 78
 - getNbBallesMinute, 79
 - getPuissance, 79
 - getRotation, 79
 - getZoneObjectif, 79
 - getZoneRobot, 79
 - id, 81
 - idChanged, 80
 - nbBalles, 81
 - nbBallesChanged, 80
 - nbBallesMinute, 81
 - nbBallesMinuteChanged, 80
 - Parametre, 77
 - puissance, 81
 - puissanceChanged, 80
 - rotation, 81
 - rotationChanged, 80
 - zoneObjectif, 81
 - zoneObjectifChanged, 80
 - zoneRobot, 82
 - zoneRobotChanged, 80
- Parametre.cpp, 144
- Parametre.h, 144
- Parametres, 82
 - ~Parametres, 84
 - bdd, 86
 - getListeParametres, 84
 - insererParametres, 85
 - listeParametres, 87
 - listeParametresUpdated, 85
 - Parametres, 84
 - parametres, 86
 - recupererParametres, 85
 - recupererParametresProfil, 86
- parametres
 - Parametres, 86
- Parametres.cpp, 144
- Parametres.h, 145
- peripheriqueBluetoothLocal
 - CommunicationBluetooth, 49
- peripheriqueEcranTTPAConnecte
 - CommunicationBluetooth, 51
- peripheriqueEcranTTPADeconnecte
 - CommunicationBluetooth, 51
- peripheriqueEcranTTPADetecte
 - CommunicationBluetooth, 51
- peripheriqueRobotTTPAConnecte
 - CommunicationBluetooth, 51
- peripheriqueRobotTTPADeconnecte
 - CommunicationBluetooth, 51
- peripheriqueRobotTTPADetecte
 - CommunicationBluetooth, 51
- peripheriqueTTPA.cpp, 145
- peripheriqueTTPA.h, 146
- peripheriqueTTPAChanged
 - PeripheriqueTTPA, 91
- PeripheriqueTTPA, 87
 - adresseMAC, 93
 - connecte, 89
 - connecter, 89
 - connexionErreur, 93
 - deconnecte, 89
 - deconnecter, 90
 - envoyer, 90
 - erreur, 90
 - estConnecte, 90
 - etatConnexion, 94
 - getAdresseMAC, 91
 - getNom, 91
 - nom, 94
 - peripheriqueTTPAChanged, 91
 - PeripheriqueTTPA, 89
 - socket, 93
 - socketConnecte, 91
 - socketDeconnecte, 92
 - socketErreur, 92
 - socketReadyRead, 92
 - trameRecue, 93
- peripheriqueTableTTPAConnecte
 - CommunicationBluetooth, 52
- peripheriqueTableTTPADeconnecte
 - CommunicationBluetooth, 52
- peripheriqueTableTTPADetecte
 - CommunicationBluetooth, 52
- peripheriquesTTPAUpdated
 - CommunicationBluetooth, 43
- peripheriquesTTPA
 - CommunicationBluetooth, 49, 51
- positionRobot
 - FenetreReglages, 62
- Profil, 94
 - getId, 95
 - getIdJoueur, 96
 - getIdParametres, 96

- getNom, 96
- id, 97
- idChanged, 96
- idJoueur, 97
- idJoueurChanged, 96
- idParametres, 97
- idParametresChanged, 97
- nom, 97
- nomChanged, 97
- Profil, 95
- Profil.cpp, 146
- Profil.h, 146
- Profils, 98
 - ~Profils, 99
 - bdd, 101
 - getListeProfils, 99
 - insererProfil, 100
 - listeProfils, 101
 - listeProfilsUpdated, 100
 - Profils, 99
 - profils, 101
 - recupererNomProfils, 100
- profils
 - Profils, 101
- Profils.cpp, 147
- Profils.h, 147
- puissance
 - Parametre, 81
- puissanceBalles
 - FenetreReglages, 62, 65
- puissanceChanged
 - Parametre, 80
- quitter
 - Ttpalhm, 129
- README.md, 148
- raccourcisClavier
 - Ttpalhm, 129
- rafraichirHeure
 - Ttpalhm, 129
- recevoirConnecte
 - CommunicationBluetooth, 43
- recevoirDeconnecte
 - CommunicationBluetooth, 43
- recevoirNumeroZone
 - Ttpalhm, 129
- recevoirTramePeripheriqueTTPA
 - CommunicationBluetooth, 43
- recherche
 - CommunicationBluetooth, 44
 - FenetreAccueil, 58
- rechercheErreur
 - CommunicationBluetooth, 44
- rechercheTerminee
 - CommunicationBluetooth, 45
- rechercherPeripheriquesTTPA
 - CommunicationBluetooth, 44
- recuperer
 - BaseDeDonnees, 16–19
- recupererJoueurs
 - Joueurs, 75
- recupererNomProfils
 - Profils, 100
- recupererParametres
 - Parametres, 85
- recupererParametresProfil
 - Parametres, 86
- repandreLaSeance
 - CommunicationBluetooth, 45
- repandreSeance
 - Ttpalhm, 130
- repriseSeance
 - CommunicationBluetooth, 46
- resetSeance
 - CommunicationBluetooth, 46
 - Ttpalhm, 130
- revenirPrecedent
 - Ttpalhm, 130
- robot
 - FenetreAccueil, 57
- robotConnecte
 - FenetreAccueil, 58
- robotDeconnecte
 - FenetreAccueil, 58
- robotDetecte
 - CommunicationBluetooth, 46
 - FenetreAccueil, 59
- rotation
 - FenetreReglages, 62
 - Parametre, 81
- rotationChanged
 - Parametre, 80
- SERVICE_UUID
 - terminal-TTPA/communicationbluetooth.h, 138
- saisirNumeroZone
 - Ttpalhm, 131
- seanceFinie
 - CommunicationBluetooth, 46
- seanceTerminee
 - CommunicationBluetooth, 52
 - FenetreAccueil, 59
- serveur
 - CommunicationBluetooth, 49
- serviceInfo
 - CommunicationBluetooth, 49
- serviceNom
 - ecran-TTPA/communicationbluetooth.h, 137
- serviceUuid
 - ecran-TTPA/communicationbluetooth.h, 137
- setZoneObjectif
 - StatistiquesSeance, 112
 - Ttpalhm, 132
- setZoneRobot
 - StatistiquesSeance, 112
 - Ttpalhm, 132
- sliderFrequenceBalles
 - FenetreReglages, 62
- sliderNombreBalles
 - FenetreReglages, 62
- sliderPuissanceBalles

- FenetreReglages, 62
- socket
 - CommunicationBluetooth, 50
 - PeripheriqueTTPA, 93
- socketConnecte
 - PeripheriqueTTPA, 91
- socketConnected
 - CommunicationBluetooth, 46
- socketDeconnecte
 - PeripheriqueTTPA, 92
- socketDisconnected
 - CommunicationBluetooth, 47
- socketErreur
 - CommunicationBluetooth, 47
 - PeripheriqueTTPA, 92
- socketReadyRead
 - CommunicationBluetooth, 47
 - PeripheriqueTTPA, 92
- StatistiquesSeance, 102
 - ~StatistiquesSeance, 105
 - affichageZone, 105
 - afficherStatistiqueBallesBonnes, 105
 - afficherStatistiquesBallesTotales, 105
 - afficherStatistiquesBallesTotalesHorsTable, 106
 - afficherStatistiquesZones, 106
 - ballePrecedenteToucheTable, 113
 - balleToucheRobot, 113
 - balleToucheTable, 113
 - ballesBonnes, 106
 - ballesTotales, 106
 - ballesTotalesHorsTable, 107
 - calculerPourcentage, 107
 - calculerPourcentageSerieMax, 107
 - calculerStatistiques, 108
 - finirSeance, 108
 - getBallesBonnes, 109
 - getBallesEnchainees, 109
 - getBallesHorsTable, 109
 - getBallesTotal, 110
 - getZoneObjectif, 110
 - getZoneRobot, 111
 - getballesPrecedentToucheTable, 109
 - getballesToucheTable, 110
 - impacterZone, 111
 - initialiserStatistiques, 112
 - nbBallesBonnes, 113
 - nbBallesEnchainees, 113
 - nbBallesEnchaineesMax, 113
 - nbBallesTotales, 113
 - nbBallesTotalesTable, 114
 - nbImpactsZone, 114
 - numeroZoneObjectif, 114
 - numeroZoneRobot, 114
 - setZoneObjectif, 112
 - setZoneRobot, 112
 - StatistiquesSeance, 104
 - Zone, 104
- statistiquesSeance
 - Ttpalhm, 133
- statistiquesseance.cpp, 148
- statistiquesseance.h, 148
- switchBluetooth
 - FenetreAccueil, 57
- switchPosition
 - FenetreReglages, 65
- TEMPS_IMPACT_ZONE
 - ttpalhm.h, 149
- table
 - FenetreAccueil, 57
- tableConnecte
 - FenetreAccueil, 59
- tableDeconnecte
 - FenetreAccueil, 59
- tableDetecte
 - CommunicationBluetooth, 48
 - FenetreAccueil, 59
- tauxReussite
 - FenetreStatistiques, 66
- terminal-TTPA/communicationbluetooth.h
 - NOM_BLUETOOTH_ECRAN, 138
 - NOM_BLUETOOTH_ROBOT, 138
 - NOM_BLUETOOTH_TABLE, 138
 - SERVICE_UUID, 138
- terminal-TTPA/main.cpp
 - main, 143
- theme
 - FenetreTTPA, 69
- timerHorloge
 - Ttpalhm, 134
- title
 - FenetreTTPA, 69
- toolButton2
 - FenetreAccueil, 58
- trameRecue
 - PeripheriqueTTPA, 93
- Ttpalhm, 115
 - ~Ttpalhm, 119
 - afficherBallesBonnes, 119
 - afficherBallesTotalEcranSeance, 120
 - afficherBallesTotalHorsTable, 120
 - afficherConnexion, 121
 - afficherDureeSeance, 121
 - afficherHeure, 121
 - afficherImpactCouleurs, 122
 - afficherStatistiqueEcranResultats, 122
 - afficherStatistiquePourcentageBallesHorstable, 123
 - afficherStatistiquePourcentageBallesObjectif, 123
 - afficherStatistiquePourcentageBallesSerie, 124
 - afficherZone, 124
 - allerSuivant, 125
 - commencerSeance, 125
 - communicationBluetooth, 133
 - compteurDureeSeance, 133
 - connecterJoueur, 126
 - deconnecterJoueur, 126
 - editNumeroZone, 133
 - effacerImpactZone, 126
 - erreurRobot, 127
 - finirSeance, 127
 - InfoRobot, 127
 - initialiserZone, 128

- mettreEnPauseSeance, 128
- numeroZoneImpacte, 133
- Page, 117
- quitter, 129
- raccourcisClavier, 129
- rafraichirHeure, 129
- recevoirNumeroZone, 129
- repandreSeance, 130
- resetSeance, 130
- revenirPrecedent, 130
- saisirNumeroZone, 131
- setZoneObjectif, 132
- setZoneRobot, 132
- statistiquesSeance, 133
- timerHorloge, 134
- Ttpalhm, 117
- ui, 134
- zonesTable, 134
- ttpalhm.cpp, 148
- ttpalhm.h, 149
 - BLUETOOTH, 149
 - PERIODE_HORLOGE, 149
 - TEMPS_IMPACT_ZONE, 149
- Ui, 10
- ui
 - Ttpalhm, 134
- valeurPuissanceBalles
 - FenetreReglages, 62
- visible
 - FenetreTTPA, 69
- vueBalayage
 - FenetreTTPA, 69
- width
 - FenetreTTPA, 69
- Zone
 - StatistiquesSeance, 104
- zone1
 - FenetreReglages, 62
- zone2
 - FenetreReglages, 63
- zone3
 - FenetreReglages, 63
- zone4
 - FenetreReglages, 63
- zone5
 - FenetreReglages, 63
- zone6
 - FenetreReglages, 63
- zone7
 - FenetreReglages, 63
- zoneObjectif
 - FenetreReglages, 63
 - Parametre, 81
- zoneObjectifChanged
 - Parametre, 80
- zoneRobot
 - FenetreReglages, 63
 - Parametre, 82
- zoneRobotChanged
 - Parametre, 80
- zones
 - FenetreReglages, 63
- zonesTable
 - Ttpalhm, 134